Accelerating Dataset Distillation via Model Augmentation

Lei Zhang[1*] Jie Zhang[1*] Bowen Lei[2] Subhabrata Mukherjee[3]
Xiang Pan[4] Bo Zhao[5] Caiwen Ding[6] Yao Li[7] Dongkuan Xu[8†]
[1]Zhejiang University [2]Texas A&M University [3]Microsoft Research
[4]New York University [5]Beijing Academy of Artificial Intelligence [6]University of Connecticut
[7]University of North Carolina, Chapel Hill [8]North Carolina State University
{zl_leizhang, zj_zhangjie}@zju.edu.cn dxu27@ncsu.edu

This supplementary material provides more details on the method and experiments, including a detailed explanation of early-stage models from gradient perspective (Sec. 1), datasets (Sec. 2), network architectures (Sec. 3), and additional experiment results (Sec. 4).

Parameter	Shape	Layer hyper-parameter
pooling.avg	[2]	stride=2, padding=0
conv1.weight	[3, 128, 3, 3]	stride=1, padding=1
conv1.bias	[128]	N/A
conv2.weight	[128, 128, 3, 3]	stride=1, padding=1
conv2.bias	[128]	N/A
conv2.weight	[128, 128, 3, 3]	stride=1, padding=1
conv2.bias	[128]	N/A
norm.group	[128, 128]	eps=1e-5, affine=True
norm.group	[128, 128]	eps=1e-5, affine=True
norm.group	[128, 128]	eps=1e-5, affine=True
pooling.avg	[2]	stride=2, padding=0
pooling.avg	[2]	stride=2, padding=0
pooling.avg	[2]	stride=2, padding=0
fc.weight	[2048, 10]	N/A
fc.bias	[10]	N/A

Table 1. Detailed information of the ConvNet-3 architecture used in our experiments for CIFAR-10 and CIFAR-100.

1. Why Early-stage Models Work Better

From the gradient perspective, early-stage models can produce diverse and large-magnitude gradients which are more effective for gradient matching. Recent studies [2, 3] demonstrate that gradient dynamically converges to a very small subspace after a short period of training. The models and synthetic data will be alternatively updated after sampling the model in the dataset distillation process. Thus, successive gradients will be used for updating synthetic data. Training with small and similar successive gradients produced by well-trained models will result in worse synthetic data. This is consistent with the finding in the previous work, DSA [8], in which utilizes data augmentation to enlarge the gradient magnitude for better gradient matching.

Parameter	Shape	Layer hyper-parameter
conv1.weight	[3, 64, 7, 7]	stride=1, padding=3
pool1.avg	[4, 4]	stride=4, padding=0
conv2.weight	[64, 64, 3, 3]	stride=1, padding=1
norm1.group	[64, 64]	eps=1e-5, affine=True
conv2.weight	[64, 64, 3, 3]	stride=1, padding=1
norm1.group	[64, 64]	eps=1e-5, affine=True
conv3.weight	[64, 128, 3, 3]	stride=1, padding=1
norm2.group	[128, 128]	eps=1e-5, affine=True
conv4.weight	[128, 128, 3, 3]	stride=1, padding=1
norm2.group	[128, 128]	eps=1e-5, affine=True
conv5.weight	[64, 128, 1, 1]	stride=1, padding=0
pool2.avg	[2, 2]	stride=2, padding=0
norm2.group	[128, 128]	eps=1e-5, affine=True
conv6.weight	[128, 256, 3, 3]	stride=1, padding=1
norm3.group	[256, 256]	eps=1e-5, affine=True
conv7.weight	[256, 256, 1, 1]	stride=1, padding=1
norm3.group	[256, 256]	eps=1e-5, affine=True
conv8.weight	[128, 256, 1, 1]	stride=1, padding=0
pool2.avg	[2, 2]	stride=2, padding=0
norm3.group	[256, 256]	eps=1e-5, affine=True
conv9.weight	[256, 512, 3, 3]	stride=1, padding=1
norm4.group	[512, 512]	eps=1e-5, affine=True
conv10.weight	[512, 512, 3, 3]	stride=1, padding=1
norm4.group	[512, 512]	eps=1e-5, affine=True
conv11.weight	[256, 512, 1, 1]	stride=1, padding=1
pool2.avg	[2, 2]	stride=2, padding=0
norm4.group	[512, 512]	eps=1e-5, affine=True
pool5.avg	[7, 7]	stride=7, padding=0
fc.weight	[512, 10]	N/A
fc.bias	[10]	N/A

Table 2. Detailed information of the ResNetAP-10 architecture used in our experiments for ImageNet.

