An Exploratory Study on Abstract Images and Visual Representations Learned from Them

Haotian Li
hxl226@student.bham.ac.uk
Jianbo Jiao
j.jiao@bham.ac.uk

The School of Computer Science
University of Birmingham
Birmingham, UK

Abstract

Imagine living in a world composed solely of primitive shapes—could you still recognise familiar objects? Recent studies have shown that abstract images—constructed by primitive shapes—can indeed convey visual semantic information to deep learning models. However, representations obtained from such images often fall short compared to those derived from traditional raster images. In this paper, we study the reasons behind this performance gap and investigate how much high-level semantic content can be captured at different abstraction levels. To this end, we introduce the **H**ierarchical **A**bstraction **I**mage **D**ataset (HAID), a novel data collection that comprises abstract images generated from common raster image datasets at multiple levels of abstraction. We then train and evaluate conventional vision systems on *HAID* across various tasks of classification, segmentation, and object detection, providing a comprehensive study between rasterised and abstract image representations. We also discuss if the abstract image can be considered as a potentially effective format for conveying visual semantic information and contributing to vision tasks. Code and models will be made publicly available.

1 Introduction

Visual components, such as primitive shapes, are vital for humans to recognise and remember objects. Infants can classify objects based on their shapes [1, 21, 22], humans can recognize objects merely based on shape information at little cost [32], and such shape cues can be quickly and efficiently extracted by the human brain [7]. As for computer vision, abstract images are generally considered as the carrier to present the shape-oriented visual information. They are typically formed by vectorised shapes to provide lossless scalability and are widely used in many scenarios due to this special property. Although shape information plays a crucial role in human visual recognition patterns, early machine learning visual tasks did not focus too much on abstract images. Nevertheless, with the rapid development of computer vision, the potential contributions of such abstract images to machine learning systems are gradually being recognised. The remarkable progress related to vectorised image generation and understanding has been achieved, for example, DeepSVG [4] successfully generated the transition animation between two Scalable Vector Graphics (SVG) icons, SVGformer [3]

further improved the performance and supported up to four different downstream tasks, recently, StarVector [26] presented the first large-scale pretraining dataset and the Multi-modal Large Language Model (MLLM) for SVG generation.

Figure 1: An overview of samples in the introduced HAID dataset and corresponding images from the raster image datasets. HAID-MiniImageNet supports the abstract level up to 1000 shapes, HAID-Caltech-256 and HAID-CIFAR-10 support the abstract level up to 100 shapes.

However, despite such great achievements, the studies related to abstract images generally stay on the high-abstract level and rarely consider the correlations with the complex visual semantic information from the real world. In the work of [2], the authors try to leverage the powerful understanding abilities of the Large Language Model (LLM) to 'see' and 'draw' the vectorised images, but there is still a significant performance gap compared with the vision experts trained on pixel-level images. Another work [29] also tried to use LLM to understand and generate code-based abstract images, then use the generated images to train the vision model and evaluate based on the real images. The result shows that LLM can understand and generate visual concepts from code-based images, yet it will fail when encountering images containing complex semantic information, and the contribution from generated images to vision systems is still limited.

For the reasons of such a performance gap, following the conclusions from some works [2, 29], we speculate that the difficulty of demonstrating the complex and fine-grained features from abstract images is one of the major problems. For most of the current vector graphics image datasets [4, 16, 26, 34], the simple and single-object icons or fonts are mainly contained. However, raster images generally have complex scenes with multiple objects and various environments. In the work of [2], they tried to provide SVG images directly converted from rasterised images, but some fine-grained features, textural features, for example, still failed to be presented. Based on these, we came up with several questions: **1) Is the unsophisticated level of abstraction a major reason for the performance gap between representations obtained from raster and abstract images? 2) To what extent do changes in fine-grained features of abstract images affect the recognition of visual semantic information?**

To answer these questions, we present a dataset called ***H**ierarchical **A**bstraction **I**mage **D**ataset (HAID)* containing various abstract levels of SVG images. The dataset is generated directly from raster image datasets [10, 14, 31] using the tool named Primitive [3]. Then, we use images with different levels of abstraction to train and evaluate models for classification, object detection, and segmentation tasks. Finally, we discuss whether the difficulty of presenting fine-grained features is a major reason leading to such a performance gap and

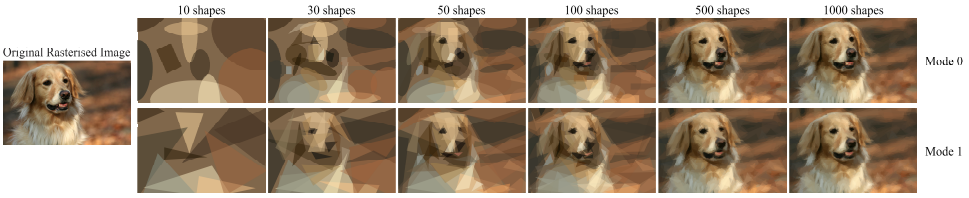


Figure 2: Sample from MiniImageNet (original rasterised image) and HAID-MiniImageNet (abstract images from different abstract levels ranging from 10 to 1,000 shapes).

whether the abstract images generated by primitive shapes could contribute to the vision tasks. To summarise, the main contributions of this study include:

