DEvS: Data Distillation Algorithm Based on Evolution Strategy

Nadiya Shvai
National University of Kyiv-Mohyla Academy
Kyiv, Ukraine
cyclope.ai
Paris, France
nadiya.shvai@cyclope.ai

Arcadi Llanza
University Paris Est, Laboratoire LISSI
Vitry-sur-Seine, France
cyclope.ai
Paris, France
arcadi.llanza@cyclope.ai

Abul Hasnat
cyclope.ai
Paris, France
hasnat.abul@cyclope.ai

Amir Nakib
University Paris Est, Laboratoire LISSI
Vitry-sur-Seine, France
nakib@u-pec.fr

ABSTRACT

The development of machine learning solutions often relies on training using large labeled datasets. This raises challenges in terms of data storage, data privacy protection, and longer model training time. One of the possible solutions to overcome these problems is called dataset distillation – a process of creating a smaller dataset while maximizing the preservation of its task-related information. In this paper, a new dataset distillation algorithm is proposed, called DEvS, which uses an evolutionary strategy approach to condense the training samples initially available for an image classification task, while minimizing the loss of classification accuracy. Experiments on CIFAR-10 demonstrate the competitiveness of the proposed approach. Also, contrary to recent trends, DEvS is derivative-free image generation, and therefore has greater scalability on larger input image sizes.

CCS CONCEPTS

• **Computing methodologies → Computer vision**; **Supervised learning by classification**; **Neural networks**; • **Mathematics of computing → Evolutionary algorithms**.

KEYWORDS

dataset distillation, image classification, neural networks, evolution strategy, optimization

ACM Reference Format:
Nadiya Shvai, Arcadi Llanza, Abul Hasnat, and Amir Nakib. 2022. DEvS: Data Distillation Algorithm Based on Evolution Strategy. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3520304.3528819

1 INTRODUCTION

The success of the recent decade of deep learning, and, in particular, its applications to computer vision, has been subject to the availability of large labeled datasets. This dependency consumes enormous amount of human and computational resources for data collection, labeling, storage, privacy protection, and neural network training time. From the perspective of human cognition, such addiction is excessive and must be resolvable. Multiple directions of research have been conducted to address this problem and its derivatives. Few-shot [18, 19, 23] and one-shot learning [7, 21] aim to train from a limited number of examples, transfer learning [24, 30] uses knowledge from one task to perform better on another but similar one, active learning [4, 9, 26] focuses on the efficient data selection-for-labeling strategy, coreset selection [1–3, 5, 6, 10, 13, 15, 17, 20] proposes strategies for choosing a representative training set, dataset distillation (condensation) [11, 12, 22, 28, 29] designs algorithms for creating a small amount of synthetic training samples etc. This research aims to address the problem using data distillation.

Recently, the data distillation technique gains enormous attention from the research community and a number of gradient-based methods have been proposed [11, 12, 22, 28, 29]. In particular, the idea of using synthetic images that look different visually from the original images as well as using non-binary [11] tags are significant. However, despite the promising performance, they face the key challenge of requiring very large computational resources [11, 12]. This research is motivated by this limitation and proposes a new efficient data distillation technique inspired by the principles of evolutionary strategies, called *DEvS*.

The main idea of DEvS is to distill the images using a multi-parents linear crossover from the original training dataset while applying appropriate regularization during the training procedure. The operation of linear crossover is implemented with the convex combination of the image tensors and their one-hot labels. We use mixup [27] regularization during training to ensure that the result of the crossover is consistent with model feature extraction. DEvS does not rely on gradient calculation for image generation and is therefore scalable with respect to the size of the input image

In order to evaluate the proposed method, we conduct the experiments on a popular computer vision benchmark dataset CIFAR-10. The comparison with competing techniques demonstrates the feasibility of DEvS and has the following advantages:

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9268-6/22/07.
https://doi.org/10.1145/3520304.3528819

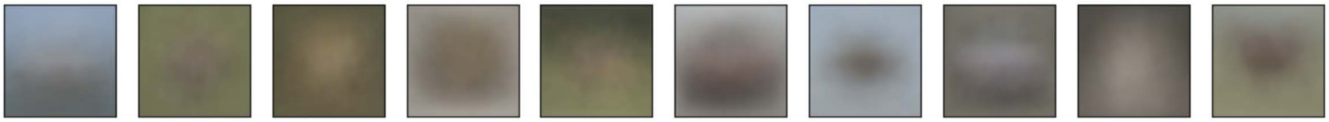

Figure 1: Dataset of size 10 distilled from CIFAR-10 using DEvS.

- low computational cost compared to the gradient-based methods;
- scalability to the larger input image sizes;
- higher performance compared to the coreset based methods.

Therefore, the contribution of this research is the proposal of a novel, scalable and computationally efficient data distillation technique.

