Multi-source Source-free Domain Adaption for Medical Image Analysis

Lu Liu

SID 520087608

Supervisor: Prof. Weidong (Tom) Cai

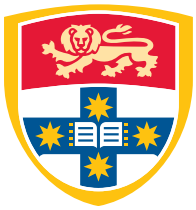
Associate Supervisor: Dr. Dongnan Liu

A thesis submitted in partial fulfillment of

the requirements for the degree of

Bachelor of Science (Honours) in Data Science

Mathematics and Statistics



THE UNIVERSITY OF SYDNEY

October 2022

Statement of originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Lu Liu

Abstract

Training deep neural network to their best performance always needs a huge amount of labelled training data. This requirement cannot be achieved in the medical image area where a lack of professional annotators results in a large number of unlabeled medical images. Unsupervised Domain Adaptation (UDA) methods are an alternative, aiming to extract knowledge from labelled source domains and transfer it to a separate target domain. Recent research focuses more on Multiple source Domain Adaptation (MUDA) as it can aggregate information from different labelled sources so to reduce the domain shift between the source and target domains. However, traditional domain adaptation method requires source data during transferring, it is not feasible in most real-world cases due to reasons like device storage limitation or privacy data protection. Our main focus then becomes: can we improve the feature extractor in the training phase so that the model can have a better performance during adaptation without access to the source data? Thus, we propose a disentanglement-based model which utilises both real data and synthetic data to enhance the feature extraction process, so that the generated feature can be not only significant to label classification, but also domain-invariant. Furthermore, two related medical image benchmark datasets are constructed based on chest lesion and diabetic retinopathy datasets. Based on extensive experiments on the datasets, the proposed model outperforms the baseline model by a large margin.

Acknowledgements

Firstly, I want to thank my supervisors Prof. Weidong Cai and Dr. Dongnan Liu. They have provided me with much help during the whole research process. Thanks for always providing me with new suggestions when I ran out of research ideas. Also thanks for always encouraging me when I was confused. Without them, I would not have been able to complete this honours degree successfully, especially in this fully remote learning mode.

Then, I want to thank my parents. Although I have faced numerous emotional breakdowns, my parents, brothers and sisters have always supported me unconditionally and helped me through my emotional difficulties.

Moreover, I want to thank all my friends. Thank you for your willingness to share the stress of school and life, to be there for me and to take me out from time to time to relieve the stress.

Finally, I would like to thank all the strangers who have helped me this year. During lockdown, it was their unconscious kindness that warmed my heart.

List of Publications

1. Liu L, Liu D, Cai W, Multi-source Source-free Domain Adaption for Medical Image Analysis, 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI). IEEE, 2023 (under preparation and submission).

Contents

1 Introduction **1**

1.1 Background 1

1.2 Main Contribution 3

1.3 Thesis Outline 3

2 Literature Review **5**

2.1 Transfer Learning 5

2.2 Domain Adaptation 6

2.3 Source-free Domain Adaptation 7

2.4 Domain Adaptation in Medical Image Area 9

3 Methodology **11**

3.1 Problem Setting 11

3.2 Architecture Overview 12

3.2.1 Synthetic Data Generation Module 13

3.2.2 Feature Extractor and Source model Training Module 16

3.2.3 Target Adapting and Distillation Module 21

4 Evaluation **25**

4.1 Datasets 25

4.1.1 Chest Lesions 25

4.1.2 Diabetic Retinopathy 26

4.2 Evaluation Metric 27

4.3 FDA Domain Adaptation Result 28

4.3.1 Chest Lesions Synthetic Data 28

4.3.2 Diabetic Retinopathy Synthetic Data 29

5 Results and Discussion **31**

5.1 Baseline 31

5.1.1 Chest Lesions 31
5.1.2 Diabetic Retinopathy 32
5.2 Model Performance 34
5.2.1 Chest Lesions 34
5.2.2 Diabetic Retinopathy 40
5.3 Ablation study 46
5.4 Hyperparameter Analysis 47
5.4.1 Learning Rate 47
5.4.2 Batch Size 48

6 Conclusion and Future Work **49**
6.1 Limitation 49
6.2 Future Work 50

Chapter 1

Introduction

1.1 Background

Deep neural networks have now achieved significant advances in different visualization tasks [13]. However, directly applying source models to test target dataset may lead to poor generalization performance due to the domain shift problem [24]. Unsupervised domain adaptation (UDA), under the setting of an unlabelled target dataset, supposes to transfer information learned from the source domain and improve the task performance on the target domain [19]. Since the UDA is a sub-field of transfer learning, there is an implicit requirement that domains are different but related and the task should be the same for both source and target domains [33].

However, the requirement to have source data during transferring in many domain adaptation methods is crucial and impossible in the real-life scenario [36]. Lack of mobile storage and computation capacity is one of the limitations when deploying domain adaptation algorithms on mobile devices [36]. From Liang et al., given the VisDA-C dataset and ResNet-101 as the backbone of the source network, the storage size for the source dataset is 7884.8 MB while the source model only requires 172.6 MB [17]. Therefore, direct transfer of the trained source model is more achievable than the source dataset. Also, due to data privacy protection [36], the source-free domain adaptation (SFDA) setting in domain adaptation areas is necessary, especially in the medical area when the patients' information must not be disclosed [1].

Source-combined DA was once proposed to solve the domain shift problem. However, the results indicate that domain shift also exists within source domains. Directly combining source to train model will bring a poorer performance [40]. Thus, multi-domain source-free domain adaptation extends the single source domain adaptation setting and

incorporates more prior knowledge from multiple domains and leverage between them [2]. Latent transformation is a popular method in multi-domain domain adaptation. It aims to align between source domains and target domains by either decreasing the discrepancy measures (eg. Maximum Mean Discrepancy [12]) or adversarial loss [40].

In the medical field, SFDA is even more challenging, as the medical domain dataset has a large data variation problem and label imbalanced problem. Different scanners can cause image-level distribution heterogeneity [11], as from Figure 1.1. Demographics of patient populations or patients' pathological conditions will also cause domain shift [7]. Research from Choudhary et al., suggests that rare diseases with fewer positive cases will bring the image dataset imbalanced, causing more difficulty for model learning [7].

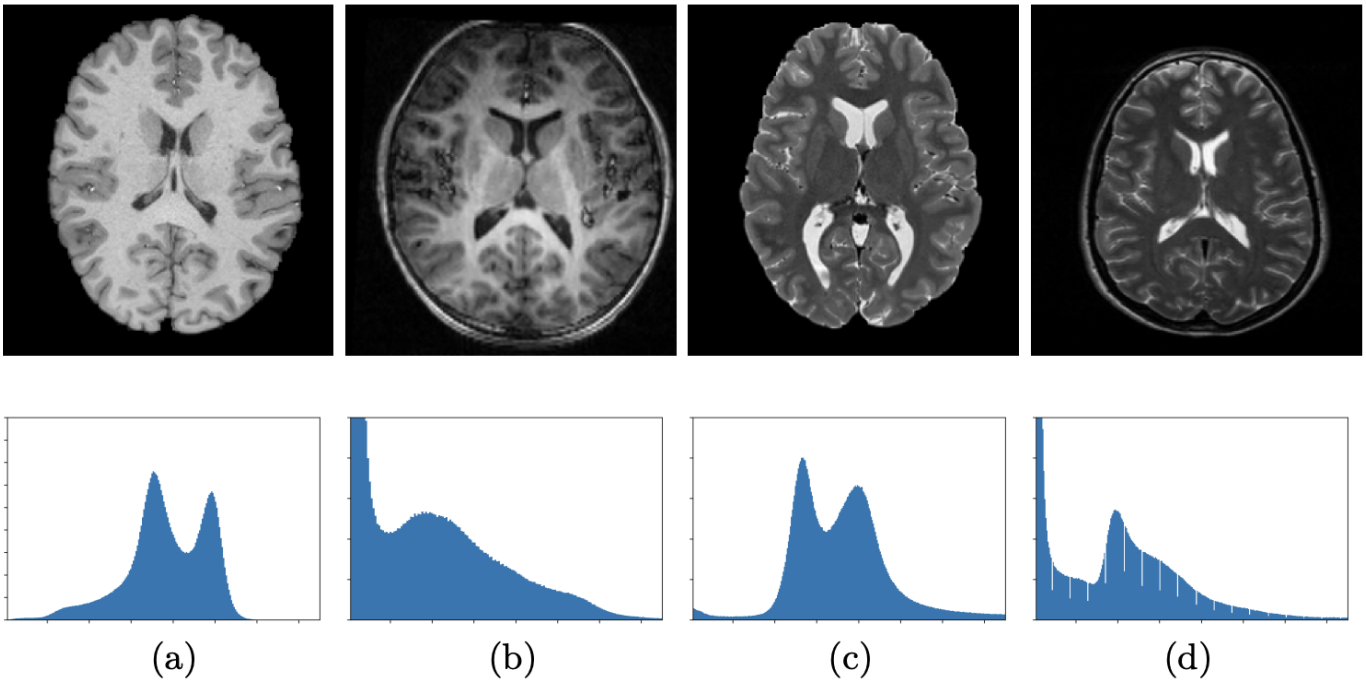


Figure 1.1: Image slices (top) and corresponding histograms (bottom) of normalized T1w (a,b) and T2w (c,d) MRIs from different scanners. There exist intensity and contrast differences between images. Source from N. Karani et al [14]

.

In this research, a novel disentanglement-based model is proposed to enhance the feature extraction step and achieve superior adaptation performance on the constructed two medical datasets. The model first utilises the unsupervised Fourier domain adaptation method to generate synthetic data. Then, the real data and synthetic data are fed into the

feature extractor network, hoping to obtain features that are significant towards classification tasks while performing as domain-invariant between the real and synthetic domains. Moreover, during target domain adaptation, the model implements a pseudo-labelling strategy with information maximisation on the target domains. Finally, we find optimal weights regarding all source models and distil them into a single target model.

1.2 Main Contribution

The contribution of this work can be concluded as follows:

- Synthetic data generation is proposed to enhance the source domain feature learning process.
- A disentanglement-based model is utilised. With real data and synthetic data incorporated into the source model, significant domain-invariant features can be extracted for later task learning and domain adaptation.
- Several state-of-art models are included and a novel model is constructed based on those baselines to address the problem of multi-source unsupervised domain adaptation without access to the source data.
- Two related medical image benchmark datasets are constructed based on chest lesion and diabetic ratinopathy datasets. Also, I validated and analysed the experimental results and the effectiveness of the proposed model. The proposed model presents superior performance over the baseline model.

1.3 Thesis Outline

In the introduction part (Chapter 1), I give a background introduction about concepts about domain adaptation, related challenges in the medical image area and a general overview of the proposed model.