- We introduce a dataset – *HAID* that comprises the different abstract levels of vectorised images generated from the raster images.
- A comprehensive study is presented on how fine-grained level of abstract images affects the ability of conventional vision systems to capture visual semantic information.
- We investigate how much the abstract image representations from different fine-grained levels contribute to downstream tasks with extensive experiments.
- We further discuss the potential benefits of abstract images to contribute to the vision tasks as well as the limitations

2 Related Works

Abstract images. Abstract images, commonly rendered using vector graphics formats such as SVG [23] and TikZ [19], have found widespread use in numerous domains due to their unique properties. Distinct from raster images, vector graphics offer lossless and infinite scalability, moreover, they are text-based, which facilitates both generation and subsequent editing. Scalable Vector Graphics (SVG) image, an XML-based format, is primarily considered in this study due to its convenience of third-party support and demonstration. Although abstract images enjoy certain advantages, their use mostly remains in presenting simple abstract information or logical relationships instead of fine features as high-resolution rasterised images.

Representation learning of vector graphics and datasets. The previous deep learning studies on vector graphics and their datasets generally focused on the generation task [3, 4, 12, 16, 24, 26, 34], the early seminal works like SVG-VAE [16] and DeepSVG [4], pioneered the ways of generating SVG images, and built the datasets that contained SVG fonts and icons. Subsequent studies further advanced these methods to improve performance as well as to broaden the scope of tasks related to vectorised images. For instance, by distilling from the powerful diffusion models, VectorFusion [12] is capable of directly generating SVG images from text instructions. Further, VGbench [34] leverages the Large Language Model (LLM) to endow the model with both visual understanding and generating abilities for vector graphics. The datasets for both works are formed as image-text pairs collected from past works or the Internet. Very recently, StarVector [26] presented a foundation model for SVG generation as well as a new large-scale dataset. However, the datasets above are mostly built for the universal utilisation of vectorised graphics, and their images are often single-object and high-abstract which is disconnected from reality.

With the remarkable progress achieved by LLMs, some studies try to utilise the textual property of vector graphics to endow the visual understanding ability to the LLM. Work [2]

treats the SVG images as the bridge between image-text and enables the LLM in a variety of visual semantic understanding tasks. Moreover, the work [29] tries to use code-based images to reveal whether the LLM can 'see' and 'draw'. Further, they use the code-based images drawn by LLM to train the vision system, which eventually demonstrates the ability to understand high-level visual semantic information from raster images. Despite all these studies exhibiting that abstract images can provide the visual semantic information for representation learning, a distinct performance gap persists between representations derived from pixel-level images and those obtained from abstract, code-based images.

Primitive. Primitive [9] is a tool to generate abstract images from raster images. Different from VTracer [30] or Potrace [28], Primitive iteratively adds primitive shapes to a canvas to approximate the original raster image. Specifically, the algorithm randomly generates candidate primitive shapes at each iteration and then uses a hill-climb-based algorithm to repeatedly mutate these shapes, choosing the one with the best score evaluated by Root Mean Square Error (RSME) as the target shape to be added to the canvas. The number of iterations is equivalent to the target number of shapes of abstract images. The details of how the Primitive generates the shape-based images are shown in fig. S4 of supplementary material. As Primitive can set different numbers and types of SVG primitive shapes to generate the target images, it is particularly well-suited in this project for simulating different abstraction levels in vectorised images. The effect of Primitive can be viewed in fig. 2.

3 Dataset

To better discuss the questions in section 1, we introduce the ***H**ierarchical **A**bstraction **I**mage **D**ataset (HAID)*, which comprises SVG images generated at multiple levels of abstraction from open-source datasets using the Primitive [9]. Specifically, the number of shapes employed during the generation process determines the fine-grained level of the SVG images; as the number of shapes increases, the depicted objects as well as fine-grained details become increasingly recognisable from human perception (see fig. 2). The dataset offers two primary advantages: *(1) a one-to-one correspondence between raster images and their corresponding SVG images, and (2) the availability of multiple abstraction levels for each raster image.* We evaluate the differences between representations learned from pixel-level images and corresponding SVG images on three conventional computer vision tasks: image classification, semantic segmentation, and object detection. For convenience, we call the subset within HAID corresponding with a specific open-source dataset as HAID-(name of the dataset), *e.g.* HAID-MiniImageNet.

3.1 Classification

To obtain representations of code-based images and compare them with those derived from raster images, we generate SVG images from three open-source datasets. In this study, we primarily consider three datasets: MiniImageNet [31], Caltech-256 [10] and CIFAR-10 [14]. An overview of the sample images generated from three datasets is shown in the fig. 1.

HAID-MiniImageNet is generated from MiniImageNet [31] using Primitive [9] with various numbers of shapes to simulate different levels of abstraction, ranging from 10 to 1000 shapes. For each abstract level, similar to MiniImageNet, HAID-MiniImageNet contains 60,000 images across 100 categories. Sample SVG images are presented in fig. 2. The two datasets are divided into training, validation, and testing sets in an 8:1:1 ratio in the same way. We select two options to generate the images by Primitive, using all types of shapes (mode 0) and using triangle shapes (mode 1). The reason we additionally generate the im-

ages constructed by triangle only is because this type of images show the lowest capacity which could be a potentially efficient form of abstract images.