Img/							CIFAR-10				
Cls	DM	DSA	CAFE	TM	IDC$_1$	IDC$_5$	Ours$_5$	IDC$_{10}$	Ours$_{10}$	IDC$_{20}$	Ours$_{20}$
1	26.0	28.2	30.3	46.3	50.6	49.5 (0.4)	**49.2 (0.4)**	49.0 (0.3)	**48.7 (0.5)**	48.6 (0.3)	**48.0 (0.3)**
10	48.9	52.1	46.3	65.3	67.5	66.2 (0.3)	**67.1 (0.2)**	65.0 (0.3)	**66.5 (0.1)**	63.7 (0.2)	**65.1 (0.1)**
50	63.0	60.6	55.5	71.6	74.5	73.3 (0.3)	**73.8 (0.1)**	72.0 (0.6)	**73.1 (0.1)**	71.1 (0.2)	**71.7 (0.3)**

(a) Results on CIFAR-10

Img/							CIFAR-100				
Cls	DM	DSA	CAFE	TM	IDC$_1$	IDC$_5$	Ours$_5$	IDC$_{10}$	Ours$_{10}$	IDC$_{20}$	Ours$_{20}$
1	11.4	13.9	12.9	24.3	25.1	24.9 (0.5)	**29.8 (0.2)**	24.7 (0.1)	**29.6 (0.1)**	24.4 (0.1)	**29.1 (0.1)**
10	29.7	32.3	27.8	40.1	45.1	44.1 (0.2)	**46.2 (0.3)**	43.1 (0.2)	**45.6 (0.4)**	41.6 (0.2)	**45.0 (0.0)**
50	43.6	42.8	37.9	47.7	-	-	**52.6 (0.4)**	-	**52.3 (0.2)**	-	**52.2 (0.1)**

(b) Results on CIFAR-100

Table 3. Comparing performance of dataset distillation methods on CIFAR-10 and CIFAR-100. We report Top-1 test accuracy of test models ConvNet-3 trained on condensed datasets. Img/Cls means the number of images per class of the condensed dataset. We evaluate each task with 3 repetitions and denote the standard deviations in the parenthesis. We compare the same acceleration levels between IDC and our method on 5×, 10×, and 20×.

Dataset	Img/Cls	DM	DSA	IDC$_1$	IDC$_5$	Ours$_5$	IDC$_{10}$	Ours$_{10}$
ImageNet-10	10	52.3	52.7	72.8 (0.6)	72.3 (0.7)	**74.6 (0.4)**	72.3 (1.0)	**74.0 (0.4)**
	20	59.3	57.4	76.6 (0.4)	74.7 (0.4)	**76.3 (1.0)**	74.7 (0.4)	**75.2 (1.0)**
ImageNet-100	10	22.3	21.8	46.7 (0.2)	-	**48.4 (0.3)**	-	-
	20	30.4	30.7	53.7 (0.9)	-	**56.0 (0.5)**	-	-

Table 4. Comparing performance of dataset distillation methods on ImageNet-10 and ImageNet-100. We report Top-1 test accuracy of test models ResNetAP-10 trained on condensed datasets. We evaluate each task with 3 repetitions and denote the standard deviations in the parenthesis.

2. Datasets

ImageNet-subset. Following previous works [5, 7], we evaluate our method on ImageNet-subset, which borrows a subclass list containing 100 classes from [7]. We use the first 10 classes from the list in our ImageNet-10 experiments and the complete list in our ImageNet-100 experiment. The images in ImageNet-subset are preprocessed to a fixed size of 224 × 224 using resize and center crop functions.

