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

ECE1512 Digital Image Processing and Applications
Siyu Zhang
1010403653
Date due: November 6th 2024
Date handed in: November 6th 2024

Introduction

Training large datasets in deep learning models often encounters challenges, primarily due to high computational demands. To address this, Hinton et al. [1] introduced the concept of model distillation, also known as knowledge distillation. Building upon this, Wang et al. [2] proposed an alternative approach called Dataset Distillation, which aims to reduce computational costs by transforming large datasets into smaller, compact, synthetic datasets that retain the essential information of the original dataset. By using these synthetic datasets with discriminative features, it is possible to achieve performance comparable to that of the original large datasets.

This project is based on the idea of enhancing data efficiency by utilizing dataset distillation as a data compression technique. The datasets used include "MNIST" [3] and "MHIST" [4]. MNIST is a well-known grayscale classification dataset of handwritten digits with 10 classes (digits 0-9), containing 60,000 training images and 10,000 test images, each 28x28 pixels in size. The MHIST dataset, formally known as the Minimalist Histopathology dataset, contains fixed-size images of colorectal polyps. It comprises 3152 images in total: 2175 for training and 977 for testing. The MHIST dataset contains two classes of colon polyp images: 162 images of hyperplastic polyps (HP), which are benign, and 990 images of sessile serrated adenomas (SSA), which are considered precancerous.

This project has two main tasks. The first task is to apply the concept of attention matching, as proposed in the paper "Efficient Dataset Distillation with Attention Matching" [5]. This involves comparing the original dataset with a synthetic dataset initialized from both real images and Gaussian noise. The synthetic dataset is then trained and evaluated against the original dataset in terms of test accuracy, performance, and training time, as well as assessed for cross-architecture generalization capabilities.

The second task builds on the first, using the approach from the paper "Prioritize Alignment in Dataset Distillation" [6]. In this task, a two-step process is employed to produce a synthetic dataset that retains high-level features from the original dataset while removing redundancy and noise. The effectiveness of the methods used in both tasks will then be compared.

Discussion

Task 1: Dataset Distillation with Attention Matching

Basic Concepts

(a) What is the purpose of using Dataset Distillation in this paper?[8]

Dataset Distillation is a data-centric approach that generates a condensed synthetic dataset from a larger real dataset. This synthetic dataset retains the essential or valuable features of the original data. It offers two main benefits: first, it reduces the training time and computational costs, and second, it allows the model to achieve high performance comparable to training on the original large dataset. The purpose of Dataset Distillation is to compress data while preserving its key features, enabling efficient model training without sacrificing performance.

(b) What are the advantages of their methodology over state-of-the-art? Explain your rationale.

In the paper proposes a frame called Data Distillation with Attention Matching (DataDAM). The advantages of this method are demonstrated in its efficiency in reducing computational cost, improving performance and accurately capturing the original data distribution without introducing biases. It used Spatial Attention Matching to check critical features between synthetic and real data, this allows it to retain essential information from the full dataset in a condensed form. It also has practical advantages in applications such as continual learning and neural architecture search.

The two main features that they used is Spatial Attention Matching (SAM) and complementary feature distribution alignment.

SAM, or Spatial Attention Matching, is designed to capture the key features and relationships in input images by aligning the spatial attention maps between real and synthetic data across multiple layers within a neural network. These spatial attention maps highlight the most discriminative regions in each image, allowing SAM to represent features from different layers, which correspond to low-, mid-, and high-level information. This layered approach helps produce a comprehensive representation of the input data, as SAM ensures that both real and synthetic datasets focus on similar critical regions across the network's layers. DataDAM using initialized convolutional neural network instead of the complex bi-level optimization to enhance scalability and reduce biases in the synthetic data.

Maximum Mean Discrepancy (MMD) loss is used to ensure that the overall feature distribution of the synthetic data matches that of the real dataset, particularly in the last layer where high-level information is captured. By combining MMD loss with SAM, DataDAM generates synthetic data that captures essential features and closely aligns with the overall distribution of the original dataset.

(c) What novelty did they contribute compared to their prior methods?

The paper outlines this methodology that begins by initializing a synthetic dataset derived from the original data, using techniques such as adding random noise, random sampling of real images, or clustering algorithms like K-Center. Next, both real and synthetic data are passed through a neural network, with probability distributions over randomly initialized weights, P_θ, transforming them into different feature maps. A spatial attention map is then created using a SAM module, which summarizes the absolute values of the features across channels, and compares the attention maps of the real and synthetic data using loss functions to minimize their differences. If the loss function is large, adjustments are made to the learning rate and task balance parameter. The feature distributions between the real and synthetic datasets are further compared using Maximum Mean Discrepancy (MMD) at a high level in the last layer. DataDAM refines the synthetic dataset by minimizing the combined SAM and MMD losses through iterative stochastic gradient descent. Finally, the synthetic dataset is evaluated for effectiveness by testing accuracy, and cross-architecture generalization tests are conducted to ensure robustness across various model architectures and datasets.

(e) Explain the usefulness of the methodology in machine learning applications (at least two applications).

Continual Learning and Neural Architecture Search are the two machine learning applications that used the methodology that proposed before.

Continual Learning due with the challenges like limited storage and catastrophic forgetting, where models forget previous knowledge when learning new tasks. Instead of using a large memory-intensive replay buffer, DataDAM produces synthetic data that preserves essential information from the original dataset. This enables models to recall important past knowledge efficiently, even with restricted storage.

Neural Architecture Search (NAS) demands large amount computational resources and it needs to evaluate over the full dataset many times due to different architectures. DataDAM addresses this by generating a compact synthetic dataset that effectively represents the original data, allowing NAS to quickly assess architectures with far less computation. This makes NAS faster and more feasible in resource-constrained settings while maintaining evaluation accuracy close to that achieved with the full dataset.