HAID-Caltech-256 is generated from Caltech-256 [10] to support the classification task. The abstract levels of HAID-Caltech-256 range from 10 to 100 shapes. Both the original and abstract datasets are partitioned into training and validation sets using a 9:1 ratio.

To comprehensively investigate the effect of image complexity, HAID-CIFAR-10, which is generated from CIFAR-10 [14] and characterised by comparatively simple images, is included, with the abstract levels similarly set between 10 and 100 shapes. The dataset splitting follows the official strategy.

3.2 Object detection & segmentation

We further investigate whether the representations can contribute to the downstream tasks of semantic segmentation and object detection. Pascal VOC 2012 dataset [8] is used in these tasks. Following the official dataset split, the segmentation task utilises 1,464 images for training and 1,449 images for validation, while the object detection task utilises 5,717 training images and 5,823 validation images.

4 Experiments

To comprehensively explore the issues outlined in the section 1, we first employ a third-party pretrained model to compare the difference between abstract and raster images, then, we evaluate model performance on HAID across three tasks: image classification, semantic segmentation, and object detection. In the classification task, we investigate whether traditional vision architectures can capture high-level semantic information from abstract images and how performance varies across different levels of abstraction. Subsequently, we employ the representations learned from the classification task as backbones and fine-tune the models for downstream tasks, assessing their contribution to enhanced performance in segmentation and object detection tasks.

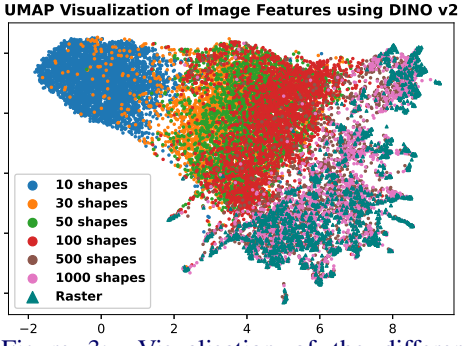


Figure 3: Visualisation of the difference across the abstract images and raster images in the embedding space by DINO v2.

4.1 Difference across the abstract levels

First, we investigate how many differences there are between abstract levels in the perspective of deep learning representations. We randomly sampled 4,000 image pairs—each consisting of an original MiniImageNet raster image and its corresponding abstract SVG versions across all abstract levels. We encoded these images using the DINO v2 [20] and applied UMAP [18] to project the resulting embeddings into two dimensions for visualisation, as shown in fig. 3. From the visualisation results, embeddings of highly abstract images (*e.g.* 10 or 30 shapes) form clusters that lie far from the raster-image cluster, indicating substantial representational differences at coarse abstraction. As the number of primitives increases, the abstract-image clusters move steadily closer to the raster-image cluster, demonstrating that higher-shape-count abstractions maintain higher fidelity of semantics than lower ones. By the 500–1000 shape levels, abstract embeddings overlap significantly with raster embeddings, suggesting near-parity in semantic content despite the vectorised input format.

This trend confirms our intuition: coarse abstractions omit fine details and are thus distinct from pixel-based representations, however, increasing the granularity of primitive shapes progressively bridges the gap. Consequently, the "distance" in embedding space represents the semantic fidelity of abstracted images relative to their raster counterparts.

4.2 How significant are fine-grained features?

Building on the previous visualisation, we next investigate how increasing abstraction—and the corresponding loss of fine-grained details—impacts visual representation learning. To provide extensive studies of these issues, we decompose the problems into two sub-questions: 1) What is the performance of representation learning, and how does it compare to that achieved using raster images? 2) Will the performance be more comparable based on the low-resolution raster images that are difficult to display the fine-grained features? To answer these questions, we design a series of classification tasks to provide a comprehensive discussion. We primarily consider two conventional vision systems, ResNet50 [11] and MobileNetv2 [27], to extract the semantic features. Our experiments utilise HAID-MiniImageNet, HAID-Caltech-256, and HAID-CIFAR-10, alongside their corresponding raster datasets, to enable a full comparison. All experiments are implemented in PyTorch and executed on an NVIDIA A100 40GB GPU.

Comparing with raster images, how good can it be? We discuss the difference between representations derived from raster images and those obtained from abstract images in this part. First, we establish baseline performance by training ResNet50 and MobileNetv2 on the original MiniImageNet dataset. Next, we train the models with the same architectures on the HAID-MiniImageNet dataset across six abstraction levels (10, 30, 50, 100, 500, and 1,000 shapes) and assess the performance on test sets corresponding to each level. Additionally, to examine the effect of training data volume on representation quality, we randomly select four subsets containing fewer training samples from the training set of HAID-MiniImageNet, ranging from 20% to 80% of the full training set.

For the training recipe, since we try to evaluate the difference between raster and abstract images rather than explore the best performance, simple hyperparameter settings are applied in our experiments. We use AdamW [17] as the optimiser with the initial learning rate of 0.0001, and set batch size to 256. We also considered the data augmentations including: Random Resized Cropping, Random Horizontal Flipping, Random Augment [6], and Random Erasing [33] (more details are in section S2.1 of supplementary material). The training recipe is shared across all the experiments in this section. The final results are presented in the fig. 4, and the specific results are shown in tables S1 and S2 of supplementary material.