2 RELATED WORK

Dataset distillation (condensation). The term dataset distillation was coined in the work of Wang et al. [22]. Its main idea lies in *synthesising* small amount of informative training samples, as opposed to *coreset selection* where the samples are selected from the initial training dataset. They used the gradient optimization in order to build synthetic images as "most useful for empirical risk minimization w.r.t model parameters" on a given batch of original training images. In order to enable generalization over different model initializations they considered random initializations from some distribution during the optimization process.

Nguen et al. [11] proposed a dataset meta-learning from kernel ridge-regression algorithm, called Kernel Inducing Points (KIP). This method used gradient descent to minimize ridge regression loss function value w.r.t. training data while iterating through random kernels and target data batches. Additionally, they proposed its variant called Label Solve (LS) based on labels learning while the training data is fixed. The latter significantly improved the results obtained with the initial KIP method. This work was continued in [12] where Nguen et al. applied a novel kernel-based meta-learning framework using infinitely wide convolutional neural networks in order to solve the data distillation problem. To the best of our knowledge this work constitutes the current state-of-art on the topic. However, computationally it is extensively heavy even for "toy" benchmark problems as the authors stated that the distilled datasets were obtained using "thousands of GPU hours".

Zhao et al. [29] formulated dataset distillation as a gradient matching problem between the gradients of the real and synthetic training loss w.r.t. the model parameters. They explored the applications of the proposed method to continual learning and neural network search. Based on this work, Zhao and Bilen proposed Differentiable Siamese Augmentation [28] in order to efficiently use data augmentation during the synthesis.

Coreset selection Coreset selection is a straightforward approach to decrease dataset size by selecting only a few of its points. The notions associated with this research directions can be found in [14]. The vast family of existing approaches include Forward Stagewise [6], Matching Pursuit [10], Orthogonal Matching Pursuit [13], Frank-Wolfe [3], Least-angle regression [5], Greedy Iterative Geodesic Ascent [2], Herding [15], K-Center [17, 25], Forgetting [20] etc. Particularly, the work from Barbiero et al. [1] is very interesting and similar to our research as it solves the coreset selection problem with evolutionary algorithms.

It is interesting to mention that a large number of coreset selection algorithms were initially developed for different purposes. For example, Forward Stagewise, Least-angle regression, Matching Pursuit and Orthogonal Matching Pursuit were originally used for dimensionality reduction, K-Center was designed for active learning, Forgetting was purposed for continual learning.

3 METHOD

3.1 Dataset distillation

Let D^{tr} be a large dataset of tuples $\{(\boldsymbol{x}_i, y_i)\}|_{i=1}^{K}$, where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d, y \in \{0, \ldots, C-1\}, \mathcal{X}$ is a d-dimensional input space, K is the number of samples in dataset D^{tr} and C is the number of classes. We assume this dataset to be associated with a classification problem of finding $f \in \mathcal{F}$ such that $f(\boldsymbol{x}) = y$ for $\boldsymbol{x} \in \mathcal{X}$. Then *data distillation* is a problem of constructing a new dataset $\tilde{D}^{tr} = \{(\boldsymbol{x}_i, y_i)\}|_{i=1}^{\tilde{K}}$ of smaller size $\tilde{K} \ll K$ such that the classification problem solution $\tilde{f}$ obtained with $\tilde{D}^{tr}$ has similar performance to the solution f obtained with D^{tr}. In particular, we expect similar performance on some validation dataset D^{val}:

$$perf(f(D^{val})) \approx perf(\tilde{f}(D^{val})).$$

Notable, for image classification problem, we assume input space $\mathcal{X}$ to be a space of (eligible) images, and when solving this problem with deep learning approach we assume $\mathcal{F} = \mathcal{F}_\theta$ to be a parametric family of functions representing some neural network architecture A and its weights $\boldsymbol{\theta}$. The process of solving classification problem will then be associated with neural network training, i.e. finding the weights $\boldsymbol{\theta}$ that minimize the empirical loss function

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(D^{tr}, \boldsymbol{\theta}).$$

3.2 Proposed method

The proposed method is inspired from the evolutionary algorithms. We consider the training dataset as a parent population that gets enlarged during the training phase via the operations of data augmentation (seen as a mutation) and mixup [27], i.e. convex combination (seen as a cross-over). Our goal is to reduce the parent population keeping the derived population diverse and representative. For the reduction step we will use crossover over similar samples in order to keep the average sample representation, where the similarity is considered with respect to the information retrieval based on the derived dataset.

