ECE1512 - Project B: Dataset Distillation: A Data-Efficient Learning Framework

Assigned: Monday, November 14, 5:00 PM EST.

Due: Thursday, December 12, 5:00 PM EST.

1 Introduction

Deep neural networks (DNNs) have been widely deployed on the cloud for a wide spectrum of applications, from computer vision to natural language processing. The great success of deep learning is mainly due to its capability to encode large-scale data and to maneuver billions of model parameters. However, for real-time inference, it is a challenge to deploy these cumbersome deep models on edge devices with limited resources (e.g., mobile phones, IoT nodes, and embedded devices) not only because of the high computational complexity but also the large storage requirements.

In order to reduce the high computing source challenge of deep learning model trained on the large-scaled datasets, the widely known term, distillation, was introduced by Hinton *et al.* [13]. Model distillation as well as knowledge distillation are the methods to distill complex models into small models. Wang *et al.* [40, 1] introduced another option to reduce computation cost based model-space approach which is **dataset distillation (DD)**. The result of dataset distillation is a distilled dataset or synthetic dataset of small size. Nevertheless, this small dataset still assures the same performance of the learned model compared to the model learned on the full dataset. Hence, the dataset distillation method can reduce the required memory and computation for a deep learning model. For example, a deep learning model can train on only 10 distilled images for 10 digits (1 image per class) and still achieve good performance instead of training on all $50,000$ images in the MNIST dataset [21]. Unlike classical data compression, dataset distillation aims for a small synthetic dataset that still retains adequate task-related information so that models trained on it can generalize to unseen test data, as shown in Figure 1. Thus, the distilling algorithm must strike a delicate balance by heavily compressing information without completely obliterating the discriminative features.

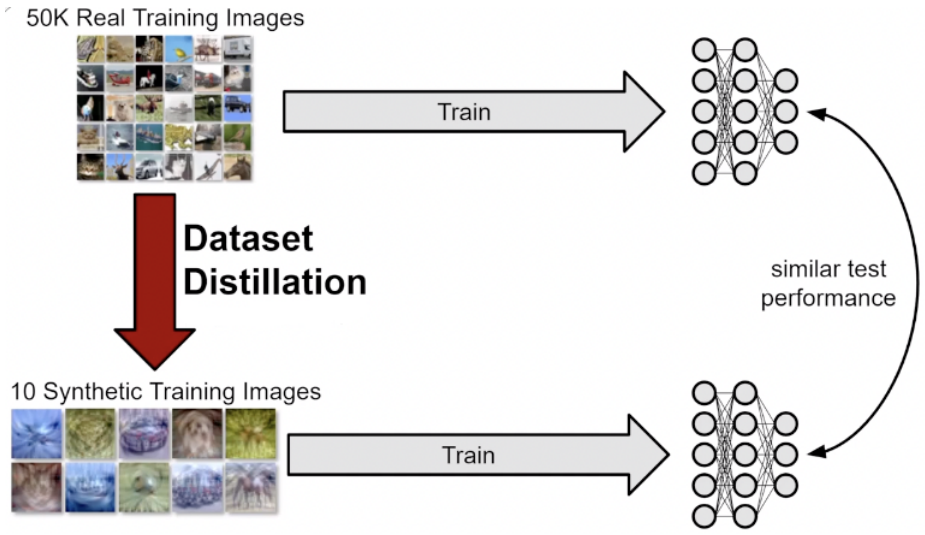

Figure 1: Dataset distillation aims to generate a small synthetic dataset for which a model trained on it can achieve a similar test performance as a model trained on the whole real training set [4].

To do this, dataset distillation methods attempt to discover exactly which aspects of the real data are critical for learning said discrimination. Mainly, the dataset distillation methods can be categorized into the following groups:

a. **Meta-loss based Dataset Distillation:** These end-to-end training methods create a synthetic dataset by solving a meta-loss, bi-objective optimization problem [40, 28, 30, 31, 38, 49, 27].

b. **Gradient/Trajectory Matching Surrogate Objective:** These methods formulate the dataset distillation as a gradient/trajectory matching problem between the gradients of deep neural network weights that are trained on the original and the synthetic data [47, 44, 4, 22, 15].

c. **Distribution/Feature Matching Surrogate Objective:** The methods that explicitly align real-feature distributions from the real and synthetic training data across various scales and sampled embedding spaces [45, 39].

d. **Distilled Dataset Parametrization:** These methods perform dataset distillation by taking into account efficient parametrization, data regularity, latent space factorization, and sharing [46, 19, 8, 18, 26, 32, 22].