Dataset Distillation Learning

For the dataset distillation learning, the architecture backbones that used in MNIST and MHIST datasets are ConvNet-3 and ConvNet-7, respectively. In the *networks.ipynb* file, there are six architecture backbones, MLP, ConvNet, LeNet, AlexNet, VGG and ResNet. The reason to choose ConvNet because the models like AlexNet, VGG and ResNet, are designed for more complex and high-resolution dataset such as ImageNet. For example, ImageNet-21k contains 14197122 images divided into 21841 classes.[7] Compare this dataset to the MNIST and MHIST, the datasets are used in the project is so small. The rest two can work well for the small dataset but ConvNet is more efficient and accurate than others.

For the both datasets, after training the dataset and record the classification accuracy, it also requires to use floating-point operations per second (FLOPs) to analyze the computational cost for test dataset and the complexity of the model. Compare the FLOPs for different models, it helps to visualize the efficient of the model with the target application. To train the datasets, Stochastic Gradient Descent (SGD) is used for better generalization in the deep learning models. Since the dataset is quite large, mini-batch SGD is used to save time and help to converge the results. In the set-up part, the batch size for training is 128, the initial learning rate is 0.01 and the number of the epoch is 20. During the training, SGD optimizer is used with 0.9 momentum and also a Cosine Annealing Scheduler. Scheduler helps the model converge effectively and prevent overfitting in the training end. Fig.1 shows 100 randomly selected images from the MNIST dataset, and Fig. 2 displays random images from the MHIST dataset.

The ConvNet-3 model on the MNIST dataset demonstrates exceptional performance, achieving a training accuracy of approximately 99% with an average loss of 0.02 per epoch. Its test accuracy reaches 99.42% and the FLOPs value 4.96×10^7, which demonstrate that ConvNet-3 can efficiently capture MNIST's patterns and it is ideally for simple and grayscale datasets like MNIST. In contrast, the ConvNet-7 model trained on the MHIST dataset starts with a lower initial accuracy of 68.59% and gradually improves to around 87.25% with a final test accuracy of 87.21%. This model shows steady learning but requires significantly higher computational resources, 8.81×10^8, which reflects the increasing complexity of the MHIST dataset. The MHIST images are more complex because it has RGB channels and detailed medical features. Additionally, the ConvNet-7 has more layers to capture relevant patterns from these images. This is a trade-off between efficiency and the learning or representational power. This highlights the importance to match the design of model to the complexity of the dataset. It is helpful to achieve a better accuracy and with efficient computation.

After training and testing on the two original datasets, a synthetic dataset is created using the attention matching algorithm. The initialization of condensed images is achieved by randomly selecting samples from the original training dataset. The model setup involves defining a structured training loop based on the attention matching strategy. The process begins with selecting random images from each class to create the initial synthetic dataset. Using the defined hyperparameters such as random weight initializations, K, the iterations, T, and learning rate for the condensed samples, η_S optimization steps for the condensed samples, ζ_s , the learning rate for the model, η_θ, and optimization steps for the model, ζ_θ, and a task balance parameter, $\lambda = 0.01$, in Attention Matching to train on the ConvNet structure. The synthetic images will be compared the difference with original training dataset through attention loss to make sure synthetic images capture the critical patterns of real images. Table 1 shows the hyperparameters for Initial Setup of Synthetic MNIST and MHIST Datasets.

The condensed images initialized by randomly selecting samples from the MNIST and MHIST datasets, are shown in Fig. 3 and Fig. 4. When comparing these condensed synthetic images to the original MNIST dataset, each digit class is easily recognizable because the synthetic images retain the essential features and shapes of each number. Similarly, for the MHIST dataset, the condensed images are also distinguishable, displaying clear and well-defined structures that preserve the essential characteristics of each class. The simple comparison suggests that training the synthetic dataset using the Attention Matching Algorithm with real images can capture the basic features and reduce dataset size. It might also reduce the computational time and cost for later training. If increasing the number of training epochs or tuning the hyperparameters could enhance the performance of the synthetic images, making them more effective for subsequent processes.

The condensed images initialized by adding Gaussian noise from the MNIST and MHIST datasets, are shown in Fig. 5 and Fig. 6. The Gaussian noise used in the project is standard, with a mean of zero and a variance of one and it adds after the randomly select the real images per class. The Gaussian noise-initialized images for MNIST dataset appear random and lack recognizable features, making it difficult to visualize specific classes compared to images initialized from real samples because they are too noisy and lack sufficient features to capture. The similar performance is also shown in the MHIST dataset, the Gaussian noise-initialized synthetic images for Class 0 (HP) and Class 1 (SSA) show no recognizable histological patterns. The images are purely random, filled with multicolored pixels, which makes class distinctions impossible at this stage. From a quantitative perspective, it would be beneficial to train a model on the Gaussian noise-initialized dataset with more iterations and to fine-tune the hyperparameters. Reducing the Gaussian noise variance could also help reveal the basic structure of the images, making it easier for the model to develop class-specific features. These adjustments would make the synthetic images more recognizable and effectively shaped to identify the classes.

Compare training original dataset and the synthetic dataset, the test accuracy exists huge difference although they use the same model. Training on the full original datasets (MHIST and MNIST) provides the highest test accuracies (87.21% for MHIST and 99.42% for MNIST) with the quite obvious training time request. For example, the training time on original MNIST dataset is 252.77s. Table 2 shows the test accuracy for MNIST and MHIST datasets on different Training Sets. Using condensed data initialized from real images to train the model, the performance of the test accuracy decrease (63.01% for MHIST and 88.42% for MNIST). It indicates that the condensed dataset retains essential features through attention matching, but it unable to fully replicate the information of the original dataset. The faster training, low computational costs and mid-level test accuracy performance is a trade-off. This trade-off shows the effectiveness of dataset distillation with attention matching in retaining core features of the data while reducing training time and computational costs, making it suitable for applications where computational efficiency is prioritized over peak accuracy. With more iterations and fine-tuning hyperparameters, the test accuracy might achieve a better performance than now.