In Chapter 2, literature reviews about different settings of domain adaptation and problems are presented. Current domain adaptation applications and challenges in the medical image area are also discussed in detail.

In Chapter 3, the specific framework of the proposed model and related state-of-art models are described. The proposed model includes three parts: synthetic data generation, feature extractor & source model training and target adapting & distillation module.

In Chapter 4, I introduced two related datasets which are constructed based on Chest Lesions and Diabetic Retinopathy. Evaluation metrics and FDA synthetic data samples are explained in this module.

In Chapter 5, extensive experiments are conducted. The performance of the proposed model and baseline are compared and analysed. Ablation studies about domain classifiers and hyperparameter analysis are also presented in the module.

In Chapter 6, I provide a general conclusion about the proposed model, along with limitations and potential future research directions.

Chapter 2

Literature Review

2.1 Transfer Learning

Transfer learning is a general machine learning concept, of which transfers information from an source domain to improve the model for the target domain, where either the tasks or the domains or both may differ [33].

The formal definition of transfer learning includes two parts: domain $\mathcal{D} = \{\mathcal{X}$, P(X)$\}$ and task $\mathcal{T} = \{\mathcal{Y}, \theta(\cdot)\}$ [32]. For domain $\mathcal{D}$, $\mathcal{X}$ represents the feature space of all possible feature vectors and P(X) represents the marginal probability distribution of a particular learning sample, of which an X $= [x_1, x_2, ..., x_n] \in \mathcal{X}$. Regarding task $\mathcal{T}$, $\mathcal{Y}$ represents the label space and $\theta(\cdot)$ is the predictive function which can be learned from the feature vector. The general definition can be also split into source and target two parts. The detailed notation is listed in the below table.

$\mathcal{D}_S$	Source domain	$\mathcal{D}_T$	Target domain
$\mathcal{T}_S$	Source task	$\mathcal{T}_T$	Target task
$x_{Si} \in \mathcal{X}_S$	Source ith data instance	$x_{Ti} \in \mathcal{X}_T$	Target ith data instance
$y_{Si} \in \mathcal{Y}_S$	Source ith data instance's label	$y_{Ti} \in \mathcal{Y}_T$	Target ith data instance's label
$\mathcal{X}_S$	Source feature space	$\mathcal{X}_\mathcal{D}$	Target feature space
$\mathcal{Y}_S$	Source label space	$\mathcal{Y}_\mathcal{D}$	Target label space
$\theta_S(\cdot)$	Source predictive function	$\theta_\mathcal{T}(\cdot)$	Target predictive function

Table 2.1: Notation for transfer Learning

When domain and task are the same, $\mathcal{D}_S = \mathcal{D}_T$, $\mathcal{T}_S = \mathcal{T}_T$, then the problem simplifies to the traditional machine learning task [19]. When the domains differ but are related, and tasks remain the same, then the problems can be transformed into domain adaptation.

2.2 Domain Adaptation

Domain Adaptation (DA) is a sub-field of domain adaptation, of which the task of source and target remains the same $\mathcal{T}_S = \mathcal{T}_T$. There are various kinds of divisions based on different characteristics of DA. Whether the feature space between the source and target is identical decides the type of homogeneous ($\mathcal{X}_S = \mathcal{X}_T$) and heterogeneous ($\mathcal{X}_S \neq \mathcal{X}_T$) domain adaptation [31] [33]. There is also categorisation based on whether the target training data is not, partially or fully labelled, corresponding to unsupervised, semi-supervised and supervised domain adaptation, respectively. However, according to the DA definition, source data is provided by default.

Unsupervised Domain Adaptation (UDA) assumes the target data is unlabelled given the labelled training domain data. Two sub-areas Single-source Unsupervised Domain Adaptation (SUDA) and Multi-source Unsupervised Domain Adaptation (MUDA) is introduced based on the number of source models. Compared to SUDA, MUDA setting is more relevant to the real-world scenario but also takes more challenges as we need to shift and match all training domains towards the target domain [42].

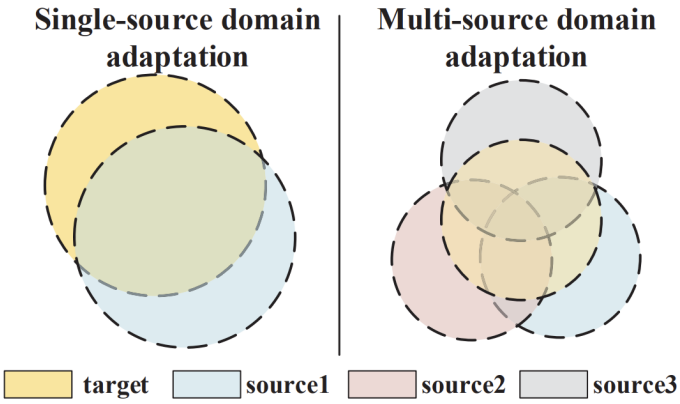


Figure 2.1: Plot of SUDA and MUDA [42]

In domain alignment, latent space transformation and alignment are often mentioned [40]. Latent space, or embedding space, is a multi-dimensional vector to capture semantic features of data to support later data analysis [18]. Discrepancy-based method and adversarial method are two mainstream ways to perform latent space alignment [40]. Discrepancy-based methods optimise the distance between different domains and utilise

various discrepancy losses to align features. Some typical discrepancy measures are maximum mean discrepancy (MMD) [12], correlation alignment [28] [33] and moment distance [20] [40]. Traditional adversarial method to align the source and target domain by generating fake data from the source data with Generative Adversarial Network (GAN) [10]. Domain-adversarial training of domain adversarial neural networks, which belongs to adversarial discriminative methods, employ the idea of gradient reversal layers to extract features which can perform well in the label classification but indistinguishable to determine the domain [9].

Furthermore, the image synthesis method has been widely used in the domain adaptation area and is declared as a critical component in visual machine learning model designing. By efficiently enlarging the training data, it can control the generating process and provide better data distribution and content variety [29]. Constrained image synthesis is a branch of image synthesis topic, with images generated under the given specified constraint, like input text, description text [34]. There are some famous applications regarding constrained image synthesis in the survey from Wu et al., like cycle-consistent Generative Adversarial Networks (CycleGAN) which implements unpaired image-to-image translation and Stacked Generative Adversarial Networks (StackGAN) which generates high-resolution synthetic images based on given text description [34].

2.3 Source-free Domain Adaptation

Hypothesis Transfer Learning (HTL) has a different setting compared to domain adaptation, as it aims to only incorporate the hypotheses trained from the source domain and adapt to learn the feature domain without the source model as Figure 2.2. However, it requires the target data to be labelled which is unrealistic in most real-world scenarios [2] [21].

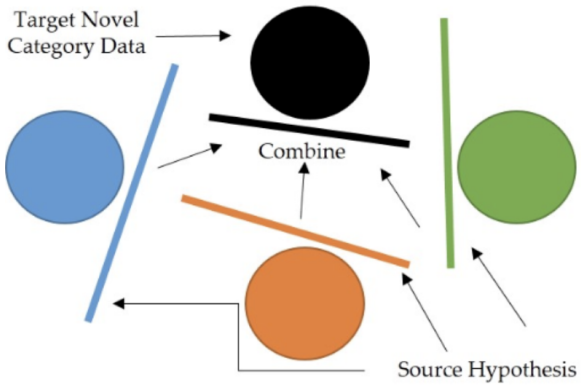


Figure 2.2: Plot of Hypothesis Transfer Learning [3]

Source-free Domain Adaptation (SFDA) setting is extended from the previous two settings. Traditional domain adaptation method requires source data when adapting, it is not feasible in most real-world cases, such as the adapting online source to mobile devices while having capacity limitations, or data privacy protection restrictions especially in the medical image domain [2] [36]. SFDA setting only has access to the pretrained source models and then adapts to the unlabeled target domain. A self-supervised learning method for deep clustering using 'pseudo-labelling' is proposed to solve the dilemma of unlabeled data in the target domain [5].

Setting	Multi-domains	No source data	Source Model	Unlabeled Target Data
SUDA	✗	✗	✓	✓
MUDA	✓	✗	✓	✓
HTL	✗	✓	✓	✗
SFDA	✗	✓	✓	✓
MSFDA	✓	✓	✓	✓

Table 2.2: Comparison between different settings. [2]

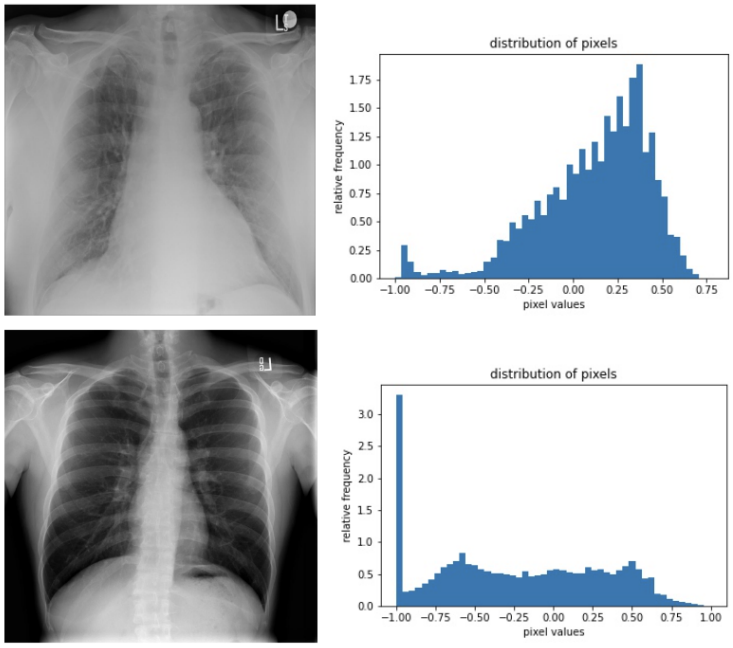


Figure 2.3: Medical Images and corresponding intensity distribution

2.4 Domain Adaptation in Medical Image Area

For traditional deep learning methods, plenty of correct labelling is critical for model training. While in the medical image area, this problem gets worse due to privacy data protection for patients [1]. Also for rare diseases, a low number of positive cases may contribute to a highly imbalanced dataset, adding difficulties to model learning [7]. Source-free unsupervised domain adaptation is proposed to alleviate the problem, as it extracts the information from the provided source domain data and applies it to improve the task performance on the target domain without requiring the target labels' information.

However, the domain shift problem between the source and target dataset may result in significant performance degradation, especially in medical image areas where different cities and scanners result in varying photo quality [11]. Figure 2.3 illustrates the sample images and intensity distribution after normalization, from which we can observe a clear distribution shift. Some research also indicates that data shift occurs due to demographics of patient populations or patients' pathological conditions and those distribution variations may bring challenges to model generalization [7][22].

