Awesome

Tomoki Tanimura (@tanimu)
d-hacks, Jin Nakazawa Lab, SFC, Keio University, M2

Created At: 2021/06/08

Awesome Researches

- Dataset Distillation
- What’s Hidden in a Randomly Weighted Neural Network?
- ImageNet-Trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness
- Are Convolutional Neural Networks or Transformers more like human vision?

How awesome are those researches?

- The point that no one address
- Everyone want to know that
- Sharp eyesight

Dataset Distillation

- Model Distillation is to distill the knowledge of the large model to small one
- Dataset Distillation is to distill the knowledge of the large **Dataset** to small one

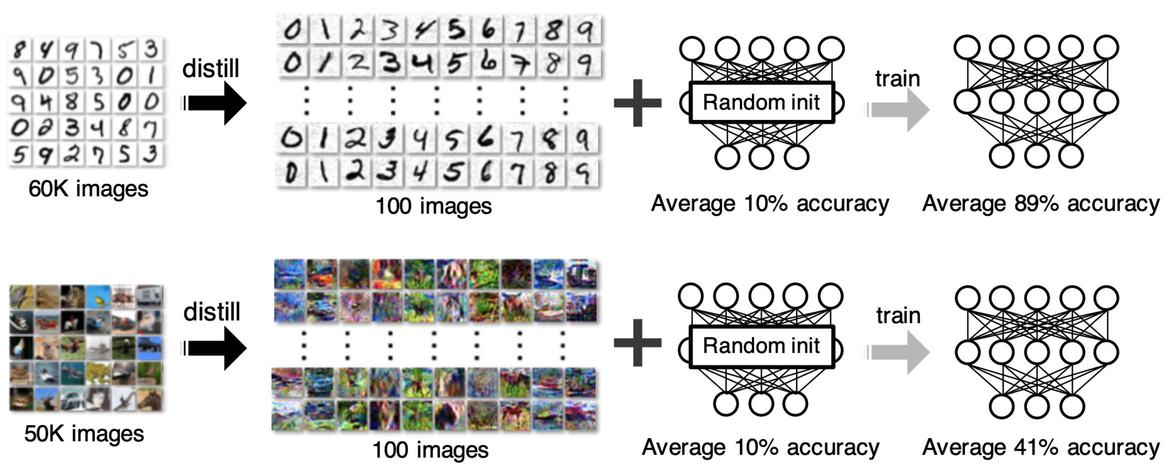


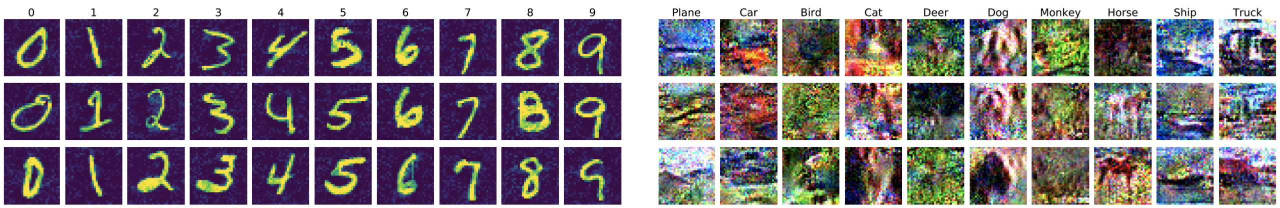
Figure 1: We distill the knowledge of tens of thousands of images into a few synthetic training images called distilled images. On MNIST, 100 distilled images can train a standard LeNet with a random initialization to 89% test accuracy, compared to 99% when fully trained. On CIFAR10, 100 distilled images can train a network with a random initialization to 41% test accuracy, compared to 80% when fully trained. In Section 3.6, we show that these distilled images can efficiently store knowledge of previous tasks for continual learning.