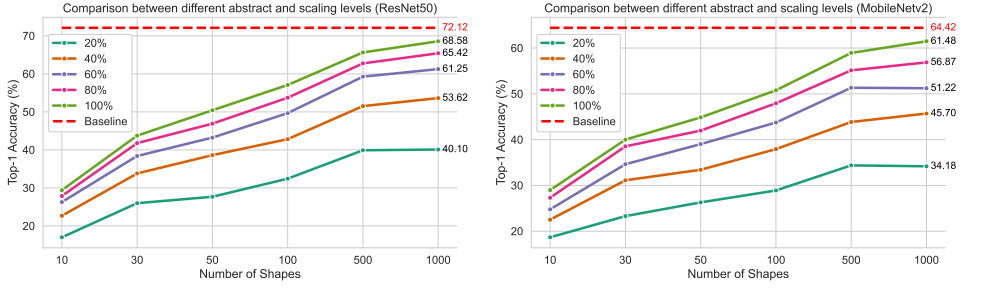


Figure 4: Comparison between representations learned from MiniImageNet and HAID-MiniImageNet across abstract levels (10–1,000 shapes) and scaling factors (20%–100%).

Our results indicate that as the level of fine-grained detail in the abstract images increases,

the learned representations are better able to understand high-level semantic information, with performance gradually approaching that of the raster image baseline. In particular, SVG images generated with 500 and 1,000 shapes yield representations that are highly comparable to those derived from raster images. Conversely, at high abstraction levels (*e.g.* 10 and 30 shapes), a pronounced performance gap is observed, which is expected given the inherent difficulty in recognising highly abstracted images even for humans. Additionally, our scaling experiments reveal that increasing the number of training samples further enhances the performance of the learned representations.

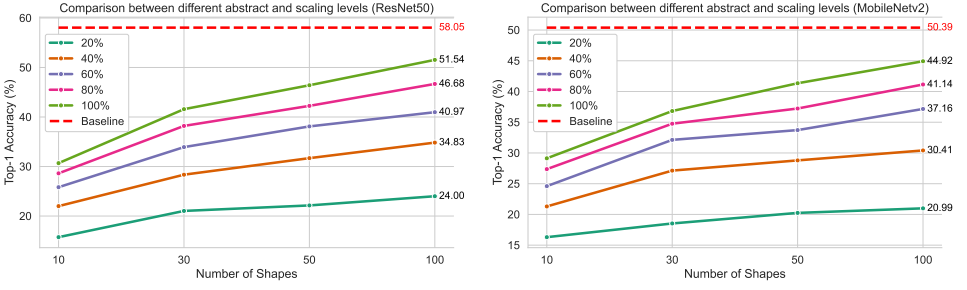


Figure 5: Comparison between representations learned from Caltech-256 and HAID-Caltech-256 across various abstract levels (10–100 shapes) and scaling factors (20%–100%).

We also evaluate the representation performance on another dataset to support our perspective. Similar to the experiments on HAID-MiniImageNet, we train ResNet50 and MobileNetv2 on the HAID-Caltech-256 across four abstraction levels (10, 30, 50, and 100 shapes). The baseline is built by training on the raster images of Caltech-256. The rest settings remain the same with the experiments on HAID-MiniImageNet.The results on HAID-Caltech-256 and Caltech-256, which are shown in fig. 5, follow the trend of the results from HAID-MiniImageNet, demonstrating once again that the representations can better understand high-level semantics on images with low abstractions.

We also measured the difference between the abstractions with two different generation modes (abstractions with all types of shapes and triangles only). However, very slight differences are observed between them compared with the differences from abstract levels. So, the discussion regarding this part is narrated in section S2.2 of the supplementary material.

Recognizing small pictures To further investigate whether the difficulty in demonstrating fine-grained details is the primary factor influencing performance, we employ the CIFAR-10 dataset [14], which contains low-resolution images which are also difficult to present visual details. A four-layer convolutional neural network is used to extract features from the images. The network is trained from scratch on HAID-CIFAR-10 at various abstraction levels for 10 epochs, using Adam optimiser [13] with 0.001 learning rate (more details are in the supplementary material), and the top-1 classification accuracy of the resulting representations is evaluated. The results are summarised in table 1.

Table 1: Accuracy on CIFAR-10 and HAID-CIFAR-10

Abstract level	**10**	**30**	**50**	**100**	**Raster**
Top-1 Acc	60.01%	67.02%	68.48%	70.17%	72.10%

Notably, although a small performance gap remains, the performance gap within highly abstracted levels area between raster images and abstract images is significantly reduced. Combined with previous experimental results, this observation suggests that the inability to

capture fine-grained details is a major factor contributing to the performance gap between representations derived from raster images and those obtained from code-based images.

4.3 Can the representations further contribute the downstream tasks?

We further investigated how much these representations can contribute to downstream tasks by examining two benchmarks: semantic segmentation and object detection.