Based on previous latent space alignment methods, different algorithms have been

proposed to minimise the source and target domain shift in the medical image area. Wachinger and Reuter introduced instance re-weighting in the supervised domain adaptation of Alzheimer's Disease dataset [30]. Zhu et al. designed a maximum mean discrepancy-based Multiple Kernel Learning method to map all the MRI and PET data to a common space and alleviate the modality heterogeneity issue [41]. Zhang et al. utilised a cycle feature adaptation module in the Unsupervised Conditional consensus Adversarial Network (UCAN) to improve the performance on brain disease identification [39]. Bousmalis et al. proposed a novel domain separation network to learn not only the shared domain features but also the domain-specific features [4].

Chapter 3

Methodology

3.1 Problem Setting

This research is based on a multi-source source free domain adaptation setting, with source models pretrained on multiple source datasets and then adapt to the target domain dataset. The adaptation process doesn't have access to the source data and the target dataset is unlabelled. Given M different source domains $\{\mathcal{D}_{\mathcal{S}j}\}_{j=1}^{M}$, with each domain containing K categories (K = 2 in our binary classification cases), we consider a sets of source models $\{f_{\mathcal{S}}^{j}(\cdot)\}_{j=1}^{M}$, with j_{th} source model $f_{\mathcal{S}}^{j}(\cdot)$ indicates mapping: $\mathcal{X} \rightarrow \mathcal{R}^{K}$ which is learned during the training of N data instances $(x_{S_j}^{i}, y_{S_j}^{i})_{i=i}^{N_j}$ in the domain j. The task is, given the target unlabelled domain dataset $\mathcal{D}_{\mathcal{T}} = \{x_T\}_{i=1}^{N_i}$, we want to learn the target predictive mapping function $f_{\mathcal{T}}(\cdot)$: $\mathcal{X} \rightarrow \mathcal{R}^{K}$ with only access to M source models.

3.2 Architecture Overview

In this section, I will first give an high-level overview of the proposed model. Then I will decompose and explain each step of the model.

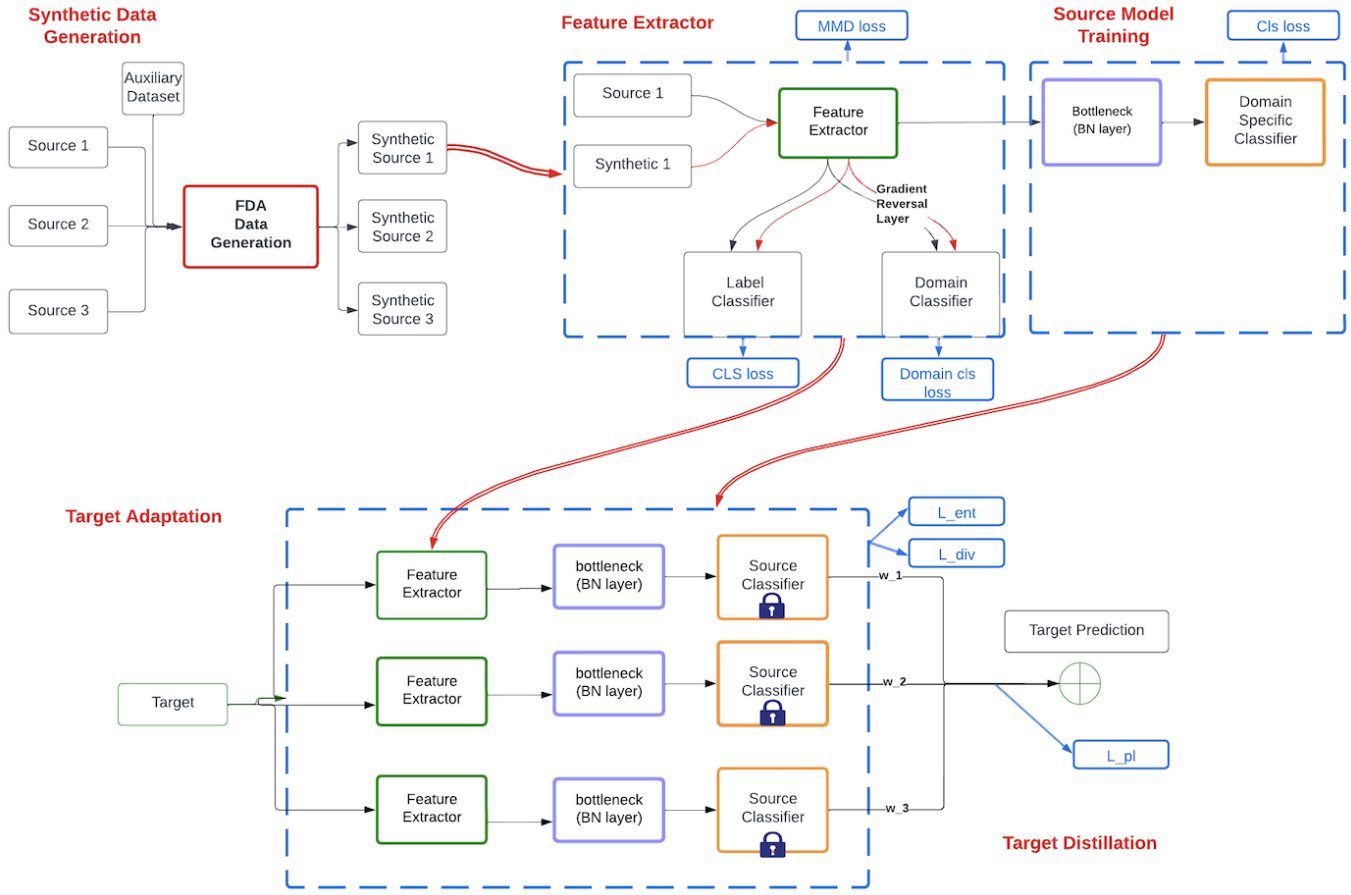


Figure 3.1: This is the high-level overview of the proposed model. The model contains three major steps: synthetic data generation, feature extractor & source model training and target adaptation & target distillation. The red dashes are implemented to explain order of the pipeline and how models are transferred between different modules.

3.2.1 Synthetic Data Generation Module

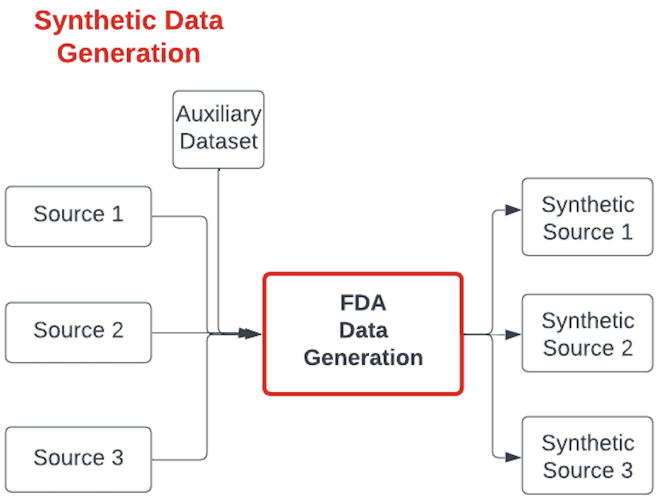


Figure 3.2: Synthetic Data Generation Module

Based on Yang and Soatto [37], an image's high-level semantic meaning should not be influenced by low-level amplitude variation, like the characteristic of scanners, sensors or illuminates. However, a learning-based model is more prone to learn this along with other types of nuisance variability. Thus, for better adaptation and generalization performance, we should introduce such variability to the training set.

So the Fourier-based domain adaptation method is designed to reduce the domain gap between the source and target dataset. Research has shown that the phase component in the Fourier spectrum retains the most of high-level semantic information of the original signal [35]. By swapping the low-frequency component of the spectrum of the source image with the selected target image, the transformation generates 'source image in target style' and narrows the domain gap between the source and target in an unsupervised way [37].

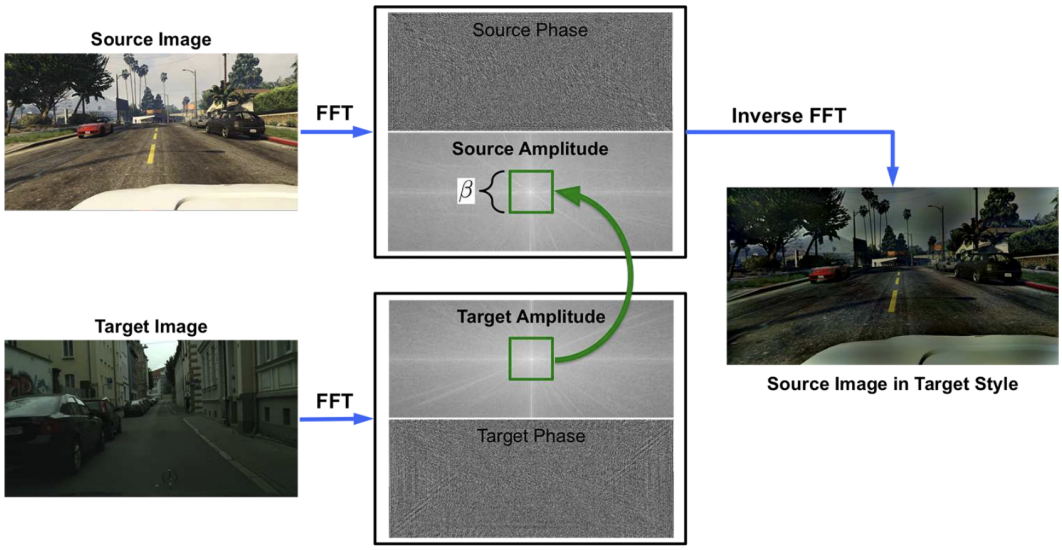


Figure 3.3: FDA structure [37]

Our model utilises the Fourier transform and inverses Fourier transform to generate synthetic data for data augmentation. The auxiliary dataset is cocoval_75 [1]. Given a sample image $x_s \sim \mathcal{D}_s$, randomly select one target image $x_a \sim \mathcal{D}_a$ from the auxiliary dataset and perform one-to-one mapping to generate corresponding synthetic data image. Since the original data source is grey-scale, so the auxiliary image and the corresponding synthetic data are then also transformed to grey-scale and have the same shape as the source image. Then, the FDA method changes the low-level spectrum (amplitude) of the source image to that of the target image and generates a new source image in the target style (the synthetic data).

For a single channel, its Fourier transform:

$$\mathcal{F}(x)(u,v) = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} x(h,w)\, e^{-j2\pi\left(\frac{h}{H}u+\frac{w}{W}v\right)} \tag{3.1}$$

and $\mathcal{F}^{-1}(x)$ is the inverse Fourier transform that maps spectral signals including phase and amplitude back to image space [37].

