Appendix for:
ImageNet-Hard: The Hardest Images Remaining from a Study of the Power of Zoom and Spatial Biases in Image Classification

A Implementation details

In this section, we provide a detailed description of our experimental setup, including the Python code for our zoom transform, the classifiers we employed, and the setup we used for zero-shot classification.

A.1 Sample Python code for zoom-based transform

```
from PIL import Image
import torchvision.transforms.functional as fv
import torchvision.transforms as transforms
from functools import partial

def crop_at(size, slice_x, slice_y):
    def slice_crop(image, size, slice_x, slice_y):
        width, height = image.size
        tile_size_x = width // 3
        tile_size_y = height // 3
        anchor_x = (slice_y * tile_size_x) + (tile_size_x // 2)
        anchor_y = (slice_x * tile_size_y) + (tile_size_y // 2)
        return fv.crop(
            image,
            anchor_y - (size // 2),
            anchor_x - (size // 2),
            size,
            size,
        )
    return partial(slice_crop, size=size, slice_x=slice_x, slice_y=
    slice_y)

zoom_scale = 255
zoom_transform = transforms.Compose(
                    [
                        transforms.Resize(
                            zoom_scale,
                            interpolation=transforms.InterpolationMode
    .BICUBIC,
                            max_size=None,
                            antialias=None,
                        ),
                        crop_at(224, i, j),
                    ]
                )
```

Figure A1: Sample python code.

A.2 Datasets' licenses

Dataset Name	License
ImageNet	Custom license, non-commercial
ImageNet-A	License
ImageNet-R	MIT License
ImageNet-Sketch	MIT License
ImageNet-C	MIT License
ObjectNet	Custom license derived from Creative Commons Attribution 4.0
ImageNet-V2	MIT License

Table A1: Dataset Licenses

A.3 Zoom Scales used

In our experiments, we tried the following zoom scales:

$$10, 16, 32, 48, 64, 96, 122, 128, 192, 224, 235, 240, 256, 288, 320, 348, 384, 448, 460, 512, 573, 576, 640, 664, 672, 680, 686, 690, 700, 720, 768, 798, 832, 896, 911, 1024.$$

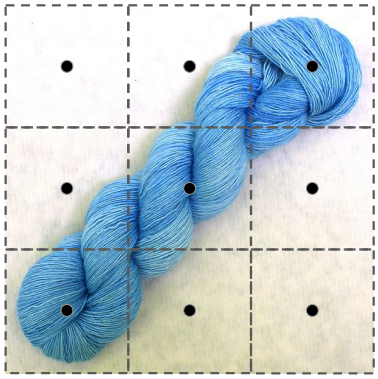
A.4 Model selection

We use the official OpenAI's official CLIP for all CLIP-related experiments. All IN-trained models are retrieved from the `torchvision` [47] library. For models from the OpenCLIP family, we utilize the OpenCLIP library version `2.20.0`. In the case of the EfficientNet-B family, we use the Hugging Face Transformers library. Lastly, for EfficientNet-L2, we use the implementation from the timm library.

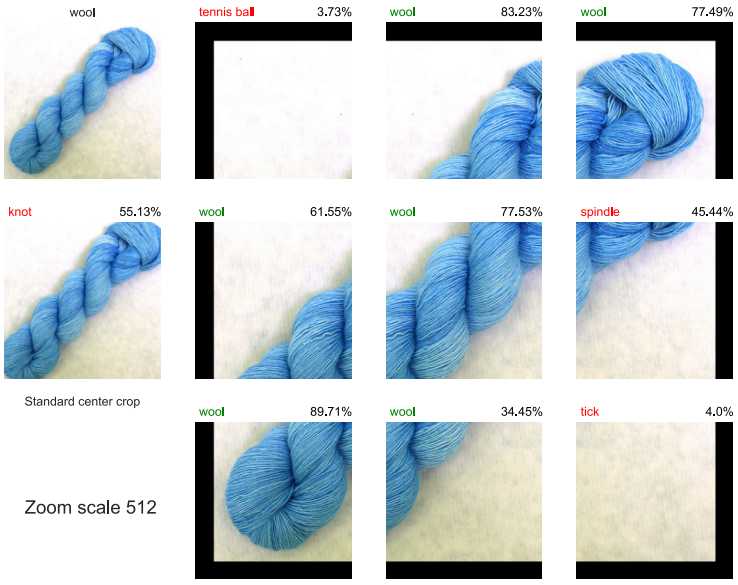
A.5 Zero-shot classification using CLIP

For CLIP, we follow the standard zero-shot classification. This involves creating a text template for each class in the dataset, which contains a generic description of an image featuring an object from that class. Then, we use CLIP's text encoder to obtain embeddings for these templates and then average them to obtain a final vector that represents the class. To classify an image, we calculate the cosine similarity between its embedding and the text vectors for each class and then select the class with the highest value.

A.6 Zoom-based transform



(a)



(b)

Figure A2: (a) Making a 3-by-3 uniform grid out of the image. We pick the center point in each region as the anchor. (b) Sample image showing how our zoom transform is applied to an image.

B Additional Results

In this section, we provide additional results for our experiments.

B.1 Zooming out is needed for a small portion of the datasets

In our approach, we leverage the power of both zoom-in and zoom-out transforms, and Tab. 1 results indicate that this combined zooming approach can be effective in classifying images from diverse datasets. Zooming in enhances texture patterns while zooming out provides a better perspective of the object's shape. The question we aim to answer is which dataset and model pairs require which type of zoom, and whether zooming is always necessary. Additionally, we investigate which types of networks are less reliant on explicit zooming, as they implicitly focus on the main object in the image.

Experiment We separate zoom transforms into three groups and report the maximum possible accuracy as defined in Sec. 3. We use transforms in the minimum set covers (as shown in Fig. A10) for each dataset and classifier pair. We then report the number of images that can only be classified using transforms in each group separately.

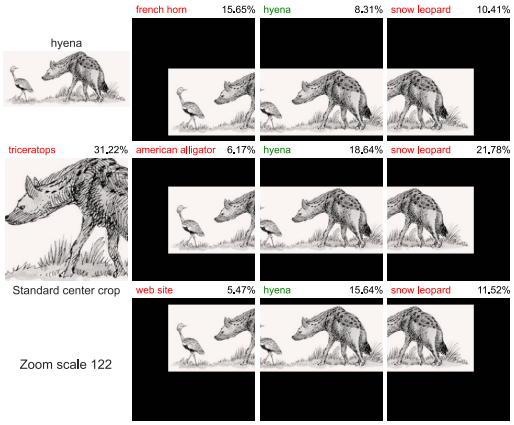


Figure A3: A sample image from the ImageNet-Sketch dataset that can only be solved by zooming out. For this image, with the standard ImageNet transform, the entire body of the animal is not visible. Instead, zooming out of the image helps you see the whole body of the animal. More samples can be found in Appendix D.3.

Results In general, we find that zooming in is more effective than zooming out. Zooming in provides two benefits: (1) it helps the model to focus on the key region where the target object is located, and (2) the model can extract features from the target object at a higher resolution. Across all methods and datasets, we can see a certain percentage of images are only classifiable using transforms of the *zoom-out* group. In particular, for ImageNet-R and ImageNet-Sketch, between $1.2\% - 3\%$ (Table A2) of the entire dataset can only be solved using a transform in the *zoom-out* group. This is especially true for drawings, where the texture may lack distinguishable features, and zooming out allows us to better perceive the shape.

Table A2: Breakdown of maximum possible accuracy by different zoom groups. In each dataset, certain images necessitate a specific zoom group for correct classification regardless of the model being used. However, CLIP performs well overall without depending heavily on a particular zoom level. On average, the percentage of datasets that can only be solved with a specific zoom group is very small for this model.

Dataset	**Model**	***zoom-in* Solve**	***zoom-out* Solves**	***zoom-224* Solves**	**Only *zoom-in* Solves**	**Only *zoom-out* Solves**	**Only *zoom-224* Solves**
ImageNet	ResNet-18	94.57	79.49	81.16	10.59	0.43	0.08
	ResNet-50	96.30	85.84	86.39	7.59	0.40	0.04
	ViT-B/32	96.83	86.18	85.12	7.59	0.30	0.02
	VGG-16	94.60	82.11	83.08	8.92	0.58	0.07
	AlexNet	89.17	62.92	67.98	18.01	0.65	0.18
	CLIP-ViT-L/14	95.82	90.80	87.04	4.81	0.83	0.05
ImageNet ReaL	ResNet-18	97.37	86.10	87.62	7.38	0.27	0.07
	ResNet-50	98.22	91.07	91.87	4.65	0.25	0.04
	ViT-B/32	98.50	90.79	88.06	4.92	0.18	0.03
	VGG-16	97.38	88.43	89.40	6.02	0.38	0.07
	AlexNet	93.15	69.58	74.85	15.47	0.45	0.19
	CLIP-ViT-L/14	98.05	94.44	91.69	3.20	0.55	0.04
ImageNet+ReaL	ResNet-18	97.16	85.51	86.77	7.72	0.28	0.05
	ResNet-50	98.25	91.10	91.77	4.60	0.24	0.03
	ViT-B/32	98.70	91.00	90.95	4.92	0.14	0.02
	VGG-16	97.12	87.88	89.09	6.25	0.42	0.06
	AlexNet	92.79	68.65	73.93	16.25	0.47	0.16
	CLIP-ViT-L/14	98.24	95.09	92.41	2.75	0.47	0.04
ImageNet-A	ResNet-18	63.66	47.95	45.37	13.97	2.75	0.21
	ResNet-50	65.28	52.36	48.59	12.05	3.13	0.22
	ViT-B/32	73.07	56.34	54.84	14.20	2.04	0.27
	VGG-16	56.67	44.95	39.35	11.80	3.85	0.24
	AlexNet	52.69	32.86	31.95	17.15	2.34	0.30
	CLIP-ViT-L/14	98.35	96.71	93.57	1.70	0.69	0.04
ImageNet-R	ResNet-18	57.07	12.19	10.07	40.67	0.92	0.19
	ResNet-50	64.52	12.95	10.36	48.72	1.00	0.23
	ViT-B/32	76.71	18.57	21.92	51.75	0.85	0.15
	VGG-16	56.59	13.15	13.27	38.24	0.93	0.29
	AlexNet	39.91	10.39	9.11	26.27	1.08	0.36
	CLIP-ViT-L/14	97.99	81.32	77.03	12.01	0.44	0.05
ImageNet-Sketch	ResNet-18	41.14	27.06	27.41	11.83	1.77	0.36
	ResNet-50	44.72	32.80	31.45	10.99	2.23	0.24
	ViT-B/32	53.45	37.43	37.38	13.11	1.83	0.36
	VGG-16	36.20	27.20	24.59	9.47	2.97	0.28
	AlexNet	27.71	13.84	15.11	11.26	1.22	0.33
	CLIP-ViT-L/14	86.20	80.67	73.94	6.64	2.38	0.12
ObjectNet	ResNet-18	68.98	38.52	37.23	25.76	1.93	0.25
	ResNet-50	74.16	51.56	47.79	19.68	2.16	0.30
	ViT-B/32	77.66	44.49	42.65	27.43	1.34	0.20
	VGG-16	69.19	41.72	39.49	23.34	2.27	0.31
	AlexNet	56.76	23.45	22.59	28.85	2.27	0.33
	CLIP-ViT-L/14	91.28	82.22	77.60	8.37	1.38	0.15
Average	ResNet-18	74.28	53.83	53.66	16.85	1.19	0.17
	ResNet-50	77.35	59.67	58.32	15.47	1.34	0.16
	ViT-B/32	82.13	60.69	60.13	17.70	0.95	0.15
	VGG-16	72.54	55.06	54.04	14.86	1.63	0.19
	AlexNet	64.60	40.24	42.22	19.04	1.21	0.26
	CLIP-ViT-L/14	95.13	88.75	84.75	5.64	0.95	0.07

B.2 Anchor-based analysis of Center bias in ImageNet and OOD datasets

91.58	93.49	91.74
93.33	95.21	93.46
91.87	93.71	92.03

(a) ImageNet-ReaL

19.35	22.21	19.21
23.87	39.09	23.41
20.20	23.21	19.88

(b) ImageNet-A

54.42	57.41	54.66
56.70	60.14	56.89
53.78	56.52	53.84

(c) ImageNet-R

34.47	37.05	34.62
36.29	39.34	36.43
33.71	36.04	33.77

(d) ImageNet-Sketch

34.15	35.87	34.34
53.14	62.66	53.39
35.11	36.66	35.06

(e) ObjectNet

Figure A4: ResNet-18

94.53	95.82	94.82
95.58	96.77	95.91
94.65	95.92	94.94

(a) ImageNet-ReaL

21.17	26.77	21.59
27.57	46.49	26.57
22.52	27.61	22.31

(b) ImageNet-A

57.55	60.28	57.59
59.49	62.52	59.62
57.09	59.60	57.19

(c) ImageNet-R

38.88	41.08	39.01
40.57	42.92	40.71
38.29	40.38	38.37

(d) ImageNet-Sketch

46.81	47.96	46.85
62.25	69.30	62.53
48.42	49.84	48.55

(e) ObjectNet

Figure A5: ResNet-50

94.45	95.55	94.14
95.61	96.86	95.41
94.70	95.80	94.26

(a) ImageNet-ReaL

32.72	38.48	33.21
40.28	59.92	39.49
34.51	40.13	33.13

(b) ImageNet-A

63.91	66.87	64.06
65.79	69.45	65.96
62.81	65.31	62.82

(c) ImageNet-R

46.29	48.88	46.29
47.86	50.84	47.80
44.83	46.91	44.73

(d) ImageNet-Sketch

40.16	41.54	40.40
59.80	69.52	59.93
41.95	43.88	42.02

(e) ObjectNet

Figure A6: ViT-B/32

92.77	94.20	92.72
94.13	95.52	94.02
92.83	94.41	93.04

(a) ImageNet-ReaL

21.83	26.35	21.95
26.97	39.36	26.19
22.03	27.37	21.99

(b) ImageNet-A

50.04	51.98	49.86
51.76	54.28	51.71
49.40	51.35	49.31

(c) ImageNet-R

32.43	34.25	32.13
33.92	35.91	33.63
31.60	33.30	31.35

(d) ImageNet-Sketch

37.42	38.70	37.42
54.41	62.45	54.66
37.74	39.07	37.86

(e) ObjectNet

Figure A7: VGG16

81.72	85.25	81.62
84.88	88.78	84.70
81.91	85.35	81.76

(a) ImageNet-ReaL

15.93	17.24	15.00
19.01	25.51	17.51
16.79	18.89	16.20

(b) ImageNet-A

40.39	43.67	40.20
43.05	47.22	42.97
39.67	42.88	39.58

(c) ImageNet-R

20.64	23.07	20.64
22.26	25.14	22.20
19.73	21.88	19.70

(d) ImageNet-Sketch

21.42	22.49	21.42
38.20	48.16	38.35
21.15	22.42	21.08

(e) ObjectNet

Figure A8: AlexNet

92.42	93.67	92.76
93.64	94.44	93.70
92.59	93.86	92.62

(a) ImageNet-ReaL

77.31	85.04	77.53
85.41	92.31	84.65
77.60	84.49	77.29

(b) ImageNet-A

94.57	95.92	94.64
95.87	97.07	95.80
94.15	95.42	94.26

(c) ImageNet-R

74.57	77.28	74.65
77.08	79.43	77.19
73.61	76.24	73.62

(d) ImageNet-Sketch

66.81	70.03	67.88
82.06	86.97	81.87
68.32	71.38	68.46

(e) ObjectNet

Figure A9: CLIP-ViT-L/14

B.3 Distribution of the minimum set cover per classifier and dataset

In this section, we provide details on the distribution of minimum set cover size.