The proposed DEvS method is an iterative method that reduces the database size at each iteration based on data distillation. Algorithm 1 provides a formal presentation of the DEvS method. It requires the initial dataset samples and the target distilled dataset size as input. Besides, optionally a decreasing sequence of intermediate distilled dataset sizes are provided as input. DEvS begins with initializing the entire training dataset as the current distilled dataset. Then it decreases the dataset size at each iteration from the existing size to the next distilled dataset size, which is predefined or provided as input. During each iteration, first the neural network model (see Section 4.2 for details) is trained on the current distilled dataset. Then this trained model is used to extract features with the convolutional blocks of this neural network. Afterward, a data clustering algorithm (see Section 4.3 for details) is applied on the extracted features, where the number of clusters is defined as a next-in-sequence intermediate distilled dataset size. Next, the clusters are reduced to their centers to generate a single sample from a set of samples belongs to the cluster. This sample construction is achieved by averaging the inputs and their corresponding labels. These synthesized samples construct the distilled dataset for the next iteration, and the algorithm continues iteration until the target distilled dataset size is reached.

Note that, in the distilled dataset the sample labels are soft probabilities rather than hard label assignment to particular classes. Indeed, multiple studies have shown that such labels provide additional advantages during the training procedure [11, 27]. Another important point is the use of step-wise dataset evolution. During our experiments we observed its advantage compared to the one-step reduction procedure in terms of the model performance that was trained on the distilled dataset.

Algorithm 1 Dataset Distillation Evolution Strategy (DEvS)

```
Require: A labeled training dataset $D^{tr}$ and a labeled validation
    dataset $D^{val}$; target dataset size $N$.
Initialize $D = D^{tr}$; sequence of intermediate dataset sizes $N_0 =
    |D| > N_1 > \ldots > N_M = N$, counter $i = 0$.
while $i < M$ do
    Train neural network $f$ using mutation (data augmentation)
        and crossover (mixup regularization). Use its truncated ver-
        sion to extract features $D^{feat}$ of dataset $D$.
    Cluster dataset $D^{feat}$ into $N_{i+1}$ clusters.
    Perform crossover by aggregating clusters with averaging
        inputs $\boldsymbol{x}$ and labels $y$ that correspond to that cluster.
    Obtain new dataset $D$ by considering the set of aggregated
        images and labels $\{(\boldsymbol{x_c}, y_c)\}$ (each corresponding to the clus-
        ter center in the feature space).
    Increase counter $i \leftarrow i + 1$
end while
```

4 EXPERIMENTS, RESULTS AND DISCUSSIONS

4.1 Datasets and associated tasks

In order to evaluate the performance of the proposed method we conduct experiments on the CIFAR-10 [8] dataset, which is a well-known computer vision benchmark dataset for image classification. It is composed of 60,000 32×32 colour images divided equally into ten different classes as shown in Figure 2. The train set is composed of 50,000 images and the test set of 10,000. The classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.

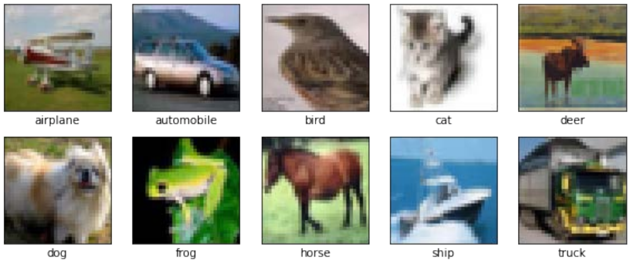

Figure 2: CIFAR-10 images examples with their class labels.

4.2 Convolutional neural network

For the experiments we follow Zhao et al. [29] and adopt a ConvNet architecture which is commonly-used in few-shot learning problems. It consists of $N = 3$ repeated blocks, each of which has a convolutional layer with 128 3×3 filters, Instance Normalization layer, and a ReLU layer. This sequence of blocks is concluded with a dense layer that has the softmax activation.

During training we use the Kaiming initialization to initialize the weights of the ConvNet architecture. We apply simple image data augmentation strategies such as shifts up to 5 pixels and horizontal flip. Additionally, we use mixup data augmentation [27] with parameter $\alpha = 0.4$ in order to assign labels in line with the samples aggregation strategy of averaging. Dropout of $p = 0.2$ is applied before the prediction layer during the training phase.

Experiments shows that it is beneficial to add a bottleneck dense layer before the prediction layer when training the network for feature extraction (final results are given for the original ConvNet architecture). Indeed, this observation supports the general intuition of applying stricter regularization when training with low number of samples.

4.3 Clustering

Clustering plays important role in the data distillation procedure w.r.t. scalability and computational efficiency. Therefore, in order to efficiently cluster the data while being consistent with computationally and scalable issues we use the mini-batch k-means algorithm [16]. The algorithm is initialized with the efficient k-means++ method to ensure the best possible initial cluster centers. The maximum number of iterations for clustering is set to 200 or a minimum tolerance 0.0001 is reached. We use 1028 number of samples in each mini-batch.