Also, the amplitude and phase components is:

$$\mathcal{A}(x)(u,v) = [R^2(x)(u,v) + I^2(x)(u,v)]^{\frac{1}{2}} \tag{3.2}$$

$$\mathcal{P}(x)(u,v) = arctan[\frac{I(x)(u,v)}{R(x)(u,v)}] \tag{3.3}$$

[1] https://cocodataset.org/#download

where R(x) and I(x) represent the real and imaginary part of $\mathcal{F}(x)$. The original method denotes M_β as a masking, whose value is zero except the central region.

$$M_\beta(h, w) = \mathbf{1}_{(h,w)\in[-\beta h:\beta h,-\beta W:\beta W]} \tag{3.4}$$

Then the full formulation of Fourier domain adaptation (FDA) can be formulated as Equation 3.5.

$$x^{s\to t} = \mathcal{F}^{-1}([M_\beta \circ \mathcal{F}^A x^t + (1 - M_\beta) \circ \mathcal{F}^A x^s, \mathcal{F}^P x^s]) \tag{3.5}$$

With phase component unchanged, the method only changes the low frequency part of the amplitude of the source image with the target image and then map back to the source image with inverse Fourier transform ($x^{s\leftarrow t}$).

The overall algorithm is shown as below.

Algorithm 1 Fourier-based Synthetic Data Generation

Input : Source Image **s** & Auxiliary Image **t**

Output: Synthetic Image

function FDA($s, t, L = 0.1$) ▷ L - mask ratio

$s_{fft} \leftarrow$ **FFT**(s) ▷ 3.1

$t_{fft} \leftarrow$ **FFT**(t)

$s_{amp}, s_{phase}, t_{amp}, t_{phase} \leftarrow$ **extract_ampl_phase**(s_{fft}, t_{fft}) ▷ 3.2, 3.3

$synthetic_{amp} \leftarrow$ **low_freq_mutate**(s_{amp}, t_{amp}, L) ▷ 3.4

$fft_{synthetic} \leftarrow$ **construct_source**($synthetic_{amp}, s_{phase}$) ▷ 3.5

$synthetic \leftarrow$ **IFFT**($fft_{synthetic}$)

return *synthetic* ▷ Source in target style

end function

3.2.2 Feature Extractor and Source model Training Module

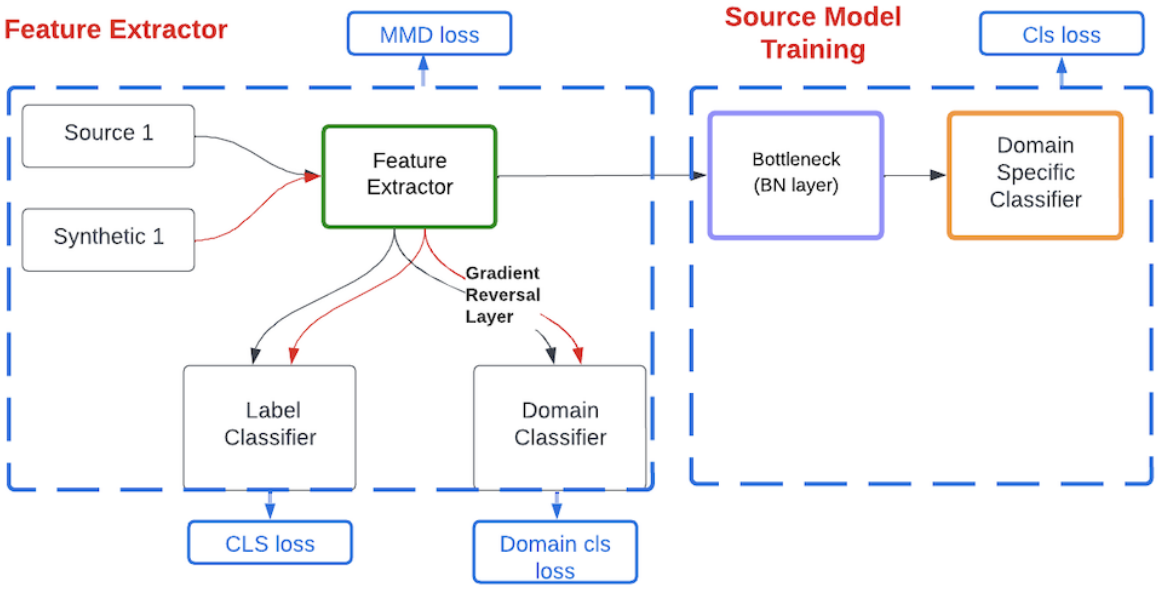


Figure 3.4: Feature Extractor and Source Training Module

This module includes two parts: feature extractor and source model training.

Multiple Feature Spaces Adaptation Network (MFSAN)

The feature extractor part is modified based on the Multiple Feature Spaces Adaptation Network (MFSAN) [42].

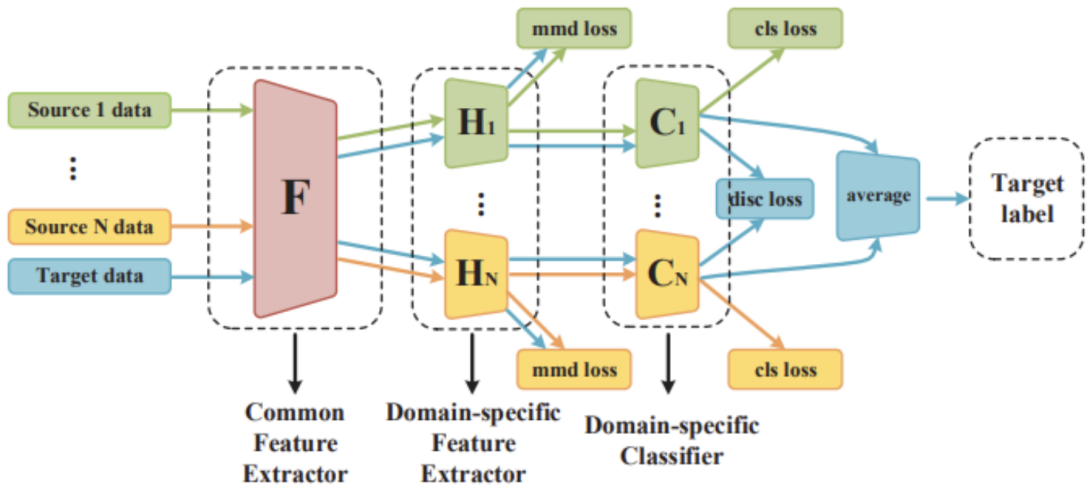


Figure 3.5: MFSAN structure [42]

The original MFSAN implements a two-stage alignment framework that consists of three components: common feature extractor, domain feature extractor and domain-specific classifier. It firstly proposes a common feature to map source and target data to a common feature space, of which the parameters are shared between each training model. After receiving the common features from the shared network, each model trained a separate feature extraction network so to map the common features to domain-specific features. To reduce the distribution discrepancy between each pair of source and target domain, the model uses maximum mean discrepancy (MMD) to align distributions. The formal definition of the MMD measure is shown below. If two distributions are identical, then all the statistics should be equal and $D_{\mathcal{H}}(p, q) = 0$ [42].

$$D_{\mathcal{H}}(p, q) \triangleq \|\mathbb{E}_p[\phi(x_s)] - \mathbb{E}_q[\phi(x_s)]\|^2_{\mathcal{H}} \tag{3.6}$$

where $\mathcal{H}$ represents the reproduced kernel Hillbert space (RKHS) with a characteristic kernel, and $\phi(\cdot)$ represents the mapping from the original sample to the RKHS.

Due to the lack of data distribution and law of large number, the implementation of the model uses empirical unbiased estimator $\hat{D}_{\mathcal{H}}(p, q)$ to estimate MMD.

$$\hat{D}_{\mathcal{H}}(p, q) = \left\| \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{H}_{\mathcal{S}}} \phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{\mathbf{x}_j \in \mathcal{H}_{\mathcal{T}}} \phi(\mathbf{x}_j) \right\|^2_{\mathcal{H}} \tag{3.7}$$

For source j, the final MMD loss between source domain (denoted as $\mathbf{s}$) and synthetic domain (denoted as $\mathbf{t}$) is defined as:

$$\mathcal{L}_{mmd} = \frac{1}{N} \sum_{j=1}^{N} \hat{D}(\, F\,(X_{s_j}),\, F\,(X_t)\,) \tag{3.8}$$

After domain alignment, models are trained to have higher label prediction in their own domains. However, different domain models may have disagreement between the prediction of some test samples, especially those samples near the classification boundary, so this step's optimisation in MFSAN includes distribution loss $\mathcal{L}_{disc}$ to minimise the difference between all domain classifiers:

$$\mathcal{L}_{disc} = \frac{2}{N \times (N-1)} \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \mathbb{E}_{x \sim X_t}[\,|\, C_i\,(H_i\,(F(x_k))) - C_j\,(H_j\,(F\,(x_k)))\,|\,] \tag{3.9}$$

After aligning those classifiers, the overall prediction is the average of all outputs. This step includes classification error which is the cross entropy loss between prediction and true target labels.

Our modified version

The MFSAN model requires both source-target pair and all source data available at the training phase. However, under the source-free setting, only one single source domain data should be available while training each source model. So in the training stage, I feed the source and synthetic dataset to the feature extraction part and discard $\mathcal{L}_{disc}$ only considering the loss which is related to those data ($\mathcal{L}_{mmd}$ and $\mathcal{L}_{cls}$). The feature extractor network is expected to extract features which are also perform significantly to label classification.

Furthermore, after mapping all features to the domain feature space, the classification result should not be influenced by whether the samples are from the generated data or the original source (domain-invariant). This problem has been solved by the Unsupervised Domain Adaptation by Backpropagation (DANN) network [9].

