Final Submission

Applicability of synthetic data in medical field

: Leveraging Synthetic Data and PneumoniaMNIST on Deep Neural Networks

11조 김정국, 박세현, 이채은, 진현빈

Index

Challenges of AI in Medical Field

Diagnosis of disease

1. Limited Availability of Real Data

- Training AI model requires diverse and numerous amount of accurately labeled medical data
- Challenges in collecting real-world medical data due to privacy concerns and the need for expert annotation
- The scarcity of high-quality datasets restricts effective model training

2. Automating Labeling of Real Medical Data

- Diagnosing diseases as well as manual labelling real-world medical data is labor-intensive, time-consuming, and expensive

Solution in real world : use **synthetic data** for training and applying the model

→ **Question**: Would using a synthetic data feasible to be used in the medical field?

→ **Experiment** : Figure out using the pretrained models to compensate limited amount of data (Transfer learning)

Paper Review 1

: Synthetic Augmentation with Large-scale Unconditional Pre-training [1]

About

- Improve classification of medical images using 'HistoDiffusion'
- 'HistoDiffusion' is trained by large-scale unlabeled synthetic data and small-scale labeled real data
- Compare the performance of real images and synthetic images generated by StyleGAN2 and HistoDiffusion

Method

- 5,000 (5%) real-images as a sample for base model
- Add synthetic images with ratio of 50%, 100%, 200%, 300% of real-images to base model

Table 1. Quantitative comparison results of synthetic image quality and augmented classification. "Random" refers to directly augmenting the training dataset with synthesized images without any image selections while "selective" indicates applying selective module [36] to filter out low-quality images. The number ($X\%$) suggests that the number of the synthesized images is $X\%$ of the original training set.

	FID↓	Accuracy↑	F1 Score↑	Sensitivity↑	Specificity↑
Baseline (5% real images)	/	0.855	0.850	0.855	0.983
StyleGAN2 [14]					
+ random 50%	5.714	0.860	0.856	0.860	0.980
+ selective [36] 50%	5.088	0.868	0.861	0.867	0.978
100%	5.927	0.879	0.876	0.879	0.982
200%	7.550	0.895	0.888	0.895	0.983
300%	10.643	0.898	0.896	0.898	0.987
HistoDiffusion (Ours)					
+ random 50%	4.921	0.870	0.869	0.870	0.982
+ selective [36] 50%	4.544	0.891	0.888	0.891	0.983
100%	**3.874**	0.903	0.902	0.903	0.991
200%	4.583	**0.919**	**0.916**	**0.919**	**0.992**
300%	8.326	0.910	0.912	0.910	0.988

> Performs best when adding selective HistoDiffusion generated synthetic data with ratio 200%

https://arxiv.org/abs/2308.04020

Paper Review 2

: Vision-Language Foundation Model for Chest X-ray Generation [2]

About

- RoentGen generated diverse and visually convincing synthetic chest X-ray images based on text prompts
- Improved accuracy training in a combination of synthetic and real images

Method

- Set baseline models and add 100%, 500% of synthetic images

Experiment	Training Data		AUROC	Accuracy
	Real	Synth.	(n = 5,000)	
Small dataset				
R (Baseline)	1.1k		0.73	0.71
1.1k S		1.1k	0.69 (↓0.04)	0.66
1.1k R/S	1.1k	1.1k	0.77 (↑0.04)	0.76
5.5k S		5.5k	0.75 (↑0.02)	0.78
R + 5.5k S	1.1k	5.5k	0.76 (↑0.03)	0.77
Big dataset				
30k R	30k		0.82 (↑0.09)	0.75
30k S		30k	0.80 (↑0.07)	0.74
30k R/S	30k	30k	**0.84 (↑0.11)**	**0.79**

> R: real data, S: synthetic data, AUROC: area under the ROC curve.
> Mixture of big dataset in Real/Synthetic showed best performance.

https://arxiv.org/abs/2211.12737

Proposal

Feasibility of synthetic data in constructing AI model for medical field

Would using a synthetic data feasible to be used in the medical field?