3. Networks

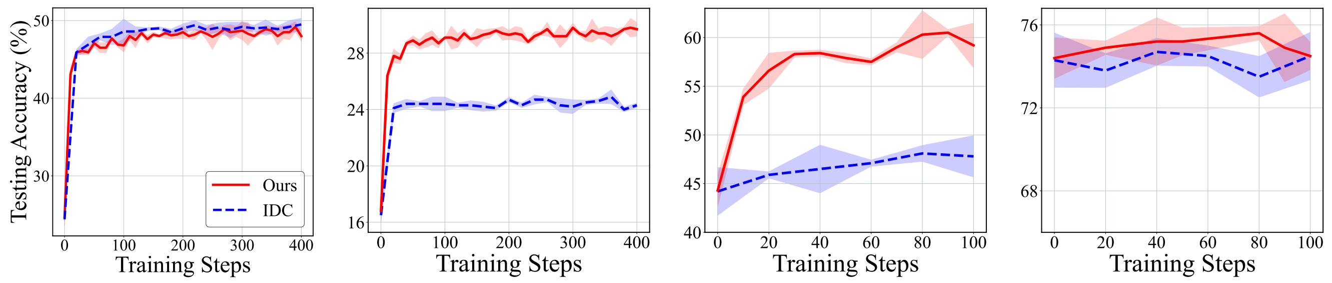
Staying with precdent [1, 5, 8, 10], we employ a simple ConvNet-3 architecture for CIFAR-10 and CIFAR-100 [6] dataset and a modified ResNet-10 [4] architecture ResNetAP-10 for ImageNet-subset. As shown in Tab. 1, ConvNet-3 consists of several convolutional tasks, each containing a 3 × 3 convolutional layer with 129 filters, Instance normalization, RELU and 2 ×2 average pooling with stride 2. We also demonstrate the detailed architecture of ResNetAP-10 in Tab. 2. ResNetAP-10 is modified on ResNet-10, which replaces strided convolution as average pooling for downsampling.

4. Additional Experiments

4.1. More Experiment results on Datasets

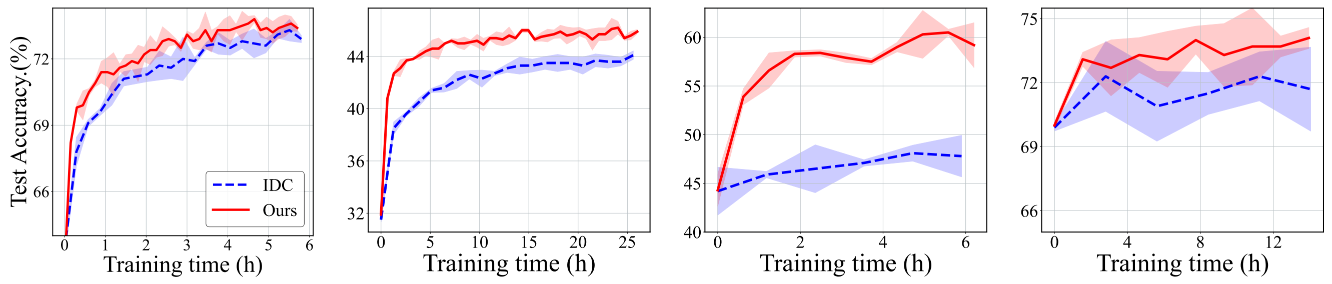
CIFAR-10 & CIFAR-100. Our method achieves a better trade-off in task performance vs. acceleration of training compared to other state-of-the-art baselines on CIFAR-10 and CIFAR-100. As shown in Tab. 3, our method consistently outperforms SOTA method IDC under 5×, 10×, and 20× speed-ups and other baselines without acceleration. This verifies the efficiency of our method, which requires less training time and computation resources to achieve comparable or surpass performances of leading baselines.

ImageNet. Apart from CIFAR-10 and CIFAR-100, we evaluate the performance of our method on large-scale and high-resolution dataset ImageNet. As shown in Tab. 4, our method significantly outperforms all the baselines across various numbers number of classes and surpasses the leading method IDC under 5× and 10× speed-ups. Existing methods perform poorly on the high-resolution datasets, such as ImageNet. However, our method not only achieves a better performance but also improves the efficiency on both low- and high-resolution datasets. This demonstrates



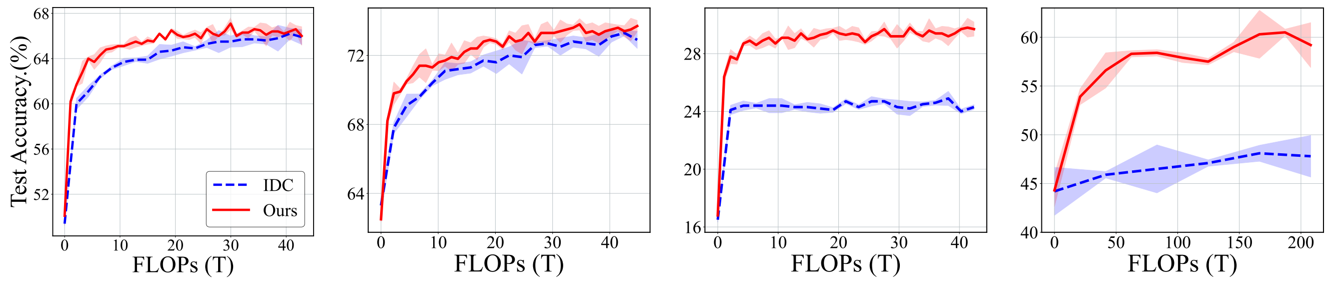
(a) CIFAR-10 (Img/Cls=1). (b) CIFAR-100 (Img/Cls=1) (c) ImageNet-10 (Img/Cls=1) (d) ImageNet-10 (Img/Cls=20)