Comparing training on the original dataset with the synthetic dataset reveals a significant difference in test accuracy, despite using the same model. Training on the full original datasets (87.21% accuracy for MHIST and 99.42% accuracy for MNIST) achieves the highest test accuracies, but it requires notably longer training times. When using condensed data initialized from real images, test accuracy decreases (63.01% for MHIST and 88.42% for MNIST) with only less than 5 seconds (3.79s for MHIST and 0.61s for MNIST). It indicates that the condensed dataset retains essential features through attention matching, but it unable to fully replicate the useful information of the original dataset. This condensed dataset's faster training time, lower computational costs, and moderate test accuracy represent a practical trade-off. This balance demonstrates the effectiveness of dataset distillation with attention matching in capturing core features while reducing training time and computational demands, making it ideal for scenarios where computational efficiency is prioritized over maximum accuracy. With further iterations and fine-tuning of hyperparameters, the test accuracy could achieve even better.

However, for the second type of condensed dataset, where initialized with standard Gaussian noise, the performance appears drastically different. The test accuracy for MHIST is only 54.70%, and for MNIST, it is as low as 9.58%. This represents a significant drop compared to both the original dataset and the first type of condensed dataset initialized from real images. But the training time on MNIST is only 0.61 seconds, while MHIST requires 5.79 seconds. This suggests that lower computational time corresponds to lower test accuracy for this type of condensed data. Since the images start with Gaussian noise with a variance of one, they lose their basic structure. For MNIST, which only has a single channel, the added noise makes it even harder for the model to distinguish classes, in contrast to MHIST, which has three channels and potentially more structural information. This indicates that standard Gaussian noise is not a suitable initialization method to replace the original dataset. It also highlights the importance of starting from real images in the attention matching process. The excessive noise in the Gaussian-initialized images leads to a lower-quality representation, which negatively impacts model performance.

Additionally, the testing experiment demonstrates that the minibatch size plays an important role. With only 100 condensed images in each case, the batch size used with the original dataset is no longer suitable. Instead, a smaller minibatch size of 10 is more appropriate for training on the condensed dataset.

Cross-architecture Generalization

Dataset distillation not only enables the generation of synthetic images from real images or Gaussian noise but also allows these synthetic datasets to be used for training across different models, such as LeNet, VGG11, and others. This approach not only demonstrates the effectiveness of the synthetic dataset in achieving high performance but also showcases its ability to generalize across various architectures.

In this part, the new architecture used is VGG11, and the datasets employed are the first version of the synthetic datasets initialized from real images. A learning rate of 0.001 was selected because VGG11 is a large model, typically used for training on larger datasets. However, since the synthetic datasets contain only 100 images with fewer classes (2 for MHIST and 10 for MNIST), a lower learning rate was appropriate. After tuning the basic parameters, the test accuracy achieved on the original MNIST test set was 87.12%, and for the MHIST dataset, it was 68.59%. During training, both datasets exhibit high initial losses. For MHIST, the training starts with a loss of 22.3543, while for MNIST, the initial loss is 2.4745. These high starting losses suggest that the model initially struggles to the model is learning and try to make right prediction. During the training, the losses substantially decrease to 0.6916 for MHIST and 0.0273 for MNIST and become quite steady. Compared to the synthetic real images on ConvNet model, both test accuracies are similar, showing only minor differences. Specifically, the test accuracy for MHIST increased by approximately 5%, while the test accuracy for MNIST decreased by around 2%. This suggests that VGG11 performs slightly better with higher-dimensional images, as seen with MHIST. However, when compared to training on the original datasets, these values are notably lower. Despite this, the results indicate that the synthetic datasets for both MNIST and MHIST successfully generalize to the VGG11 architecture, achieving reasonably high accuracy across architectures.

Application

The two main machine learning application mentioned before is Continual Learning and Neural Architecture Search (NAS). In this part, the two sets of condensed images that initialized from the real images and the two original test datasets are the datasets. NAS is used to help to find the model that can help synthetic dataset to perform well with limited information.

NAS explores different possible configurations for the customized Basic CNN model to find the best structure for synthetic dataset. It varies the number of convolutional layers (2, 3, or 4) to capture features at different levels of details, adjusts the number of filters in each layer (16, 32, or 64) to control the model's capacity, and tests different number of units in the fully connected layer (64, 128, or 256) to learn high-level patterns. Additionally, it experiments with two types of activation functions (ReLU or Leaky ReLU) to add non-linearity, helping the model learn more complex relationships in the data. The reason to choose these parameters is to show the complexity and adaption of the NAS, if needed, the search space can be expanded and customized based on the scratch of the model. The customized the model has 3 channels, for each convolutional layer has a fixed kernel size 3x3 and a padding 1 and use 2x2 max pooling layer with stride of 2.

The six architectures get good results on condensed MNIST dataset, five is 100% and one with high parameter value is 98%. The evaluation values on condensed MHIST dataset is also quite high, but also get a low value on architecture with more layer and filters. The performance on both original test datasets is in the mild, 70.38% for MNIST and 63.39% for MHIST. This architecture experiment demonstrates that these features with these values might not be the best one to train on the condensed datasets. If the number of convolutional layers and number of filters in each layer can be increased, it would be more helpful to discover more about the architectures that can fit the datasets better. Notice that for the MHIST, the test performance becomes a little bit better than on ConvNet-7 with 20 epochs, this suggest that maybe some parts of the basic CNN model are more suitable to gather the important information on the condensed dataset. This also shows the flexibility of NAS on the architectural design of the model and it enable it to adapt to different dataset characteristics by adjusting structural elements.

Task 2: Comparison with State-of-the-arts Methods

Basic Concepts

(a) What knowledge gap did your one/two chosen dataset distillation methods fill?[8]

Prioritize Alignment in Dataset Distillation (PAD) is used to address the issue of misaligned information in dataset distillation processes, which can lower the quality of distilled datasets. It has two steps. First, PAD aligns information extraction by selecting samples that match the intended compression ratio, ensuring only relevant data is included. Second, it aligns information embedding by using only deep layers of the agent model, which captures high-level, meaningful features and avoids introducing redundant low-level information. This approach results in higher-quality synthetic datasets optimized for model performance.

(b) What novelty did they contribute compared to their prior methods?

The main novelty in PAD is a two-step approach to addressing misaligned information in dataset distillation processes. Before these main steps, PAD add a step, trajectory matching to ensure that the data aligns with sample difficulty.