4.4 Results

For comparison we consider coreset selection methods as Herding, K-Center and Forgetting, and data condensation (DC) with gradient matching method of [29]. Additionally, we use Random coreset selection as a baseline. We do not consider here KIP and LS [11] due to their high computation load.

Table 1: Test accuracies of the ConvNet model trained on the distilled (or core) datasets obtained by competitive methods.

Dataset	Nb of distilled	% of distilled	Method					
	images	images	Random	Herding	K-Center	Forgetting	DC	DEvS
	10	0.02	14.4±2.0	21.5±1.2	21.5±1.3	13.5±1.2	28.3±0.5	25.77±1.1
CIFAR-10	100	0.2	26.0±1.2	31.6±0.7	24.7±0.9	23.3±1.0	44.9±0.5	38.97±1.0
	500	1	43.4±1.0	40.4±0.6	27.0±1.4	23.3±1.1	53.9±0.5	52.41±1.0

The results are presented in Table 1. We observe that the proposed algorithm outperfoms the coreset selection methods. In particular, it is interesting to compare it to Herding method as it also relies on clustering: the coreset is composed out of samples closest to the cluster centers.

The examples of distilled images are presented in Figure 1. They are not associated with a particular class label but with a target probability vector describing distribution over classes (for example, $0.26, 0.02, 0.04, 0.0, 0.03, 0.01, 0.0, 0.02, 0.55, 0.07$).

Although our method concedes to gradient methods such as DC, it proposes a viable alternative in terms of computation time: for CIFAR-10 it takes only a couple of GPU-hours while state-of-the-art KIP requires thousands. Moreover, its complexity does not depend on input image size, but on extracted features dimension size. Thus it scales well to larger image input size, which is to be expected in real-life scenario. For the future research steps we plan to study the effect of feature space regularization and prediction layer loss function on the proposed method performance.

5 CONCLUSIONS

In this paper, we have proposed a novel low-complexity and derivative-free dataset distillation method based on the evolution strategy. Unlike gradient-based competitive methods it is easily scalable w.r.t. the input image size. Conducted experiments demonstrate feasibility of the proposed approach. In terms of performance to computation costs ratio DEvS lies in the middle between traditional coreset selection methods and the gradient-based dataset distillation methods with easy implementation. In work under progress, we are studying the effect of feature space regularization, and of introducing a new prediction loss function to enhance the performance of the proposed method.

REFERENCES

[1] Pietro Barbiero, Giovanni Squillero, and Alberto Tonda. 2020. Uncovering Coresets for Classification With Multi-Objective Evolutionary Algorithms. *arXiv preprint arXiv:2002.08645* (2020).

[2] Trevor Campbell and Tamara Broderick. 2018. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*. PMLR, 698–706.

[3] Kenneth L Clarkson. 2010. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)* 6, 4 (2010), 1–30.

[4] Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei. 2008. Towards scalable dataset construction: An active learning approach. In *European conference on computer vision*. Springer, 86–98.

[5] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. 2004. Least angle regression. *The Annals of statistics* 32, 2 (2004), 407–499.

[6] MA Efroymson. 1960. Multiple regression analysis. *Mathematical methods for digital computers* (1960), 191–203.

[7] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence* 28, 4 (2006), 594–611.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[9] Xin Li and Yuhong Guo. 2013. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 859–866.

[10] Stéphane G Mallat and Zhifeng Zhang. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing* 41, 12 (1993), 3397–3415.

[11] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. 2020. Dataset Meta-Learning from Kernel Ridge-Regression. *arXiv preprint arXiv:2011.00050* (2020).

[12] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. 2021. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems* 34 (2021).

[13] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 40–44.

[14] Jeff M Phillips. 2016. Coresets and sketches. *arXiv preprint arXiv:1601.00617* (2016).

[15] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2001–2010.

[16] David Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*. 1177–1178.

[17] Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489* (2017).

[18] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems* 30 (2017).

[19] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1199–1208.

[20] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159* (2018).

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

[22] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959* (2018).

[23] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* 53, 3 (2020), 1–34.

[24] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[25] Gert W Wolf. 2011. Facility location: concepts, models, algorithms and case studies. Series: Contributions to Management Science: edited by Zanjirani Farahani, Reza and Hekmatfar, Masoud, Heidelberg, Germany, Physica-Verlag, 2009, 549 pp.,€ 171.15, $219.00,£ 144.00, ISBN 978-3-7908-2150-5 (hardprint), 978-3-7908-2151-2 (electronic).

[26] Donggeun Yoo and In So Kweon. 2019. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 93–102.

[27] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[28] Bo Zhao and Hakan Bilen. 2021. Dataset Condensation with Differentiable Siamese Augmentation. *arXiv preprint arXiv:2102.08259* (2021).

[29] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929* (2020).

[30] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.