Dataset Distillation / Motivation

- How small can we compress the dataset knowledge?
- Distilled dataset can be used for continuous learning
- Can the model learn by the data sampled from incorrect distribution

But why is dataset distillation interesting? First, there is the purely scientific question of how much data is encoded in a given training set and how compressible it is? Second, we wish to know whether it is possible to “load up" a given network with an entire dataset-worth of knowledge by a handful of images. This is in contrast to traditional training that often requires tens of thousands of data samples. Finally, on the practical side, dataset distillation enables applications that require compressing data with its task. We demonstrate that under the continual learning setting, storing distilled images as memory of past task and data can alleviate catastrophic forgetting (McCloskey and Cohen, 1989).

Dataset Distillation / Result



(a) MNIST. These distilled images can train unknown random initializations to $88.51\% \pm 1.11\%$ test accuracy in 2000 GD steps.

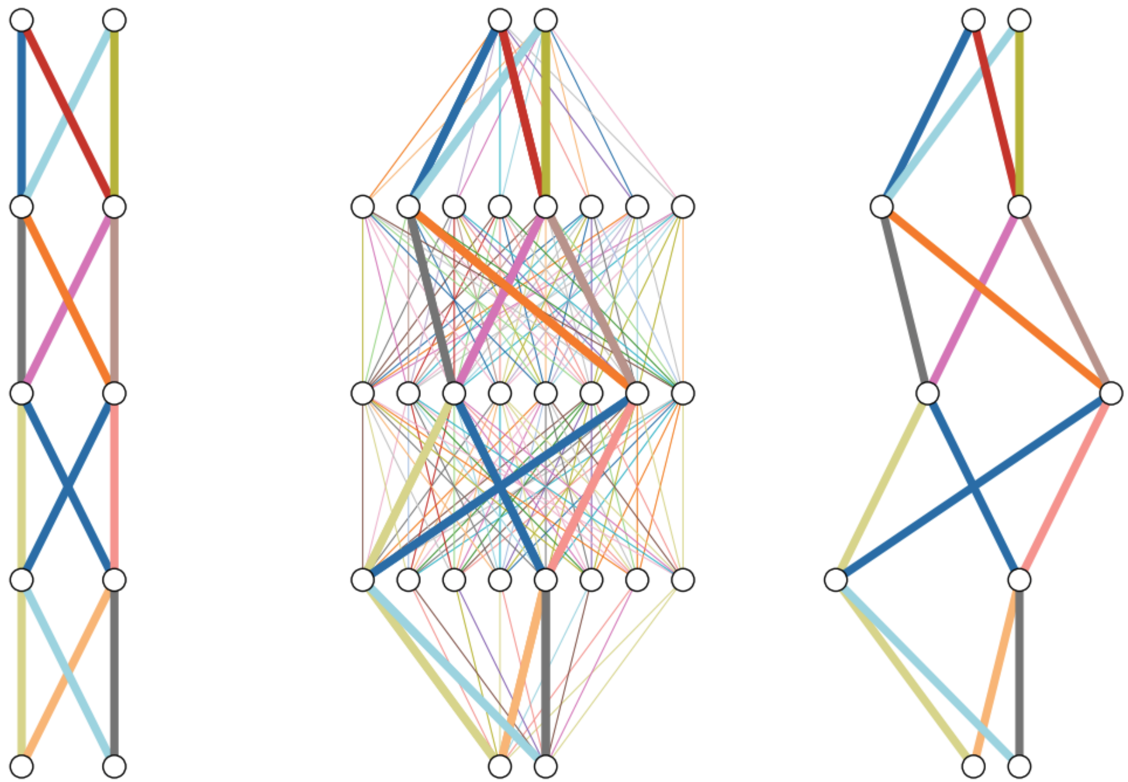
(b) CIFAR10. These distilled images can train unknown random initializations to $41.23\% \pm 0.88\%$ test accuracy in 50 GD steps.

Figure 2: Distilled images trained for *random initialization* a batch of 100 distilled images (ten per class). Only 30 of 100 distilled images are shown here. Please see the appendix for the full result.