Table A3: Distribution of the minimum set cover per classifier and dataset. (ZI: *zoom-in*, ZO: *zoom-out*, ZL: *zoom-224*)

	ReaL				IN-A				IN-R				IN-Sketch				ON			
	ZI	ZO	ZL	Total	ZI	ZO	ZL	Total	ZI	ZO	ZL	Total	ZI	ZO	ZL	Total	ZI	ZO	ZL	Total
ResNet-18	160	33	8	201	174	31	6	211	204	65	9	278	209	51	9	269	191	54	9	254
ResNet-50	136	33	9	178	165	42	7	214	200	62	9	271	216	56	9	281	187	63	9	259
ViT-B/32	134	30	4	168	167	19	7	193	196	52	9	257	218	46	9	273	206	58	9	273
VGG-16	158	34	9	201	181	33	8	222	214	66	9	289	210	54	9	273	198	52	9	259
AlexNet	191	40	8	239	170	33	9	212	212	51	9	272	217	49	9	275	201	58	9	268
CLIP-ViT-L/14	141	48	8	197	75	14	4	93	76	33	5	114	142	61	9	212	205	66	9	280

B.4 Only 70% of all transforms are needed to reach maximum possible accuracy

In Sec. 4.1, we first pre-define all 324 zoom transforms and then compute the *maximum* possible accuracy to ensure the predicted labels were the results of models looking at a controlled zoomed region (*i.e.* not because a model was given 324 arbitrary trials per image). Here, we aim to compute the minimum number of zoom settings required for a model to reach the same upper-bound accuracy. Evaluating this minimum set may reveal spatial biases of a dataset (Sec. 4.2) as well as the implicit zoom operation that a state-of-the-art model (*e.g.* CLIP) may have learned.

Experiment Given a (dataset, classifier) pair, each zoom transform among the 324 will result in a set of correctly classified images. We employ a greedy minimum-set cover algorithm [61, 35] to find a minimum subset of transforms that lead to the correct prediction for all classifiable images in Sec. 4 (*i.e.* those that make up the accuracy scores in Tab. 1c).

For each dataset and classifier pair, we construct a bipartite graph, consisting of transforms and images as distinct groups of nodes. We connect a node from the transform group to an image node, if that transform leads to the correct classification of that particular image. Finding a minimum set cover in this graph is the same as finding the aforementioned subset of transforms. During each iteration of the greedy minimum set cover algorithm, the transform that yields the highest number of correct classifications for the remaining images is selected. This process continues until all of the images have been "covered", *i.e.* all images have connected to a transform with at least one edge. The result is a subset of transforms that can classify images without sacrificing accuracy.

Results Fig. A10 shows the minimum number of transforms per dataset required to reach the maximum possible accuracy. Although this number varies depending on the dataset and classifier, on average, the size of the minimum cover is 229, which is ∼70% of all 324 pre-defined transforms. We evaluate the maximum possible accuracy using the top 36 transforms, the same number as the number of zoom scales and report the results in Tab. 1b. This set of transforms is achieved by stopping the algorithm after 36 iterations, which provided us with 36 high-performing transforms. The maximum possible accuracy using only 36 crops is only slightly lower than that when using all 324 crops but is substantially higher than the standard 1-crop, *e.g.* 85.19% vs. 56.16% for AlexNet on IN (Tab. 1b). Also, the upper-bound accuracy for 36 crops being much higher than the random baseline (*i.e.* 3.6% for IN) confirms that the pre-defined zoom transforms are important to classification (not because models are given 36 random trials per image). The top-36 zoom transforms for ResNet-50 on ImageNet contain zooms at various locations in the image (see the visualizations in Appendix D.1).

Figure A10: The minimum number of zoom transforms (out of 324) required to achieve the maximum possible accuracy scores reported in Tab. 1c.

	IN	ReaL	IN+ReaL	IN-A	IN-R	IN-S	ON	μ
AlexNet	255	239	246	212	272	275	268	252
VGG-16	242	201	201	222	289	273	259	241
ResNet-18	250	201	208	211	278	269	254	239
ResNet-50	234	178	183	214	271	281	259	231
ViT-B/32	233	168	173	193	257	273	273	224
CLIP-ViT-L/14	251	197	186	93	114	280	212	190

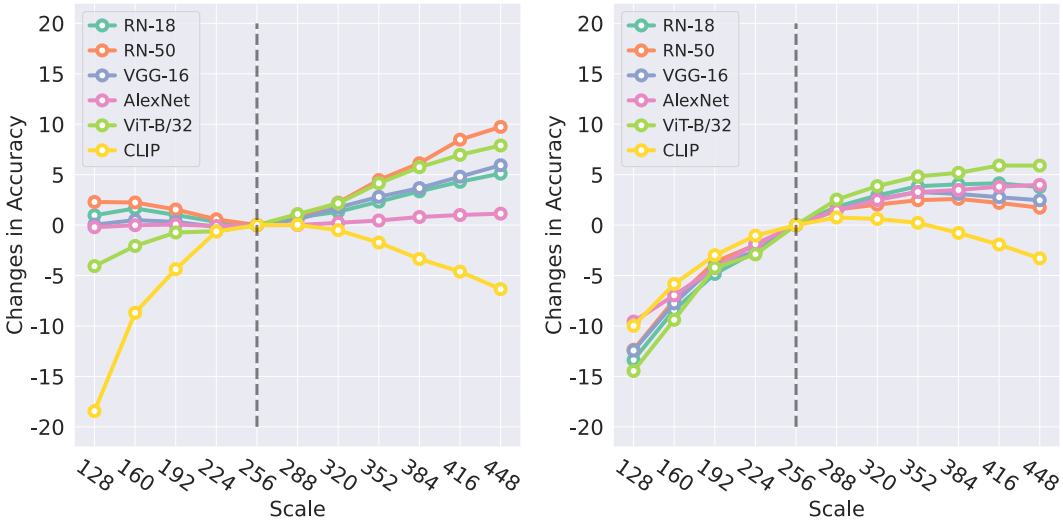
Remarkably, CLIP requires 190 transforms on average, which is fewer than every other model (Fig. A10; μ column). This can be attributed to either the implicit zoom power of CLIP or the fact it has a stronger feature extractor.

B.5 Center-zooming increases the accuracy of all ImageNet-trained models but not CLIP

Previously, we have found that CLIP obtains the best accuracy on all six datasets (Tab. 1a) and also requires the smallest minimum set of zoom transforms to obtain the upper-bound accuracy (Appendix B.4). It is important to understand what classification strategy a CLIP classifier internally performs to classify better. Here, we test the hypothesis that the state-of-the-art CLIP is already performing an implicit zoom on images. If that is true, directly zooming to the center, exploiting the strong center bias of ImageNet-A and ObjectNet, will not improve CLIP accuracy.

Experiment We evaluate the accuracy of all models when center-zooming on IN-A and ON images at 11 different scales $S \in \{128, 160, 192, ..., 448\}$ (Fig. A11). That is, center-zooming at S first resizes the input image so that the smaller dimension becomes S and then takes a 224×224 center crop (zero-padding is applied when necessary).

Results In Fig. A11, we show the changes in the top-1 accuracy (1-crop) when varying the center-zoom scales away from the default ImageNet transform scale ($S = 256$) for both ImageNet-A and ObjectNet. While IN-trained networks exhibit consistent improvement as the zoom scale increases, CLIP shows a monotonic decrease in performance (Fig. A11; yellow curves decreasing on both sides of $S = 256$). This result is surprising but consistent with our hypothesis that CLIP internally performs implicit zooming to reach its peak accuracy and therefore manually zooming (either in or out) at the center mostly ruins its performance.



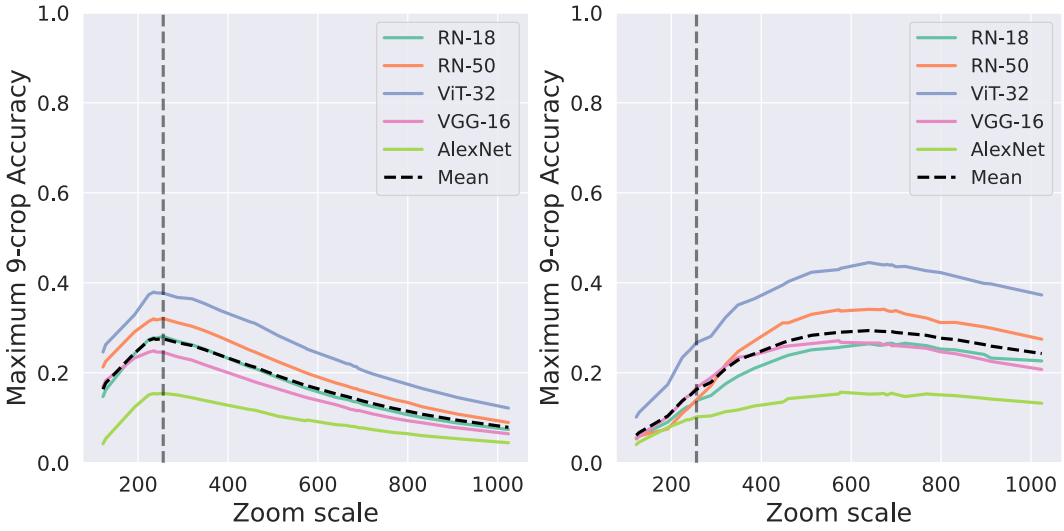
(a) ImageNet-A (b) ObjectNet

Figure A11: Absolute changes in the top-1 accuracy (%) of 6 models on ImageNet-A (a) and ObjectNet (b) when center-zooming images at various scales. Interestingly, center-zooming helps IN-trained networks but hurts CLIP.

B.6 Zoom-in is more useful than zoom-out, which is most important to abstract images

Zooming in enhances texture patterns while zooming out provides a better perspective of the object's shape, which is known to be useful to image classification [12, 19]. Results in Sec. 4.1 and Appendix B.4 indicate that this combined zooming approach can be effective in classifying images from diverse datasets. Here, we test which dataset and model pairs require which type of zoom, and whether zooming in or out is always necessary.

Experiment To better understand the effectiveness of each zoom group, we calculate the maximum possible accuracy using all nine locations and different zoom scales S to show per-dataset trends. Additionally, we examined the percentage of images within each dataset that required a specific zoom group to be accurately classified. This analysis allowed us to gain a more comprehensive understanding of the role that each zoom group played in reaching the maximum possible accuracy reported in Tab. 1.



(a) ImageNet-Sketch (b) ImageNet-A

Figure A12: Maximum possible accuracy using nine crops at varying scales. The vertical line represents the standard ImageNet zoom scale ($S = 256$). While for ImageNet-Sketch (a), zooming out marginally improves the accuracy, for scale factors larger than 256, ImageNet-A (b) exhibits an increase in accuracy. See Appendix B.9 for details.

Results The maximum possible accuracy for different zoom scales reveals a clear trend for each dataset. For instance, a slight zoom-**out** enhances accuracy for abstract image datasets like IN-Sketch (Fig. A12a). Conversely, for adversarial image datasets such as IN-A, zooming **in** improves accuracy (Fig. A12b) This pattern is also evident in evaluations using standard 1-crop accuracy (Appendix B.9). Furthermore, the percentage of images that are *exclusively* classifiable with the *zoom-in* group is consistently higher than the other two groups, *i.e.* using ViT-B/32 51.75% on IN-A, and 13.11% on IN-S (Tab. A4a). This shows that most datasets necessitate focusing on the object of interest in the image to both see texture patterns better and reduce background clutter (see Tab. A2 for full results). However, we also find that the *zoom-out* group is also necessary for the correct classification of a small portion of each dataset. For instance, $1.22\% - 2.97\%$ of IN-S images (Tab. A4b) require a *zoom-out* transform to be correctly labeled (*i.e. zoom-in* does not help at all).

Table A4: % of images in the entire dataset that require a particular zoom group to be classified correctly. See Tab. A2 for full results.

	zoom-in (a)		***zoom-out*** (b)		***zoom-224*** (d)	
	IN-A	IN-S	IN-A	IN-S	IN-A	IN-S
ResNet-18	40.67	11.83	0.92	1.77	0.19	0.36
ResNet-50	48.72	10.99	1.00	2.23	0.23	0.24
ViT-B/32	51.75	13.11	0.85	1.83	0.15	0.36
VGG-16	38.24	9.47	0.93	2.97	0.29	0.28
AlexNet	26.27	11.26	1.08	1.22	0.36	0.33
CLIP-ViT-L/14	12.01	6.64	0.44	2.38	0.05	0.12

B.7 Simple aggregation of the zoom transforms can improve accuracy on some datasets but not all

Sec. 4.1 and Appendix B.5 show that using the same feature extractors (even as old as AlexNet), it is possible to achieve higher image classification accuracy if we know where to zoom and at which scale. A practical follow-up question is: How to build a classifier that knows how to zoom given a test image? In this section, we establish simple baselines that aggregate predictions over a set of zoom transforms.

Experiment We employ the mean method from prior work [58, 45], and the max method to aggregate output marginal distributions. For a given image, we get N output distributions over classes from a classifier, in which N is the total number of used transforms. The aggregation process combines these N distributions and outputs a final prediction for the given image. In the aggregation step, we use the mean or max method to infer the final confidence for each class along N distributions. Finally, we select the class that has the highest confidence score. Additionally, we test 5-crop and 10-crop evaluation [38, 60, 23] and compare them with our methods. We use the transforms in the minimum set found for IN-ReaL to evaluate the remaining datasets. The purpose is to reduce the number of augmentations and prevent training on OOD benchmarks.

Results max aggregation of zoom-in transforms results in the largest improvements on ImageNet-A. That is, on IN-A, ViT-B/32 reaches a top-1 accuracy of 24.69% (+15.05) (Tabs. A5 and A6) and a ResNet-50 accuracy increases by +13.03 points from 16.62% to 29.65% (Appendix C.3)–a surprisingly strong baseline for future studies. On ObjectNet, max aggregation of zoom-in transforms also yields +1.99 improvement over the 1-crop ViT-B/32 baseline.