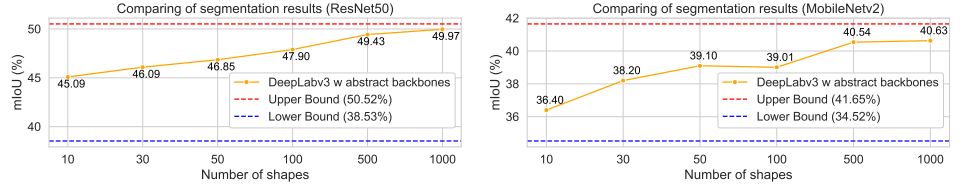


Figure 6: Semantic Segmentation results of DeepLabv3 with backbones and two baselines, upper bound refers to the model initialised with backbone pretrained on MiniImageNet, lower bound refers to the model with random initialisation.

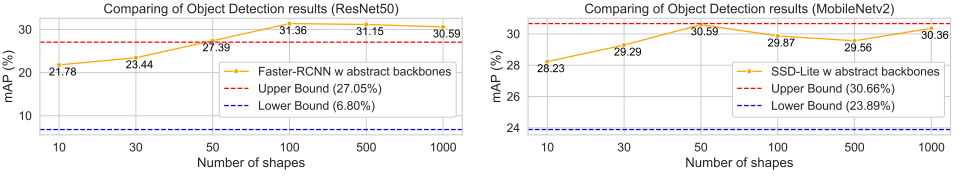


Figure 7: Object Detection results of Faster R-CNN and SSD-Lite with backbones and two baselines. The upper bound refers to the model initialised with a backbone pretrained on MiniImageNet, and the lower bound refers to the model with random initialisation.

Semantic segmentation. To evaluate if the abstract image representations can contribute to the downstream task of semantic segmentation, we utilise backbones derived from models trained on both MiniImageNet and HAID-MiniImageNet (see section 4.2). DeepLabv3 [5] is considered the framework for the segmentation tasks. For comprehensively measuring the contributions, we also set two performance baselines, one using the backbone pretrained on MiniImageNet, and another without any initialisation from pretrained backbone.

The models are trained using the AdamW optimiser with an initial learning rate of 0.0009, batch size of 8, for 200 epochs. We compared the results of DeepLabv3 models with both ResNet50 and MobileNetv2 backbones in fig. 6. The numerical differences between different model performances and upper and lower bounds are presented in table S3.

From the results, initialising the network weights from pretrained abstract backbones shows an increasing performance trend as the descending of abstraction level. Notably, regardless of the specific abstraction level employed, the contributions of these representations are evident, demonstrating that such representations can further contribute to segmentation tasks, even if such tasks strongly rely on fine-grained features capturing ability.

Object detection. After evaluating the results from the downstream task, which challenges the pixel-level visual understanding, we then discuss how abstract image representation contributes to the spatial visual understanding by object detection task. Two architectures: SSD-Lite [27] and Faster R-CNN [25] are considered. For SSD-Lite—a lighter variant of SSD [15]—we initialised the weights from MobileNetv2 backbones. The model was trained using the stochastic gradient descent (SGD) optimiser with an initial learning rate of 0.001 and a weight decay of 0.0005 for 120 epochs. In addition, we used Faster R-CNN with a ResNet50

backbone for object detection. This model was trained for 20 epochs using the SGD optimiser with an initial learning rate of 0.005 and a weight decay of 0.0005. The detail setting refers to section S2.1 of supplementary material. fig. 7 compares the difference between the model initialised by the abstract image backbone and two baselines. The specific numerical differences are shown in table S4 of the supplementary material.

The results from object detection demonstrate the same trend as the results from semantic segmentation. Moreover, the performance for some models initialised by representations from abstract images surprisingly exceeds the performance from the raster image representation. These results once again exhibit that representations obtained from abstract images can contribute to downstream tasks, and more than that, Compared with the results from previous segmentation tasks, we can observe that tasks relying on spatial perception, such as object detection, seem to better reflect the advantages of abstract image representation compared to tasks that rely more on fine-grained features.

4.4 Potential benefits of abstract images

In this section, we discuss whether the abstract images can be a potentially effective format to contribute to the vision tasks as well as their limitations. As the previous results demonstrated, representations learned from sufficiently detailed abstract images (*e.g.* those generated with 500 and 1,000 shapes) approach—and in some object-detection scenarios even exceed—the performance of raster-trained representations. Considering the pixel-level images take advantage of using CNN-based models, which are designed for the rasterised images, not abstract images, such results are very promising to further explore how the abstract images can contribute to the vision tasks.

However, two key limitations remained. First, highly abstract images (fewer than 100 shapes) lack critical fine-grained features—such as texture, small edges, or subtle shading—that raster data naturally provides. As a result, performance gaps persist in tasks heavily dependent on such details (*e.g.* semantic segmentation). Second, in this study, we use Primitive to generate the abstract images that approximate the original raster images, despite the resulting images being visually appealing, the redundant shapes may be introduced during the generation process, which leads to unnecessary code fields and increases the capacity of the file. In section S2.3 of supplementary material, we observe a strong correlation between perceptual similarity and image entropy, which shows that under the same abstract level, images enjoying low entropy generally have better perceptual loss on related abstract images, in other words, entropy can be considered as the metric to provide the trade-off between capacity and performance to further improve the efficiency. Moreover, considering that abstract images enjoy code-based format, the keywords of the SVG code can be further compressed and thus benefiting data transmission.