What’s Hidden in a Randomly Weighted Neural Network? (Hidden)

- Q. Does random network contains the subnetwork which can achieve good performance
- Find such subnetwork in large network with random weight
- Propose the efficient method to explore hidden subnetwork

Hidden / Overview



A neural network τ which achieves good performance

Randomly initialized neural network N

A subnetwork τ' of N

Hidden / Related works

- "The Lottery Ticket Hypothesis"
  - Large random initialized NN has the subnetwork which has the potential to achieve the same performance as the entire network
- Some algorithms are proposed for finding WINiing-ticket effectively
- "WANNs (Weight Agnosstic Neural Network)"
  - Subnetwork of NN has same weight value can achieve almost same accuray as the trained one
  - But, this paper said we cannot when random initialization

Hidden / Algorithm

- Set the score to each path
- Select the optimal path based on the score
- Update the scores with GD

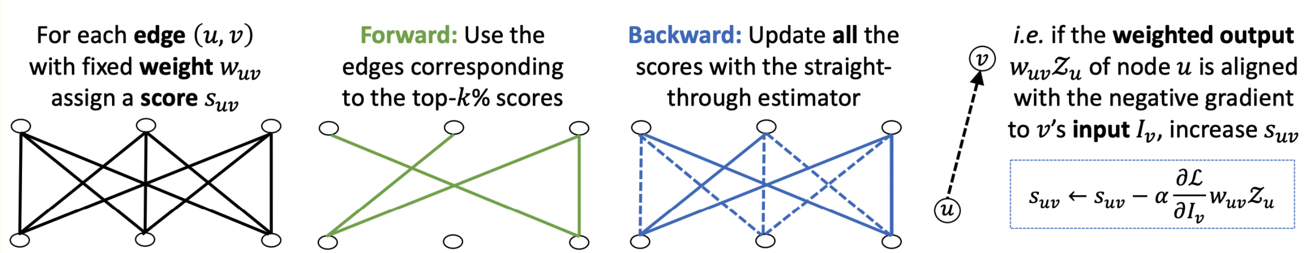


Figure 2. In the `edge-popup` Algorithm, we associate a score with each edge. On the forward pass we choose the top edges by score. On the backward pass we update the scores of all the edges with the straight-through estimator, allowing helpful edges that are "dead" to re-enter the subnetwork. We never update the value of any weight in the network, only the score associated with each weight.

Hidden / Result

Method	Model	Initialization	% of Weights	# of Parameters	Accuracy
Learned Dense Weights (SGD)	ResNet-34 [8]	-	-	21.8M	73.3%
	ResNet-50 [8]	-	-	25M	76.1%
	Wide ResNet-50 [28]	-	-	69M	78.1%
`edge-popup`	ResNet-50	Kaiming Normal	30%	7.6M	61.71%
	ResNet-101	Kaiming Normal	30%	13M	66.15%
	Wide ResNet-50	Kaiming Normal	30%	20.6M	67.95%
`edge-popup`	ResNet-50	Signed Constant	30%	7.6M	68.6%
	ResNet-101	Signed Constant	30%	13M	72.3%
	Wide ResNet-50	Signed Constant	30%	20.6M	73.3%

Table 2. ImageNet [3] classification results corresponding to Figure 8. Note that for the non-dense models, # of Parameters denotes the size of the subnetwork.