2 Objective

The goal of this project is to equip you with the tools and technologies required to create a synthetic small $\mathcal{S}$ that has the most discriminative features of the original large-scale dataset $\mathcal{T}$. Specifically, the project will focus on the setting up of dataset distillation as a data compression technique. The project is divided into 2 tasks, (1) using the prior dataset distillation with Gradient Matching as a data compression method for two popular computer vision classification dataset, namely **"MNIST"** [21] and **"CIFAR10"** [20]; (2) using two/one state-of-the-art methods to further explore the effect of dataset distillation framework on visual classification tasks. The detailed descriptions of the datasets and tasks are included in the next sections. To begin with this project, you must create a GitHub repository for Project B as follows:

1. Create a group (individual) project personal GitHub repository. Use a proper naming convention, *ECE1512_2022F_ProjectRepo_NameStudent(s).*
2. Create a Project B sub-directory.
3. Populate your project B sub-directories with project related material.

Datasets

The following is a detailed description of two datasets that you will need for Project B. Both datasets can be found in the ***Project_B_Supp.zip*** file, which has been uploaded to Quercus.

MNIST:

- **Paper:** Gradient-based learning applied to document recognition [21]: `http://yann.lecun.com/exdb/mnist/`
- **Description:** This dataset is a popular digit classification dataset which is a subset of a larger set available from NIST. The MNIST dataset is divided into 10 classes, each of which represents a digit between 0-9. The digits have been size-normalized and centered in a fixed-size image.
- **Availability:** Publicly available (for academic purposes).
- **Resources needed:** CPU
- **Data size:** 60000 train data + 10000 test data.

CIFAR10:

- **Paper:** Learning multiple layers of features from tiny images [20]: `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`
- **GitHub link:** `https://github.com/wikiabhi/Cifar-10`

- **Description:** The CIFAR10 dataset consist of 60000 32×32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The classes are "Airplane", "Automobile", "Bird", "Cat", "Deer", "Dog", "Frog", "Horse", "Ship", "Truck", which are completely mutually exclusive. For example, there is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, and things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.
- **Availability:** Publicly available (for academic purposes).
- **Resources needed:** CPU
- **Data size:** 50000 train data + 10000 test data.

Experimental Setup – What you need

1. Prerequisites:
   (a) Python 3.X
   (b) PyTorch
   (c) Sci-kit-learn (suggested)
   (d) NumPy (suggested)
   (e) Matplotlib (suggested)
2. Download ***Project_B_Supp.zip*** from Quercus.
3. The networks.ipynb is a Python notebook that provides six different networks, including MLP, Convent [10], LeNet [21], AlexNet [20], VGG11 [34], and ResNet-18 [12]. You will use this file to learn synthetic dates on the MNIST and CIFAR10 datasets.
4. The utils.ipynb is a Python notebook that provides utilities such as access to datasets, preprocessing, and data augmentation. You will use this file to learn synthetic dates on the MNIST and CIFAR10 datasets.
5. The Project_B_FAQs.pdf is a list of frequently asked questions that try to shed light on (almost) all of your questions and concerns that you may have during Project B.

GPU Requirements

Note that you **do not need a GPU to successfully complete Project B. You should be able to train your models on your local (personal) machines.** However, if you so wish, you can use a GPU by accessing Google Colab. Following are the steps to enable a GPU using Colab:

1. Upload the code base to Colab using your Google Drive.
2. Navigate to *Runtime* → *Change runtime* type in the top bar.
3. Change runtime accelerator to *GPU* and click *Save*.
4. Use *device = torch.device("cuda:0")* after importing libraries and call *.to(device)* function to transfer your model and tensors to GPU. Make sure that model and tensor both are placed on GPU.

3 Task 1: Dataset Distillation with Gradient Matching [25 Marks]

In this task, you will use the dataset distillation with gradient matching [47] to learn a synthetically small dataset for the MNIST and CIFAR10 dataset, train networks from scratch on the condensed images, and then evaluate them on the real testing data. This is one of the fundamental frameworks for dealing with dataset distillation in computer vision classification tasks while decrease the computational costs. Please follow the instructions below to become acquainted with this algorithm.