On the other hand, mean aggregation results in smaller but more consistent improvements over the 1-crop baseline for many datasets (+3.56 on IN, +4.08 on ReaL, +4.65 on IN-A, and +3.03 on ON; Tab. A5). mean aggregation (Tab. A5b) also outperforms the standard 5-crop and 10-crop [38, 23] aggregation on these four datasets (Tab. A5e–f).

In contrast, for all 6 datasets, aggregating zoom-out and *zoom-224* transforms consistently worsen the performance over the 1-crop baseline (Tab. A5c–d). That is, we find that for a few dozen images (*e.g.* sketches and abstract visuals; Fig. 1ac), interestingly, only zooming out can lead to a correct classification (Appendix B.6), yet for most images in these 6 benchmarks, zooming out hurts the accuracy.

In summary, based on the insights from Sec. 4.1, showing that zooming could help classification, we find that simple methods for aggregating zoom-in transforms at test-time can directly improve model accuracy over the 1-crop and *zoom-224* baselines on four benchmarks, *i.e.* all except IN-R and IN-S, which contain abstract images.

Table A5: Top-1 accuracy (%) of aggregation methods on an IN-trained ViT-B/32 model. Compared to the 1-crop baseline, aggregating zoom-in transforms consistently yields improved accuracy on IN-A, ON but worse accuracy on IN-R and IN-S. *zoom-224* refers to the set of zoom transforms at $S = 224$. See Tab. A6 for more results.

	(a)	(b) *zoom-in*		(c) *zoom-out*		(d) *zoom-224*		(e) 5-crop		(f) 10-crop [38]	
Dataset	1-crop	max	mean	max	mean	max	mean	max	mean	max	mean
IN	75.75	74.35 (-1.40)	**79.31** (+3.56)	71.48	69.47	72.66	73.67	77.33	77.73	77.30	77.87
ReaL	81.89	80.22 (-1.67)	**85.97** (+4.08)	77.95	76.28	79.25	80.31	83.24	83.80	83.17	83.87
IN-A	9.64	**24.69** (+15.05)	14.29 (+4.65)	7.79	5.48	8.12	7.39	12.19	9.88	12.32	9.67
IN-R	41.29	39.90 (-1.39)	40.06 (-1.23)	39.05	36.21	39.52	39.28	43.90	43.17	**44.31**	43.28
IN-S	26.83	19.74 (-7.09)	20.89 (-5.94)	22.37	19.25	25.06	25.21	28.72	28.66	**28.94**	28.76
ON	30.89	32.88 (+1.99)	**33.92** (+3.03)	22.56	19.51	22.75	22.72	26.96	24.98	27.14	24.97

Table A6: Performance of various aggregating methods (%) – The bold numbers show maximum accuracy per model/dataset. CLIP strongly and consistently favors 10-crop over other settings.

		(a)	(b) *zoom-in*		(c) *zoom-out*		(d) *zoom-224*		(e) 5-crop		(f) 10-crop [38]	
	Dataset	**1-crop**	***Max***	***Mean***	***Max***	***Mean***	***Max***	***Mean***	***Max***	***Mean***	***Max***	***Mean***
ResNet-18	**IN**	69.45	68.45 (-1.00)	71.45 (+2.00)	60.33	56.79	67.85	68.70	70.61	71.32	70.83	**71.85**
	ReaL	76.94	76.33 (-0.61)	**79.94** (+3.00)	67.64	63.92	75.73	76.74	78.26	79.01	78.42	79.46
	IN-A	1.37	**11.68** (+10.31)	5.48 (+4.11)	2.44	2.19	3.41	2.69	3.16	2.13	3.28	1.87
	IN-R	32.14	30.60 (-1.54)	28.95 (-3.19)	29.08	27.28	32.29	32.54	33.99	33.38	**34.59**	33.83
	IN-S	19.41	14.86 (-4.55)	14.34 (-5.07)	14.48	11.49	17.80	17.83	20.83	20.70	**21.39**	21.06
	ON	27.59	**28.21** (+0.62)	25.92 (-1.67)	16.11	14.10	22.82	22.86	24.77	20.91	25.47	21.03
ResNet-50	**IN**	75.75	73.24 (-2.51)	77.30 (+1.55)	69.06	66.42	74.45	75.39	76.67	77.13	76.89	**77.43**
	ReaL	82.63	80.36 (-2.27)	**84.68** (+2.05)	76.35	73.85	81.96	82.85	83.67	84.06	83.82	84.31
	IN-A	0.21	**16.11** (+15.9)	6.23 (+6.02)	2.79	2.19	3.04	2.11	2.28	0.95	2.43	1.00
	IN-R	35.39	33.58 (-1.81)	32.73 (-2.66)	35.85	33.22	36.64	36.44	37.47	36.50	**38.23**	36.86
	IN-S	22.91	16.89 (-6.02)	17.80 (-5.11)	19.51	17.12	21.60	21.66	24.71	24.51	**24.94**	24.74
	ON	**36.18**	34.56 (-1.62)	34.22 (-1.96)	27.10	25.32	31.78	31.98	33.34	29.58	33.93	29.86
ViT-B/32	**IN**	75.75	74.35 (-1.40)	**79.31** (+3.56)	71.48	69.47	72.66	73.67	77.33	77.73	77.30	77.87
	ReaL	81.89	80.22 (-1.67)	**85.97** (+4.08)	77.95	76.28	79.25	80.31	83.24	83.80	83.17	83.87
	IN-A	9.64	**24.69** (+15.05)	14.29 (+4.65)	7.79	5.48	8.12	7.39	12.19	9.88	12.32	9.67
	IN-R	41.29	39.90 (-1.39)	40.06 (-1.23)	39.05	36.21	39.52	39.28	43.90	43.17	**44.31**	43.28
	IN-S	26.83	19.74 (-7.09)	20.89 (-5.94)	22.37	19.25	25.06	25.21	28.72	28.66	**28.94**	28.76
	ON	30.89	32.88 (+1.99)	**33.92** (+3.03)	22.56	19.51	22.75	22.72	26.96	24.98	27.14	24.97
VGG-16	**IN**	71.37	69.60 (-1.77)	72.46 (+1.09)	64.75	59.95	69.51	70.48	72.31	73.09	72.67	**73.53**
	ReaL	78.90	77.23 (-1.67)	80.59 (+1.69)	72.55	67.68	77.48	78.58	79.80	80.42	80.13	**80.80**
	IN-A	2.69	**11.55** (+8.86)	6.24 (+3.55)	3.33	2.77	4.69	3.87	4.87	3.19	5.09	3.19
	IN-R	26.98	26.18 (-0.80)	24.74 (-2.24)	28.01	25.62	27.76	27.78	28.75	27.95	**29.23**	28.35
	IN-S	16.78	13.30 (-3.48)	13.05 (-3.73)	15.18	13.37	15.82	15.97	17.80	17.63	**18.28**	17.92
	ON	**28.32**	26.96 (-1.36)	26.15 (-2.17)	19.88	16.42	23.47	23.60	26.21	21.65	26.52	21.80
AlexNet	**IN**	56.16	54.74 (-1.42)	56.98 (+0.82)	40.78	27.09	51.80	51.50	57.86	58.60	58.26	**59.11**
	ReaL	62.67	61.46 (-1.21)	64.35 (+1.68)	45.84	30.58	58.25	58.16	64.53	65.39	64.98	**65.94**
	IN-A	1.75	**4.65** (+2.90)	3.27 (+1.52)	1.56	1.23	2.31	1.97	2.53	2.04	2.64	2.03
	IN-R	21.10	20.65 (-0.45)	17.97 (-3.13)	15.72	11.25	19.91	19.55	22.79	21.86	**23.26**	22.16
	IN-S	10.05	7.94 (-2.11)	6.54 (-3.51)	5.82	2.72	8.29	7.39	10.84	10.65	**11.20**	10.80
	ON	14.23	**14.91** (+0.68)	11.80 (-2.43)	6.11	3.75	9.65	9.01	12.63	9.57	12.84	9.58
CLIP-ViT-L/14	**IN**	75.03	70.01 (-5.02)	74.45 (-0.58)	72.01	72.21	74.45	76.04	76.77	76.91	76.72	**77.00**
	ReaL	80.68	76.37 (-4.31)	81.31 (+0.63)	78.28	78.93	81.45	82.05	82.26	**82.55**	82.26	**82.55**
	IN-A	71.28	76.57 (+5.29)	68.16 (-3.12)	60.71	49.51	71.69	70.04	77.80	76.61	**78.25**	76.83
	IN-R	87.74	84.12 (-3.62)	83.54 (-4.20)	86.84	86.29	88.12	88.24	89.64	89.66	**90.01**	89.94
	IN-S	58.23	51.88 (-6.35)	56.06 (-2.17)	57.14	57.43	59.00	59.90	61.28	61.61	61.59	**62.07**
	ON	66.32	60.20 (-6.12)	58.10 (-8.22)	56.57	58.11	62.44	62.65	66.70	64.88	**66.87**	64.97

B.8 Runtime analysis of MEMO

Another benefit of `RRC` compared to AugMix is faster inference time. Table A7 shows the runtime analysis of MEMO. Typically, TTA methods suffer from slow runtime due to augmentation and test-time training processes. We find that MEMO + `RRC` consistently leads to an average 1.6× speed-up compared to MEMO + AugMix (Tab. A7; 0.65s / image vs. 1.15s / image), providing more evidence to support this transformation as a viable option for test-time augmentations.

Table A7: Average runtime per query image (in seconds). Using `RandomResizedCrop` in MEMO speed ups the runtime by an average factor of 1.6×.

Runtime (in seconds)	IN	IN-A	IN-R	IN-S	ON
MEMO + AugMix [79]					
ResNet-50 [23]	1.24	1.12	1.12	1.32	1.51
DeepAug+AugMix [26]	1.19	1.07	1.12	1.23	1.55
MoEx+CutMix [40]	1.15	1.16	1.11	1.31	1.53
MEMO + `RRC` (**Ours**)					
ResNet-50 [23]	0.64	0.60	0.65	0.88	1.19
DeepAug+AugMix [26]	0.62	0.62	0.64	0.87	1.18
MoEx+CutMix [40]	0.65	0.62	0.66	0.88	1.19

B.9 1-crop accuracy with different zoom scales

In this section, we demonstrate the performance of various models when zooming in or out of an image. In other words, we utilize the standard 1-crop ImageNet transform while altering the initial scale of the image.

In this section, we are conducting experiments using the following models: AlexNet [38], ConvNext (Base, Large, Small, Tiny) [43], DenseNet-161 [29], EfficientNet-B7 [66], MobileNet (V2, V3 Large) [57, 28], ResNet (50, 101) [23], ResNeXt-50 (32x4d) [76], ShuffleNet V2 x1.0 [46], VGG-19 [60], Vision Transformer (ViT-B/16, ViT-B/32, ViT-L/16, ViT-L/32) [17], and Wide ResNet-50-2 [78].

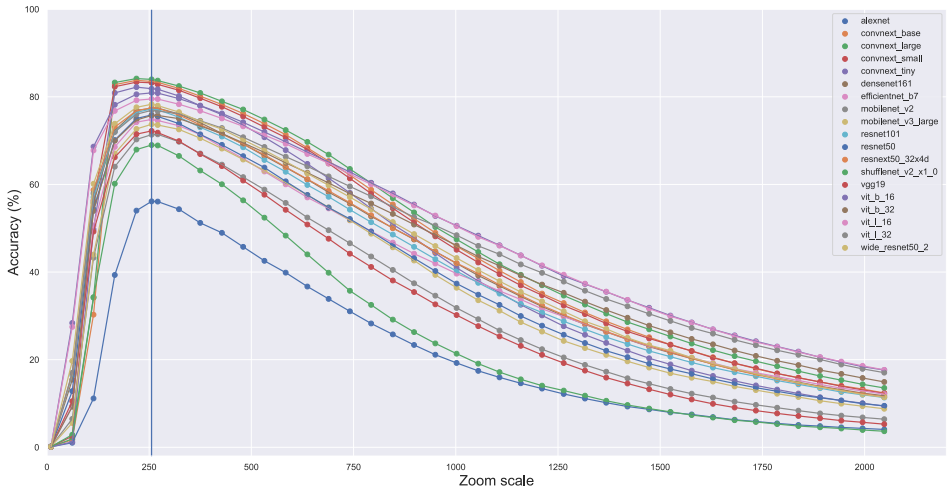


Figure A13: ImageNet accuracy using a 1-crop transform (the vertical line represents the standard ImageNet transform scale factor).

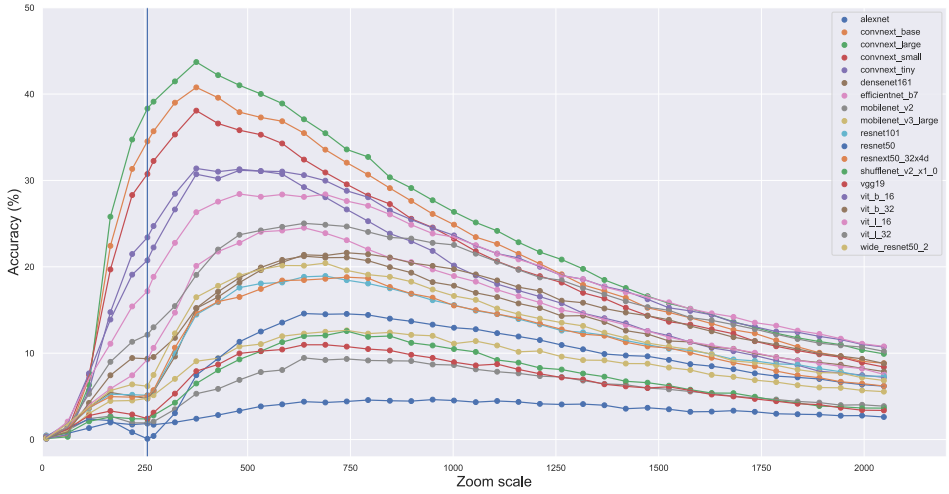


Figure A14: ImageNet-A accuracy using a 1-crop transform (the vertical line represents the standard ImageNet transform scale factor).

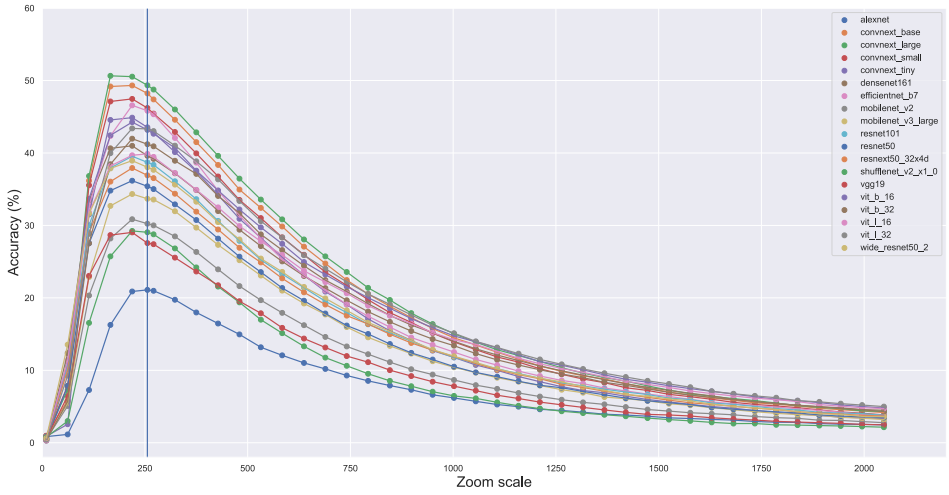


Figure A15: ImageNet-R accuracy using a 1-crop transform (the vertical line represents the standard ImageNet transform scale factor).