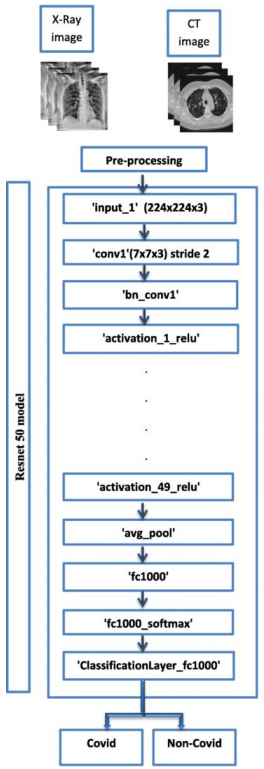
- **Now:** A number of studies in medical field heavily studying on the potential of synthetic data
- **Problem:** However, use of synthetic data might possess high risk of malfunction
- **How?** Using CNN-based model & transformer-based models with settings from previous works

	Name	Train	Train	Test
Model 1	Base	Real	-	Real
Model 1-2	Synthetic base	Synthetic	-	Real
Model 2	Synthetic augmentation	Real+Synthetic	-	Real
Model 3	Synthetic Fine-tuning	Real	Synthetic	Real

* Real(Real world images), Synthetic(cycleGAN synthetic images)

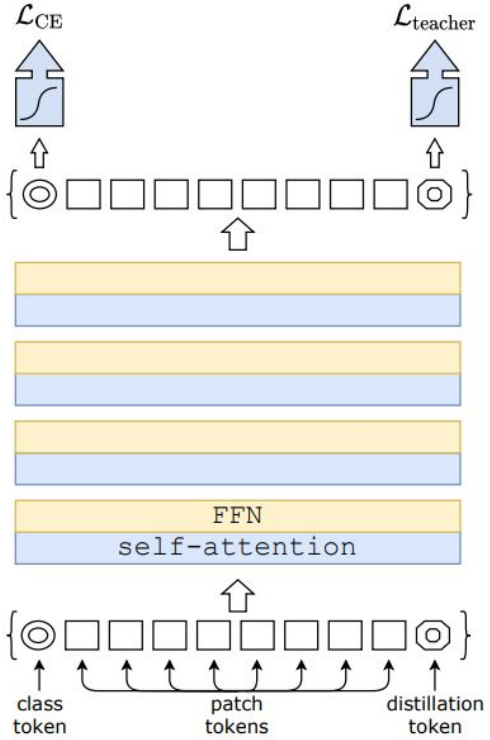
Experimental Setting - Backbone Model

ResNet 50



- CNN-based model
- Widely used in the medical domain

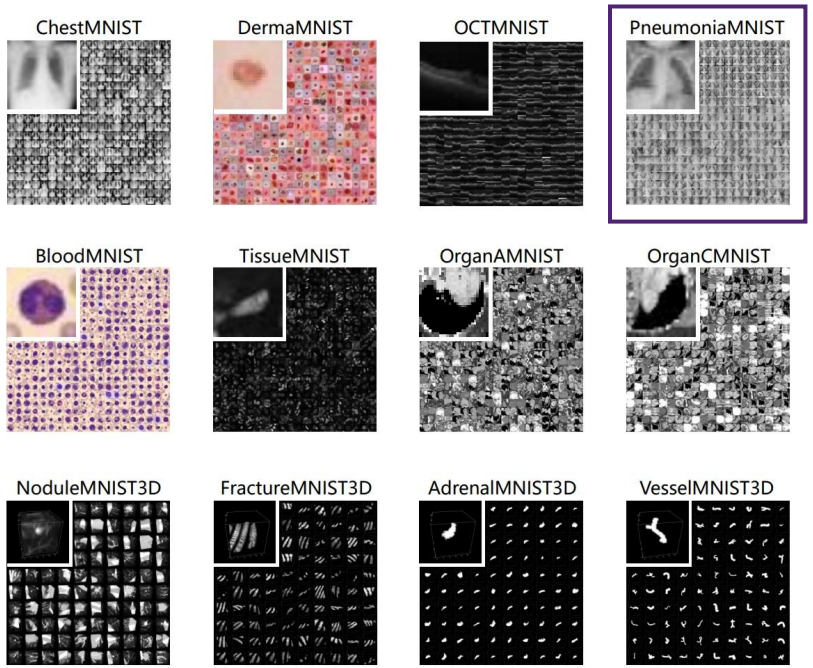
DeiT



- Transformer-based model for vision tasks
- Training data-efficient ViT with optimized hyperparameters