1. **Basic Concepts.** Read the paper **"Dataset Condensation with Gradient Matching"** [47] carefully, and then answer the following questions: [5 Marks]
   (a) What is the purpose of using Dataset Distillation in this paper? [1 Marks]
   (b) What are the advantages of their methodology over state-of-the-art? Explain your rationale.[1 Marks]
   (c) What novelty did they contribute compared to their prior methods? [1 Marks]
   (d) Explain in full detail the methodologies of the paper. [1 Marks]
   (e) Explain the usefulness of the methodology in machine learning applications (at least two applications). [1 Marks]

2. **Dataset Distillation Learning.** Select one of the six architecture backbones provided in the *networks.ipynb* file and follow the below steps on the MNIST and CIFAR10 datasets to implement Dataset Condensation with Gradient Matching algorithm: [20 Marks]
   (a) Train the selected model with the original dataset and report the classification accuracy along with floating-point operations per second (FLOPs) for the test set. Use SGD as an optimizer with a cosine annealing scheduler with an initial learning rate of 0.01 for 20 epochs. (For more information on experimental setting, look at the implementation details of [47]) These scores gives you the uppper bound benchmark evaluation. [2 Marks]
   (b) Learn the synthetic dataset $\mathcal{S}$ using the selected model and Gradient Matching algorithm. For initialization of condensed images, randomly select from real training images. The experimental setup could be found in Table 1. [5 Marks]

Table 1: Experimental setup for learning the synthetic dataset $\mathcal{S}$ with Gradient Matching algorithm. The hyper-parameters K, T, $\eta_{\mathcal{S}}$, $\zeta_{\mathcal{S}}$, η_θ, and ζ_θ denote the number of random weight initializations, number of iterations, the learning rate for the condensed samples, the number of optimization steps for the condensed samples, the learning rate for the model, and the number of optimization steps for the model, respectively.

Dataset	K	T	$\eta_{\mathcal{S}}$	$\zeta_{\mathcal{S}}$	η_θ	ζ_θ	Optimizer	# images/class	minibatch size
MNIST & CIFAR10	100	10	0.1	1	0.01	50	SGD	10	256

   (c) Provide the visualization of condensed image per class for both MNIST and CIFAR10 datasets. Do you think these condensed images are recognizable? Support your explanations. [1 Marks]
   (d) Repeat parts 2b and 2c while the condensed images are initialized with Gaussian noise. Discuss in full detail the qualitative and quantitative results you have achieved. Are the results and visualizations are comparable with parts 2b and 2c. [2 Marks]
   (e) Now that you have had a chance to understand, learn, and visualize the condensed dataset, we can train the selected network from scratch on the condensed images. Train the selected network on a learned synthetic dataset (with 100 training images), then evaluate it on the real testing data. Compare the test accuracy performance and the training time with part 2a. Explain your results. (For the fair comparison, you should use the exact same experimental setting as part 2a) [2 Marks]

3. **Cross-architecture Generalization.** Another key advantage of the dataset distillation approaches is that the condensed images learned using one architecture can be used to train another, unseen one. Here you learned the synthetic datasets for the MNIST and CIFAR10 over the selected model in part 2b. Once the condensed sets are synthesised, train another network from *networks.ipynb* file and evaluate its cross-architecture performance in terms of classification accuracy on the test sets. Were your condensed datasets successful in cross-architecture generalization on the MNIST and CIFAR10 datasets? Support all your answers with detailed reasons and results. (Hint: use the *utils.ipynb* file) [3 Marks]

4. **Application.** Apply your synthetic small datasets to one of the machine learning applications you proposed in part 1e (These papers are helpful: [33, 5, 37, 6, 9, 24, 11, 48, 43, 29, 25, 35, 14, 17, 16, 36, 23, 42]). Discuss in full detail the qualitative and quantitative results you have achieved. [5 Marks]

5. **(Optional) Dataset Distillation for Histopathological Classification Task.** Apply the dataset distillation with gradient matching for a clinical histopathology dataset, **"MHIST"** [41]. [Up to 3 Marks]

4 Task 2: Comparison with State-of-the-arts Methods [15 Marks]

In this task, we would like to use one/two of the state-of-the-art methods described in the Introduction section 1 and compare them with the Gradient Matching algorithm that you have used in Task 1 to further explore the effect of dataset distillation methods in visual classification tasks. The group with one student should use the paper **"Dataset Distillation by Matching Training Trajectories"** [4, 2, 3] while the group with two students should use that paper along with another one from the shortlist containing the references for the state-of-the-art methods on page 2. **Further, considering the categorization provided in the introduction paragraph, you should not select the method from the category b..**