The first step, Filtering Information Extraction, introduces a data selection strategy in PAD that matches the data's complexity with the desired distillation compression ratio, Images Per Class. Using a scoring function to assess sample difficulty, PAD emphasizes easier samples for lower IPCs and harder samples for higher IPCs, where preserving more detailed information is advantageous.

The second step, Filtering Information Embedding, improves the embedding process by using only the deep layers of the agent model, leaving out the shallow layers that add unnecessary, basic details. By focusing on the deeper layers, PAD generates a synthetic dataset that captures high-level features, resulting in a cleaner and more effective dataset for accurate distillation.

(c) Explain in full detail the methodologies of your selected methods.

Step1, initialize a synthetic dataset by selecting samples from different difficulty levels in the original dataset, using the EL2N scoring metric to evaluate each data point's difficulty

Step2, pass the real dataset through an agent model and record its parameter trajectory. Then, pass the synthetic dataset through a student agent model and optimize its parameters to align with the recorded trajectory from the real dataset

Step3, use a data scheduler to control training. The model initially trains on easier samples and

gradually introduces harder samples, ensuring the difficulty levels match the intended complexity for each IPC (Images Per Class) setting.
Step4, focus primarily on deep layers that contain high-level information. Remove shallow-layer parameters from the optimization process to avoid introducing misaligned, low-level information into the synthetic dataset.
Step5, train the synthetic dataset iteratively to achieve optimized values where its parameters closely resemble those of the real dataset
Step6, evaluate the synthetic dataset by testing its effectiveness and accuracy against the original dataset. Conduct cross-architecture generalization tests to ensure robustness across various models and datasets

(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? Why?

The main advantage of PAD is that it enhances the quality and performance of distilled datasets by using a two-step approach to reduce noise and redundancy effectively. By filtering data based on difficulty and focusing on deep-layer information, PAD captures high-level features, making the synthetic dataset closer to the original.
The disadvantage is the increase in computational time and complexity, as PAD requires scoring each sample's difficulty and gradually increasing difficulty levels during training. This process can be challenging and time-consuming, especially for large datasets with many samples.
I believe PAD can reproduce the performance of original dataset if it does not have large amounts of data. It uses difficulty scores to match complexity levels per image class and records parameter trajectories, training the synthetic dataset to match these values. Filtering only deep-layer information helps capture essential features, bringing the synthetic dataset closer to the original. I think PAD could likely work on large-scale datasets like ImageNet with enough computational resources.

PAD on MNIST Dataset

In this section, a different approach called Process of Adversarial Distillation (PAD) is used to create condensed images on the original MNIST dataset. As before, the first initialization involves randomly selecting real images from each class, followed by applying the EL2N scoring metric to the condensed images to assess their complexity. Images with higher EL2N scores, indicating richer information, are selected for further training. The training function is structured to focus on deeper layers, capturing more abstract features from the synthetic images with high information content. In the code, the last layer of the model is extracted, and only activations from this deep layer are used during training, effectively aligning information embedding by emphasizing meaningful, high-level representations. Additionally, the optimized PAD training parameters, including image gradients and model states, are used same parameter as Task 1 to allow the consistency when compare with the results for Task 1 and Task 2. This approach enhances the quality of the synthetic dataset, ensuring that the most relevant and information-dense aspects are preserved, ultimately improving the model's performance on the original MNIST dataset.

Comparing the performance of PAD method with Attention Matching in the task 1, PAD works a little better than the method used in task 1. The PAD synthetic dataset achieves a test accuracy of 90.66% and 0.2972 loss. During the training epoch with high accuracy, it is clear to see that PAD approach effectively maintain the key feature of the dataset.

Overall, both PAD and Attention Matching methods effectively generated synthetic datasets that captured the core patterns of MNIST classes and show the quite high accuracy. Since PAD focus on filtering high-level information and pay more attention on deep layer features, the generalization is better than Attention Matching a little bit. Both methods are suitable to create compact condensed images and maintain the essential features. With more detailed parameters training and more iterations, the condensed images might represent original images better.

Conclusion

This project focuses on the potential of Dataset Distillation techniques, specifically Attention Matching and Prioritize Alignment (PAD), and explores their application in Neural Architecture Search (NAS). By using these two methods to create compact synthetic datasets, the project demonstrates the effectiveness of Dataset Distillation frameworks in maintaining high model performance while significantly reducing dataset size and computational time. The Attention Matching method captures key features of each class through an attention-based training loop. It updates synthetic images by comparing them with the original dataset, guiding the condensed images to retain high-level discriminative features. PAD, on the other hand, employs a different approach to preserve high-level information. It assigns scores to define the complexity of randomly selected images per class, then focuses training on deeper layers for high-score images, minimizing noise and redundancy in the process. In terms of test accuracy on condensed MNIST datasets initialized from real images, Attention Matching achieves 88.42%, while PAD reaches 90.66%. Both methods successfully maintain fair test accuracy while significantly reducing the original dataset size. However, a limitation arises when Gaussian noise with excessively high variance is used for initialization. Reducing the variance levels and tuning the parameters, this condensed images can reach a good test accuracy.

Codes

https://github.com/Alicesyz/ECE1512_2024_DatasetDistillation_SiyuZhang

Result

Table 1: Hyperparameters for Initial Setup of Synthetic MNIST and MHIST Datasets

Dataset	K	T	η_s	ζ_s	η_θ	ζ_θ	Optimizer	#image/class	Minibatch size
MNIST	100	10	0.1	1	0.01	50	SGD	10	256
MHIST	200	10	0.1	1	0.01	50	SGD	50	128

Table 2: Test Accuracy for MNIST and MHIST Datasets on Different Training Sets

Dataset	Original Training Dataset	Condensed Images from Real Images	Condensed Images by Added Gaussian Noise
MNIST	99.42%	88.42%	9.58%
MHIST	87.21%	63.01%	54.70%

