An Exploratory Study on Abstract Images and Visual Representations Learned from Them (Supplementary Material)

S1 Dataset

S1.1 More examples

In here, we demonstrate more examples of our dataset. More image pairs of MiniImageNet and HAID-MiniImageNet are shown in fig. S7 and more image pairs of CIFAR-10 and HAID-CIFAR-10 are shown in fig. S8.

S1.2 Configurations of HAID

HAID contains three sub-datasets generated from MiniImageNet [32], Caltech-256 [10], CIFAR-10 [14]. We applied different abstract levels when generating these three datasets, specifically, HAID-CIFAR-10 and HAID-Caltech-256 support the SVG images with abstract levels up to 100 shapes; however, the HAID-MiniImageNet supports two extra abstract levels: 500 and 1,000 shapes. The reason for applying two more abstract levels for HAID-MiniImageNet is that they can provide comprehensive results about how primitive-based images perform on images with complex scenes. We observed that, compared with Caltech-256, MiniImageNet contains many more samples with multiple objects and complex backgrounds. As for CIFAR-10, generating images with shapes of 500 and 1,000 will far exceed the size of the original pixel image (~1 KB vs. ~60 KB). We believe that, for this study, it is meaningless to compare the performance under significant file size differences.

S1.3 Primitive *vs.* VTracer

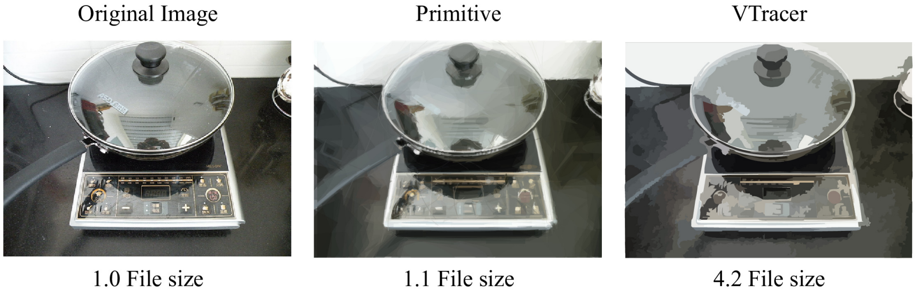
VTracer [31] is a tracing-based algorithm that can convert pixel images to SVG images. Although it can faithfully restore the details of pixel images, the file size of the generated SVG images is generally much larger than the original pixel images. Figure S1 compares the differences in generation effect and file size between images generated by the Primitive and VTracer algorithms.

S2 Supplementary Experiment Content

S2.1 Experiment setting

In here, we supplement the experimental details in section 4, including the settings of hyperparameters and network architectures. The parameter may not be the best setting since we focus more on comparing the performance difference under the same situations rather than exploring the best performance.

Classification. For the experiments of training ResNet50 [11] and MobileNetv2 [27] on the MiniImageNet and HAID-MiniImageNet, we use AdamW as the optimiser with an initial learning rate of 0.0001, and we use random resize crop to frame the image size into 224×224.



Original Image Primitive VTracer

1.0 File size 1.1 File size 4.2 File size

Figure S1: The comparison between the images generated by the Primitive and VTracer. We applied the highest capacity setting (1,000 shapes) to generate the Primitive-based SVG images. For the image generated by the VTracer, the hyperparameters will be: Filter Speckle = 15, Colour Precision = 6, and Gradient Step = 16. Based on the file size of the original image, we show the difference in file size between the generated images and their original image in the form of multiples.

We use a series of data augmentation strategies, including: Random Horizontal Flipping, Random Augment [6], and Random Erasing [33], training for 120 epochs. For the HAID-CIFAR-10 experiments, we only use the Adam [13] optimiser with an initial learning rate of 0.001 and training for 10 epochs.