Experimental Setting - Dataset

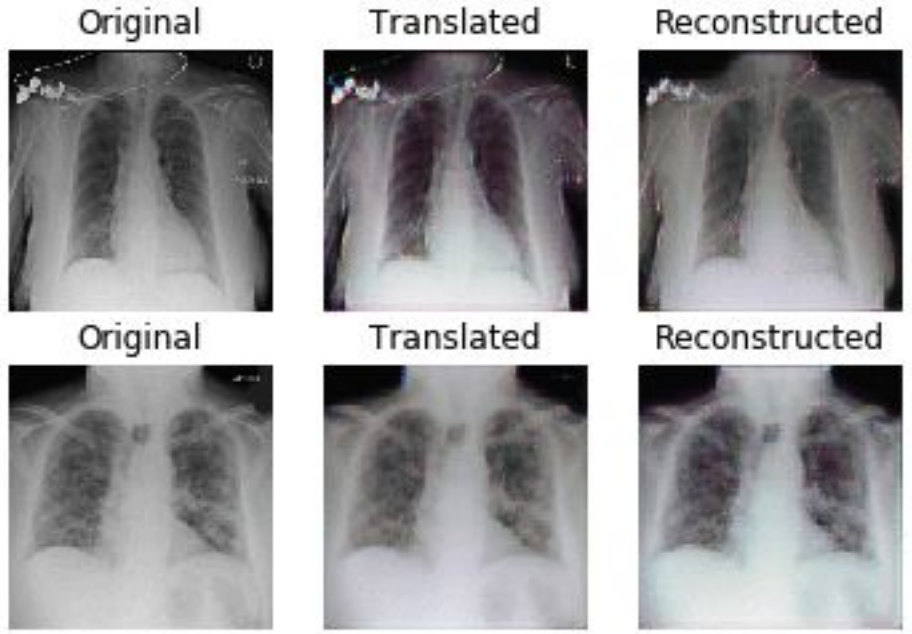
PneumoniaMNIST [3] (Real Data)



- Number of Images : 5,856
- Number of labels : 2 (pneumonia/no)
- Type : Real-world data

https://medmnist.com/

Synthetic Chest X-Rays Dataset [4]



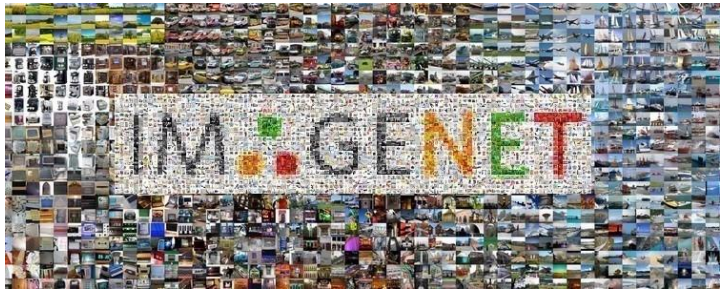
- Number of Images : 21,295
- Number of labels : 2 (pneumonia/no)
- Type : Synthetic data (by CycleGAN)

https://github.com/hasibzunair/synthetic-covid-cxr-dataset

Experimental Setting - Pre-trained Model

Effect of pre-trained model on medical field

ImageNet



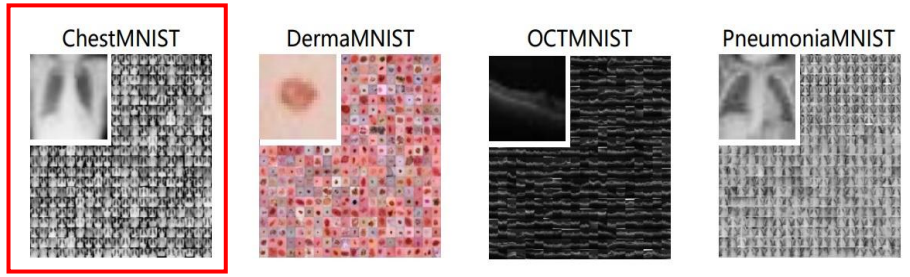
Generalized dataset to compare domain-specific features in the medical field.

- Number of Images: 14,197,122 images
- Number of Labels: 21841 synsets indexed

→ Using the pre-trained checkpoint given by HuggingFace

https://www.image-net.org/index.php

chestMNIST



Domain-specific dataset to check the necessities in the medical field.