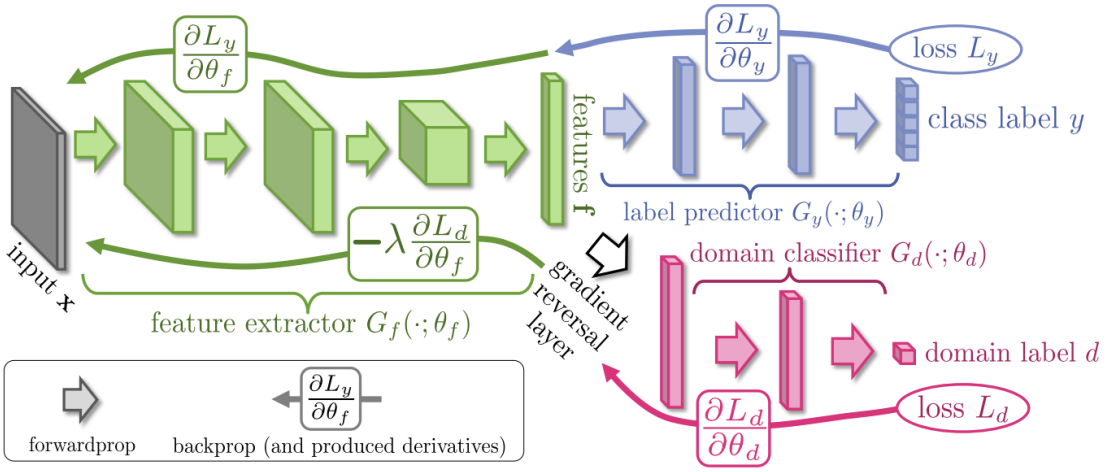


Figure 3.6: DANN structure[9]

The main idea in DANN is to seek parameters of feature extractor θ_f, to minimise the loss for label predictor as well as maximise the loss for the domain classifier. This design

aims to find domain-invariant features for label classification.

$$E(\theta_f, \theta_y, \theta_d) = \sum_{i=1...N, di=0} L_y(G_y(G_f(\mathbf{x_i}; \theta_f); \theta_y), y_i) - \lambda \sum_{i=1,...,N} L_d(G_d(G_f(\mathbf{x_i}; \theta_f); \theta_d), y_i)$$
$$= \sum_{i=1...N, di=0} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1,...,N} L_d^i(\theta_f, \theta_d)$$

where $L_y(\cdot, \cdot)$ is the loss for label prediction and Ł$_d(\cdot, \cdot)$ is the loss for the domain classification. DANN also proposes Gradient Reversal Layer (GRL) for backpropogation. It performs as identity mapping during forward transformation. While in backward propagation, it takes the gradient from the gradient, multiplies it by -λ, and passes it to the previous layer [9]. Inspired from DANN, our proposed model also includes domain classifier as well as the gradient reversal layer, hoping to extract domain invariant features in the training phase.

The overall loss function includes three part: maximum mean discrepancy loss $\mathcal{L}_{mmd}$ to align the feature distribution between the source and synthetic domains, classification loss between the prediction and the true label $\mathcal{L}_{cls}$ and the domain classification loss between the predicted domain and true domain $\mathcal{L}_{domain}$.

For source classifier j, the classification loss is formulated as below, where $J(\cdot, \cdot)$ is the cross-entropy loss function.

$$\mathcal{L}_{cls} = \sum_{i=1}^{N} \mathbb{E}_{x \sim s_j} J(\, C_j \, F(\mathbf{x}_i^{s_j}), \, \mathbf{y}_i^{s_j}; \theta_c) \tag{3.10}$$

Regarding the domain classification loss, I calculate the additive domain loss using the cross entropy method, $J(\cdot, \cdot)$ is the cross-entropy loss function, and $\mathbf{d}s_i$ is the one-hot domain representation of the ith instance of the source data.

$$\mathcal{L}_{domain} = \mathcal{L}_{domain_s} + \mathcal{L}_{domain_t}$$
$$= \sum_{i=1...N_s} L_d(\mathbf{x}_i^{s_j}; \theta_f, \theta_d) + \sum_{i=1...N_t} L_d(\mathbf{x}_i^{t_j}; \theta_f, \theta_d)$$
$$= \sum_{i=1...N_s} J(C_{d_j}(F(\mathbf{x}_i^{s_j})), \mathbf{d}_i^{s_j}) + \sum_{i=1...N_t} J(C_{d_j}(F(\mathbf{x}_i^{t_j})), \mathbf{d}_i^{t_j})$$

The overall loss function is formulated as below. Therefore, the model mainly wants to seek optimal parameters $\hat{\theta_f}$, $\hat{\theta_d}$ and $\hat{\theta_c}$ that minimise the loss, as Equation 3.13 and 3.14:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \gamma(\mathcal{L}_{mmd} - \mathcal{L}_{domain}) \tag{3.12}$$

$$\hat{\theta_f}, \hat{\theta_c} = \underset{\theta_f, \theta_c}{\operatorname{argmin}}(L_{total}; \theta_f, \theta_c, \hat{\theta_d}) \tag{3.13}$$

$$\hat{\theta_d} = \arg\max_{\theta_d}(L_{total}; \theta_d, \hat{\theta_f}, \hat{\theta_c}) \tag{3.14}$$

Algorithm 2 Feature Extractor

Input : Number of iterations T

Output: Feature Extractor net **F** for j_{th} source domain

for *i in 1: T* **do**

$sample_{s_j} = \{x_i^{s_j}; y_i^{s_j}\}_{i=1}^m \leftarrow$ **Sample**(s_j) ▷ where source domain $s_j = (X_{s_j}, Y_{s_j})_{i=1}^N$

$sample_{t_j} = \{x_i^{t_j}; y_i^{t_j}\}_{i=1}^m \leftarrow$ **Sample**(t_j) ▷ where synthetic domain $t_j = (X_{t_j}, Y_{t_j})_{i=1}^N$

Get $\mathbf{F}(x_i^{s_j}), \mathbf{F}(x_i^{t_j})$ from feature extractor net **F**

Calculate $\mathcal{L}_{mmd}$ for $F(x_i^{s_j}), F(x_i^{t_j})$ as Equation 3.8

Feed the features to label classifier C_j and domain classifier C_{d_j}

Calculate domain loss $\mathcal{L}_{domain}$ as Equation 3.11.

Calculate label classification loss $\mathcal{L}_{cls}$ as Equation 3.10.

Update the parameters in **F**, C_j and C_{d_j} by minimizing the total loss in Equation 3.12.

end

After the feature extractor network completes its training, the feature extractor part of the source model is replaced with the network that has already been trained. Then along with the batch normalization layer and the domain-specific classification layer, all of the layers form the source training model. The only data fed into training the model is only the source data. During training, the parameters of the feature extractor network are frozen and only update the parameters of the remaining batch normalization and classification layers so that the parameters of those layers will only keep the information about the source data and not the noisy synthetic data. The loss function for this step is just the label classification loss between the prediction and the true labels, as Equation 3.15.

$$\mathcal{L}_{cls} = \sum_{i=1}^{N} \mathbb{E}_{x \sim s_j} J(\, C_j\, (BN_j(F(\mathbf{x}_i^{s_j}))),\, \mathbf{y}_i^{s_j}; \theta_{c_j}) \tag{3.15}$$

where BN_j is the batch normalization layer for the j th source model.

3.2.3 Target Adapting and Distillation Module

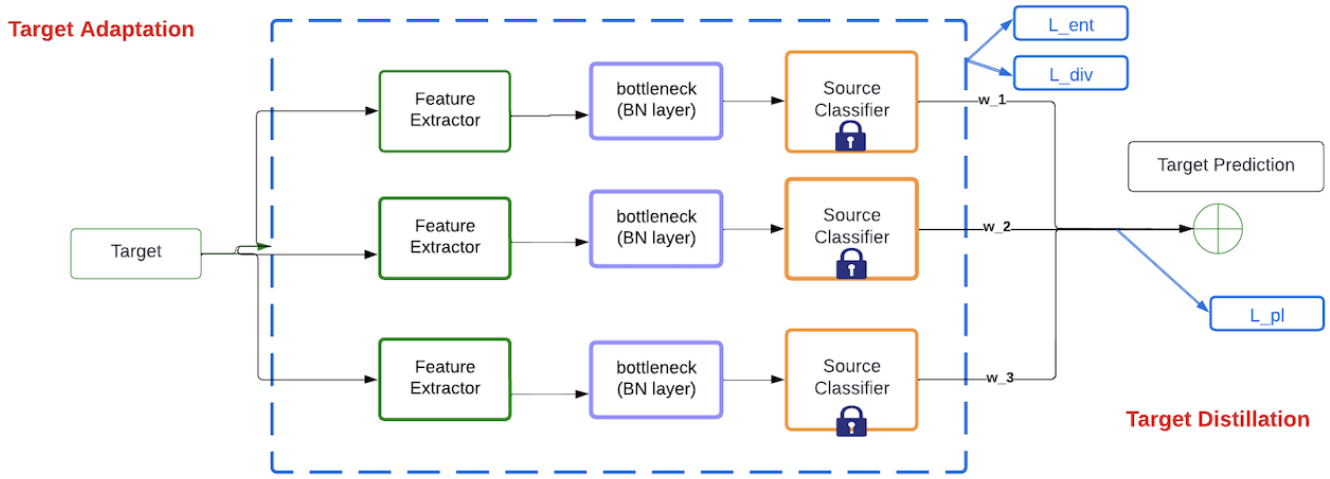


Figure 3.7: Target Adapting and Distil Module

Since the model is constructed based on the Data free Multi-source Unsupervised Domain Adaptation (DECISION) model, so I retain their logic and construction about the adaptation and distil module.

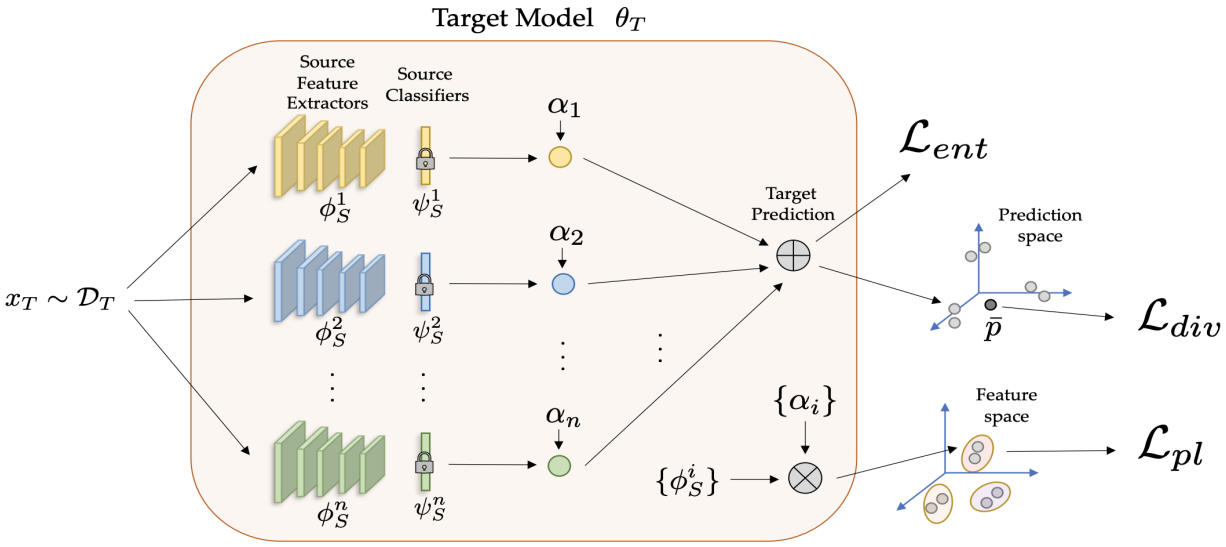


Figure 3.8: DECISION framework[2]

The overall framework of DECISION is split into three stages: source model training, target domain adaptation and target model distillation. Each source model consists of a source feature extractor and source classifier.