1. Read the papers and answer the following questions for each of them: (Groups of one and two students must choose one and two papers, respectively, as illustrated above—one student, one paper.) [4 Marks]
   (a) What knowledge gap did your one/two chosen dataset distillation methods fill? [1 Marks]
   (b) What novelty did they contribute compared to their prior methods? [1 Marks]
   (c) Explain in full detail the methodologies of your selected methods. [1 Marks]
   (d) Discuss the main advantages and disadvantages of your selected methods. Do you think these methods can concretely distill the original datasets? Do you think your selected methods can analyze and inspect the cases of large-scale datasets like ImageNet [7]? Why? [1 Marks]

2. Apply their methods to the selected architecture in part 2 on one of the MNIST or CIFAR10 datsets. [11 Marks]
   (a) Similar to Task 1, the learning pipeline has two stages: (1) learn the condensed images using the selected method(s); (2) train your network from scratch on the condensed images, then evaluate them on the real testing data. (Make sure the comparison is fair.) [8 Marks]
   (b) Report your findings in both quantitative and qualitative manners and compare them with the results of Task 1. Discuss your results. Do you think the selected methods outperform the Gradient Matching algorithm in terms of test accuracy? Explain the effect of the dataset distillation methods in terms of generalization and recognition abilities. [3 Marks]

Notes

1. This project should be completed in groups of one or two students.
2. Coding is expected for this project, and your code must be included in your submitted report – use the Python programming language.
3. External code may be used only if properly cited.
4. Include as many data visualization results as possible – a good visual is worth more than a thousand words (as per your ECE1512 Lecture 1 handout – "One picture is worth more than ten thousand words" (anonymous))

What to submit & Evaluation

1. **Project evaluation will be based on a submitted written report.** You need to submit a written report (one per group) **in PDF format using the ECE1512 Q page submission utility.** Your report should be approximately **20 pages in IEEE style** (either one column or two columns). List any additional references you have used in your work. You need to provide the code, data, figures and any other auxiliary material) used in your work. To that end, your written report should include a link to your Project B GitHub repository.
2. **Project evaluation will include an examination of the code, notebooks, figures, and other auxiliary material posted on your GitHub repo.** It is highly recommended that you include a README document on your project's GitHub page.
3. **Ouriginal "plagiarism" scores** will be visible upon (report) submission, so make sure that you are not penalized for plagiarism in uncited text and code portions (if applicable).
4. Make sure your report submission is desigend to be used simultaneously with the Github-Repo. To do this, use **Machine Learning Paper Reproducibility Checklist** that clearly acknowledge that the report and the code are two separate artefacts, each with their own checklist.
5. **Late submissions will not be accepted.**

Resources

Certain concepts and methods used in this project may be unfamiliar to you. Refer to these online resources for more details (cite if code is used):

- Basic Concepts of Dataset Distillation
- Dataset Distillation by Matching Training Trajectories
- A Review of Dataset Distillation for Deep Learning
- Awesome-Dataset-Distillation
- Dataset Condensation Benchmark: DCBench
- GitHub repo for Dataset Condensation Benchmark: DCBench
- PyTorch documentation
- Creation a GitHub repo
- Measuring FLOPs for a given model
- Training Machine Learning Models More Efficiently with Dataset Distillation
- Dataset Distillation using Neural Feature Regression

References

[1] Dataset distillation,. https://www.tongzhouwang.info/. Available access on 2022 Jun 20.

[2] Dataset distillation by matching training trajectories,. https://georgecazenavette.github.io/mtt-distillation/.

[3] Github repo of dataset distillation by matching training trajectories,. https://github.com/GeorgeCazenavette/mtt-distillation.

[4] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. https://arxiv.org/abs/2203.11932.

[5] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. *arXiv preprint arXiv:2211.04446*, 2022. https://arxiv.org/pdf/2211.04446.pdf.

[6] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. *arXiv preprint arXiv:2207.09639*, 2022. https://openreview.net/pdf?id=Bs8iFQ7AM6.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. https://ieeexplore.ieee.org/document/5206848.

[8] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *arXiv preprint arXiv:2206.02916*, 2022. https://arxiv.org/abs/2206.02916.

[9] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? *arXiv preprint arXiv:2206.00240*, 2022. https://proceedings.mlr.press/v162/dong22c/dong22c.pdf.

[10] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018. https://arxiv.org/abs/1804.09458.

[11] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020. https://arxiv.org/abs/2008.04489.

[12] 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. https://arxiv.org/abs/1512.03385.

[13] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. https://arxiv.org/abs/1503.02531.

[14] Shengyuan Hu, Jack Goetz, Kshitiz Malik, Hongyuan Zhan, Zhe Liu, and Yue Liu. Fedsynth: Gradient compression via synthetic data in federated learning. *arXiv preprint arXiv:2204.01273*, 2022. https://arxiv.org/pdf/2204.01273.pdf.