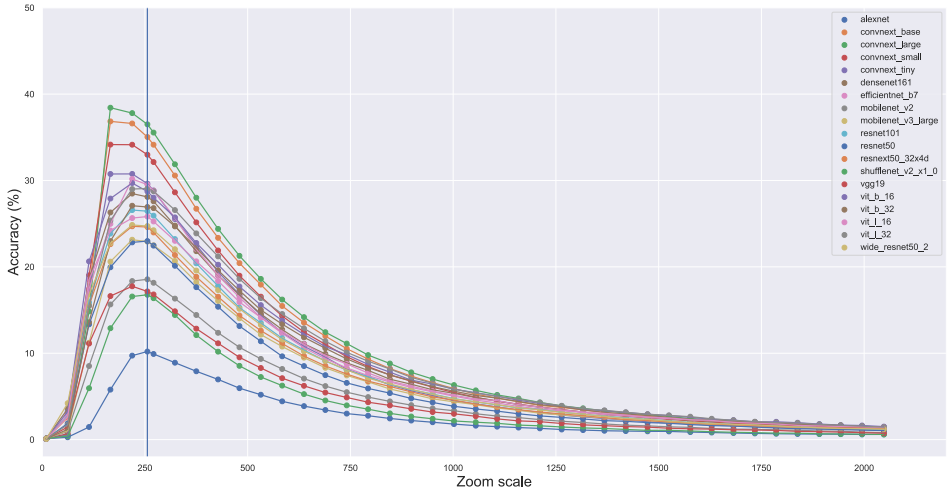


Figure A16: ImageNet-Sketch accuracy using a 1-crop transform (the vertical line represents the standard ImageNet transform scale factor).

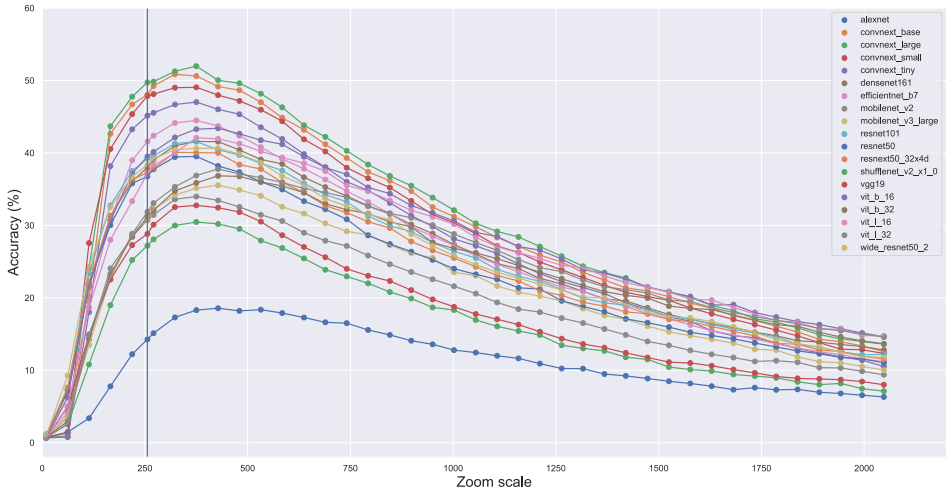


Figure A17: Accuracy using a 1-crop transform on 5K random images of the ObjectNet dataset (the vertical line represents the standard ImageNet transform scale factor).

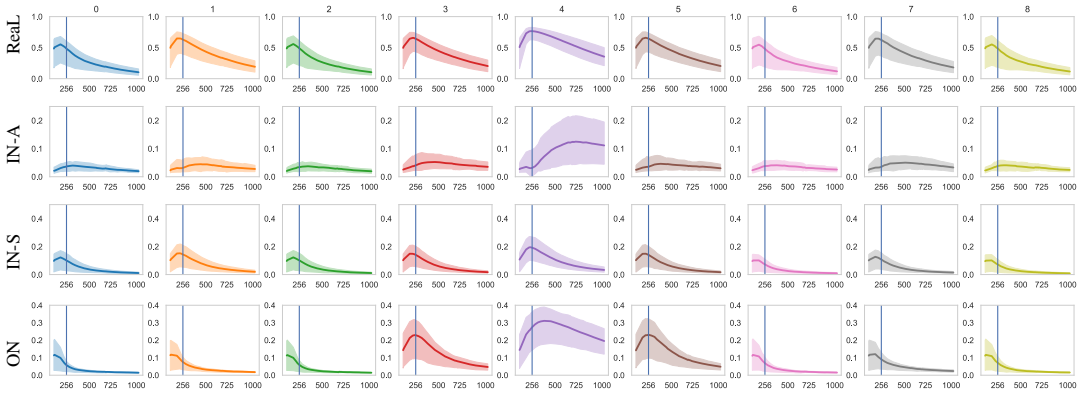


Figure A18: Breakdown of the accuracy of IN-trained models at different crop locations and scale size – Analysis of accuracy across various crop locations and scale sizes reveals that different datasets exhibit distinct optimal conditions. For instance, the IN-A dataset experiences a considerable increase in accuracy when zoomed in, while ImageNet-R yields better results when zoomed out.

B.10 Background occlusion in ImageNet dataset

Sample images for images with and without occlusion.

(a) A sample image of the `Tank` class without occlusion.

(b) Image with heavy background occlusion.

(c) A clean sample image of the `Four Poster` class.

(d) A low-quality image with background occlusion.

Figure A19: Background occlusion examples.

C Additional Experiments

In this section, we provide additional experiments with the proposed zoom-based transform.

C.1 Zooming is similarly important to the foreground and background contents

Background pixels, despite often being neglected in image classification, can contain predictive signals [82, 74, 20, 55]. It has remained largely unknown how much the image context (background) could contribute to the model performance. While Zhu et al. [82] disentangle the predictiveness of background (BG) and foreground (FG) via model training, we directly measure how pretrained models perceive these two signals.

Experiment Using bounding-box annotations provided by Russakovsky et al. [56], we create two dataset variations of ImageNet: *FGSet* and *BGSet*, following Zhu et al. [82]. We mask all the background for *FGSet* as in Fig. A20b, and for *BGSet* we mask all the main objects, as depicted in Fig. A20d & Fig. A20f. After that, we compute the accuracy of these two sets with all tested classifiers using ImageNet and ImageNet-ReaL labels as in Tab. A8.

Results Our results suggest that zooming is important to ImageNet regardless of whether foreground or background features are used, with the difference for *FGSet* and *BGSet* on average being similar (Tab. A8). Additionally, when only the background features were available, almost half of ImageNet images (45.23%) could be correctly classified if optimal Zoom was used. Finally, we found that with only foreground information, ViT-B/32 could achieve a maximum possible accuracy of 95.50% given an optimal zooming method, suggesting that only $98.75\% - 95.50\% = 3.25\%$ of images (Tab. 1) required the background information. These findings suggest that both foreground and background features are important for ImageNet classification, but that an optimal zooming method can considerably improve performance even in the absence of one of these feature sets.

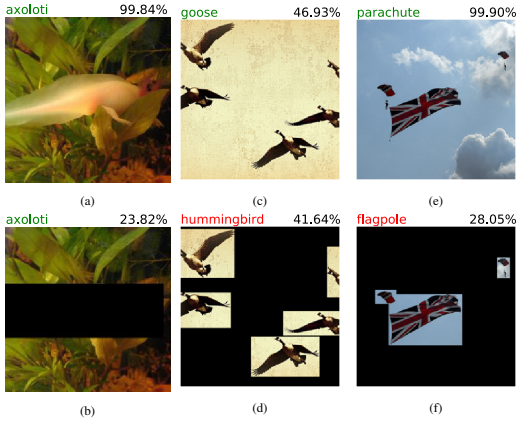


Figure A20: The foreground and the background both contain predictive signals. A ResNet-50 classifier can detect `axolotl` (a), even when the main object is masked (b). Removing the background from images of 'goose' (c) and 'parachute' (e) causes misclassification (d, f).

C.2 Adversarial datasets contain more objects compared to ImageNet

So far, our findings indicate that if we apply the zoom-in operation to the two datasets of ImageNet-A and ObjectNet, the performance of conventional vision models improves consistently up to a certain threshold (Sec. 4 and Appendix B.5). This suggests that the initial images contain distracting elements that impede the model from correctly identifying the object of interest. Both ImageNet-A and ObjectNet are considered out-of-distribution datasets, which are specifically designed to evaluate a vision model's ability to withstand natural adversarial and pose attacks. We hypothesize that the primary reason that these datasets are hard can be attributed to background clutter, multiple objects, and the presence of a positional bias in these images.

Table A8: ImageNet classification from object-only and background-only signals. Numbers show the maximum possible top-1 accuracy (%) using zoom-based transforms for minimum set covers in Appendix B.4. We discover that background signals potentially hold significance for image classification. The bold numbers show the highest possible accuracy per dataset and group.

	1-crop				Max possible using zooming			
	FGSet		BGSet		FGSet		BGSet	
	IN	ReaL	IN	ReaL	IN	ReaL	IN	ReaL
ResNet-18	59.77	64.97	4.91	7.84	89.89	92.04	25.81	31.33
ResNet-50	68.02	72.90	6.18	9.83	93.45	94.89	30.30	35.98
ViT-B/32	67.46	71.78	**9.72**	13.38	94.40	95.50	**39.70**	**45.23**
VGG-16	63.78	69.09	5.36	8.59	91.01	92.91	26.98	32.62
AlexNet	42.38	46.54	3.66	5.46	80.20	83.25	22.02	27.04
CLIP-ViT-L/14	**74.46**	**78.62**	9.49	**13.80**	**96.14**	**97.35**	36.85	42.51
mean	62.65	67.32	6.55	9.82	90.85	92.66	30.28	35.79

Experiment We use OWL-ViT [48], an open vocabulary object detection model, to quantify the number of objects present in three datasets of ImageNet, ImageNet-A, and ObjecNet. The OWL-ViT expects an input image with a set of object names and will determine if any object instances are present in the image. To specify object names, we use LVIS vocabulary [22], which encompasses a comprehensive list of 1203 distinct objects. The OWL-ViT model includes a threshold parameter that reflects its confidence level in its predictions. To assess whether different threshold values would affect our results, we conducted our experiment using both 0.1 and 0.05 as threshold values.
After calculating the distribution of the number of objects in images, we perform a Mann-Whitney U test to determine whether there is a statistically significant difference in this distribution between datasets. As each dataset has a different number of classes, we limited our analysis to shared classes between any two datasets.

Results The results of our study reveal a contrast between ImageNet and ImageNet-A, as well as ImageNet and ObjectNet. This finding implies a dissimilarity between the images in the original ImageNet dataset and its OOD datasets that might arise from the presence of background clutter. Specifically, on average, images in ImageNet-A and ObjectNet datasets tend to feature more objects, which can pose more significant distractions for image classification models.
The results of the Mann-Whitney U test also reflect this finding, the p-value for both thresholds was found to be less than 0.05, which is statistically significant at the 95% confidence level (Tab. A9).

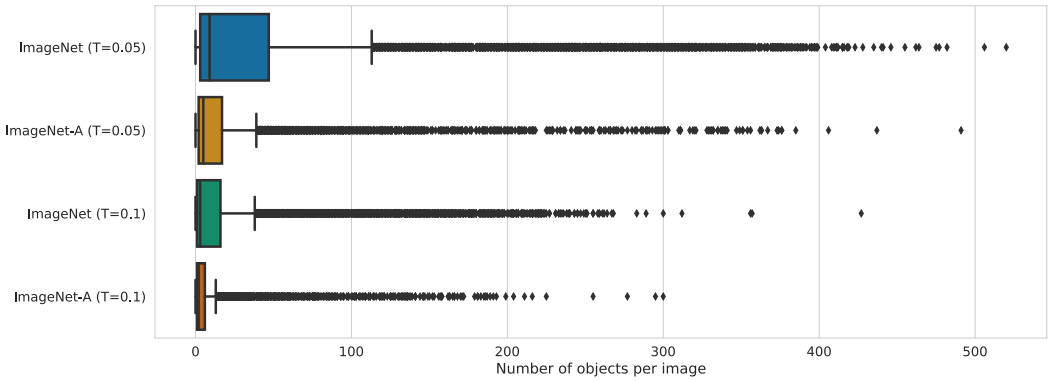


Figure A21: Comparison of the number of objects in two datasets of ImageNet and ImageNet-A using OWL-ViT [48] – T denotes the classification's threshold

C.2.1 p-values for Mann Whitney U test

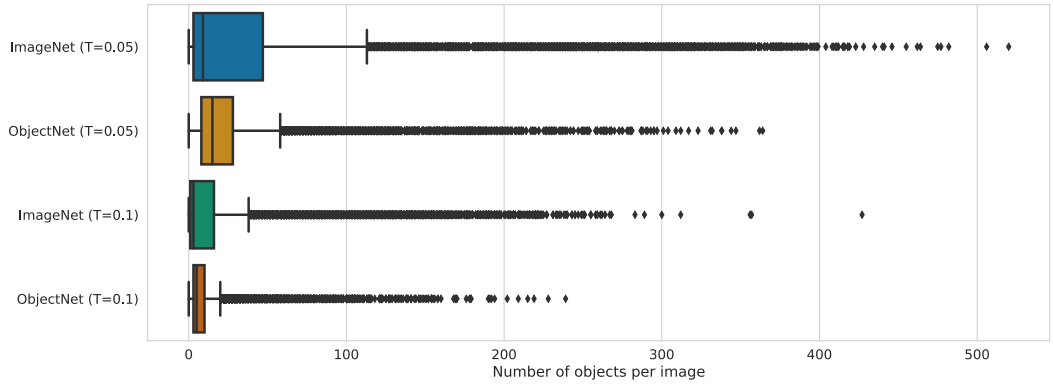


Figure A22: Comparison of the number of objects in two datasets of ImageNet and ObjectNet using OWL-ViT [48] – T denotes the classification's threshold

Table A9: The result of the Mann-Whitney U test to compare ImageNet with ImageNet-A and ObjectNet

	$T = 0.05$	$T = 0.01$
ImageNet-A	6.27E-265	1.71E-235
ObjectNet	1.80E-02	3.66E-02

C.3 Zooming further improves robustified models on ImageNet-A

Intensive data augmentations have been proven to significantly boost CNNs' performance [73, 63] on ImageNet. Motivated by these previous successes and the fact that neural networks trained on diverse augmentations are able to learn robust representations [41], we want to know if robustified pretrained models (*i.e.* trained with intensive augmentations) could reach higher accuracy on ImageNet-A using zooming in.