Figure 1. Performance comparison across varying training steps. The batch size is set the same as 64. Our method consistently outperforms the leading method IDC [5] on various training steps.



(a) CIFAR-10 (Img/Cls=50). (b) CIFAR-100 (Img/Cls=10) (c) ImageNet-10 (Img/Cls=1) (d) ImageNet-10 (Img/Cls=10)

Figure 2. Performance comparison across varying training time. Our method significantly requires less time and achieves better performance. The result reported by our method is 5 $\times$ acceleration.



(a) CIFAR-10 (Img/Cls=10). (b) CIFAR-10 (Img/Cls=50) (c) CIFAR-100 (Img/Cls=1) (d) ImageNet-10 (Img/Cls=1)

Figure 3. Performance comparison across varying FLOPs. Our method significantly requires less computation resource and achieves better performance. The result reported of our method is 5 $\times$ acceleration.

the effectiveness and scalability of our method and makes it more appealing from all practical purposes.

4.2. Comparison on Training Budgets

We further investigate the efficiency of our method by considering various amounts of training budgets. We consider training budgets from two perspectives, time efficiency, including training steps and training times as shown in Fig. 1 and Fig. 2, and computation efficiency, including FLOPs as shown in Fig. 3. We remark that our method provides significantly better performance than the leading method IDC across all ranges of budgets. Our method requires fewer training steps, consumes shorter training time, and fewer computation resources to reach comparable or better performance, which demonstrates the efficiency of our method.

Method	Acc.(%) / Time(h)	Speed Up	Acc. Gain
DC	26.0 / 1.38	**4.95×**	**1.11×**
DC + Ours	**28.9 / 0.27**		
DM	48.9 / 0.26	**4.92×**	**1.02×**
DM + Ours	**49.5 / 0.05**		
IDC	**67.5** / 22.2	**4.90×**	**0.99×**
IDC + Ours	67.1 / **4.45**		

Table 5. Applying our method to different dataset distillation methods on CIFAR-10 (10 images per class).

4.3. Our method with Other Algorithms

Our weight perturbation strategy can be orthogonally applied to other dataset distillation methods. To verify the generality of our strategy, we apply the weight perturbation on other DD methods, gradient-matching DC [10] and distribution-matching DM [9] to accelerate the training $5\times$ faster. Tab. 5 summarizes the test performance of condensed data on CIFAR-10. The table shows that our method can also improve the performance of DC and DM.

4.4. Model Selection Effect on Distillation

We conduct an ablation study investigating the effect of model selection in Tab. 6. Instead of randomly selecting a model from pre-trained models, we consider to average the weights of all the pre-trained models at the beginning of each outer loop in our method. The table shows that the gap between random selection and weight average is no more than 1% in most of datasets. It indicates that both methods preserve the relevant information of feature spaces and verifies that our method does not mainly rely on the way of model selection.

Dataset	Img/Cls	Random Selection	Weight Average
CIFAR-10	1	**49.2 (0.4)**	48.7 (0.4)
	10	**67.1 (0.2)**	66.1 (0.2)
CIFAR-100	1	**29.8 (0.2)**	28.5 (0.1)
	10	**46.2 (0.3)**	43.6 (0.4)
ImageNet-10	1	**60.5 (0.2)**	59.7 (1.2)
	10	74.6 (0.3)	**74.8 (0.4)**

Table 6. Comparing performance of dataset distillation on different methods of method selection. The results are reported under $5\times$ acceleration with an identical training strategy.

4.5. Visual Examples

We provide visual examples of our method on CIFAR-10, ImageNet under $5\times$ acceleration in the following pages. In Fig. 4 and Fig. 5, we compare our synthetic data samples to the real training data samples, which we used as initialization of the synthetic data. From the figure, we remark that our condensed data looks abstract, yet still recognizable, representative of each class. We also provide the full condensed data in Fig. 6 and Fig. 7, under the storage of 10 images per class.