[15] Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z Pan. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*, 2022. https://arxiv.org/abs/2208.00311.

[16] Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 720–730, 2022. https://arxiv.org/abs/2206.07746.

[17] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*, 2021. https://arxiv.org/pdf/2110.07580.pdf.

[18] Balhae Kim, Jungwon Choi, Seanie Lee, Yoonho Lee, Jung-Woo Ha, and Juho Lee. On divergence measures for bayesian pseudocoresets. *arXiv preprint arXiv:2210.06205*, 2022. https://arxiv.org/abs/2210.06205.

[19] 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. *arXiv preprint arXiv:2205.14959*, 2022. https://proceedings.mlr.press/v162/kim22c.html.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf.

[22] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. *arXiv preprint arXiv:2202.02916*, 2022. https://arxiv.org/abs/2202.02916.

[23] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Soft-label anonymous gastric x-ray image distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 305–309. IEEE, 2020. https://ieeexplore.ieee.org/document/9191357.

[24] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Dataset distillation using parameter pruning. *arXiv preprint arXiv:2209.14609*, 2022. https://arxiv.org/abs/2209.14609.

[25] Ping Liu, Xin Yu, and Joey Tianyi Zhou. Meta knowledge condensation for federated learning. *arXiv preprint arXiv:2209.14851*, 2022. https://arxiv.org/pdf/2209.14851.pdf.

[26] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *arXiv preprint arXiv:2210.16774*, 2022. https://arxiv.org/abs/2210.16774.

[27] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *arXiv preprint arXiv:2210.12067*, 2022. https://openreview.net/forum?id=h8Bd7Gm3muB.

[28] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR, 2020. https://arxiv.org/abs/1911.02590.

[29] Wojciech Masarczyk and Ivona Tautkute. Reducing catastrophic forgetting with learning on synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 252–253, 2020. https://arxiv.org/abs/2004.14046.

[30] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. https://arxiv.org/abs/2011.00050.

[31] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021. https://arxiv.org/abs/2107.13034.

[32] Parsa Nooralinejad, Ali Abbasi, Soheil Kolouri, and Hamed Pirsiavash. Pranc: Pseudo random networks for compacting deep models. *arXiv preprint arXiv:2206.08464*, 2022. https://arxiv.org/abs/2206.08464.

[33] Mattia Sangermano, Antonio Carta, Andrea Cossu, and Davide Bacciu. Sample condensation in online continual learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08. IEEE, 2022. https://ieeexplore.ieee.org/document/9892299.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. https://arxiv.org/abs/1409.1556.

[35] Rui Song, Dai Liu, Dave Zhenyu Chen, Andreas Festag, Carsten Trinitis, Martin Schulz, and Alois Knoll. Federated learning via decentralized dataset distillation in resource-constrained edge environments. *arXiv preprint arXiv:2208.11311*, 2022. https://arxiv.org/pdf/2208.11311.pdf.

[36] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pages 9206–9216. PMLR, 2020. http://proceedings.mlr.press/v119/such20a/such20a.pdf.

[37] Ilia Sucholutsky and Matthias Schonlau. Secdd: Efficient and secure method for remotely training neural networks. *arXiv preprint arXiv:2009.09155*, 2020. https://arxiv.org/pdf/2009.09155.pdf.

[38] Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR, 2022. https://proceedings.mlr.press/v162/vicol22a.html.

[39] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022. https://arxiv.org/abs/2203.01531.

[40] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. https://arxiv.org/abs/1811.10959.

[41] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Naofumi Tomita, Lorenzo Torresani, et al. A petri dish for histopathology image analysis. In *International Conference on Artificial Intelligence in Medicine*, pages 11–24. Springer, 2021. https://arxiv.org/abs/2101.12355.

[42] Felix Wiewel and Bin Yang. Condensed composite memory continual learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. https://ieeexplore.ieee.org/document/9533491.

[43] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. *arXiv preprint arXiv:2207.09653*, 2022. https://arxiv.org/abs/2207.09653.

[44] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. https://arxiv.org/abs/2102.08259.

[45] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021. https://arxiv.org/abs/2110.04181.

[46] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022. https://arxiv.org/abs/2204.07513.

[47] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *ICLR*, 1(2):3, 2021. https://arxiv.org/abs/2006.05929.

[48] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020. https://arxiv.org/pdf/2009.07999.pdf.

[49] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *arXiv preprint arXiv:2206.00719*, 2022. https://arxiv.org/abs/2206.00719.