ImageNet-Trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness

- CNN see texture rather than shape

Texture / Overview

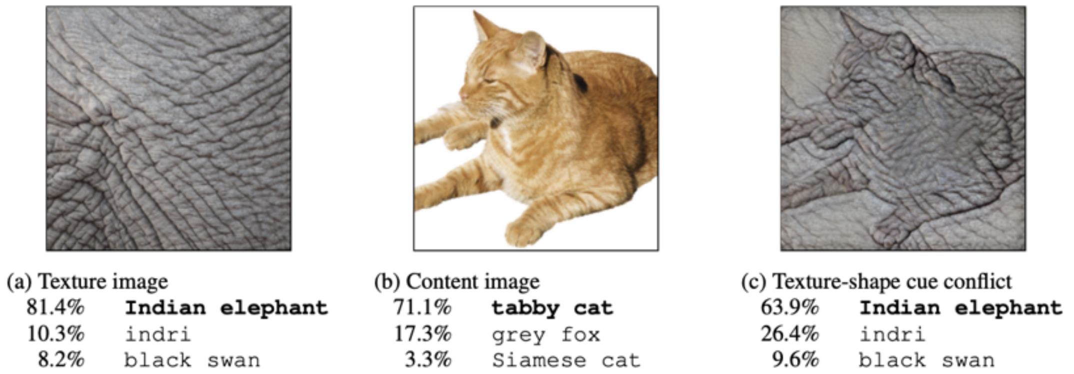


Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

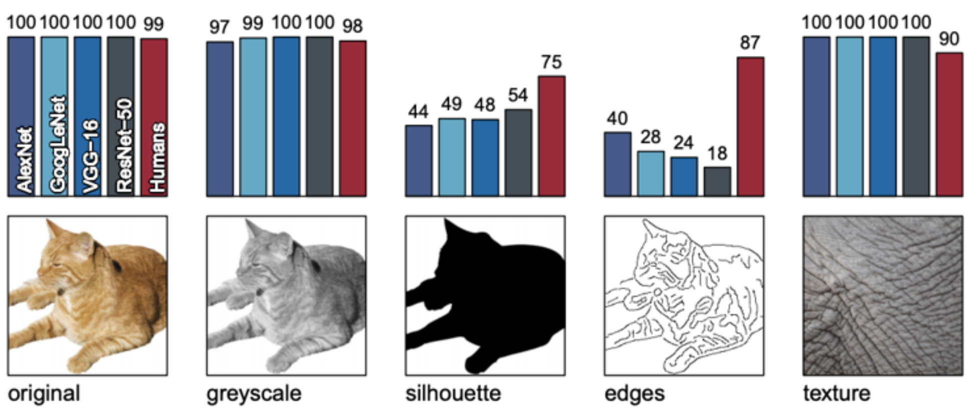
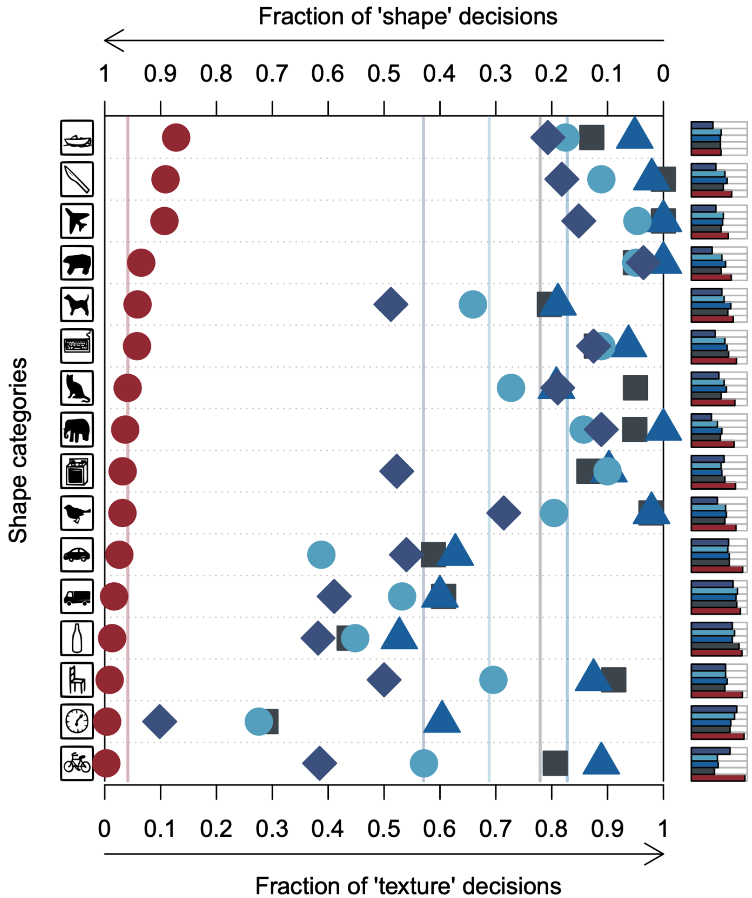


Figure 2: Accuracies and example stimuli for five different experiments without cue conflict.