References

[1] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Wearable imagenet: Synthesizing tileable textures via dataset distillation. In *CVPR Workshops*, pages 2277–2281, 2022. 2

[2] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *ICLR*, 2020. 1

[3] Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[5] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *ICML*, volume 162, pages 11102–11118, 2022. 2, 3

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[7] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, volume 12356, pages 776–794, 2020. 2

[8] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICML*, volume 139, pages 12674–12685, 2021. 1, 2

[9] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *WACV*, pages 6503–6512, 2023. 4

[10] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021. 2, 4

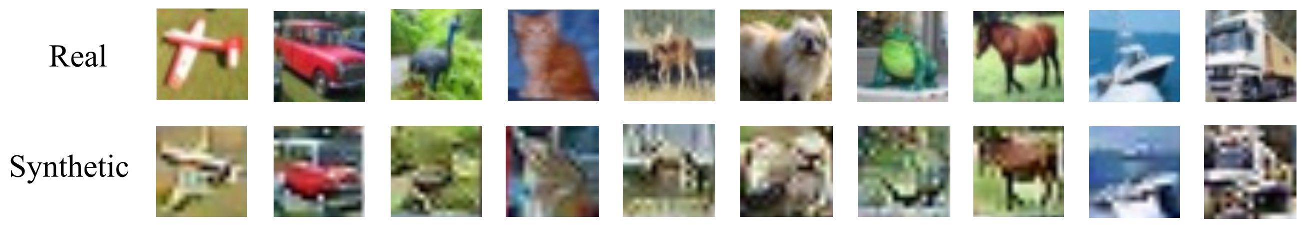


Figure 4. Comparison of real and synthetic images on CIFAR-10.