Experiment We test 4 different ResNet-50 classifier versions that have been trained with different data augmentation procedures. From the the `torchvision` library, we select two sets of model weights; trained with (V2[2]) and without (V1[3]) data augmentations. We also take two other models trained with DeepAugmentation+AugMix [26] and MoEx+CutMix [40]. The second column in Tab. A10 represents the accuracy of models using 1-crop.

Results Zooming in consistently helps ResNet-50 networks, with improvements varying from +13 to +24 points. The best-performing network is `torchvision`-V2 which uses the max aggregator and achieves 29.65%. These results suggest that simple aggregation over the proposed zoom transform is effective for datasets that have dominant center bias.

Table A10: The results of different aggregation functions on four ResNet-50 variants when tested on ImageNet-A (%). Each model has been trained using different training-time augmentation techniques. Improvements values in parentheses are with respect to the 1-crop baseline.

ResNet-50	Baseline	*Max*	*Mean*
`torchvision` V1	0.21	16.11 (+15.90)	6.23
MoEx+CutMix [40]	8.60	24.72 (+16.12)	15.32
DeepAug+AugMix [26]	3.94	27.93 (+23.99)	13.16
`torchvision` V2	16.62	29.65 (+13.03)	22.08

[2] `ResNet50_Weights.IMAGENET1K_V2`
[3] `ResNet50_Weights.IMAGENET1K_V1`

D Visualization

In this section, we provide several visualizations of zooming transforms.

D.1 Visualizations for 36 top performing zoom transforms

Figure A23: Different framing of an image of a `lorikeet` according to 36 high-performing transforms of a ResNet-50 model

Figure A24: Different framing of an image of a `lorikeet` according to 36 high-performing transforms of a ViT/B-32 model

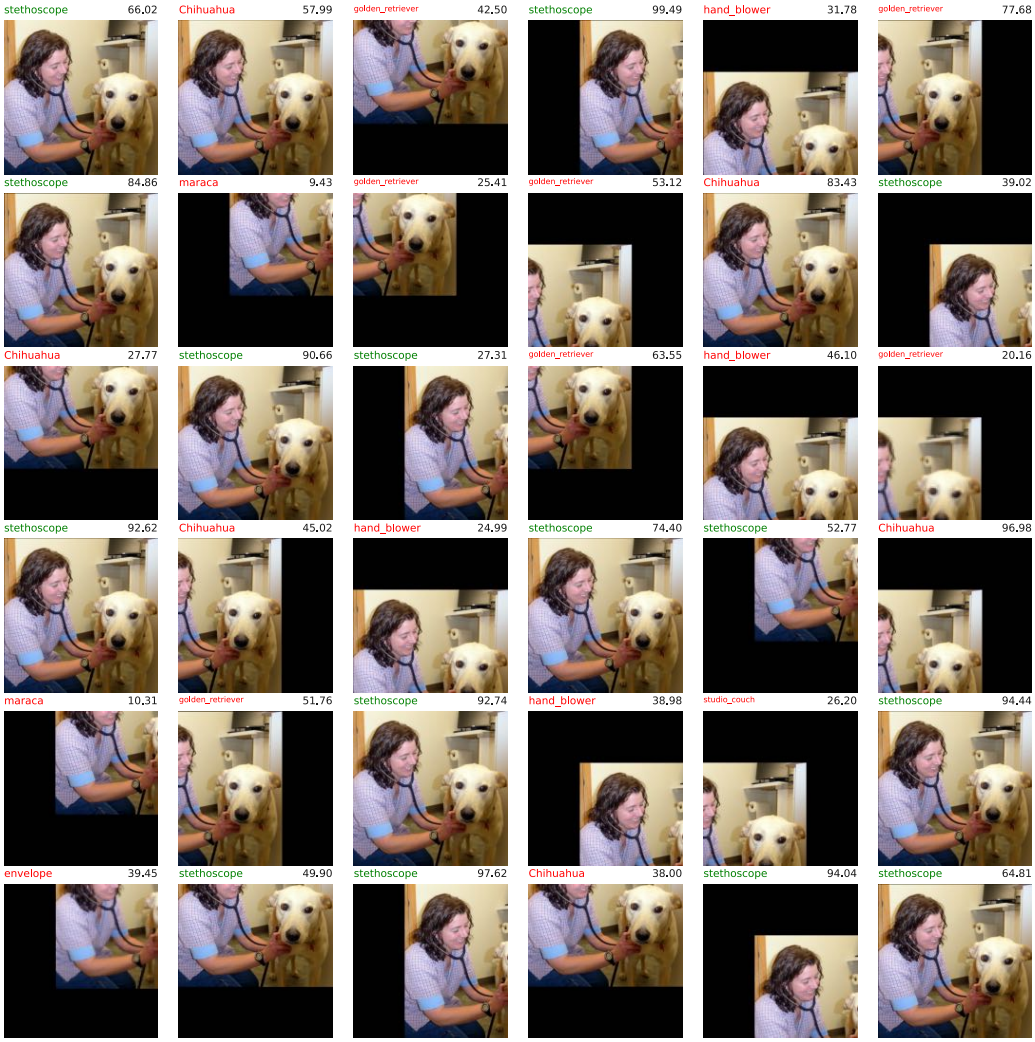


Figure A25: Different framing of an image of a `stethoscope` according to 36 high-performing transforms of a ResNet-50 model

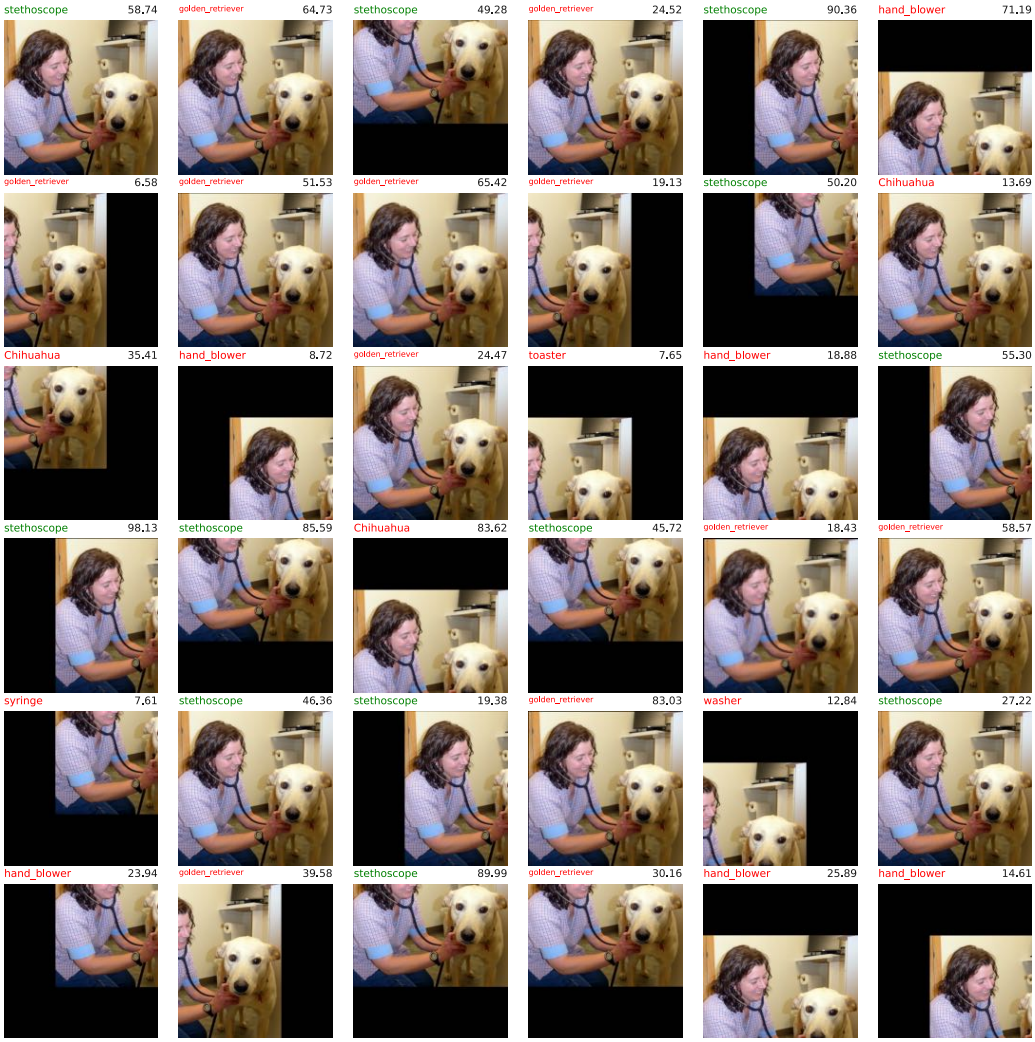


Figure A26: Different framing of an image of a `stethoscope` according to 36 high-performing transforms of a ViT/B-32 model

D.2 Overview of 324 transforms

The visualizations below illustrate the transforms that result in the correct prediction of the query image, using ViT-B/32 [17] and CLIP-ViT-L/14 [52]. Each circle represents a transform, with the initial zoom scale indicated in the accompanying text. The green circles represent the transformations that lead to correct classification, while the red circles indicate incorrect ones.

Figure A27: Visualization of effective transforms that lead to the correct classification of an image containing `scorpion`, using a ViT-B/32 model.

Figure A28: Visualization of effective transforms that lead to the correct classification of an image containing `scorpion`, using a CLIP-ViT-L/14 model.

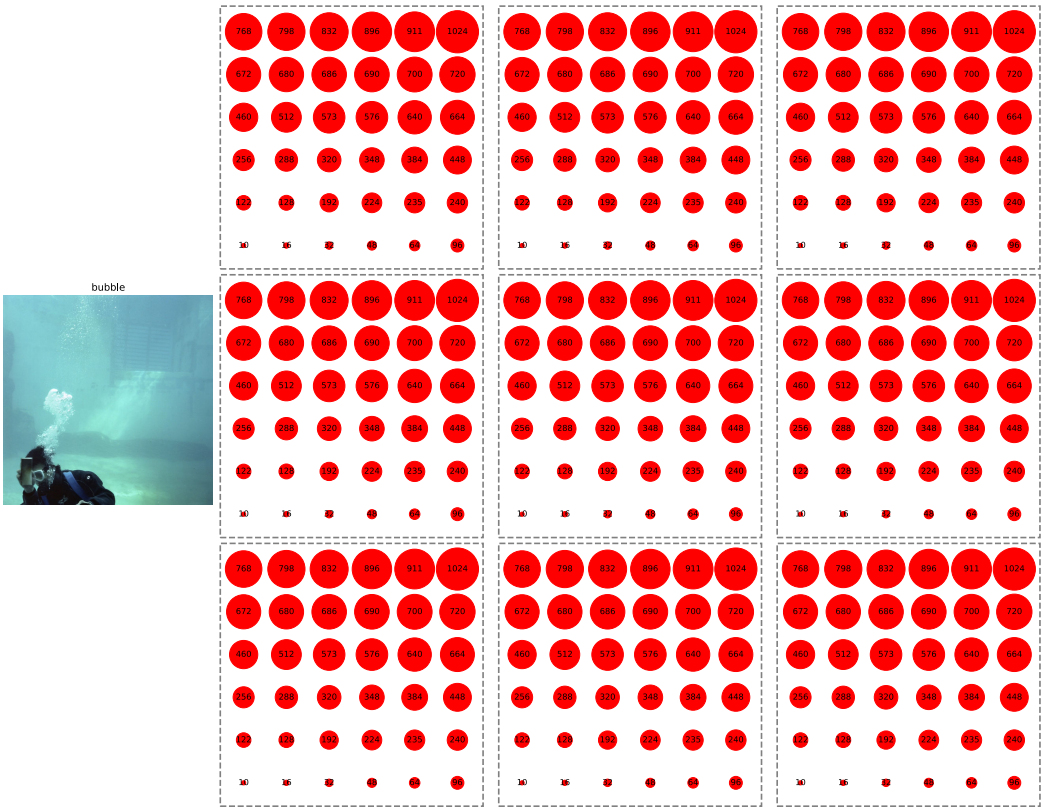


Figure A29: Visualization of effective transforms that lead to the correct classification of an image containing `bubble`, using a ViT-B/32 model.

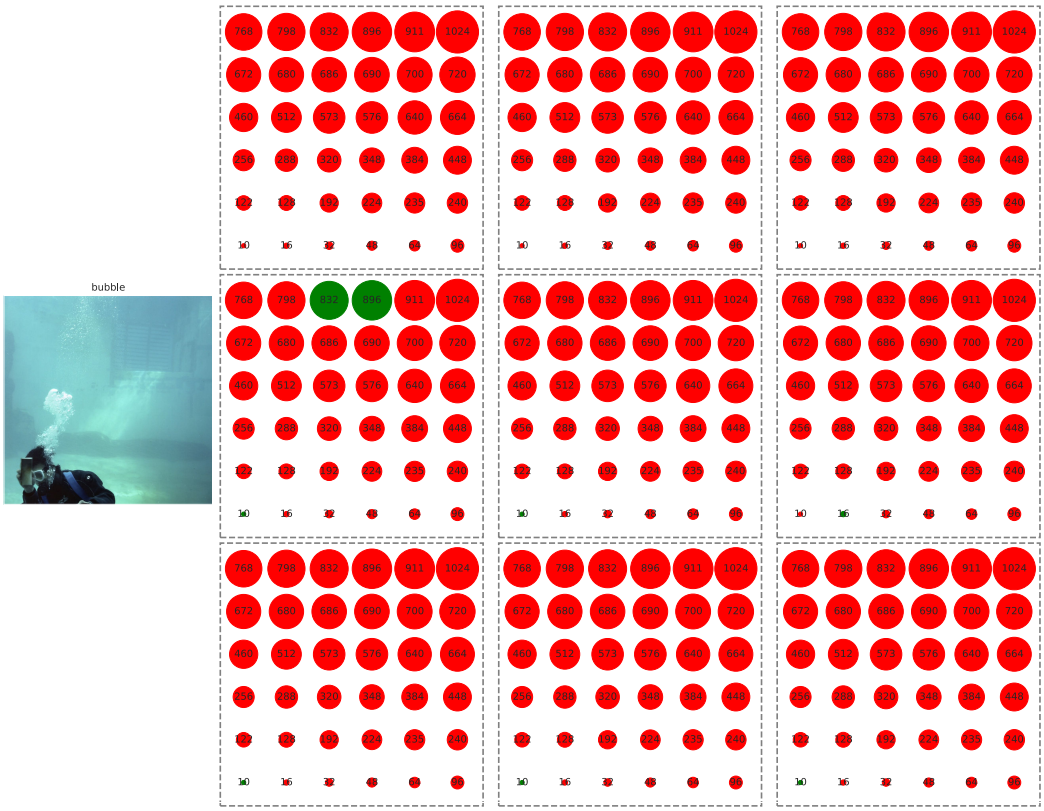


Figure A30: Visualization of effective transforms that lead to the correct classification of an image containing `bubble`, using a CLIP-ViT-L/14 model.