- Number of Images : 78468
- Number of labels : 14 (chest-related diseases)

Trained by ourselves following hyperparameters as given in [3]

- BATCH_SIZE = 128
- NUM_EPOCHS = 100
- delaying the learning rate by 0.1 after 50 and 75 epochs.
- Learning rate = 0.001
- Adam optimizer, cross-entropy loss

https://medmnist.com/

Experimental Setting - Dataset Construction

Real Images 4.7k

Synthetic Images 17k

	Name	Train Set	Fine-tuning Set	Test Set
Setting 1	Base	Real (4.7k)	-	Real
Setting 1-2	Synthetic base	Synthetic (17k)	-	Real
Setting 2	Synthetic augmentation_ratio 100%	Real (4.7k) +Synthetic (4.7k)	-	Real
	Synthetic augmentation_ratio 200%	Real (4.7k) +Synthetic (9.4k)	-	Real
	Synthetic augmentation_ratio 300%	Real (4.7k) +Synthetic (14.1k)	-	Real
Setting 3	Synthetic Fine-tuning	Real (4.7k)	Synthetic	Real

Experimental Setting - Overview

Pre-trained Model	
Backbone	No pretrained / From scratch
ImageNet	Generalized dataset to Identify the need for domain-specific features in medical field
ChestMNIST	Domain-specific dataset to Identify the need for domain-specific features in medical field

Dataset

Model's setting

ResNet 50 & DeiT (ICML'21)

- Conducted following the outline from the paper [5]
- Both of the models were trained by following hyperparameters

BATCH_SIZE = 128
NUM_EPOCHS = 50
lr = 0.0005 *
BATCH_SIZE/512
weight_decay = 0.05
warmup_steps = 5

Methods	ViT-B [15]	DeiT-B
Epochs	300	300
Batch size	4096	1024
Optimizer	AdamW	AdamW
learning rate	0.003	$0.0005 \times \frac{\text{batchsize}}{512}$
Learning rate decay	cosine	cosine
Weight decay	0.3	0.05
Warmup epochs	3.4	5
Label smoothing ε	✗	0.1
Dropout	0.1	✗
Stoch. Depth	✗	0.1
Repeated Aug	✗	✓
Gradient Clip.	✓	✗
Rand Augment	✗	9/0.5
Mixup prob.	✗	0.8
Cutmix prob.	✗	1.0
Erasing prob.	✗	0.25

e 9: Ingredients and hyper-parameters for our method and Vit-B.

ResNet50

Report of inference accuracy on Pneumonia MNIST

	Size of dataset (real:synthetic)	Backbone	ImageNet	ChestMNIST
Base	4.7k (1:0)	0.83654	0.87500	**0.88462**
Synthetic base	17.1k (0:1)	0.66186	0.63942	0.71795
Synthetic augmentation	9.4k (1:1)	0.83013	**0.87821**	0.88462
	14.1k (1:2)	0.84455	0.85577	0.88462
	18.8k (1:3)	**0.85256**	0.84615	0.88462
Synthetic FT	4.7k (1: 0)	0.66506	0.72115	0.65064

1. **ChestMNIST pretrained model** (large dataset) performs the best
2. **Backbone model benefits from large dataset** while pretrained models not exactly do so

DeiT

Report of inference accuracy on Pneumonia MNIST

	Size of dataset (real:synthetic)	Backbone	ImageNet	ChestMNIST
Base	4.7k (1:0)	0.83173	**0.86859**	0.80929
Synthetic base	17.1k (0:1)	0.36699	0.41186	0.39423
Synthetic augmentation	9.4k (1:1)	0.80609	0.86218	0.84455
	14.1k (1:2)	**0.83494**	0.85256	0.76923
	18.8k (1:3)	0.83173	0.84615	0.77564
Synthetic FT	4.7k (1: 0)	0.55929	0.40705	0.41506

1. **ImageNet pretrained model** (large dataset) performs the best
2. **Backbone model benefits from large dataset** while pretrained model do not

Evaluation

- **Can we use synthetic data?**
  - It's not a good idea to solely use synthetic data for both backbone model and pretrained model
  - Use the synthetic data with some portion of real data
- **How can we use synthetic data for pre-trained model?**
  - Pretrained model might perceive new data, even the synthetic data specifically for the task, as a noise resulting some degree of degradation of performance
  - Carefully construct a dataset when we are using synthetic data by mixing it with real data
- **CNN vs DeiT**
  - In medical field, CNN still works better with equal amount of data due to its advantages in locality