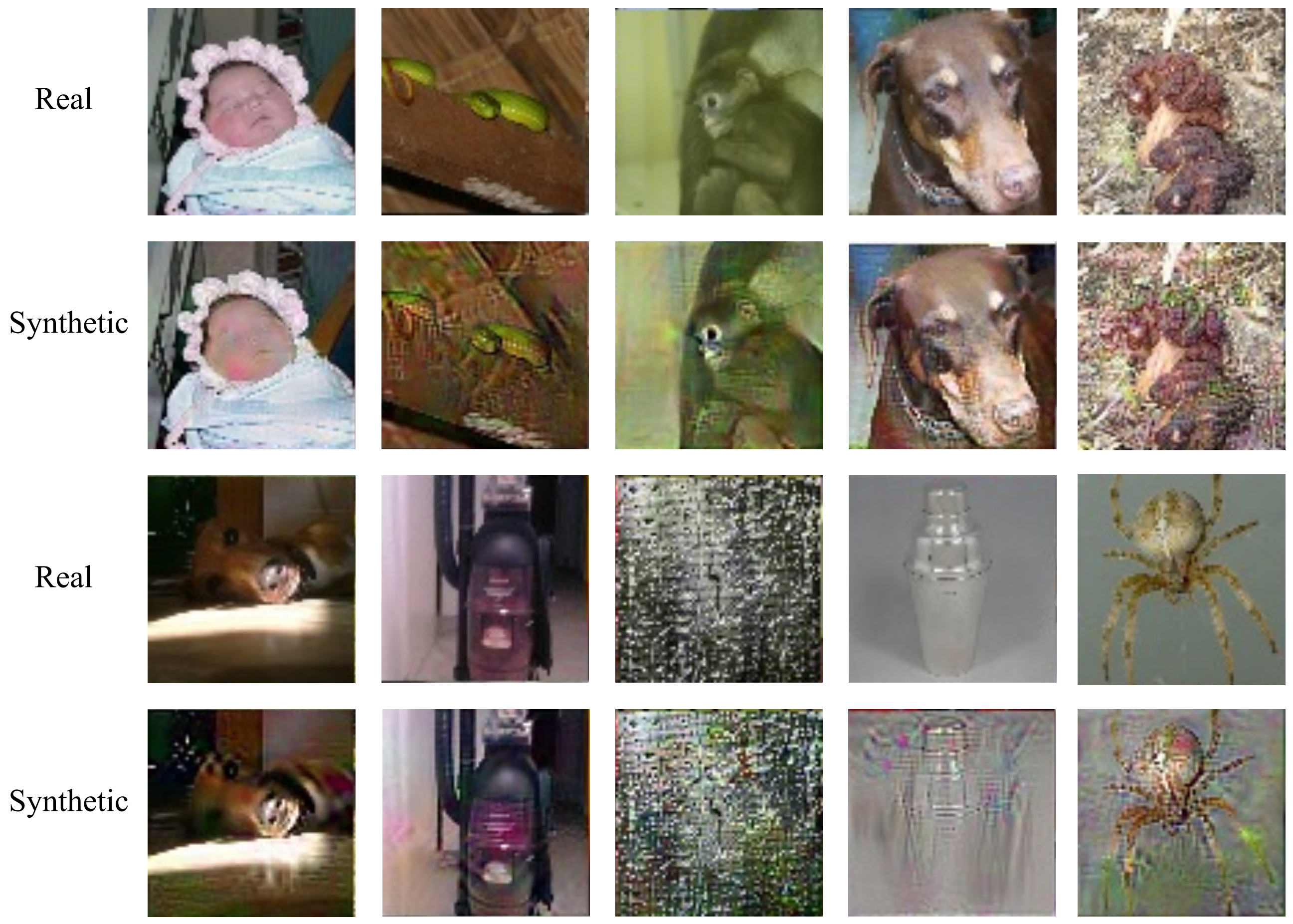


Figure 5. Comparison of real and synthetic images on ImageNet.

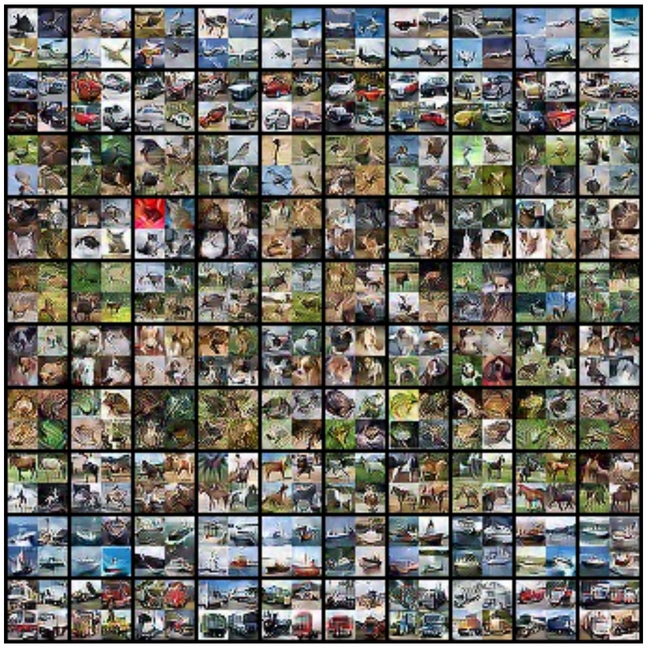


Figure 6. Condensed images of CIFAR-10 dataset 10 Img/Cls. Each row corresponds to the condensed class of a single class.

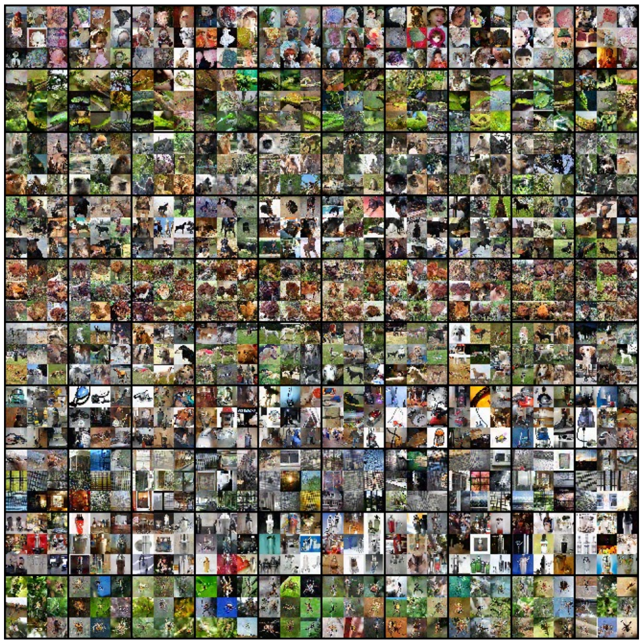


Figure 7. Condensed images of ImageNet-10 dataset 10 Img/Cls.