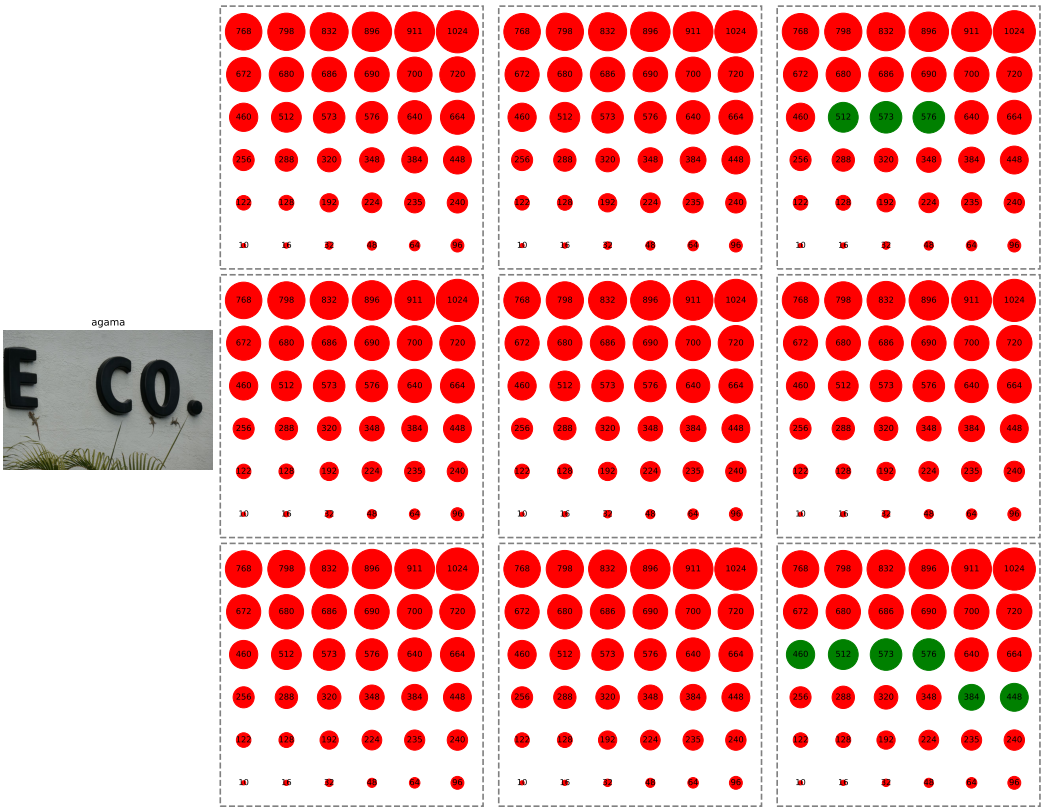


Figure A31: Visualization of effective transforms that lead to the correct classification of an image containing `agama`, using a ViT-B/32 model.

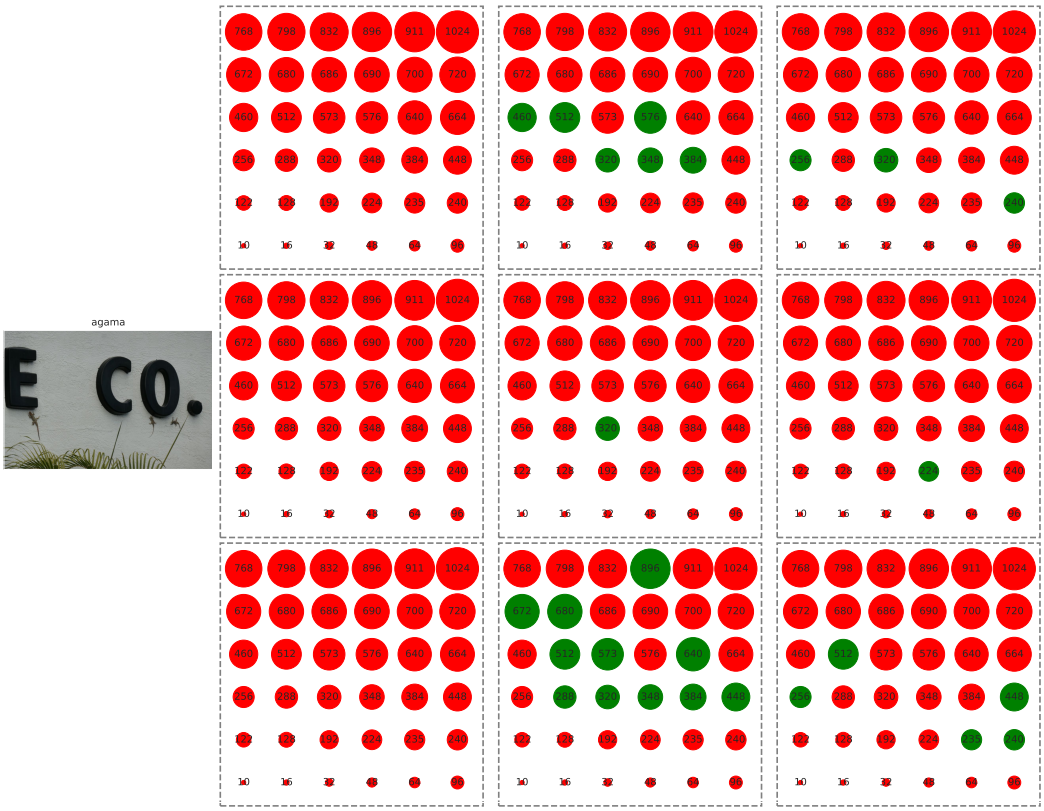


Figure A32: Visualization of effective transforms that lead to the correct classification of an image containing `agama`, using a CLIP-ViT-L/14 model.

Figure A33: Visualization of effective transforms that lead to the correct classification of an image containing `acorn`, using a ViT-B/32 model.

Figure A34: Visualization of effective transforms that lead to the correct classification of an image containing `acorn`, using a CLIP-ViT-L/14 model.

D.3 Only *zoom-out* solves

Sample images that required zooming out to be classified correctly.

D.3.1 ImageNet-Sketch

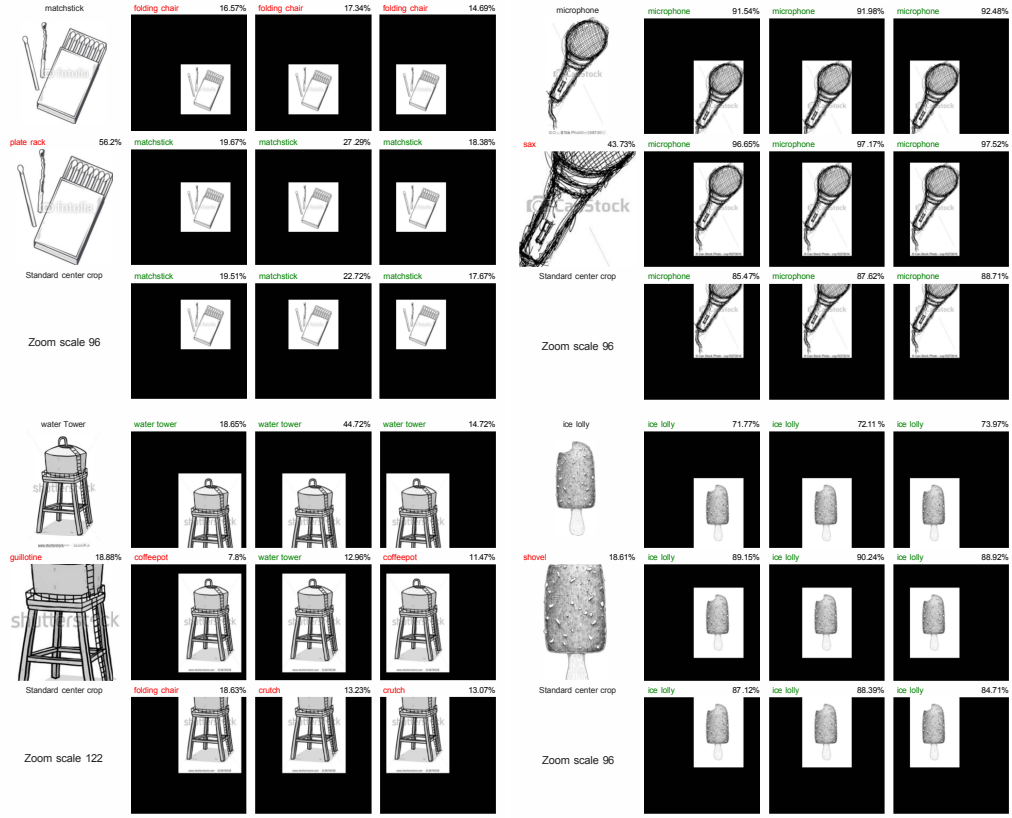


Figure A35: ImageNet-Sketch images that can only be solved using *zoom-out*. Predictions are from a ResNet-50 classifier.

D.3.2 ImageNet-R

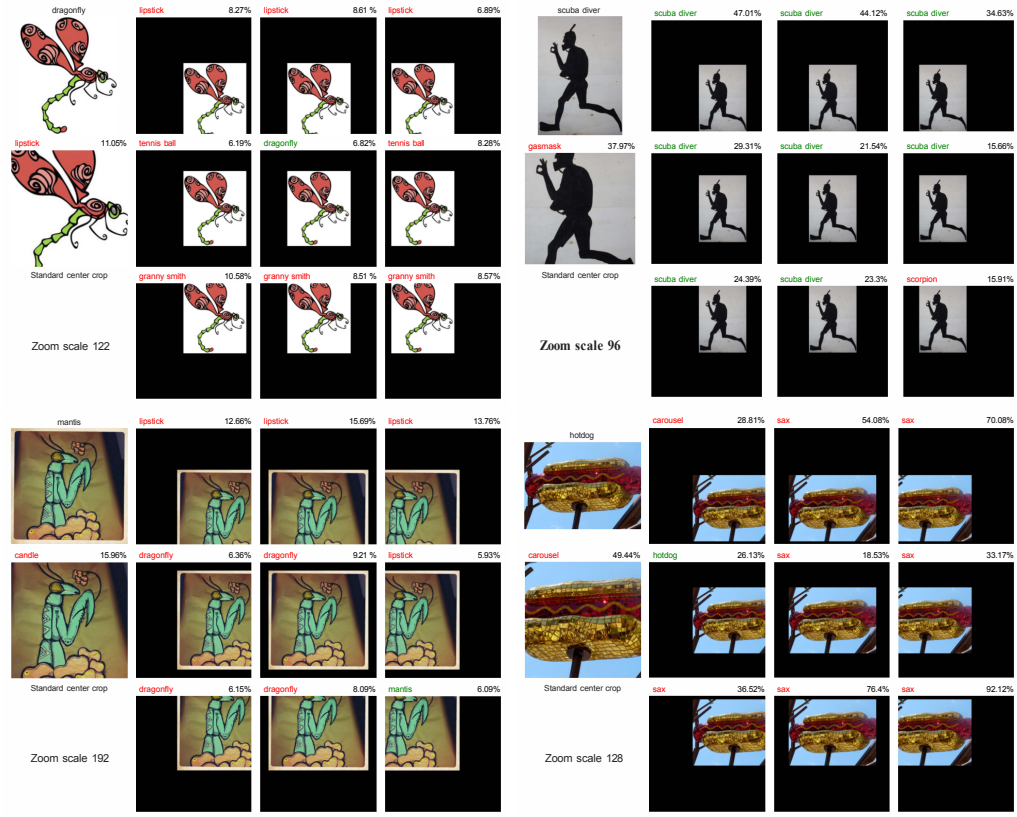


Figure A36: ImageNet-R images that can only be solved using *zoom-out*. Predictions are from a ResNet-50 classifier.

D.3.3 ObjectNet

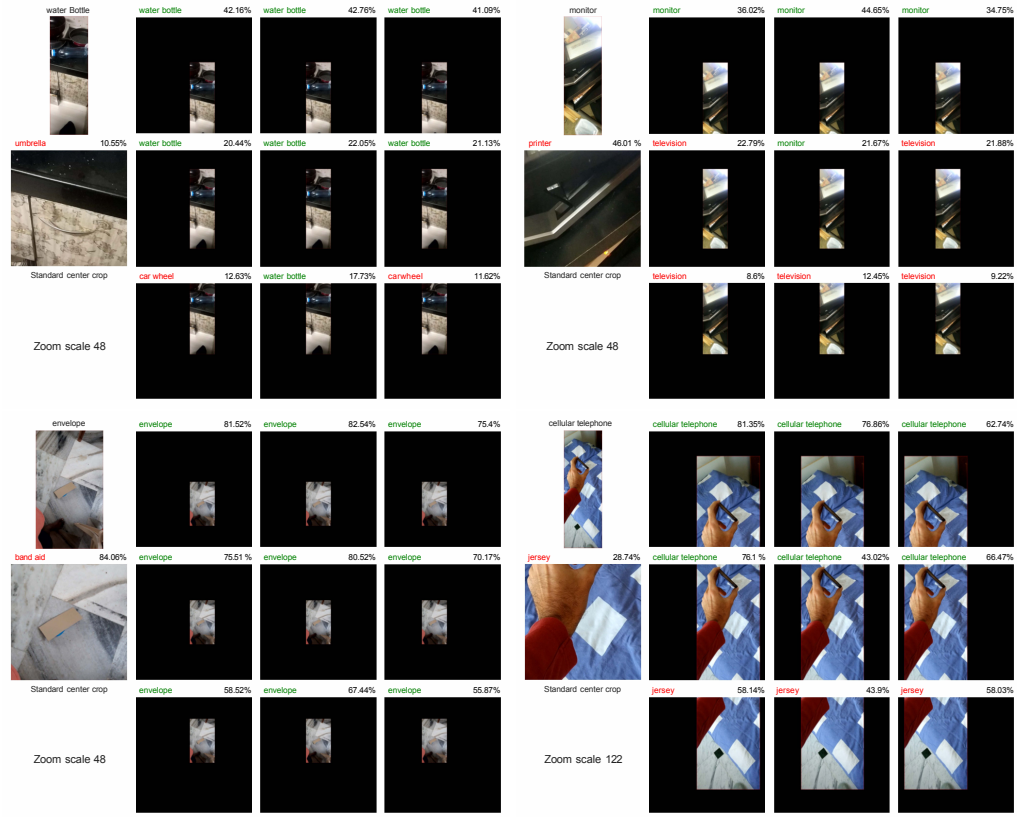


Figure A37: ObjectNet images that can only be solved using *zoom-out*. Predictions are from a ResNet-50 classifier.