To distil the source model into a single target model, DECISION model implements a weight information maximisation idea. Each source model has different transferability concerning the target domain, higher transferability model should account for a higher weight α in the final model's composition. So the ensemble of the source model includes optimisation over the term $\{\theta_S^j, \alpha_j\}$. During adapting, the model assumes the input distribution satisfies the cluster assumption. The decision boundaries of labels is not located in the place where data is densely distributed [6] [2]. So the $\mathcal{L}_{ent}$ defined a cross-entropy loss term of the label distribution given the input instance x. The lower of $\mathcal{L}_{ent}$ means the target model has a higher confidence about the label.

$$\mathcal{L}_{ent} = -\mathbb{E}_{X_T \in \mathcal{D}_\mathcal{T}}[\sum_{j=1}^{K} \delta_j(\theta_T(x_T)) \, log\left(\delta_j(\theta_T(x_T))\right)] \tag{3.16}$$

where $\theta_T(x_T) = \sum_{j=1}^{n} \alpha_j \, \theta_S^j(x_T)$ denotes the target model, and $\theta(\cdot)$ denotes the softmax function.

However, some degenerate solutions may occur that the target model always predicts a single model in order to minimise the conditional entropy, so the model also consider target label balance with $\mathcal{L}_{div}$.

$$\mathcal{L}_{div} = \sum_{j=1}^{K} -\bar{p}_j \, log\bar{p}_j \tag{3.17}$$

where $\bar{p} = \mathbb{E}_{x_T \in \mathcal{D}_\mathcal{T}}[\delta(\theta_T(x_T))]$.

Based on information maximisation, confirmation bias problem may occur as instances get wrong cluster prediction and the prediction get reinforced during adaptation. The model suggested to use pseudo-labelling and consider the loss $\mathcal{L}_{pl}$ which is the cross entropy loss with respect to the assigned pseudo-labels.

During the iterations, the cluster centroids for the target samples are calculated from the previous iteration. The cluster centroid is calculated based on a weighted aggregation on each source model and the pseudo-label of each sample is assigned based on its nearest cluster centroid. Finally, under the unsupervised setting, the classification loss function can be constructed based on the prediction outputs and the pseudo-labels.

Logic of weighted Pseudo-labellling

Firstly, each source model calculated their centroids for the whole target dataset. Here I use the first iteration as example.

$$\mu_{k_j}^{(0)} = \frac{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j)\hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j)} \tag{3.18}$$

where $\mu_{k_j}^{(0)}$ is the cluster of class k for source j at i_{th} iteration, $(\hat{\theta}_S^j)$ is the previous source model from the last iteration, and $(\hat{\phi}_S^j)$ is the source-specific centroid. Then, the weighted aggregation of each source centroid is shown as follow.

$$\mu_k^{(i)} = \sum_{j=1}^{n} w_j \mu_{k_j}^{(0)} \tag{3.19}$$

$$\hat{y}_T^{(0)} = \arg\min_k \left\| \hat{\theta}_T(x_T) - \mu_k^{(0)} \right\|_2^2 \tag{3.20}$$

Equation 3.20 denotes the pseudo-label of each sample and is assigned to its nearest cluster during the first iteration. Then during the next iteration, the centroid is updated follow the rule.

$$\mu_{k_j}^{(1)} = \frac{\sum_{x_T \in \mathcal{D}_T} \mathbf{1}\{y_T^{(0)} = k\}\hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \mathbf{1}\{y_T^{(0)} = k\}} \tag{3.21}$$

Then the calculation of cluster centroid and each sample's pseudo-label are based on Equations 3.19 and 3.20. Such iteration will be repeated for multiple times until getting stationary pseudo-labels. Then the $\mathcal{L}_{pl}$ is the classification cross entropy error between the predictions and stationary pseudo-labels.

$$\mathcal{L}_{pl} = -\mathbb{E}_{x_T \in \mathcal{D}_T} \sum_{k=1}^{K} \mathbf{1}\{\hat{y}_T = k\} \, log\left(\delta_k(\, \theta_T\,(x_T))\right) \tag{3.22}$$

Overall Optimisation

Following the previous DECISION, our model is also desired to have target predictions that are confident and globally diverse. The weighted information maximisation strategy is also utilised in the final loss computation. So, the overall optimisation objective function is

$$\mathcal{L}_{total} = \mathcal{L}_{ent} - \mathcal{L}_{div} + \gamma \mathcal{L}_{pl} \tag{3.23}$$

where $\mathcal{L}_{ent}$ refers to Equation 3.16 and $\mathcal{L}_{div}$ refers to Equation 3.17.

The overall optimisation for loss function is subject to the constraint that weight is greater or equal to 0 and all weights sum to 1, which is:

$$\underset{\{\phi_S^j\}_{j=1}^n, \{w_S^j\}_{j=1}^n}{\text{minimize}} \quad \mathcal{L}_{total} \tag{3.24}$$

$$w_j \geq 0 \quad \forall j \in 1, 2, .., n$$

$$\sum_{j=1}^{n} w_j = 1$$

Algorithm 3 Target Adaptation

Input : Number of epoch E, number of batches B, input source models $\{\theta_S^j\}_{i=1}^n$, weighted parameter$\{w_j\}_{j=1}^n$ and unlabeled target $\{x_T^i\}_{i=1}^{N_T}$

Output: Optimal Source model weight $\{\theta_S^{j\star}\}_{i=1}^n$ and optimal source weight$\{w_j^\star\}_{j=1}^n$

Initialize: Freeze last classification layer of all source models and set all weights to 1

for *epoch in 1: E* **do**

Calculate pseudo-labels from Equation 3.20

Calculate the mean embedding p from Equation 3.17

for *batch in 1: B* **do**

Sample target data and pass it through every source models

Calculate the $\mathcal{L}_{ent}$, $\mathcal{L}_{div}$ and $\mathcal{L}_{pl}$ based on Equation 3.16, 3.17 and 3.22

Calculate the total loss based on Equation 3.23

Update the parameters in source models $\{\theta_S^{j\star}\}_{i=1}^n$ and weight $\{w_j^\star\}_{j=1}^n$ based on Equation 3.24

end

end

Chapter 4

Evaluation

4.1 Datasets

Datasets for multi-source domain adaptation always include domains with different styles. Since the paper's main focus is to align models trained from multi-sources with medical images, so two datasets *chest lesions* and *diabetic retinopathy* are constructed by referring to other medical articles.

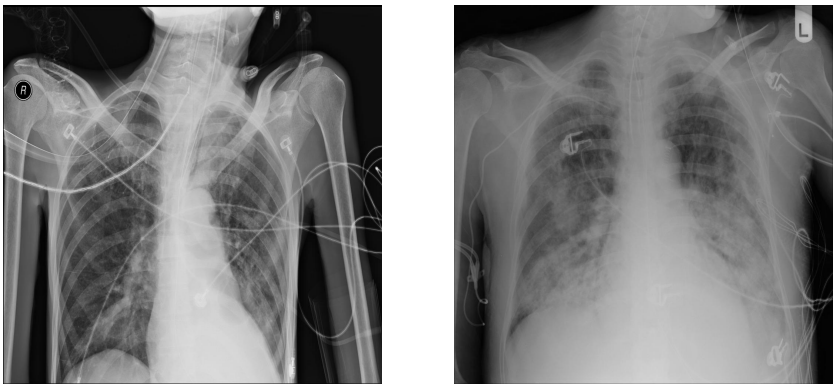
4.1.1 Chest Lesions

Based on paper from Shan et. al [27], the dataset *chest lesions* is constructed with sub-dataset Chest X-Ray (*chest*) [1], Single lesion (*rsna*)[2] and Multiple lesions (*multiple_lesions*)[3]. The labels of the dataset were rearranged into dichotomous labels (2 classes). If the data instance was labeled as at least one lesions, it is relabeled as **Pneumonia**, otherwise the label is **Normal**. Sample images from two labels are shown in Figure 4.1 and the overall information is in Table 4.1.

Dataset	total	Normal	Pneumonia
chest	5856	1583	4273
rsna	26684	20672	6012
multiple_lesions	20013	15504	4509

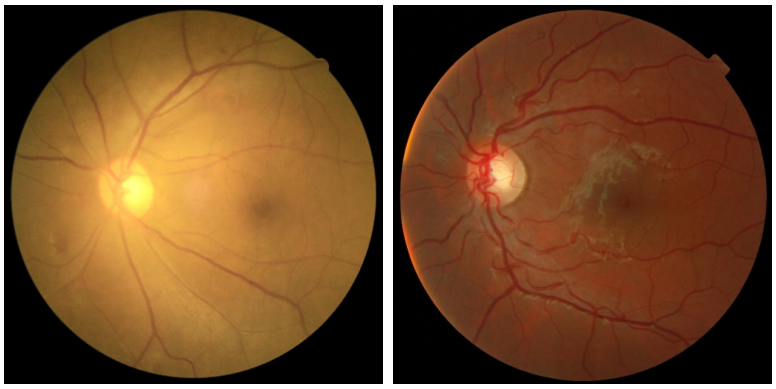
Table 4.1: Label distribution for chest lesion

[1] https://data.mendeley.com/datasets/rscbjbr9sj/2/files/
[2] https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data
[3] https://god.yanxishe.com/18



(a) Normal (b) Pneumonia

Figure 4.1: Samples from Chest Lesions



(a) non-referable (b) referable

Figure 4.2: Samples from Diabetic Retinopathy

4.1.2 Diabetic Retinopathy

The *diabetic retinopathy* dataset refers to the paper from Zhang et. al [38]. The original paper consists of five public available datasets (DDR [16], IDRiD [23], Messidor [8], Messidor-2 [8] and APTOS 2019 [15]). All domains are graded according to the severity of ICDR, from 0 to 4 representing no DR, mild DR, moderate DR, severe DR, and proliferative DR, respectively. Because the original Messidor domain missed some of the labeled information, I manually merged it with Messidor-2 to form a new Messidor. Also based on the original paper, the dataset was relabelled dichotomous class Non-referable and referable. The overall label distribution for each sub-dataset is listed in Table 4.2 and sample instances are shown in Figure 4.2.

Domain	Total	Non-referable		Referable		
		no DR	mild DR	moderate DR	severe DR	proliferative DR
DDR	12522	6266	630	4477	236	913
IDRiD	516	168	25	168	93	62
Messidor	2944	1563	423	594	329	35
APTOS	3662	1805	370	999	193	295

Table 4.2: Label distribution for Diabetic Retinopathy

4.2 Evaluation Metric

Since the overall classification label is binary, so we use the evaluation measures from the confusion matrix.