Texture / Compare CNN and Human

Figure 4: Classification results for human observers (red circles) and ImageNet-trained networks AlexNet (purple diamonds), VGG-16 (blue triangles), GoogLeNet (turquoise circles) and ResNet-50 (grey squares). Shape vs. texture biases for stimuli with cue conflict (sorted by human shape bias). Within the responses that corresponded to either the correct texture or correct shape category, the fractions of texture and shape decisions are depicted in the main plot (averages visualised by vertical lines). On the right side, small barplots display the proportion of correct decisions (either texture or shape correctly recognised) as a fraction of all trials. Similar results for ResNet-152, DenseNet-121 and Squeezenet1_1 are reported in the Appendix, Figure 13.

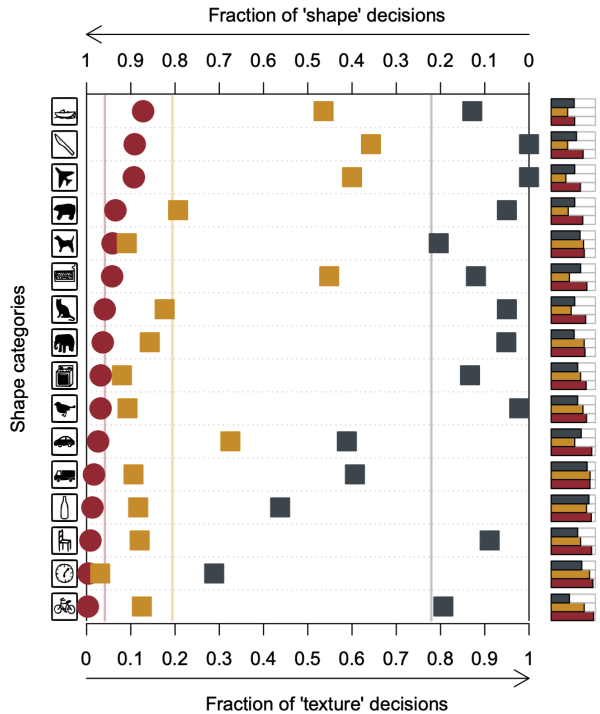


Texture / Stylized dataset

Figure 3: Visualisation of Stylized-ImageNet (SIN), created by applying AdaIN style transfer to ImageNet images. Left: randomly selected ImageNet image of class `ring-tailed lemur`. Right: ten examples of images with content/shape of left image and style/texture from different paintings. After applying AdaIN style transfer, local texture cues are no longer highly predictive of the target class, while the global shape tends to be retained. Note that within SIN, every source image is stylized only once.

Texture / Performance of SIN

Figure 5: Shape vs. texture biases for stimuli with a texture-shape cue conflict after training ResNet-50 on Stylized-ImageNet (orange squares) and on ImageNet (grey squares). Plotting conventions and human data (red circles) for comparison are identical to Figure 4. Similar results for other networks are reported in the Appendix, Figure 11.



name	training	fine-tuning	top-1 IN accuracy (%)	top-5 IN accuracy (%)	Pascal VOC mAP50 (%)
vanilla ResNet	IN	-	76.13	92.86	70.7
	SIN	-	60.18	82.62	70.6
	SIN+IN	-	74.59	92.14	74.0
Shape-ResNet	SIN+IN	IN	**76.72**	**93.28**	**75.1**

Table 2: Accuracy comparison on the ImageNet (IN) validation data set as well as object detection performance (mAP50) on PASCAL VOC 2007. All models have an identical ResNet-50 architecture. Method details reported in the Appendix.