D.4 Only *zoom-in* solves

Sample images that required zooming in to be classified correctly.

D.4.1 ObjectNet

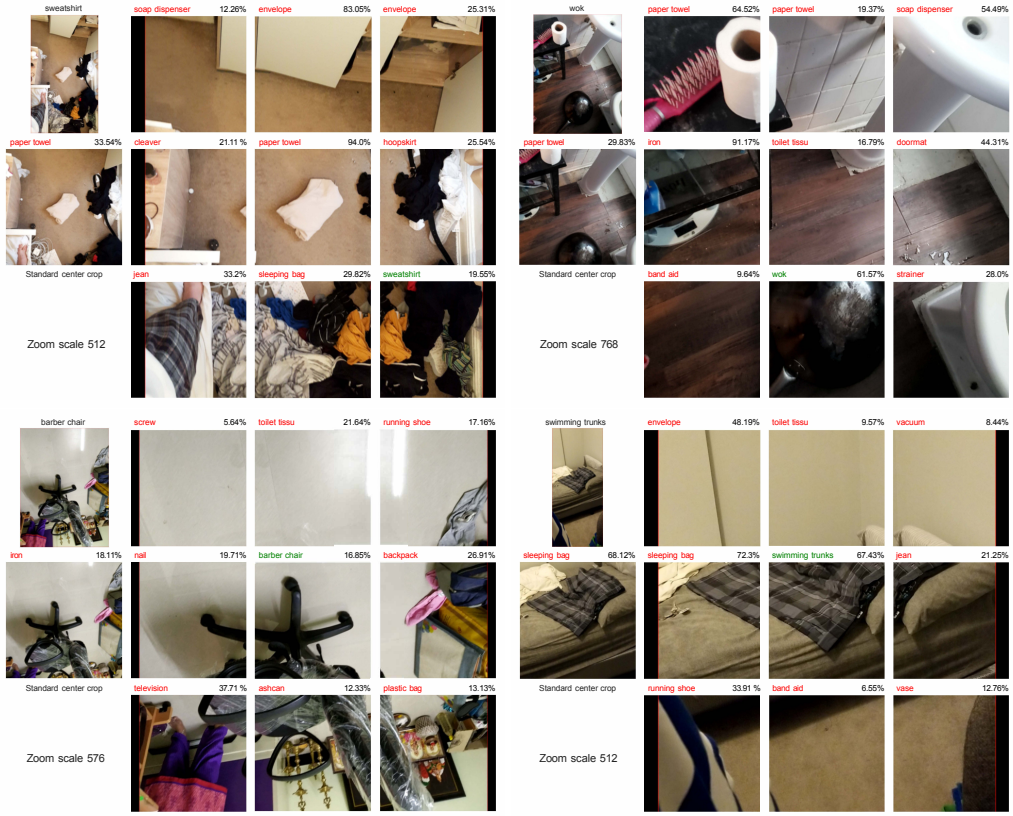


Figure A38: ObjectNet images that can only be solved using *zoom-in*. Predictions are from a ResNet-50 classifier.

D.5 AugMix and `RandomResizedCrop`

Figure A39: $K = 16$ sample outputs from AugMix [25] (which yields the results of random sampling from 13 transformations that include both spatial and color distortions).

Figure A40: $K = 16$ sample outputs from `RandomResizedCrop` (`RRC`), which basically randomly zooms into an arbitrary region in the input image.

Figure A41: $K = 16$ sample outputs from AugMix [25] (which yields the results of random sampling from 13 transformations that include both spatial and color distortions).

Figure A42: $K = 16$ sample outputs from `RandomResizedCrop` (`RRC`), which basically randomly zooms into an arbitrary region in the input image.

E ImageNet-Hard

In this section, we provide details about the ImageNet-hard dataset.

E.1 Distribution

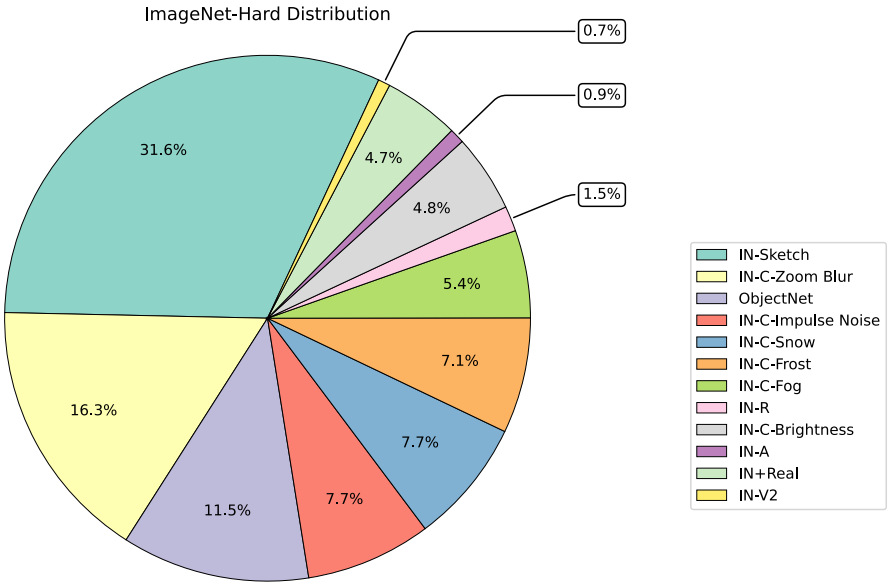


Figure A43: The distribution of the dataset within the ImageNet-Hard Dataset.

E.2 Samples images

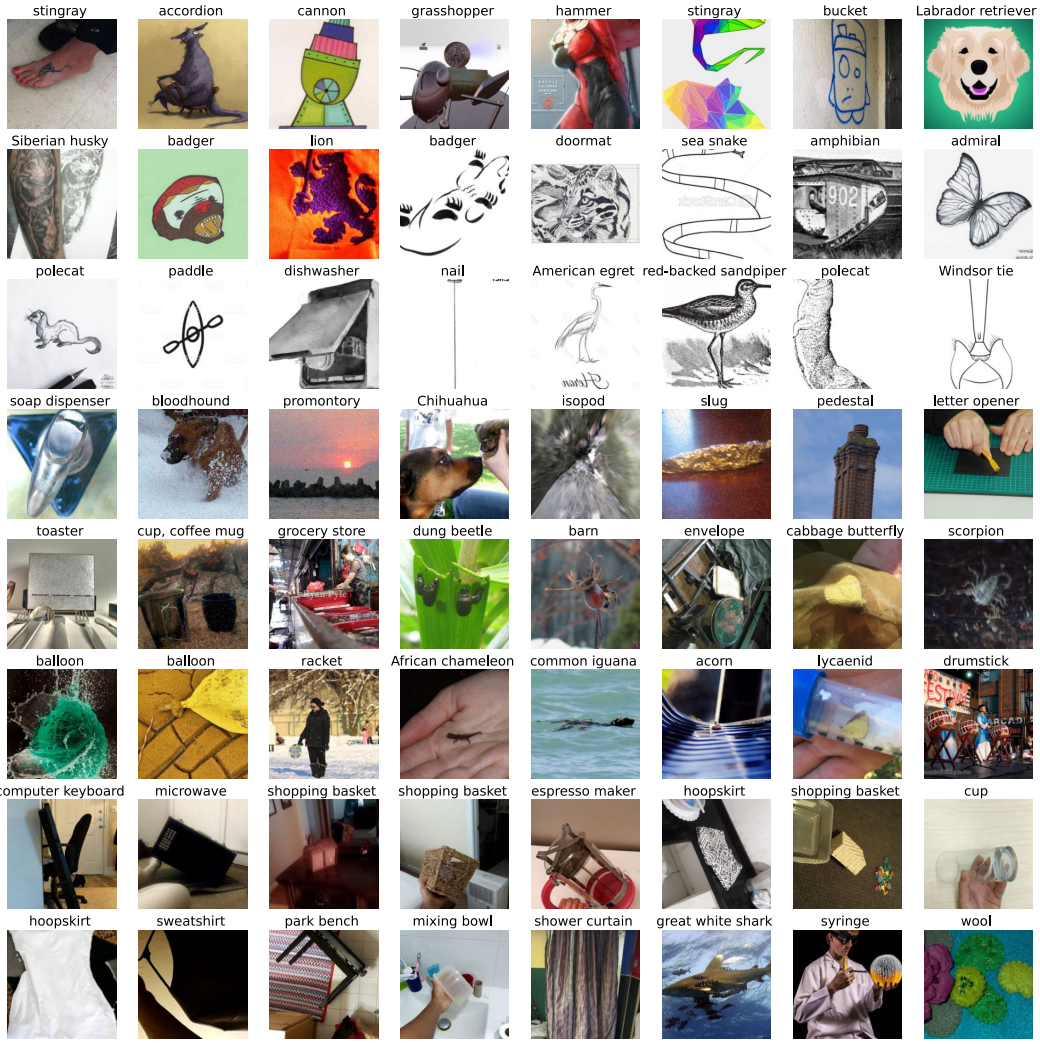


Figure A44: Sample images from ImageNet-Hard dataset with groundtruth labels.

E.3 Analysis of wrong predictions

We used `gpt-3.5-turbo` to categorize each misprediction made by EfficientNet-L2 into two classes: plausible and implausible, based on the semantic distance between the groundtruth label and the predicted label.

System Message: You are a helpful assistant tasked with evaluating an image classifier by reviewing its predictions.

User: I am looking at a picture of a `curly coated retriever`. The model predicted it is a `flat coated retriever`. Can you categorize this prediction as plausible mistake or implausible mistake? provide a one line description.

Assistant: This prediction can be categorized as a *plausible* mistake, as curly coated and flat coated retrievers can look similar to an untrained eye.

Figure A45: Sample prompt and response of `gpt-3.5-turbo` for a plausible classification. The text in the **Assistant** block is the generated response.

System Message: You are a helpful assistant tasked with evaluating an image classifier by reviewing its predictions.

User: I am looking at a picture of a "**ostrich**". The model predicted it is a "**sea anemone**". Can you categorize this prediction as plausible mistake or implausible mistake? provide a one line description.

Assistant: This is an *implausible* mistake as ostriches are large flightless birds and not related to sea anemones in any way.

Figure A46: Sample prompt and response of `gpt-3.5-turbo` for an implausible classification. The text in the **Assistant** block is the generated response.

E.4 Confusing classes

In this section, we present a selection of examples highlighting the errors made by our highest-performing model, EfficientNet-L2.