5 Conclusion

In this paper, we investigated abstract images and performed a study on the representations learned from them. Experiments showed that fine-grained detail is one of the main factors leading to the performance gap between representations learned from raster images and those learned from abstract images. On the other hand, as the level of fine-grain increases, the capacity of representation to capture high-level semantic information improves, thereby narrowing such performance differences. Moreover, our downstream task experiments revealed that representations derived from abstract images can effectively contribute to visual tasks, even achieving comparable performance on tasks that are less reliant on fine-grained

details. Given the inherent advantages- including lossless scalability, a compact textual format, and ease of editing the abstract images show significant promise as a novel data form for visual representation learning and related vision tasks.

References

[1] Vladislav Ayzenberg and Stella Lourenco. Perception of an object's global shape is best described by a model of skeletal structure in human infants. *elife*, 11:e74943, 2022.

[2] Mu Cai, Zeyi Huang, Yuheng Li, Utkarsh Ojha, Haohan Wang, and Yong Jae Lee. Leveraging large language models for scalable vector graphics-driven image understanding. *arXiv preprint arXiv:2306.06094*, 2023.

[3] Defu Cao, Zhaowen Wang, Jose Echevarria, and Yan Liu. Svgformer: Representation learning for continuous vector graphics using transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10093–10102, 2023.

[4] Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. *Advances in Neural Information Processing Systems*, 33:16351–16361, 2020.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[7] James H Elder and Ljiljana Velisavljević. Cue dynamics underlying rapid detection of animals in natural scenes. *Journal of Vision*, 9(7):7–7, 2009.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[9] Michael Fogleman. Primitive: Reproducing images with geometric primitives, 2016. URL https://github.com/fogleman/primitive. GitHub repository.

[10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, April 2022.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2023.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, 2009.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[16] Raphael Gontijo Lopes, David Ha, Douglas Eck, and Jonathon Shlens. A learned representation for scalable vector graphics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7930–7939, 2019.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[19] Andrew Mertz and William Slough. Graphics with tikz. *The PracTEX Journal*, 1:1–22, 2007.

[20] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[21] Paul C Quinn, Peter D Eimas, and Michael J Tarr. Perceptual categorization of cat and dog silhouettes by 3-to 4-month-old infants. *Journal of experimental child psychology*, 79(1):78–94, 2001.

[22] Paul C Quinn, Alan M Slater, Elizabeth Brown, and Rachel A Hayes. Developmental change in form categorization in early infancy. *British Journal of Developmental Psychology*, 19(2):207–218, 2001.

[23] Antoine Quint. Scalable vector graphics. *IEEE MultiMedia*, 10(3):99–102, 2003.

[24] Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7342–7351, 2021.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[26] Juan A Rodriguez, Shubham Agarwal, Issam H Laradji, Pau Rodriguez, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images. *arXiv preprint arXiv:2312.11556*, 2023.

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[28] Peter Selinger. Potrace: Transforming bitmaps into vector graphics. https://potrace.sourceforge.net/, 2003. GNU General Public License version 2.0, accessed March 04, 2025.

[29] Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14410–14419, 2024.

[30] The Vision Cortex Research Group. Vtracer: Raster to vector graphics converter. https://github.com/visioncortex/vtracer, 2024. Version 0.6.4, accessed March 04, 2025.

[31] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[32] Johan Wagemans, Joeri De Winter, Hans Op de Beeck, Annemie Ploeger, Tom Beckers, and Peter Vanroose. Identification of everyday objects on the basis of silhouette and outline versions. *Perception*, 37(2):207–244, 2008.

[33] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[34] Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. Vgbench: A comprehensive benchmark of vector graphics understanding and generation for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3647–3659, 2024.

Supplementary Material

S1 Dataset

In here we demonstrate more examples of our dataset. More image pairs of MiniImageNet and HAID-MiniImageNet are shown in fig. S5 and more image pairs of CIFAR-10 anf HAID-CIFAR-10 are shown in fig. S6.

S2 Supplementary experiment content

S2.1 Experiment setting

In here, we supplement the experimental details in section 4, including the settings of hyper-parameters and network architectures. The parameter may be not 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 optimizer with initial learning rate of 0.0001, we use random resize crop to frame the image size into 224×224. We use a series data augmentation strategies, including: Random Horizontal Flipping, Random Augment, and Random Erasing, training for 120 epochs. For the HAID-CIFAR-10 experiments, we only use Adam optimizer with initial learning rate of 0.001 training for 10 epochs.

	Mode 0						
	10	**30**	**50**	**100**	**500**	**1000**	**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 S1: Top-1 Accuracy of ResNet 50 with all shapes (mode0) on MiniImageNet and HAID-MiniImageNet

Semantic Segmentation During training the DeepLabv3 for Semantic Segmentation, we randomly crop the image size as 480×480 from the based size 520×520. We use the AdamW optimizer with 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-RCNN, We use Stochastic Gradient Descent (SGD) as the optimizer with 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

	Mode 0						
	10	**30**	**50**	**100**	**500**	**1000**	**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%

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

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 optimizer with 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 implementing 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 initialized 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.