Table S1: Top-1 Accuracy of ResNet 50 with all shapes (mode0) on MiniImageNet and HAID-MiniImageNet

	Mode 0						
	10	**30**	**50**	**100**	**500**	**1,000**	**Raster**
20%	17.00%	25.98%	27.67%	32.43%	39.90%	40.10%	42.67%
40%	22.67%	33.80%	38.62%	42.82%	51.52%	53.62%	55.78%
60%	26.27%	38.40%	43.23%	49.67%	59.27%	61.25%	63.88%
80%	27.87%	41.78%	46.90%	53.72%	62.75%	65.42%	67.92%
100%	29.40%	43.75%	50.42%	57.07%	65.63%	68.58%	72.12%

Table S2: Top-1 Accuracy of MobileNetv2 with all shapes (mode 0) on MiniImageNet and HAID-MiniImageNet

	Mode 0						
	10	**30**	**50**	**100**	**500**	**1,000**	**Raster**
20%	18.67%	23.30%	26.30%	28.90%	34.38%	34.18%	34.12%
40%	22.52%	31.12%	33.42%	37.93%	43.85%	45.70%	49.00%
60%	24.78%	34.63%	39.02%	43.73%	51.33%	51.22%	55.72%
80%	27.32%	38.53%	41.97%	47.93%	55.13%	56.87%	61.05%
100%	28.98%	40.00%	44.85%	50.77%	58.93%	61.48%	64.42%

Semantic Segmentation. During training the DeepLabv3 for Semantic Segmentation, we randomly crop the image size to 480×480 from the base size 520×520. We use the AdamW optimiser with an initial learning rate of 0.0009, momentum of 0.9, and weight decay of 0.01 to train the models for 200 epochs.

Object Detection. For Faster R-CNN, we use Stochastic Gradient Descent (SGD) as the optimiser with an initial learning rate of 0.005, momentum equals 0.9, and weight decay is 0.0005. The learning rate is reduced by a factor of 0.95 every 5 epochs, and we trained Faster R-CNN for 20 epochs. For the model architecture of Faster R-CNN, the Region Proposal Network (RPN) uses an anchor generator with scales of 32, 64, 128, 256, 512 and aspect ratios of 1:2, 1:1, 2:1. For the Region of Interest (ROI) Pooling, a single-scale of RoIAlign is employed with an output size of 7 and a sampling ratio of 2.

For SSD-Lite, we use SGD as the optimiser with an initial learning rate of 0.001, momentum equals 0.9, and weight decay equals 0.0005. The learning rate was reduced by a factor of 0.1 at 80 and 100 epochs. The implementation detail follows this repository.

S2.2 Supplementary results

Classification. The full results from HAID-MiniImageNet are shown in table S1 and table S2.

Semantic Segmentation & Object Detection. In section 4, we showed the experiment results in fig. 6 and fig. 7. In here, we further demonstrate the full comparison of performance between models initialised by the backbone pretrained on HAID-MiniImageNet and two baselines (Upper Bound and Lower Bound) on two downstream tasks. Table S3 shows the results from semantic segmentation task and table S4 shows the results from object detection task.

Table S3: Comparing the performances by Mean Intersection over Union (mIoU) of DeepLabv3 with backbones of ResNet 50 (up) and MobileNetv2 (down) from MiniImageNet and HAID-MiniImageNet. The number in the first line represents the abstract level applied for backbones, from 10 to 1,000 shapes. We also provide the upper bound (UB: fine-tuning with backbone from raster images) and lower bound (LB: training without backbone), and compare them with fine-tuning models with backbones gained from HAID-MiniImageNet.