Figure A47: Images misclassified into `coffemaker` by EfficientNet-L2

Figure A48: Images misclassified into `strainer` by EfficientNet-L2

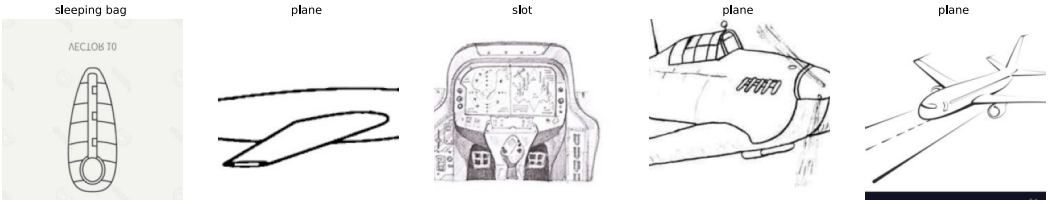


Figure A49: Images misclassified into `space shuttle` by EfficientNet-L2

Figure A50: Images misclassified into `safety pin` by EfficientNet-L2

E.5 Common and rare misclassification

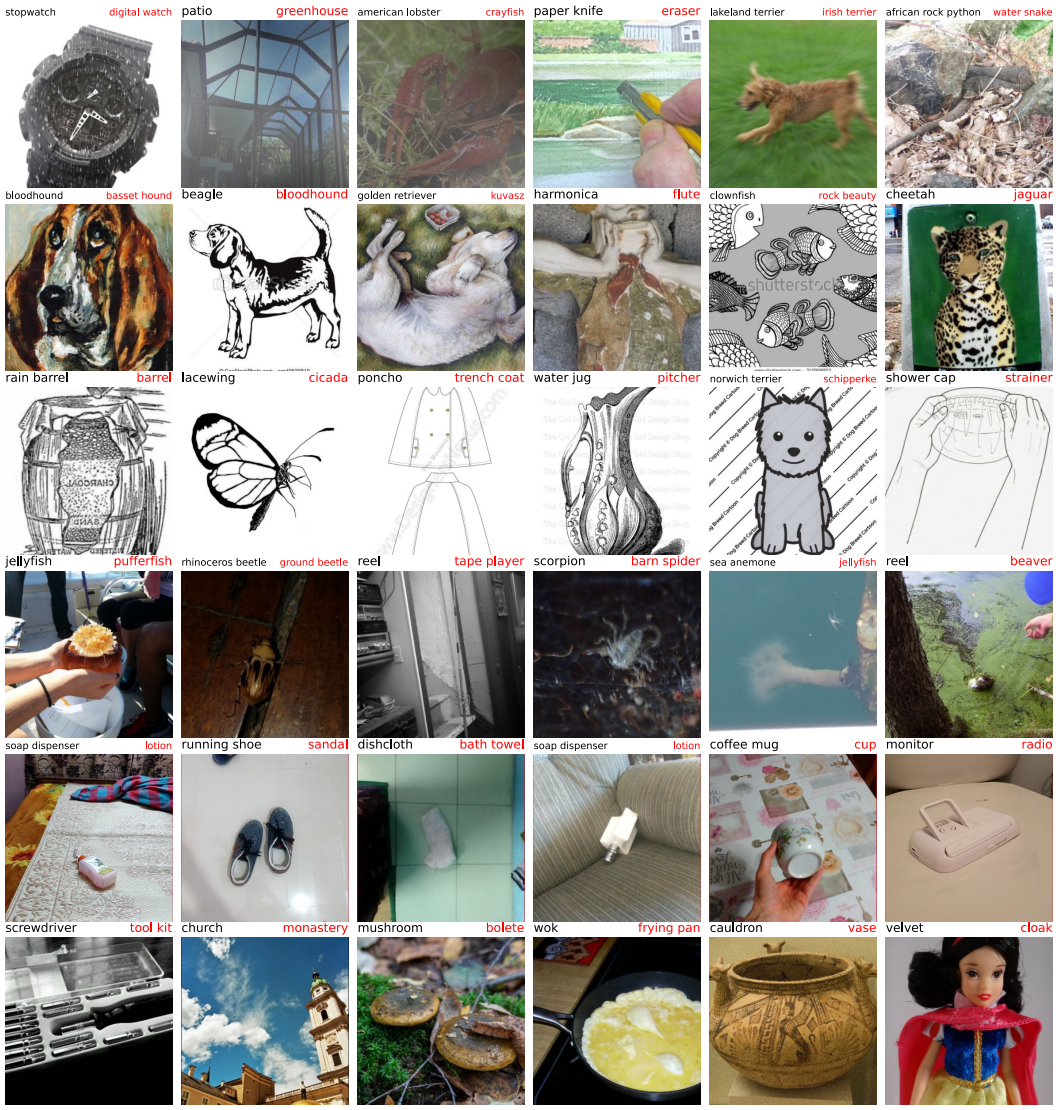


Figure A51: Samples for misclassification of type *Common*

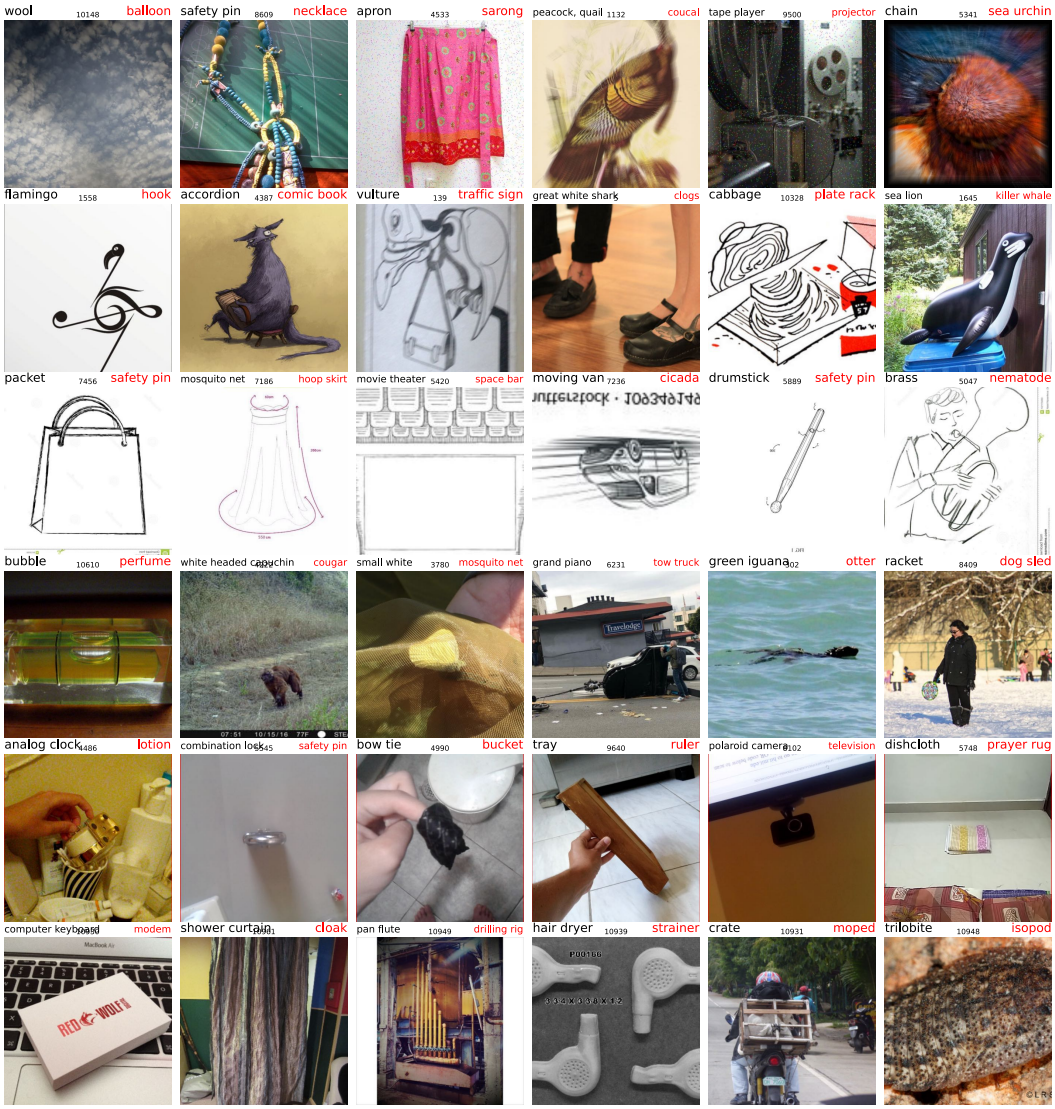


Figure A52: Samples for misclassification of type *Rare*

E.6 Evaluating OpenCLIP models' performance on ImageNet-Hard

All the models in this section are downloaded and used from the *OpenCLIP* library version `2.20.0`.

Table A11: Zero-shot performance of OpenCLIP on ImageNet-Hard (%)

Model	Pre-trained Dataset	Top-1 Accuracy
RN50	`yfcc15m`	0.80
RN50	`cc12m`	1.18
RN50-quickgelu	`yfcc15m`	0.75
RN50-quickgelu	`cc12m`	1.08
RN101	`yfcc15m`	0.65
RN101-quickgelu	`yfcc15m`	0.62
ViT-B/32	`laion400m_e31`	5.34
ViT-B/32	`laion400m_e32`	5.41
ViT-B/32	`laion2b_e16`	5.66
ViT-B/32	`laion2b_s34b_b79k`	6.13
ViT-B/32	`datacomp_m_s128m_b4k`	2.79
ViT-B/32	`commonpool_m_clip_s128m_b4k`	2.50
ViT-B/32	`commonpool_m_laion_s128m_b4k`	2.41
ViT-B/32	`commonpool_m_image_s128m_b4k`	2.72
ViT-B/32	`commonpool_m_text_s128m_b4k`	2.46
ViT-B/32	`commonpool_m_basic_s128m_b4k`	2.23
ViT-B/32	`commonpool_m_s128m_b4k`	1.73
ViT-B/32	`datacomp_s_s13m_b4k`	0.61
ViT-B/32	`commonpool_s_clip_s13m_b4k`	0.84
ViT-B/32	`commonpool_s_laion_s13m_b4k`	0.66
ViT-B/32	`commonpool_s_image_s13m_b4k`	0.61
ViT-B/32	`commonpool_s_text_s13m_b4k`	0.77
ViT-B/32	`commonpool_s_basic_s13m_b4k`	0.75
ViT-B/32	`commonpool_s_s13m_b4k`	0.43
ViT-B/32-quickgelu	`laion400m_e31`	5.34
ViT-B/32-quickgelu	`laion400m_e32`	5.28
ViT-B/16	`laion400m_e31`	6.31
ViT-B/16	`laion400m_e32`	6.46
ViT-B/16	`laion2b_s34b_b88k`	7.18
ViT-B/16	`datacomp_l_s1b_b8k`	5.98
ViT-B/16	`commonpool_l_clip_s1b_b8k`	4.92
ViT-B/16	`commonpool_l_laion_s1b_b8k`	4.44
ViT-B/16	`commonpool_l_image_s1b_b8k`	4.75
ViT-B/16	`commonpool_l_text_s1b_b8k`	5.63
ViT-B/16	`commonpool_l_basic_s1b_b8k`	4.44
ViT-B/16	`commonpool_l_s1b_b8k`	3.83
ViT-B/16-plus-240	`laion400m_e31`	6.65
ViT-B/16-plus-240	`laion400m_e32`	6.69
ViT-L/14	`laion400m_e31`	8.83
ViT-L/14	`laion400m_e32`	8.72
ViT-L/14	`laion2b_s32b_b82k`	10.13
ViT-L/14	`datacomp_xl_s13b_b90k`	15.60
ViT-L/14	`commonpool_xl_clip_s13b_b90k`	11.58
ViT-L/14	`commonpool_xl_laion_s13b_b90k`	11.42
ViT-L/14	`commonpool_xl_s13b_b90k`	12.44
ViT-H/14	`laion2b_s32b_b79k`	13.01
ViT-g/14	`laion2b_s12b_b42k`	11.47
ViT-g/14	`laion2b_s34b_b88k`	14.03
ViT-bigG-14	`laion2b_s39b_b160k`	15.93
roberta-ViT-B/32	`laion2b_s12b_b32k`	5.21
xlm-roberta-base-ViT-B/32	`laion5b_s13b_b90k`	5.72
xlm-roberta-large-ViT-H/14	`frozen_laion5b_s13b_b90k`	12.95

convnext_base	laion400m_s13b_b51k	4.74
convnext_base_w	laion2b_s13b_b82k	6.09
convnext_base_w	laion2b_s13b_b82k_augreg	7.25
convnext_base_w	laion_aesthetic_s13b_b82k	5.57
convnext_base_w_320	laion_aesthetic_s13b_b82k	5.50
convnext_base_w_320	laion_aesthetic_s13b_b82k_augreg	7.14
convnext_large_d	laion2b_s26b_b102k_augreg	10.39
convnext_large_d_320	laion2b_s29b_b131k_ft	10.69
convnext_large_d_320	laion2b_s29b_b131k_ft_soup	11.20
convnext_xxlarge	laion2b_s34b_b82k_augreg	14.27
convnext_xxlarge	laion2b_s34b_b82k_augreg_rewind	14.23
convnext_xxlarge	laion2b_s34b_b82k_augreg_soup	14.68
coca_ViT-B/32	laion2b_s13b_b90k	5.83
coca_ViT-B/32	mscoco_finetuned_laion2b_s13b_b90k	0.20
coca_ViT-L/14	laion2b_s13b_b90k	10.79
coca_ViT-L/14	mscoco_finetuned_laion2b_s13b_b90k	9.28

Table A12: Zero-shot performance of CommonPool and DataComp models on ImageNet-Hard (%)

Scale	Model	Pretrained	Top-1 Accuracy
xlarge	ViT-L/14	datacomp_xl_s13b_b90k	15.60
	ViT-L/14	commonpool_xl_clip_s13b_b90k	11.58
	ViT-L/14	commonpool_xl_laion_s13b_b90k	11.42
	ViT-L/14	commonpool_xl_s13b_b90k	12.44
large	ViT-B/16	datacomp_l_s1b_b8k	5.98
	ViT-B/16	commonpool_l_clip_s1b_b8k	4.92
	ViT-B/16	commonpool_l_laion_s1b_b8k	4.44
	ViT-B/16	commonpool_l_image_s1b_b8k	4.75
	ViT-B/16	commonpool_l_text_s1b_b8k	5.63
	ViT-B/16	commonpool_l_basic_s1b_b8k	4.44
	ViT-B/16	commonpool_l_s1b_b8k	3.83
medium	ViT-B/32	datacomp_m_s128m_b4k	2.79
	ViT-B/32	commonpool_m_clip_s128m_b4k	2.50
	ViT-B/32	commonpool_m_laion_s128m_b4k	2.41
	ViT-B/32	commonpool_m_image_s128m_b4k	2.72
	ViT-B/32	commonpool_m_text_s128m_b4k	2.46
	ViT-B/32	commonpool_m_basic_s128m_b4k	2.23
	ViT-B/32	commonpool_m_s128m_b4k	1.73
small	ViT-B/32	datacomp_s_s13m_b4k	0.61
	ViT-B/32	commonpool_s_clip_s13m_b4k	0.84
	ViT-B/32	commonpool_s_laion_s13m_b4k	0.66
	ViT-B/32	commonpool_s_image_s13m_b4k	0.61
	ViT-B/32	commonpool_s_text_s13m_b4k	0.77
	ViT-B/32	commonpool_s_basic_s13m_b4k	0.75
	ViT-B/32	commonpool_s_s13m_b4k	0.43

E.7 Evaluating classifiers on ImageNet-Hard-4K

Table A13: Top-1 accuracy (%) on ImageNet-Hard-4K. Most models obtain a lower accuracy compared to their corresponding accuracy on ImageNet-Hard.

Classifier	Accuracy	Classifier	Accuracy	Classifier	Accuracy
AlexNet	7.08 (-0.16)	ViT-B/32	18.12 (-0.40)	CLIP-ViT-L/14@224px	1.81 (-0.05)
VGG-16	11.32 (-0.68)	EfficientNet-B0@224px	12.94 (-3.63)	CLIP-ViT-L/14@336px	1.88 (-0.14)
ResNet-18	10.42 (-0.44)	EfficientNet-B7@600px	18.67 (-4.53)	OpenCLIP-ViT-bigG-14	14.33 (-1.60)
ResNet-50	13.93 (-0.81)	EfficientNet-L2@800px	28.42 (-10.58)	OpenCLIP-ViT-L-14	13.04 (-2.56)

E.8 Obviously ill-posed samples from ImageNet-Sketch

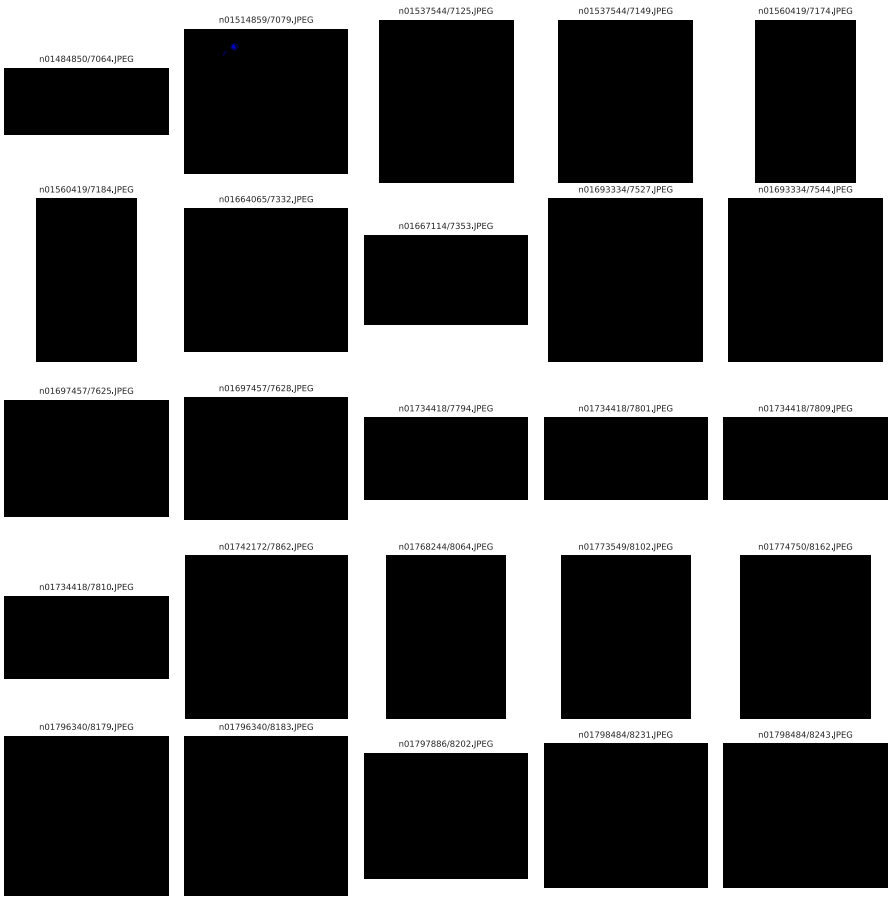


Figure A53: Sample images from ImageNet-Sketch that are completely black.