		Predicted	
		1 (Positive)	**0 (Negative)**
Actual	**1(Positive)**	True Positive (TP)	False Negative (FN)
	0 (Negative)	False Positive (FP)	True Negative (TN)

Accuracy, precision, and recall rate are the main evaluation metrics. Accuracy defines the ratio of correctly predicted observations to the total observations. The precision rate defines as the true positive cases out of all predicted positive cases. Recall rate (or sensitivity) defines the correctly predicted positive cases out of the total positive cases. F1 score

combines the precision and recall score into a single metric by taking their harmonic mean.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} * 100\% \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} * 100\% \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} * 100\% \tag{4.3}$$

$$F1 = 2\frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.3 FDA Domain Adaptation Result

While the original assumption for Fourier Domain Adaptation is to augment source data and put more variation and noise in the training data so that the extracted features and trained model will be more robust and can have better generalization performance during adaptation. However, randomly selected auxiliary datasets may not only cover some key disease determinants of the disease, but may also generate a lot of unnecessary noise that interferes with the learning of the model. For each of the two focused datasets, I list examples that may enhance or impede the training effect and analyze them specifically.

4.3.1 Chest Lesions Synthetic Data

Previous research has shown that the diagnosis of Pneumonia is challenging, as its appearance is vague, can often overlap with other diagnoses and sometimes mimic other benign abnormalities [25]. The identification of these chest lesions usually requires a thorough examination of the lung, such as the capacity, and nodule. Thus, there is a high requirement for the clarity of x-ray images.

From Figure 4.3, the noise added does not affect the clarity of the chest, and most of the chest features of the original image are preserved. So this generated image has a positive effect on model learning.

However, from Figure 4.4, because of the problem of randomly selected images, the synthetic data generated obscures the features of both sides of the lungs, especially the

right side, where most of the lung structures in the original image are blurred and dimmed. This noise generation will negatively interfere with the model's learning of disease features.

(a) Sample 1 (b) cocoval sample image (c) Synthetic 1

Figure 4.3: Synthetic sample 1 (*chest lesions*)

(a) Sample 2 (b) cocoval sample image (c) Synthetic 2

Figure 4.4: Synthetic sample 2 (*chest lesions*)

4.3.2 Diabetic Retinopathy Synthetic Data

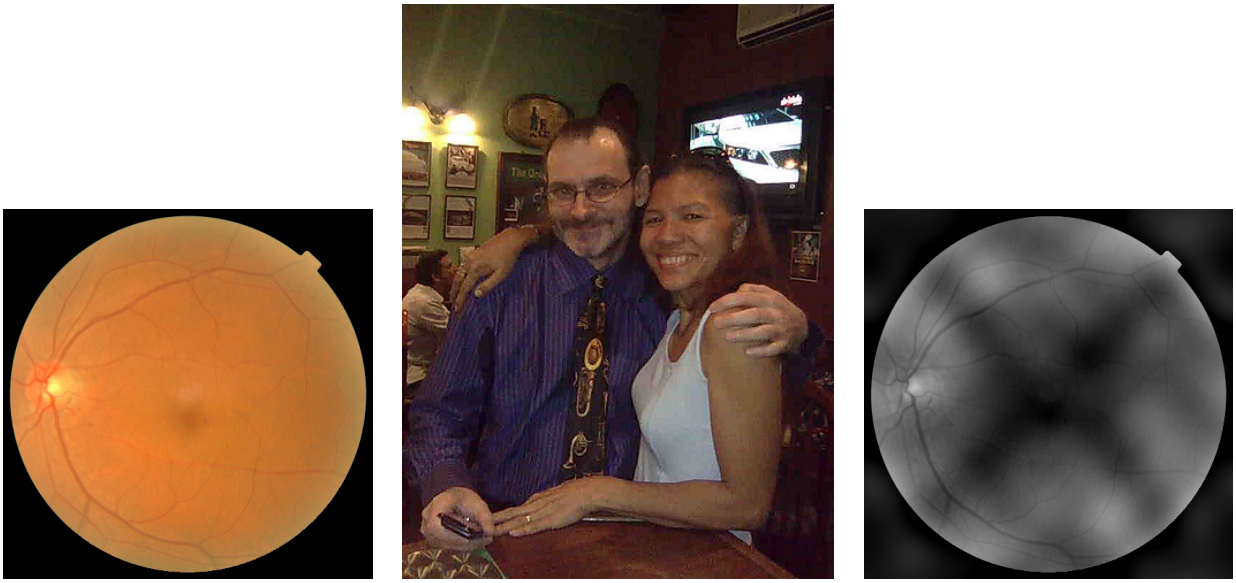
Diabetic Retinopathy is caused by a change in the blood vessel of the retinas. Patients with diabetic retinopathy with have swell blood vessels of retinas and leak fluid [26]. Therefore, in model learning, a clear understanding of the ocular structure and vascular distribution is important for the final determination of diabetes.

From Figure 4.5, although the generated images have a lot of added black noise, the overall vascular structure and distribution are well preserved, and even more contrasted with the surrounding area by conversion to grey-scale images.

However, concerning Figure 4.6, although the blood vessels in the eyes are still preserved, the noise generated is much larger and obscures most of the content of the screening. Such poor synthetic data may blur the focus of the training model to impede the performance of training and adaptation.

(a) Sample 1 (b) cocoval sample image (c) Synthetic 1

Figure 4.5: Synthetic sample 1 (*diabetic retinopathy*)



(a) Sample 2 (b) cocoval sample image (c) Synthetic 2

Figure 4.6: Synthetic sample 2 (*diabetic retinopathy*)

Chapter 5

Results and Discussion

5.1 Baseline

To gain a better understanding of how well my architecture performs, I compare it to the baseline DECISION model concerning the constructed two datasets.

5.1.1 Chest Lesions

For simplicity, I used abbreviations to denote each domain (chest - C, rsna - R, multiple_lesions - M).

Source	Training accuracy	Validation 1	Validation 2
C	98.29%	C-R: 39.89%	C-M: 38.14%
M	81.77%	M-C: 38.29%	M-R: 91.17%
R	82.88%	R-C: 38.61%	R-M: 92.54%

Table 5.1: Training performance on DECISION (*chest lesions*)

Source	Adapting acc (average)	Adapting acc 1	Adapting acc 2
R + M $\rightarrow$ C	75.72%	M-C: 76.01%	R-C: 77.27%
C + R $\rightarrow$ M	70.64%	C-M: 71.22%	R-M: 71.60%
M + C $\rightarrow$ R	67.70%	C-R: 67.53%	M-R: 69.03%
Average	71.35%	—	—

Table 5.2: Adapting performance on DECISION (*chest lesions*)

Source	Distilling Accuracy (overall)	Increase from Adaptation
R + M → C	76.62%	0.9%
C + R → M	70.57%	-0.07%
M + C → R	68.78%	1.08%
Average	71.99%	0.64%

Table 5.3: Distilling performance on DECISION (*chest lesions*)

From Table 5.1, all three source models have training accuracy over 80%. When simply applying the source model to other domains, domains M and R have more information intersection, as the validation accuracy between them exceeds 90%. However, both R and M source models have poor generalization performance on C as the validation accuracy is only around 38%.

After the adaptation process, the accuracy of source models R and M adapting to domain C improved the most, with distilling accuracy reaching 76.62%. While the adaptation accuracy between source models C and R to domain M or source models C and M to domain R both have different degrees of degradation, with distilling accuracy around 70%.

5.1.2 Diabetic Retinopathy

For simplicity, I used abbreviations to denote each domain (APTOS - A, DDR - D, IDRiD - I, Messidor - M).

Source	Training accuracy	Validation 1	Validation 2	Validation 3
A	92.64%	A-D: 68.24%	A-I: 84.11%	A-M: 75.00%
D	87.15%	D-A: 83.40%	D-I: 80.81%	D-M: 70.65%
I	94.23%	I-A: 80.91%	I-D: 70.62%	I-M: 70.92%
M	89.83%	M-A: 85.17%	M-D: 68.29%	M-I: 82.95%

Table 5.4: Training performance on DECISION (*diabetic retinopathy*)

Source	Adapting accuracy (average)	Adapting acc 1	Adapting acc 2	Adapting acc 3
D + I + M → A	89.49%	D-A: 89.60%	I-A 88.97%	M-A: 89.51%
A + I + M → D	76.32%	A-D: 76.70%	M-D: 76.47%	I-D: 75.83%
A + D + M → I	85.85%	A-I: 86.43%	D-I: 88.76%	M-I: 86.43%
A + D + I → M	70.99%	A-M: 70.41%	D-M: 71.47%	I-M: 68.34%
Average	80.66%	—	—	—

Table 5.5: Adapting performance on DECISION (*diabetic retinopathy*)

Source	Distilling Accuracy (overall)	Increase from Adaptation
D + I + M → A	89.49%	0%
A + I + M → D	76.62%	+0.30%
A + D + M → I	86.63%	+0.78%
A + D + I → M	71.23%	+0.24%
Average	80.99%	+0.33%

Table 5.6: Distilling performance on DECISION (*diabetic retinopathy*)

Compared to the previous dataset, domains within the diabetic retinopathy dataset have more information intersection. From Table 5.4, the overall validation accuracy is above 68%. In the training process, all models have relatively high accuracy, indicating that the models fit the original data very well. Each source domain that adapts to target domains A, D, and I has some degree of performance boosting. The only degradation that occurs during the adaptation process is from I-M (from 70.92% to 68.34%) and A-M (from 75% to 70.41%). The possible reason is that the M domain dataset is relatively large, resulting in the underfitting of other source models and failure to capture the information which is specific to domain M. But on average, the distillation accuracy has 0.33% increase from the previous adaptation process and results in an 80.99% accuracy.

5.2 Model Performance

5.2.1 Chest Lesions

Feature Extractor

Source	Feature extractor classification result
C	81.56%
M	80.00%
R	79.80%

Table 5.7: Feature extractor classification result(*chest lesions*)

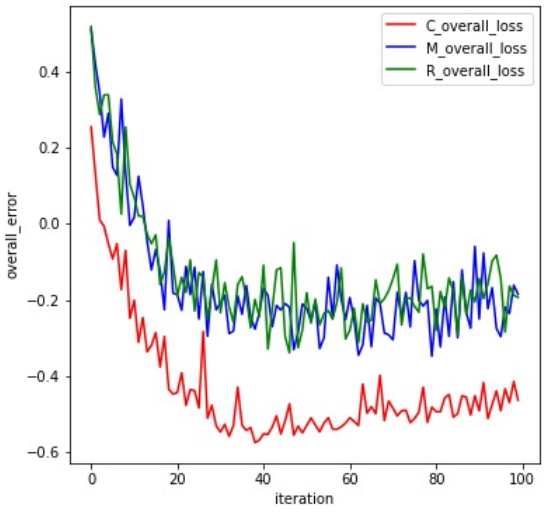


Figure 5.1: Feature extractor overall loss

For all three source datasets of Chest Lesions, the training of the feature extractor network results in an average of 80% accuracy which sounds plausible. When we have a further inspection of the overall loss, all three loss curves decrease in the beginning but fluctuate after around 30 iterations. Some possible reasons are:

- The batch size is small. The original batch size of our network is 64, which also included sets of synthetic data. During the previous discussion about synthetic data

generation, some bad cases containing high noise after FDA generation may also interfere with the training effect of the network. Also, the feature extractor network contains pretrained Resnet50, which is a huge network. So with a small dataset fed into train large network, this may result in loss fluctuation.