	10	30	50	100	500	1,000
DeepLabv3-ResNet50	45.09	46.09	46.85	47.90	49.43	49.97
Compar w LB	+6.56	+7.56	+8.32	+9.38	+10.91	+11.44
Compar w UB	-5.44	-4.43	-3.67	-2.62	-1.09	-0.56
DeepLabv3-MobileNetv2	36.40	38.20	39.10	39.01	40.54	40.63
Compar w LB	1.87	+3.68	+4.58	+4.49	+6.02	+6.10
Compar w UB	-5.44	-4.43	-3.67	-2.62	-1.09	-0.56

To further discuss the reason for the surprisingly better results from Faster R-CNN, we apply the Grad-CAM [29] to visualise the attention of the feature map from Faster R-CNN. The CAM visualisation is shown in the fig. S2. From the visualisation results, we can observe that the attention maps of Faster R-CNN models with abstract backbones initialised

Table S4: Comparing the performances by Mean Average Precision (mAP) of SSD-Lite with MobileNet v2 backbones (up) and Faster R-CNN with ResNet 50 backbones (down) from MiniImageNet and HAID-MiniImageNet. The number in the first line represents the abstract level applied for backbones, from 10 to 1,000 shapes. We also provide the upper bound (UB: fine-tuning with backbone from raster images) and lower bound (LB: training without backbone), and compare them with fine-tuning models with backbones gained from HAID.

	10	30	50	100	500	1,000
SSD-Lite	28.23	29.29	30.59	29.87	29.56	30.36
Compar w LB	+4.34	+5.40	+6.70	+5.98	+5.66	+6.47
Compar w UB	-2.44	-1.37	-0.07	-0.79	-1.11	-0.30
Faster R-CNN	21.78	23.44	27.39	31.36	31.15	30.59
Compar w LB	+14.97	+16.63	+20.59	+24.56	+24.35	23.79
Compar w UB	-5.27	-3.61	+0.34	+4.31	+4.10	+3.55

are more tightly concentrated on object geometry and core semantic regions than backbones pre-trained on raster images. As the input detail level of backbones increases to 100 shapes, such geometric focus is the most obvious, which means that it focuses more on the contour or structure representations that benefit box regression, even when fine appearance cues are reduced. Interestingly, this effect weakens as the input detail level of pre-trained backbones increases further (*e.g.* 500 and 1,000 shapes), with attention patterns and localisation performance approaching those of models pre-trained on original raster images. These observations may explain the evaluation results in table S4.

Abstraction with all shapes versus triangle only. In here, we discuss how much difference could be based on the different types of shapes by examining the performance difference between two generation modes: one that employs all available SVG primitives (mode 0) and another that exclusively uses triangles (mode 1). We focus on the triangle-only configuration since it produces images with the lowest file size among the available shape types while maintaining high image quality. Figure 2 illustrates the comparisons between mode 0 and mode 1, and more samples can be found in fig. S7. We trained the networks separately on HAID-MiniImageNet images generated under mode 0 and mode 1 and evaluated them on test sets corresponding to the same levels of abstraction. The performance differences across varying numbers of shapes are presented in fig. S3.

As a result, training on the triangle-only images shows a slight performance drop in most cases. Notably, both ResNet50 and MobileNet v2 exhibit a distinct performance gap between mode 0 and mode 1 when using images with 1,000 shapes; as observed in fig. S4, some fine-grained features are not adequately captured in triangle-only images, even at high fine-grained levels. Therefore, considering the advantages of representation performance and displaying details, images with all types of primitive shapes are a priority choice, however, with slight toleration of performance drop, triangle-only images offer a viable alternative with the advantage of lower file capacity cost.

User study. We select 36 images in total across six levels (30, 50, 100, 500, 1,000 shapes, and original images) from HAID-MiniImageNet and MiniImageNet (each level contains 6