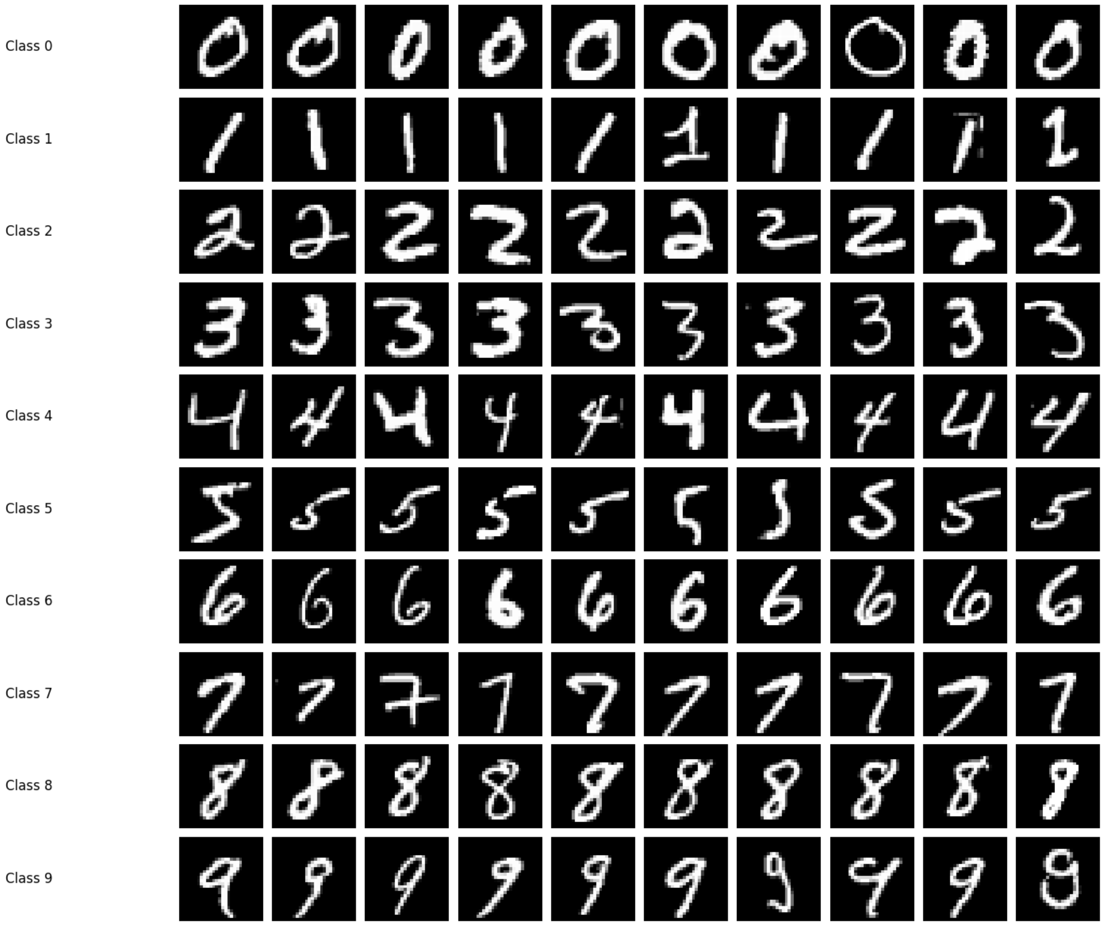

Fig.1 Randomly selected images from the MNIST dataset

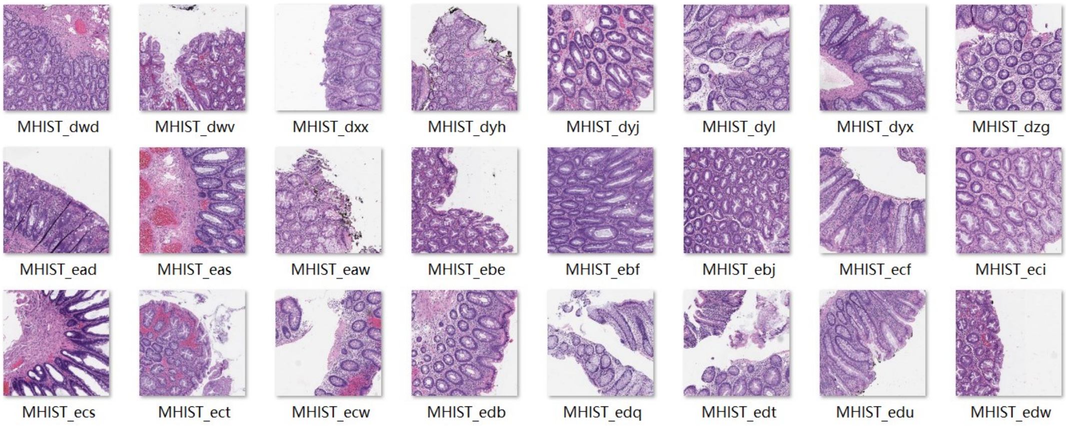

Fig.2 Randomly selected images from the MHIST dataset

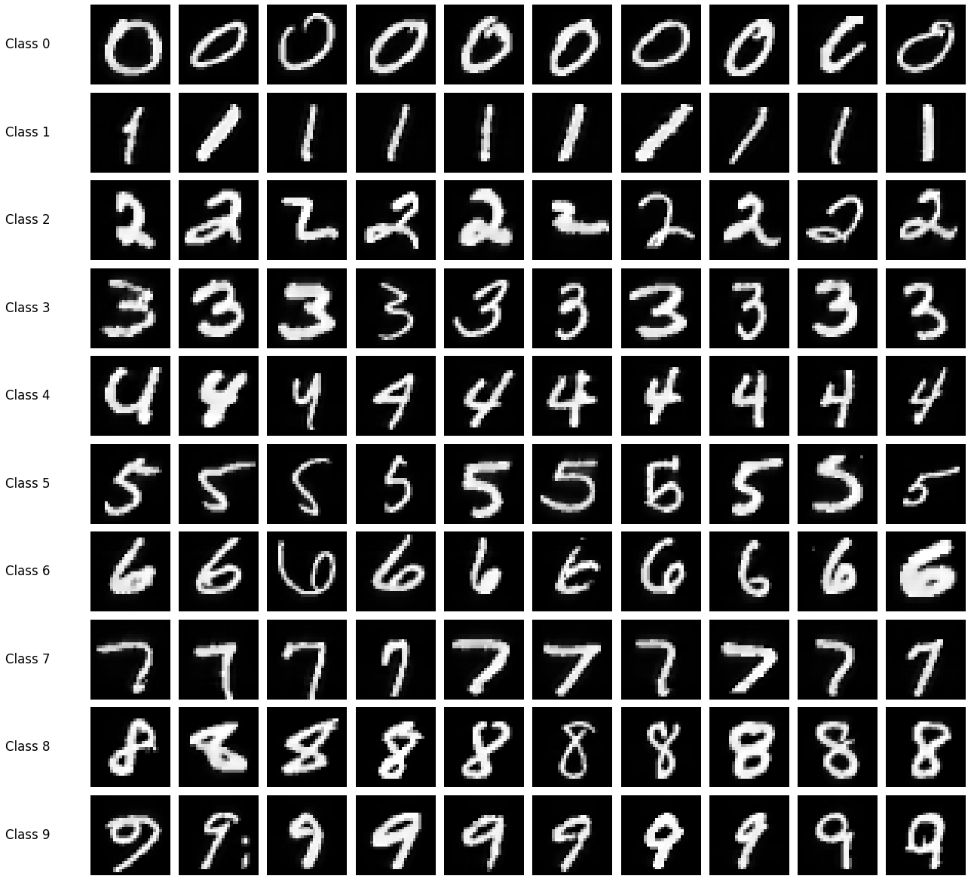