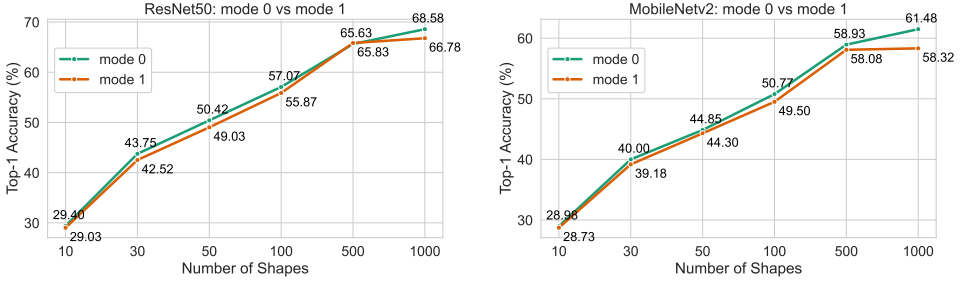
	10	30	50	100	500	1000
DeepLabv3-ResNet50	45.09	46.09	46.85	47.90	49.43	49.97
Compar w LB	+6.56	+7.56	+8.32	+9.38	+10.91	+11.44
Compar w UB	-5.44	-4.43	-3.67	-2.62	-1.09	-0.56
DeepLabv3-MobileNetv2	36.40	38.20	39.10	39.01	40.54	40.63
Compar w LB	1.87	+3.68	+4.58	+4.49	+6.02	+6.10
Compar w UB	-5.44	-4.43	-3.67	-2.62	-1.09	-0.56

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 1000 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	1000
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-RCNN	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

Table S4: Comparing the performances by Mean Average Precision (mAP) of SSD-Lite with MobileNet v2 backbones (up) and Faster-RCNN 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 1000 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

Abstraction with all shape 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. fig. 2 illustrate the comparisons between mode 0 and mode 1, and more samples can be found in fig. S5. 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. S1.



(a). ResNet50 (b). MobileNet v2

Figure S1: Comparison of ResNet50 and MobileNetv2 performance training on all types of SVG primitives (mode 0) and training on triangular primitives only (mode 1).

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 1000 shapes; as observed in fig. S2, 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.

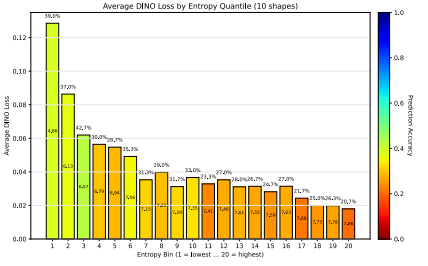
Figure S2: Detailed features comparison: raster image (left), abstract image containing all types of primitive shape (middle), and abstract containing triangle only (right).

S2.3 The correlation between image entropy and abstract images

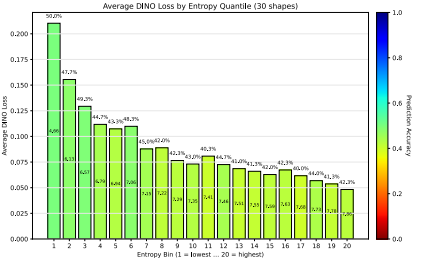
Entropy of the image serves as a quantitative measure of its information content as well as the complexity. We observed that, in the human perceptive, for the simple images (e.g. with single object and unsophisticated background), such as the last row images in fig. S5, are generally recognizable with lower number of shapes than the complex images, such as the second row images in fig. S5. Therefore, we speculate there is a correlation between the information complexity of the original image and levels of its abstractions. To prove this hypnosis, we randomly sampled 4000 images 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 small to large. We also labeled each groups with the perceptual loss (DINO loss specifically) between abstract and original images, their prediction accuracy and mean entropy value. The results are shown in fig. S3,

S3 Primitive

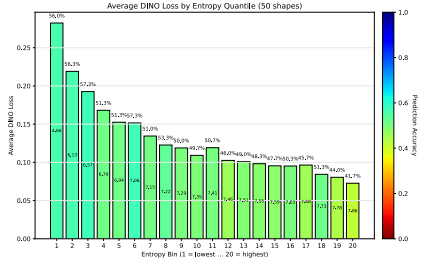
The fig. S4 specifically explained how the Primitive generate the shape-based images from the provided raster images.



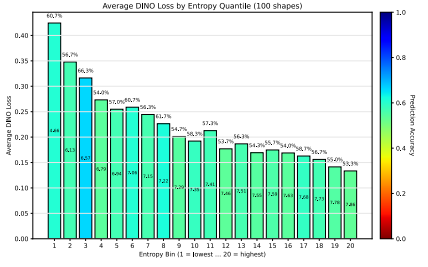
(a). 10 shapes



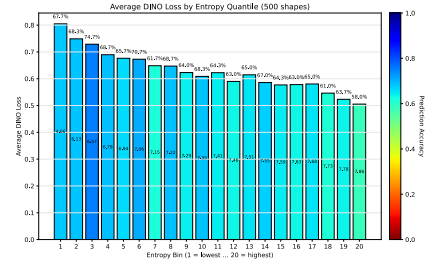
(b). 30 shapes



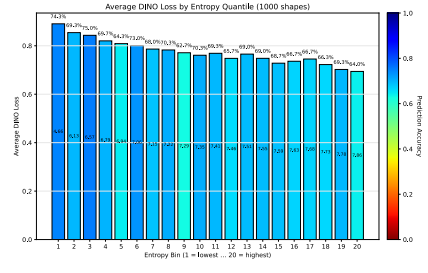
(c). 50 shapes



(d). 100 shapes



(e). 500 shapes



(f). 1000 shapes

Figure S3: The correlation between entropy and DINO loss, each diagram has 20 groups sorted by entropy value from small to large. The color and the value above each bin represent the prediction accuracy of each bin, the value inside of each bin represents the average entropy value of each group.