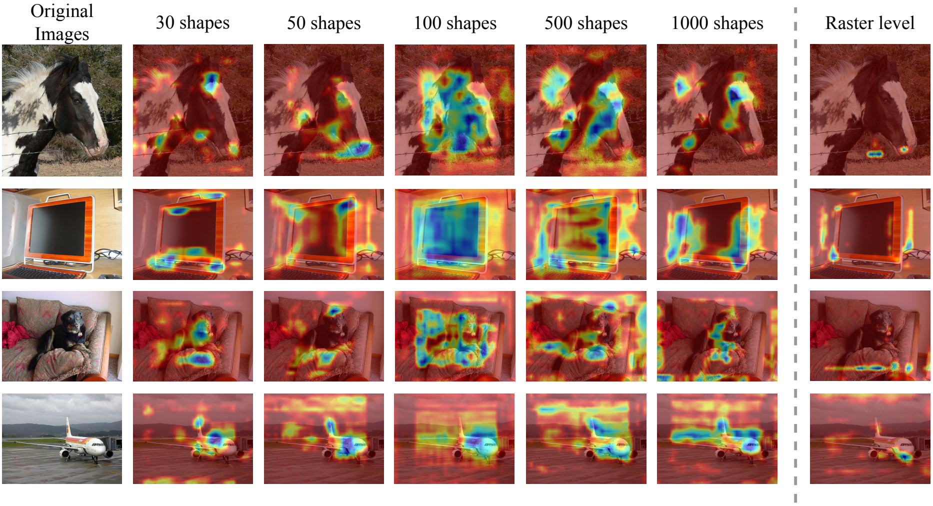


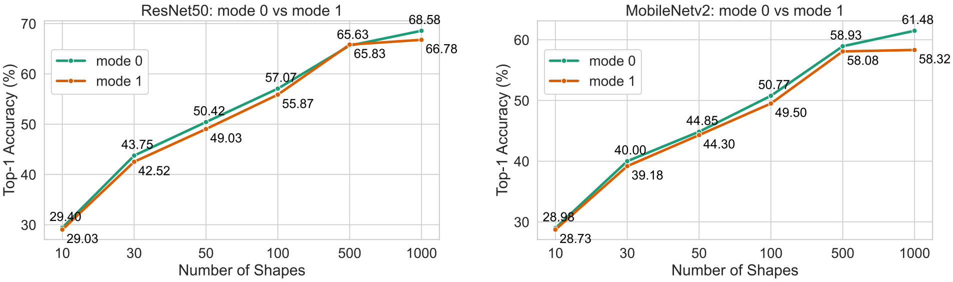
Figure S2: Grad-CAM visualisations from the Faster R-CNN using pretrained backbone. The comparison across the abstract levels of backbones from 30 to 1,000 shapes and original images.

images). Images for each abstract level were balanced by a priori difficulty: three single-object images with the simple background (labelled “easy samples”) and three images with multiple objects, complex textures, or cluttered scenes (labelled “hard samples”). Participants viewed all 36 images in randomised order and provided a single 1–5 rating per image reflecting how confident they are to perceive the object(s) (1 = cannot recognise at all, 5 = extremely confident). We provide the explicit instruction: *"Your rating should reflect* ***how clearly you perceive*** *the object in each image— you* ***do not*** *need to know the specific name of the object."*, as the task measured perceptual clarity rather than knowledge.

In section 4.4, we discussed the user study to evaluate the dataset from a human perceptive. In here, the comprehensive results are demonstrated in table S5. We also recorded the gender information of participants. Out of the 12 responses, 6 recorded gender as male, and 6 recorded gender as female. From the results, there is no significant difference in the rating results between the two genders.

Table S5: The table compares the Mean Opinion Score (MOS) from participants of the user study. We also compared the MOS for different genders, which are demonstrated in the rows of ‘Male’ and ‘Female’.

	Abstract Levels (number of shapes)					
	30	50	100	500	1,000	Original images
Easy samples	2.22	3.39	3.81	4.75	4.89	4.97
Hard samples	1.81	1.5	3.11	4.44	4.67	4.94
All samples	2.01	2.44	3.46	4.6	4.78	4.96
Male	2.20	2.47	3.47	4.75	4.81	4.92
Female	1.84	2.42	3.44	4.45	4.75	5.00



(a). ResNet50 (b). MobileNet v2

Figure S3: Comparison of ResNet50 and MobileNetv2 performance training on all types of SVG primitives (mode 0) and training on triangular primitives only (mode 1).