Are Convolutional Neural Networks or Transformers more like human vision?

- How about ViT?

Transformer / Attention-based Networks

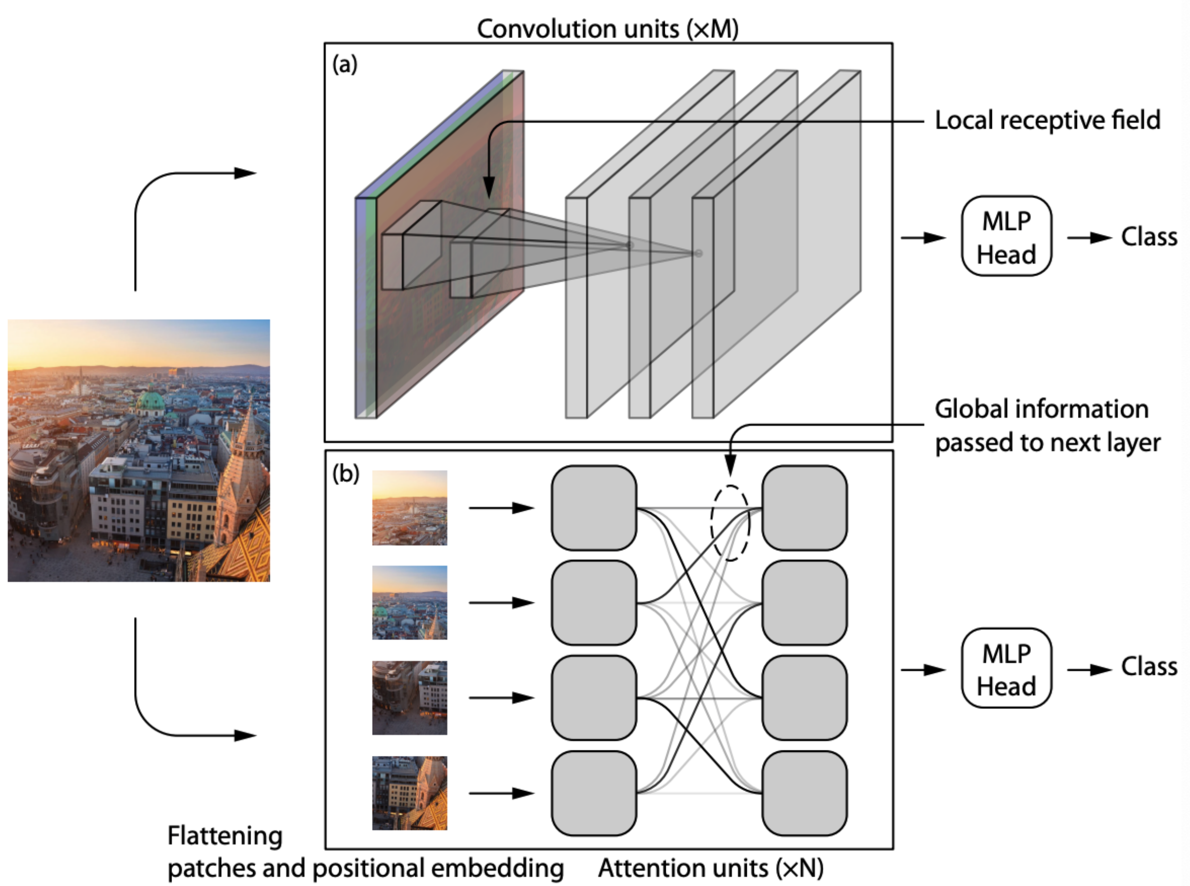


Fig. 2: Bird's eye view of (a) convolutional and (b) attention-based networks

Transformer / Shape vs. Texture

- Finetune with effective data augmentation