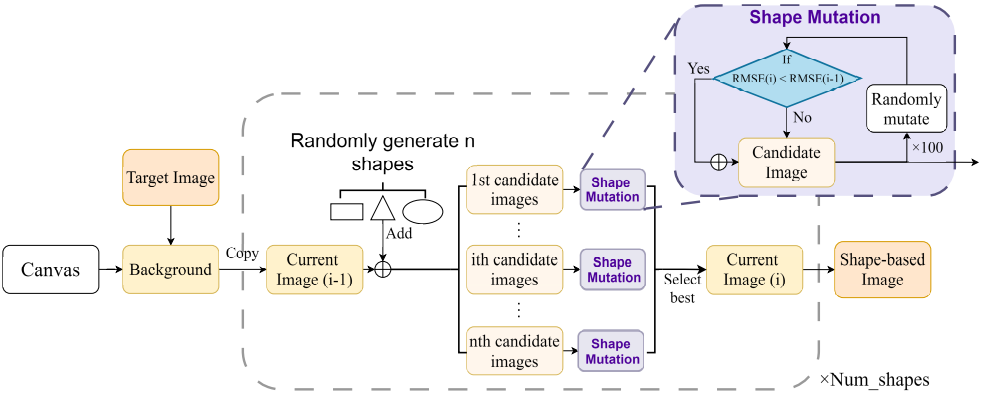


Figure S4: The generating procedure of Primitive.

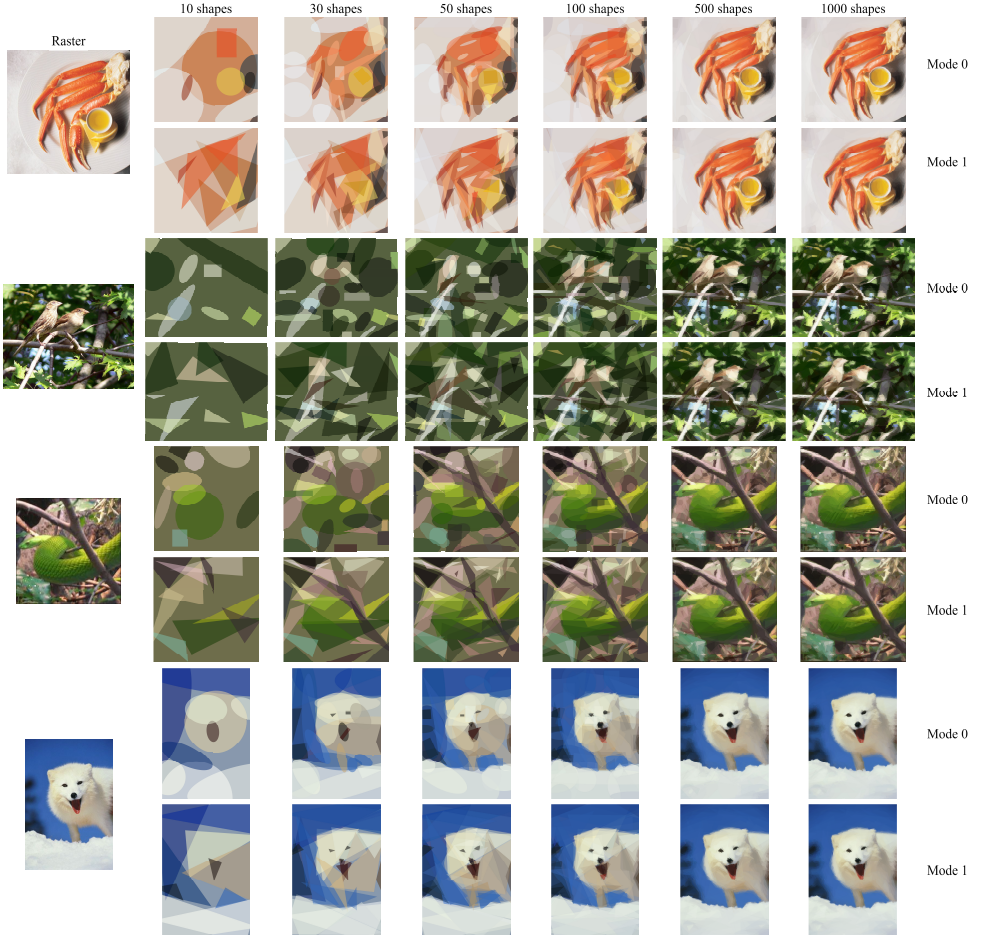


Figure S5: More examples of HAID-MiniImageNet

Figure S6: Example image pairs between CIFAR-10 and HAID-CIFAR-10