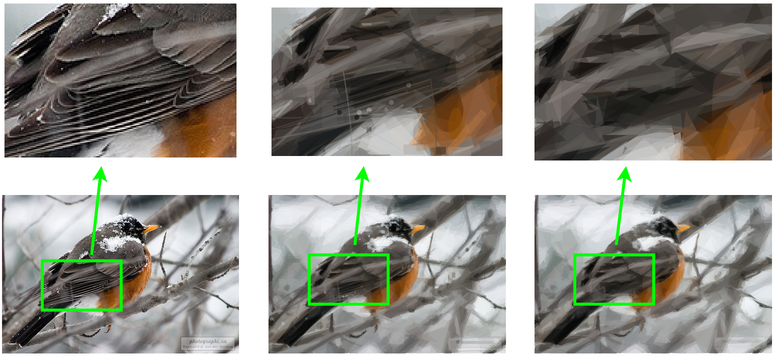


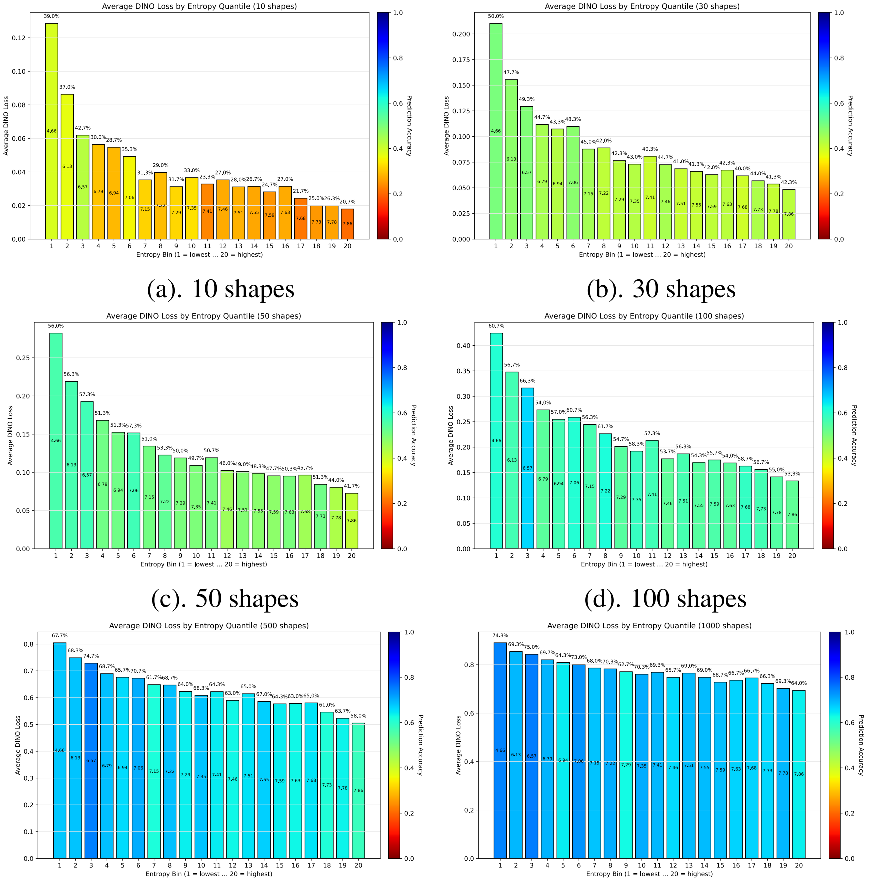
Figure S4: Detailed features comparison: raster image (left), abstract image containing all types of primitive shape (middle), and abstract image containing triangle only (right).