→ **We need enough understanding of model architecture & task to use synthetic data**

Limitations

1. **Limited amount of data**
2. **Simplistic experimental setting**
    a. Binary classification
3. **DeiT not appropriate for medical field**
    a. Emergence of ViT architectures specifically designed for medical purpose

Future Work

1. **Domain Adaptation**
    a. Implement domain adaptation scheme so that the model can also adapt to synthetic data
2. **Diverse experimental settings**
    a. MedViT: use transformer-based model that is designed for medical area

Citations

[1] Ye, J., Ni, H., Jin, P., Huang, S. X., & Xue, Y. (2023, October). Synthetic Augmentation with Large-Scale Unconditional Pre-training. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 754-764). Cham: Springer Nature Switzerland.

[2] Chambon, P., Bluethgen, C., Delbrouck, J. B., Van der Sluijs, R., Połacin, M., Chaves, J. M. Z., ... & Chaudhari, A. (2022). RoentGen: vision-language foundation model for chest x-ray generation. *arXiv preprint arXiv:2211.12737*.

[3] MedMNIST, “GitHub - MedMNIST/experiments: Codebase for reproducible benchmarking experiments in MedMNIST v2,” *GitHub*. https://github.com/MedMNIST/experiments

[4] Zunair, Hasib, and A. Ben Hamza. "Synthetic COVID-19 chest X-ray dataset for computer-aided diagnosis." *arXiv preprint arXiv:2106.09759* (2021).

[5] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou. (2020). Training data-efficient image transformers & distillation through attention. *arXiv:2012.12877*.

Contribution

박세현

- Ideation, Experimental Setting, Experiments(ResNet Fine-tuning, DeiT DA), Slides

이채은

- Experimental Setting, Paper review, Experiments(DeiT), Slides

김정국

- Experiments(ResNet DA), Slides

진현빈

- Experimental Setting, Experiments(ResNet Fine-tuning), Slides

Thank You

Appendix: Preliminary Studies

			ResNet 50 (ImageNet pretrained)	ResNet50 (torch backbone)	ResNet50 (Backbone + chestMNIST)	DeiT (ImageNet pretrained)	DeiT (Backbone)	DeiT (Backbone + chestMNIST)
		Model Specification	Pytorch pretrained model (*repo 참고)		resnet50_224_1.pth	deit_tiny_224, pretrained=true	deit_tiny_224, pretrained=false	직접 train
						CDTrans		
Fine-tuning X, Training X	`Test`	Real World Data (Pneumonia MNIST)	test loss: 0.75910 acc: 0.25763	test loss: 3.17991 acc: 0.25763	test loss: 0.66526 acc: 0.74809	test_loss: 0.63811, test_accuracy: 0.62500	test_loss: 0.67158, test_accuracy: 0.45833	test_loss: 0.66086, test_accuracy: 0.62500
	`Test`	Synthetic Data	test loss: 0.79496 acc: 0.22099	test loss: 7.63844 acc: 0.22343	**test loss: 0.64296 acc:** `0.64757`	test_loss: 0.97081, test_accuracy: 0.32576	test_loss: 1.37888, test_accuracy: 0.25706	**test_loss: 0.60475, test_accuracy:** `0.77657`
Fine-tuning X, Training O	Train	Real World Data (Pneumonia MNIST)	Avg Val Loss: 0.12448 Avg Val Acc: 0.96947	Avg Val Loss: 0.19245 Avg Val Acc: 0.95061	Avg Val Loss: 0.81291 Avg Val Acc: 0.25103	Avg_val_loss: 0.11753, Avg_val_acc: 0.63069	Avg_val_loss: 0.20964, Avg_val_acc: 0.61420	Avg_val_loss: 0.35968, Avg_val_acc: 0.67099
	`Test`	Synthetic Data	Test Loss: 6.92525 Test Acc: 0.25715	test loss: 6.17713 acc: 0.43005	test loss: 0.67433 acc: `0.60582`	test_loss: 1.26457, test_accuracy: 0.39737	test_loss: 1.27508, test_accuracy: 0.28805	test_loss: 0.71988, test_accuracy: `0.60225`
Fine-tuning O, Training O	Train	Real World Data (Pneumonia MNIST)	Avg Val Loss: 0.11690 Avg Val Acc: 0.96897	Avg Val Loss: 0.34736 Avg Val Acc: 0.83924	Avg Val Loss: 0.32567 Avg Val Acc: 0.85172	Avg_val_loss: 0.11980, Avg_val_acc: 0.61214	Avg_val_loss: 0.19916, Avg_val_acc: 0.60412	test_loss: 0.68993, test_accuracy: 0.61888
	`Test`	Synthetic Data	test loss: 6.53974 acc: 0.28453	test loss: 1.12690 acc: `0.56844`	test loss: 1.46570 acc: 0.42625	test_loss: 1.53560, test_accuracy: 0.29406	test_loss: 1.11288, test_accuracy: 0.35971	test_loss: 0.68993, test_accuracy: `0.61888`