Fig.3 Synthetic MNIST dataset initialized by randomly selecting real images

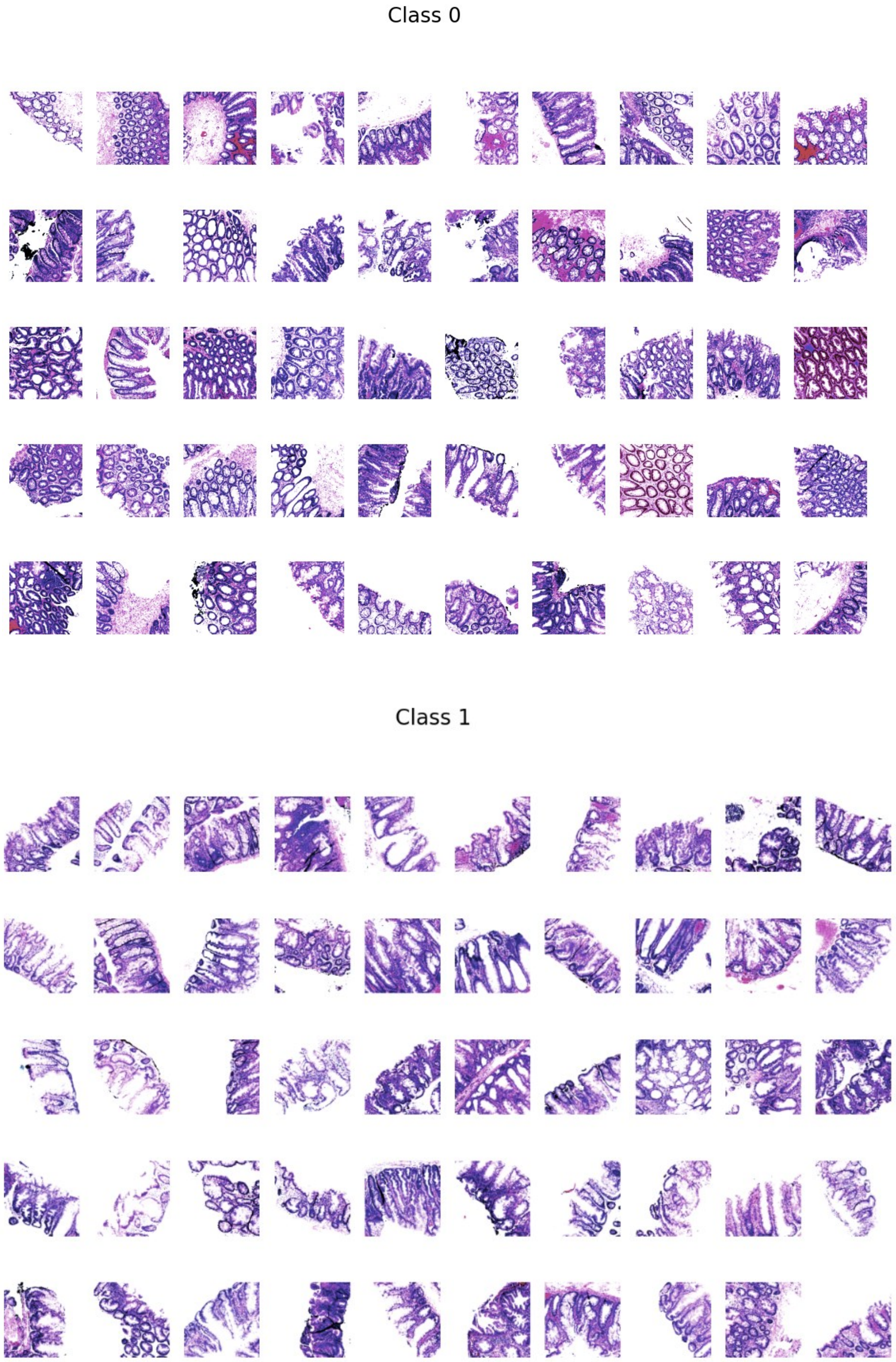

Fig.4 Synthetic MHIST dataset initialized by randomly selecting real images with class 0 (HP) and class 1 (SSA)