S2.3 The correlation between image entropy and abstract images

Entropy of the image serves as a quantitative measure of its information content as well as its complexity. We observed that, in the human perceptive, for the simple images (*e.g.* with single object and unsophisticated background), such as the last row images in fig. S7, are generally recognisable with a lower number of shapes than the complex images, such as the second row images in fig. S7. Therefore, we speculate there is a correlation between the information complexity of the original image and the levels of its abstractions. To prove this hypothesis, we randomly sampled 4000 image pairs from the MiniImageNet and HAID-MiniImageNet, calculating the entropy values of these 4000 images, and sorted them into 20 groups based on the entropy from smallest to largest. We also labelled each group with the perceptual loss (DINO loss specifically) between abstract and original images, their prediction accuracy, and the mean entropy value. The results are shown in fig. S5,

S3 Primitive

The fig. S6 specifically explained how the Primitive generates the shape-based images from the provided raster images.



(a). 10 shapes

(b). 30 shapes

(c). 50 shapes

(d). 100 shapes

(e). 500 shapes

(f). 1,000 shapes

Figure S5: The correlation between entropy and DINO loss, each diagram has 20 groups sorted by entropy value from small to large. The colour and the value above each bin represent the prediction accuracy of each bin, and the value inside each bin represents the average entropy value of each group.

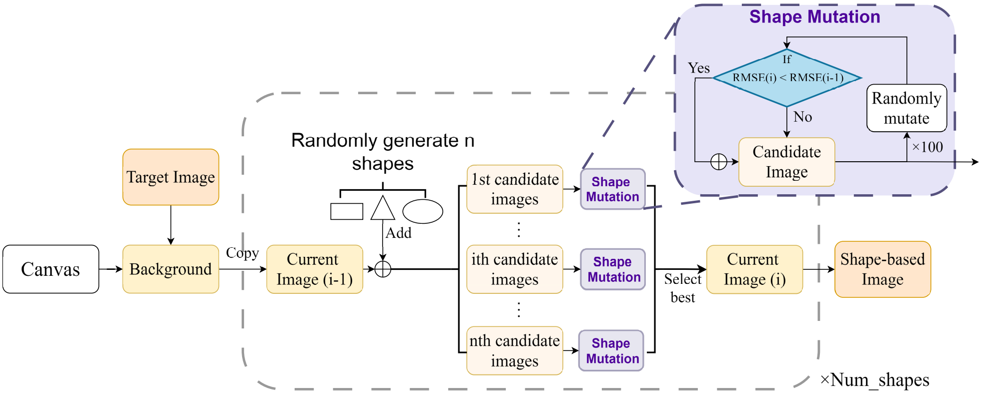


Figure S6: The generating procedure of Primitive.