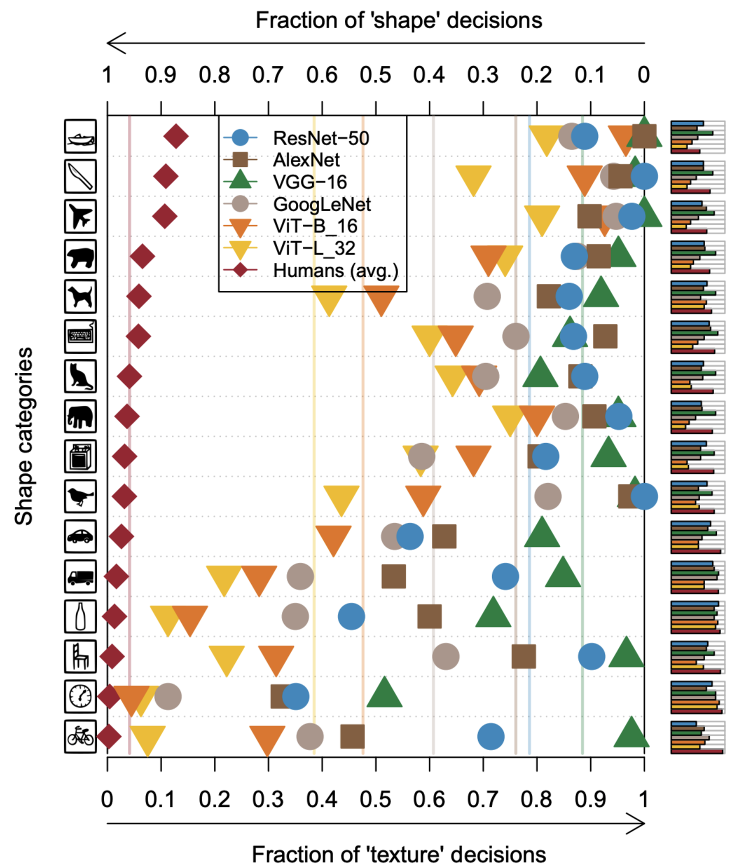


Fig. 5: Shape bias for different networks for the SIN dataset (Geirhos et al., 2019). Vertical lines indicate averages.

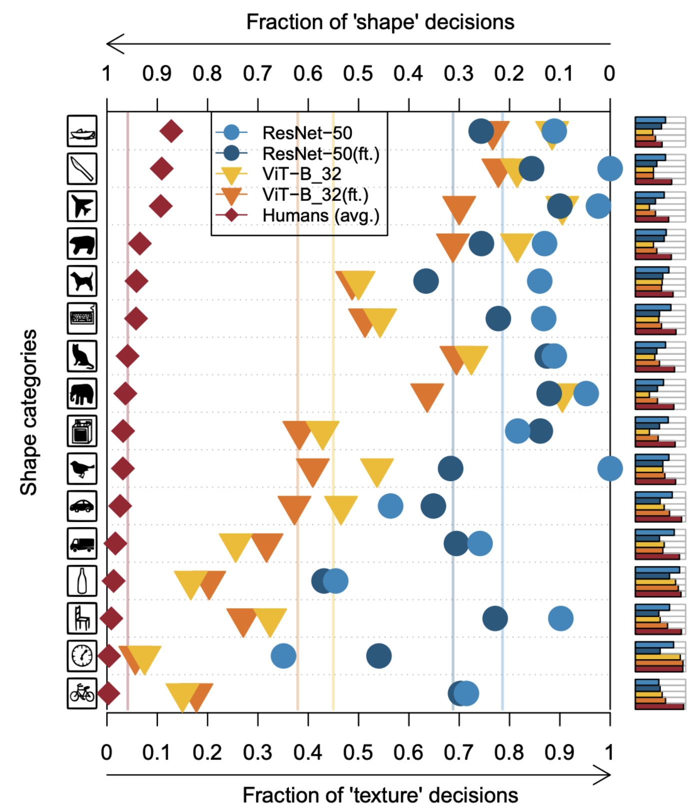


Fig. 7: Shape bias for ResNet and ViT before and after fine-tuning. Vertical lines indicate averages.

Summary

- Research what you wanna know
- Sharp eyesight
- ?