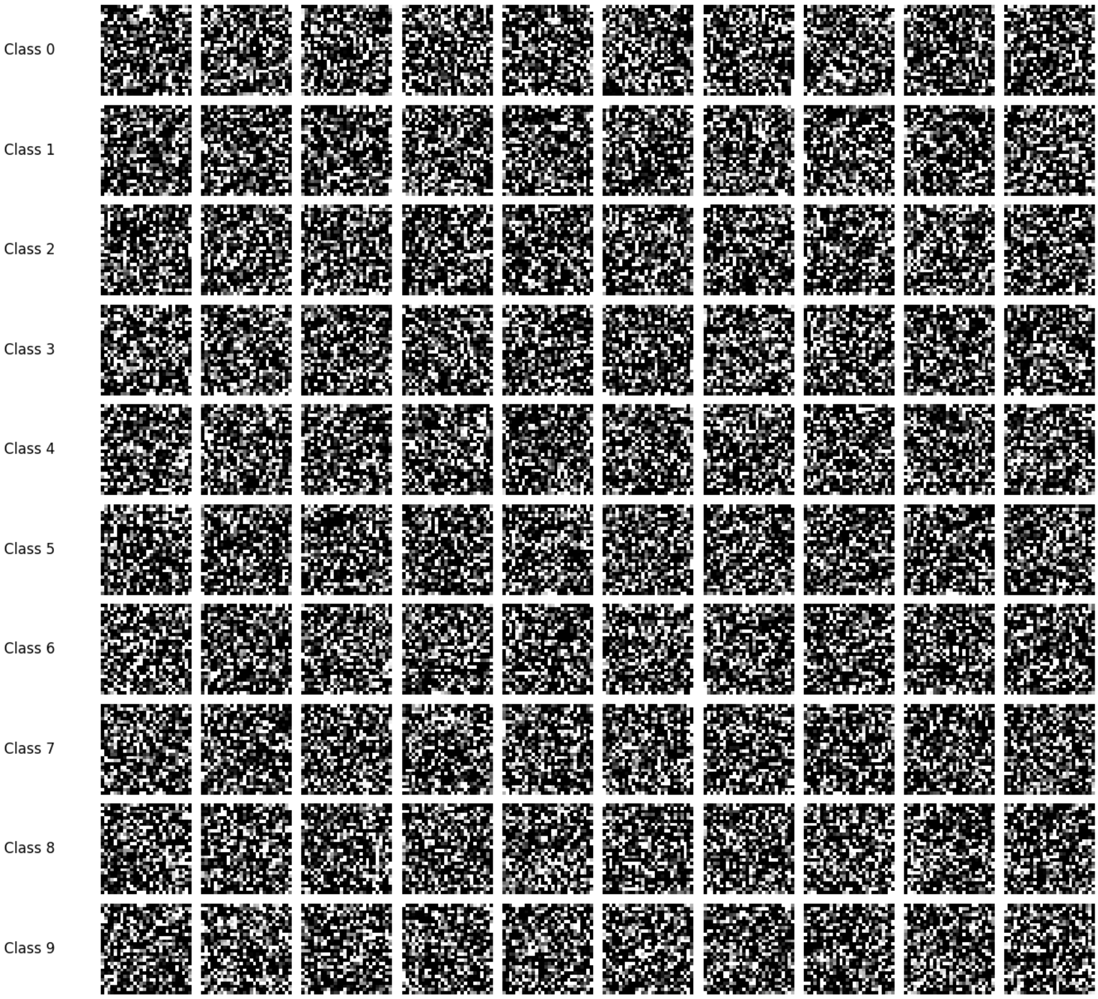

Fig.5 Synthetic MNIST dataset initialized by adding gaussian noise

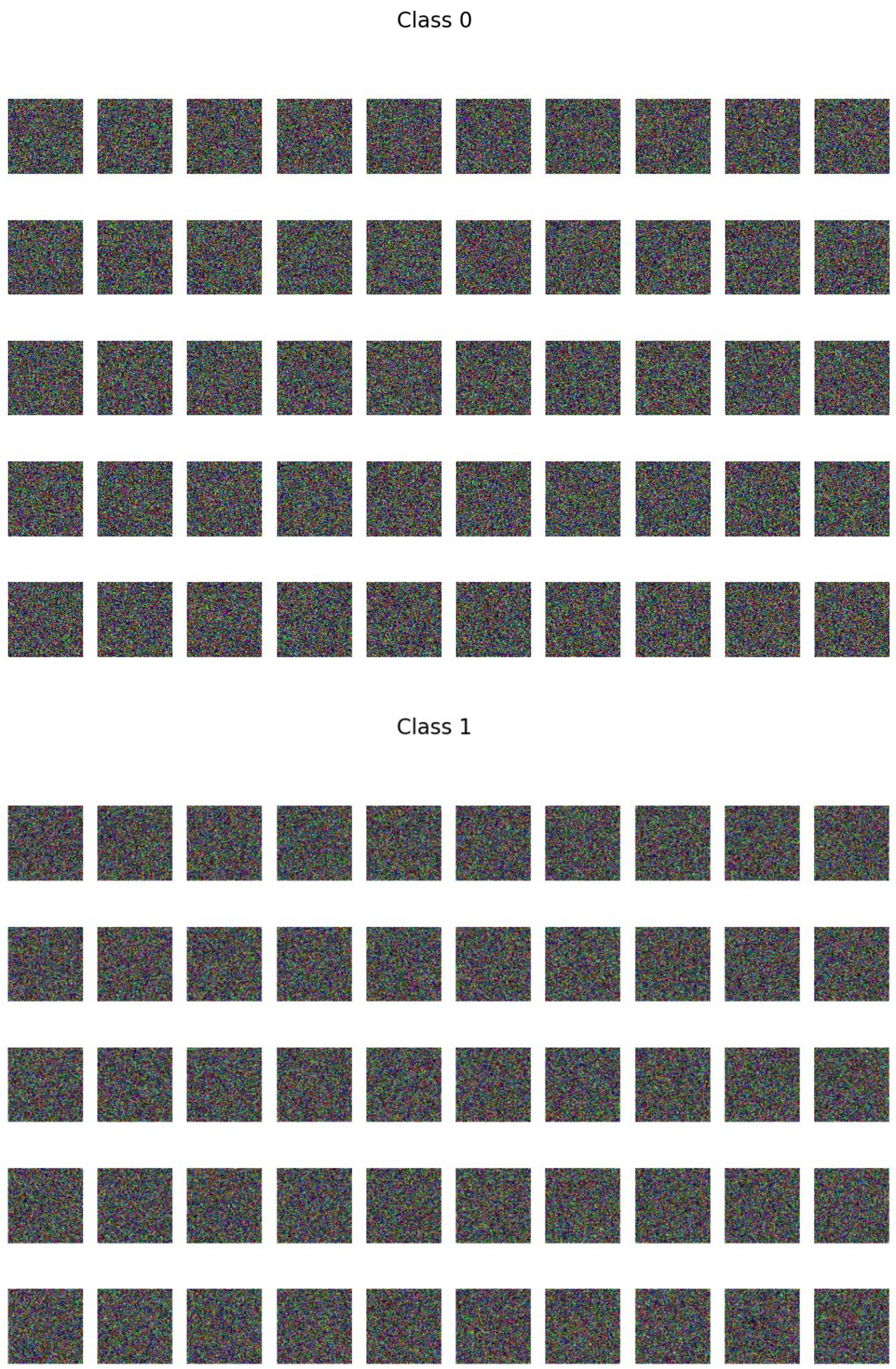

Fig.6 Synthetic MHIST dataset initialized by adding gaussian noise with class 0 (HP) and class 1 (SSA)

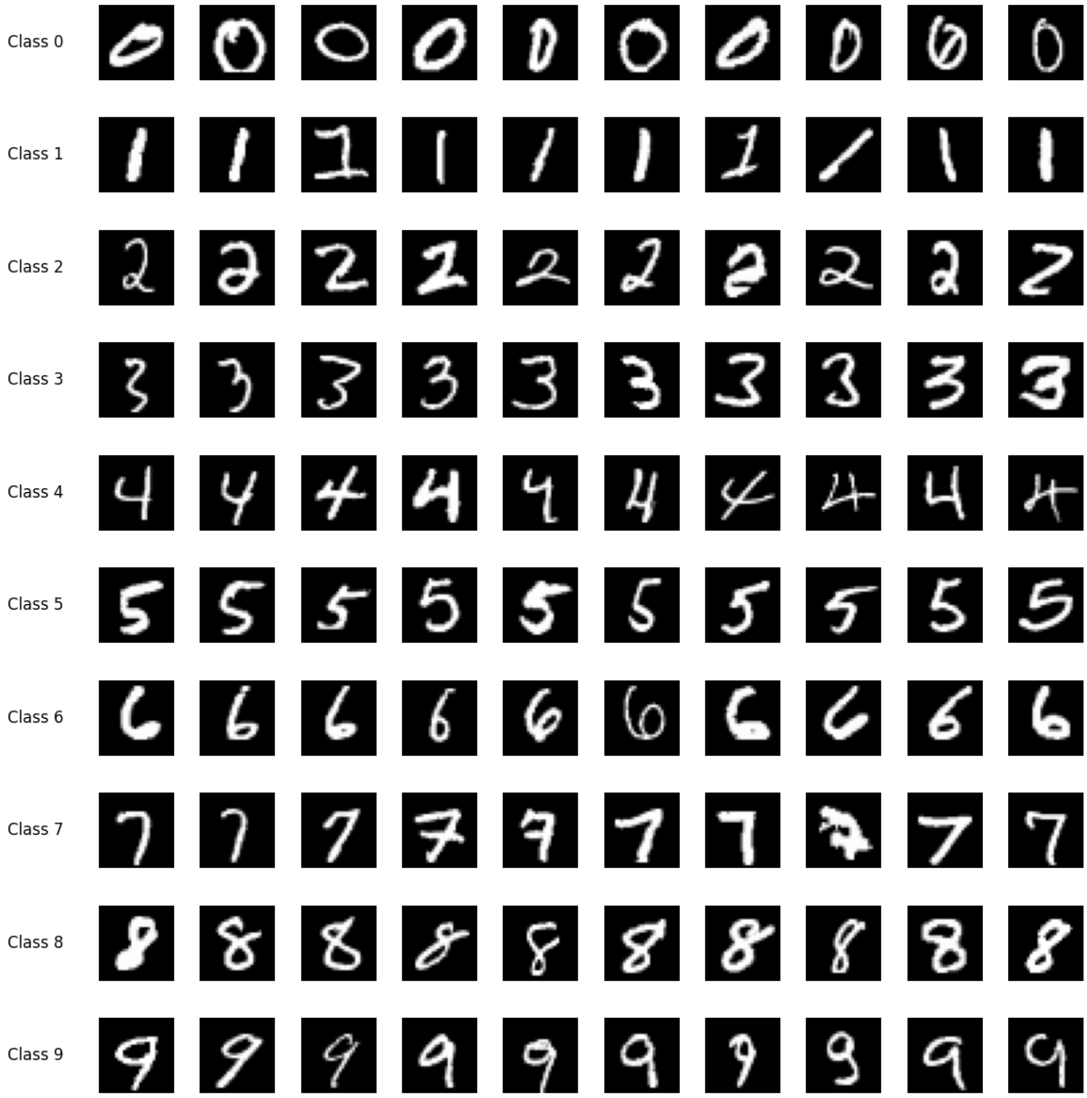

Fig.7 Synthetic MNIST dataset after PAD oepration initialized by randomly selecting real images

Appendix

[1] 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.

[2] 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.

[3] Yann LeCun, L´eon 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.

[4] 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.

[5] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In Proceedings of the *IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.

[6] Zekai Li, Ziyao Guo, Wangbo Zhao, Tianle Zhang, Zhi-Qi Cheng, Samir Khaki, Kaipeng Zhang, Ahmad Sajedi, Konstantinos N Plataniotis, Kai Wang, and Yang You. Prioritize alignment in dataset distillation, 2024.

[7] Ridnik, Tal; Ben-Baruch, Emanuel; Noy, Asaf; Zelnik-Manor, Lihi (5 August 2021). "ImageNet-21K Pretraining for the Masses". arXiv:2104.10972

[8] ECE1512_2024F_ProjectA_Assignment