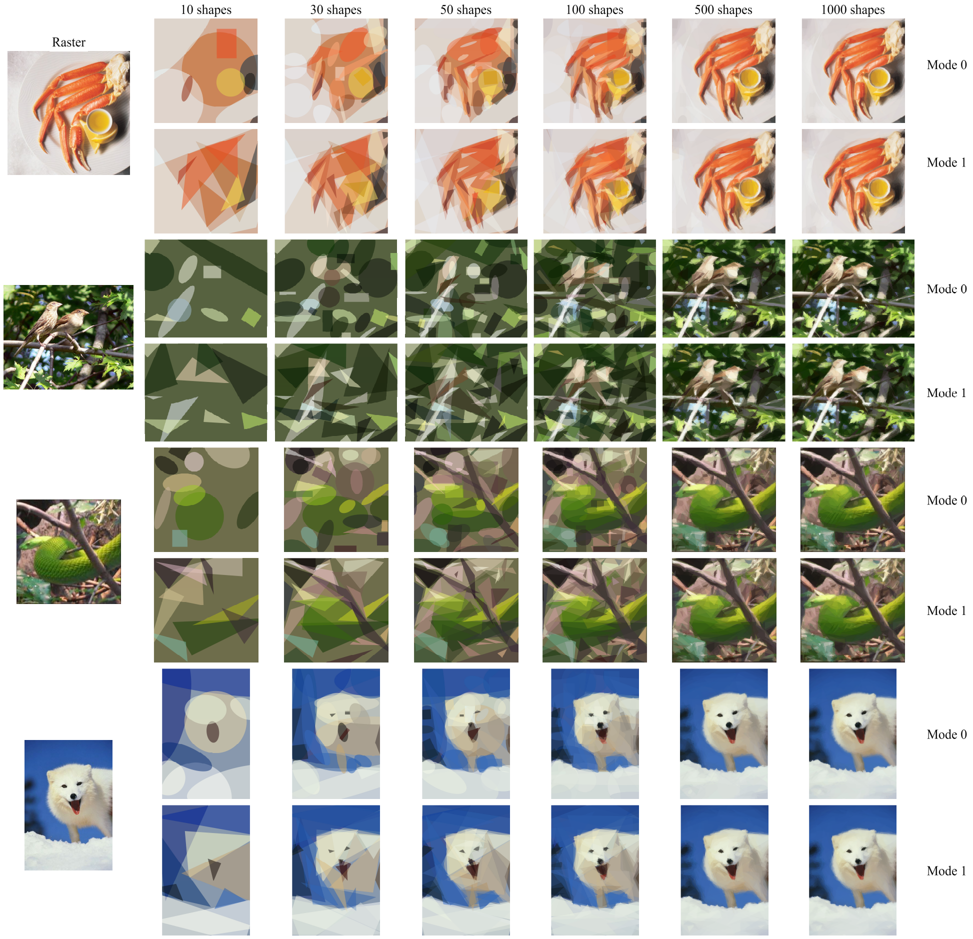


Figure S7: More examples of HAID-MiniImageNet

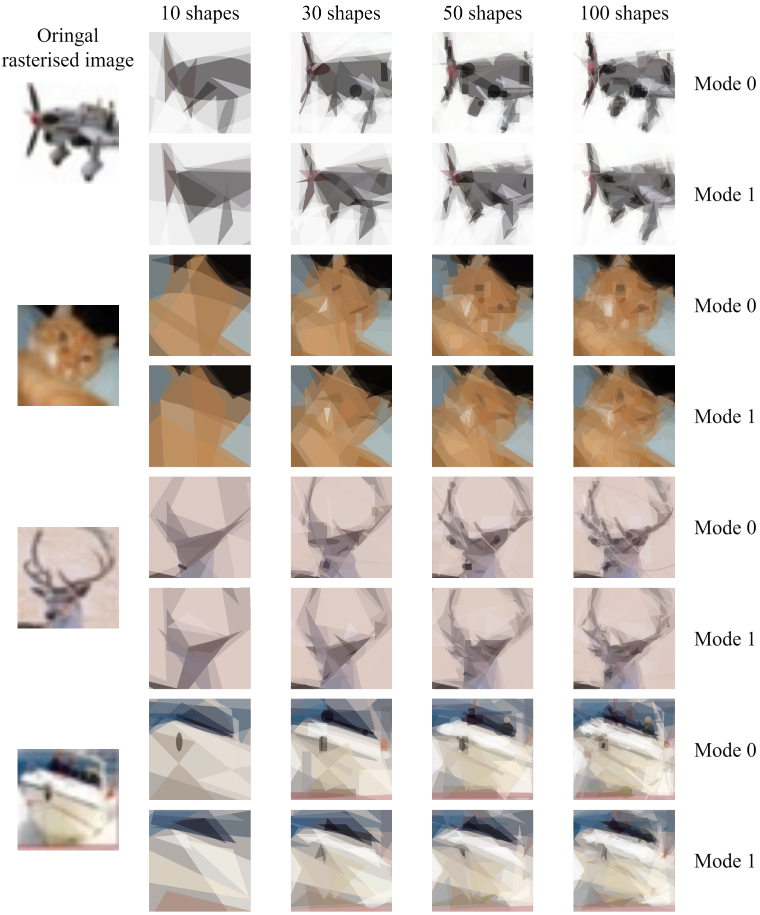


Figure S8: Example image pairs between CIFAR-10 and HAID-CIFAR-10