Appendix: Preliminary Studies

- Additional Experiments

: basic classifier vs deeper fc layer

```
fc layer
- default
    model.fc = nn.Linear(in_features, 2)
- simple ver
    model.fc = nn.Sequential(
                nn.Linear(in_features, 512),
                nn.ReLU(),
                nn.Dropout(0.5),
                nn.Linear(512, 2))
- complex ver
    model.fc = nn.Sequential(
                nn.Linear(in_features, 512),
                nn.ReLU(),
                nn.Dropout(0.5),
                nn.Linear(512, 128),
                nn.ReLU(),
                nn.Dropout(0.5),
                nn.Linear(128, 32),
                nn.ReLU(),
                nn.Dropout(0.5),
                nn.Linear(32, 2))
```

			ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch Backbone)	ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch backbone)	ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch backbone)
		Model Specification	Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth		Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth		Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth	
fc layer			default	default	default	simple ver	simple ver	simple ver	complex ver	complex ver	complex ver
Fine-tuning X, Training X	Train	Real World Data (Pneumonia MNIST)									
	`Test`	Synthetic Data	test loss: 0.79496 acc: 0.22099	test loss: 7.63844 acc: 0.22343	test loss: 0.64296 acc: 0.64757	Test Loss: 0.68940 Test Acc: 0.57497	test loss: 0.50886 acc: 0.77657	test loss: 3.06866 acc: 0.22343	Test Loss: 0.78214 Test Acc: 0.22343	test loss: 0.67642 acc: 0.77657	test loss: 0.54014 acc: 0.77657
Fine-tuning X, Training O	Train	Real World Data (Pneumonia MNIST)	Avg Val Loss: 0.12448 Avg Val Acc: 0.96947	Avg Val Loss: 0.19245 Avg Val Acc: 0.95061	Avg Val Loss: 0.81291 Avg Val Acc: 0.25103	Avg Val Loss: 0.13156 Avg Val Acc: 0.96267	Avg Val Loss: 0.19881 Avg Val Acc: 0.95500	Avg Val Loss: 0.26602 Avg Val Acc: 0.94427	Avg Val Loss: 1.00801 Avg Val Acc: 0.95908	Avg Val Loss: 0.73597 Avg Val Acc: 0.23889	Avg Val Loss: 0.37378 Avg Val Acc: 0.93813
	`Test`	Synthetic Data	Test Loss: 6.92525 Test Acc: 0.25715	test loss: 6.17713 acc: 0.43005	test loss: 0.67433 acc: 0.60582	test loss: 10.07361 acc: 0.24827	Test Loss: 5.61334 Test Acc: 0.27312	test loss: 5.89895 acc: 0.46898	test loss: 27.99608 acc: 0.22526	test loss: 0.72137 acc: 0.27293	test loss: 8.19861 acc: 0.44015
Fine-tuning O, Training O	Train	Real World Data (Pneumonia MNIST)	Avg Val Loss: 0.11690 Avg Val Acc: 0.96897	Avg Val Loss: 0.34736 Avg Val Acc: 0.83924	Avg Val Loss: 0.32567 Avg Val Acc: 0.85172	Avg Val Loss: 0.23693 Avg Val Acc: 0.92439	Avg Val Loss: 0.41489 Avg Val Acc: 0.80733	Avg Val Loss: 0.37567 Avg Val Acc: 0.82126	Avg Val Loss: 0.29344 Avg Val Acc: 0.91706	Avg Val Loss: 0.38853 Avg Val Acc: 0.81664	Avg Val Loss: 0.38814 Avg Val Acc: 0.82019
	`Test`	Synthetic Data	test loss: 6.53974 acc: 0.28453	test loss: 1.12690 acc: 0.56844	test loss: 1.46570 acc: 0.42625	test loss: 23.52054 acc: 0.22343	Test Loss: 0.64857 Test Acc: 0.77657	test loss: 0.51407 acc: 0.76163	test loss: 22.15548 acc: 0.22339	`test loss: 0.57433 acc: 0.78145`	`test loss: 0.56862 acc: 0.78399`

- Additional Experiments

: alternative setting (train : synthetic / test : real-world)

			ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch backbone)	ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch backbone)	ResNet 50 (ImageNet pretrained)	ResNet50 (Backbone + MedMNIST)	ResNet50 (torch backbone)
		Model Specification	Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth		Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth		Pytorch pretrained model (*repo 참고)	resnet50_224_1.pth	
fc layer			default	default	default	simple ver	simple ver	simple ver	complex ver	complex ver	complex ver
Fine-tuning X, Training X	Train	Synthetic Data									
	Test	Real World Data (Pneumonia MNIST)	Test Loss: 0.71194 Test Acc: 0.55449	Test Loss: 0.86126 Test Acc: 0.37500	Test Loss: 0.79564 Test Acc: 0.62500	Test Loss: 0.73270 Test Acc: 0.45353	Test Loss: 0.73546 Test Acc: 0.37500	Test Loss: 0.69206 Test Acc: 0.62340	Test Loss: 0.70279 Test Acc: 0.31731	Test Loss: 0.70030 Test Acc: 0.37500	Test Loss: 0.69995 Test Acc: 0.37500
Fine-tuning X, Training O	Train	Synthetic Data	Avg Val Loss: 0.13632 Avg Val Acc: 0.96541	Avg Val Loss: 0.06176 Avg Val Acc: 0.97816	Avg Val Loss: 0.17799 Avg Val Acc: 0.94839	Avg Val Loss: 0.37151 Avg Val Acc: 0.93376	Avg Val Loss: 0.11251 Avg Val Acc: 0.96365	Avg Val Loss: 0.18880 Avg Val Acc: 0.92914	Avg Val Loss: 0.68885 Avg Val Acc: 0.89552	Avg Val Loss: 0.27329 Avg Val Acc: 0.91176	Avg Val Loss: 0.24768 Avg Val Acc: 0.92353
	Test	Real World Data (Pneumonia MNIST)	Test Loss: 10.07782 Test Acc: 0.37500	Test Loss: 16.68277 Test Acc: 0.45353	Test Loss: 10.20715 Test Acc: 0.53846	Test Loss: 1.79577 Test Acc: 0.70513	Test Loss: 17.53240 Test Acc: 0.40064	Test Loss: 11.38501 Test Acc: 0.40545	Test Loss: 6.48878 Test Acc: 0.66186	Test Loss: 14.77841 Test Acc: 0.53205	Test Loss: 30.48742 Test Acc: 0.47276
Fine-tuning O, Training O	Train	Synthetic Data	Avg Val Loss: 0.05449 Avg Val Acc: 0.98500	Avg Val Loss: 0.37981 Avg Val Acc: 0.83757	Avg Val Loss: 0.38192 Avg Val Acc: 0.83383	Avg Val Loss: 0.03954 Avg Val Acc: 0.98525	Avg Val Loss: 0.47171 Avg Val Acc: 0.78469	Avg Val Loss: 0.47751 Avg Val Acc: 0.78469	Avg Val Loss: 0.03728 Avg Val Acc: 0.98730	Avg Val Loss: 0.50338 Avg Val Acc: 0.78469	Avg Val Loss: 0.51695 Avg Val Acc: 0.78469
	Test	Real World Data (Pneumonia MNIST)	Test Loss: 7.07542 Test Acc: 0.52404	Test Loss: 0.99175 Test Acc: 0.53846	Test Loss: 1.12320 Test Acc: 0.57853	Test Loss: 14.84227 Test Acc: 0.49840	Test Loss: 0.96332 Test Acc: 0.37500	Test Loss: 0.90863 Test Acc: 0.37500	Test Loss: 11.66845 Test Acc: 0.51122	Test Loss: 1.00683 Test Acc: 0.37500	Test Loss: 0.99428 Test Acc: 0.37500

→ Simple fine-tuning results in decreased model performance

Domain Adaptation is necessary, even in subtle domain shifts