- The learning rate is large. For simplicity, I fix the feature extractor network with a learning rate of 1e-3, it turns out that such a learning rate can help the network decrease error quickly in the beginning but may not be the optimal learning rate for all sources. For the C domain, the loss rate finally converges to around -0.5 with less fluctuation. However, for M and R domains, such a learning rate may be large so that the network may just wander around rather than decrease to local minimal points.

Training Source Models

Source	Training Accuracy	Improvement (w.r.t DECISION)	Training val1	Training val 2
C	100.00%	+1.71%	C-R: 31.06%	C-M: 31.24%
M	88.31%	+6.54%	M-C: 31.81%	M-R: 85.82%
R	85.01%	+2.13%	R-C: 30.17%	R-M: 85.55%

Table 5.8: Training performance (ours) (*chest lesions*)

With more data fed into the source model and pre-training on the feature extractor network, the overall training accuracy on the source model has a significant increase compared to the baseline DECISION model (on average increased by 3.46 percent).

Meanwhile, the accuracy of training validation is also greatly reduced. Also, compared with training performance on DECISION (Table 5.1), our trained source model has a poorer validation accuracy. It is reasonable since after pre-training of feature network, the features extracted by each model will be more domain-specific and significant for the domain label classifier, thus causing poorer validation accuracy without the later adaptation step.

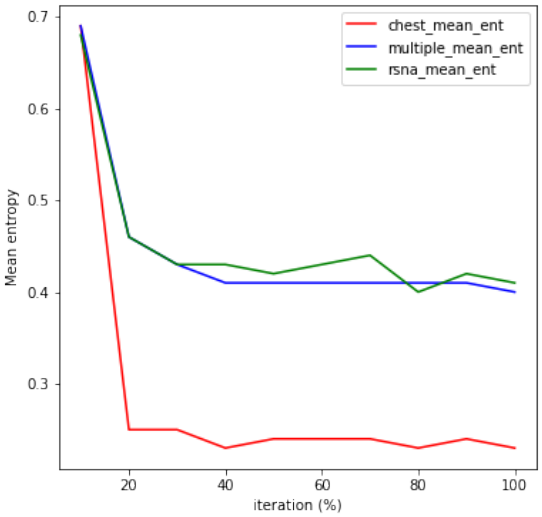


Figure 5.2: Mean entropy (*chest lesions*)

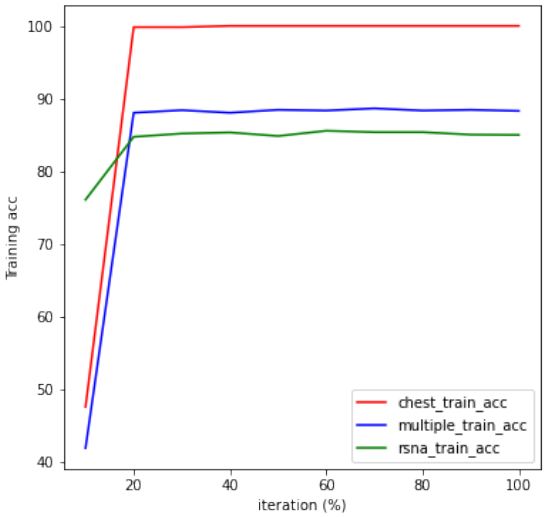


Figure 5.3: Training acc (*chest lesions*)

Based on the mean entropy curve (Figure 5.2) and training accuracy curve (Figure 5.3), chest domain (C) has a lower mean entropy (about 0.2 lower) and higher training accuracy (about 12% higher) than that of multiple_lesion (M) and rsna (R) domains. The mean entropy is the average of label entropy for each instance ($mean\ (H(Y \mid X))$) so a lower mean entropy means the model has more confidence in the prediction, which also corresponds to a higher accuracy.

Furthermore, based on the Figures 5.2 and 5.3, the curve doesn't fluctuate and has a fast convergence after 20% of total iterations.

Adaptation

Source	Adapting Accuracy	Improvement from DECISION Adaptation
R + M → C	83.28%	+7.56%
C + R → M	72.54%	+1.90%
M + C → R	74.64%	+6.94%
Average	71.35%	+5.47%

Table 5.9: Adapting performance (ours) (*chest lesions*)

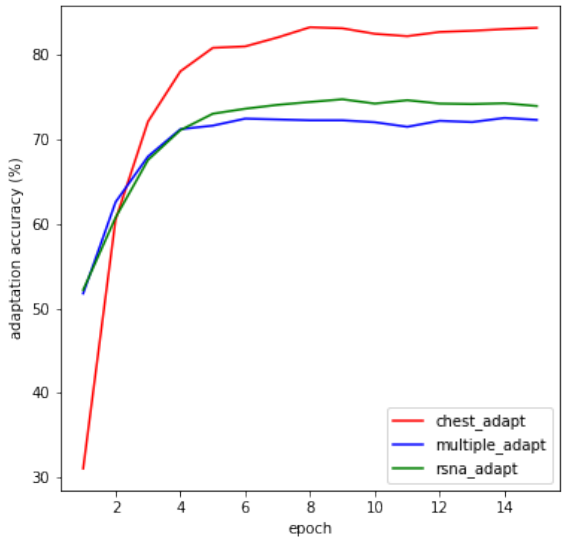


Figure 5.4: Adaptation accuracy

Compared with baseline, our model has a 5.47% adaptation accuracy improvement on average. The improvement while adapting to domain C and R was particularly significant, about 7.56% and 6.94% respectively. Regarding the accuracy curve, the original training validation accuracy scores of source model M and R on domain C are small (with M-C: 31.81% and R-C: 30.17%). However, during the adaptation step, the accuracy curve has a rapid growth from around 30% to around 80% in the first 4 epochs, indicating the good generalization performance of the source models.

Distillation

Source	Distilling Accuracy	Increase from Adaptation	Increase from DECISION
R + M → C	86.34%	+3.06%	+9.72%
C + R → M	72.95%	+0.41%	+2.38%
M + C → R	75.49%	+0.85%	+6.71%
Average	71.99%	+0.64%	+6.27%

Table 5.10: Distil performance comparison (*chest lesions*)

Based on Table 5.10, all source models have an average accuracy of 71.99% during distillation. On average, the models increased by 0.64% during the adaptation step, with domain C increasing the most (3.06%). Compared with the baseline DECISION model, our model shows a significant growth of accuracy in all domains, with an average increase of 6.27%.

Source	Distill Accuracy	Precision	Recall (Sensitivity)	F1 score
R + M → C	86.34%	97.88%	83.08%	89.87%
C + R → M	72.95%	44.21%	76.58%	56.06%
M + C → R	75.49%	46.94%	67.18%	55.27%
Average	71.99%	63.01%	75.61%	68.74%

Table 5.11: Distil performance (ours)(with lr: 1e-2) (*chest lesions*)

When breaking down the results into different metrics, target model C has a better generalization performance, with accuracy, precision, recall and f1 scores all higher than 80%. This indicates that the model can accurately predict positive cases and recall most of them from the actual positive cases. However, regarding target model M and R, even though they have a plausible accuracy rate and recall rate (around 70%), the precision rate is much lower than that of model C, which also indicates that those models have a relatively high false positive rate. In the medical area, a false positive means a patient

may be diagnosed with a disease but actually they don't have, while a false negative means a patient fails to be diagnosed with disease but actually they do have. Therefore, the test importance of false positive and false negative are not equivalent. It is preferable to detect as many positive cases as possible so that more patients can be treated in a timely manner. The overall accuracy, precision and recall curve are shown in Figure 5.5, 5.6 and 5.7 respectively.

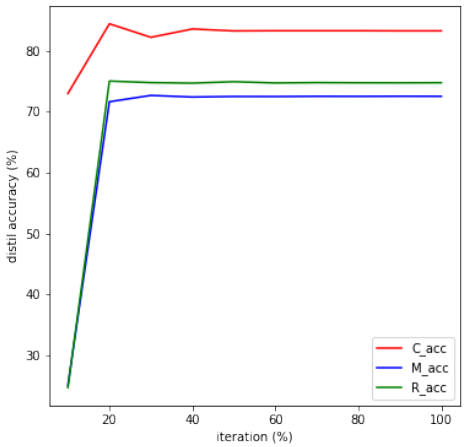


Figure 5.5: Distillation accuracy

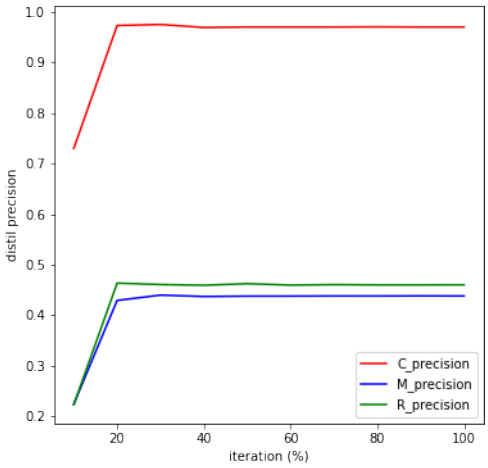


Figure 5.6: Precision

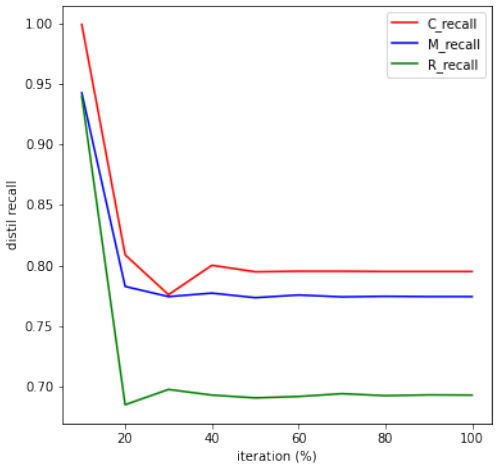


Figure 5.7: Recall

5.2.2 Diabetic Retinopathy

Feature Extractor

Source	Feature extractor classification result
A	81.48%
D	72.43%
I	65.50%
M	70.43%

Table 5.12: Feature extractor classification result(*diabetic retinopathy*)

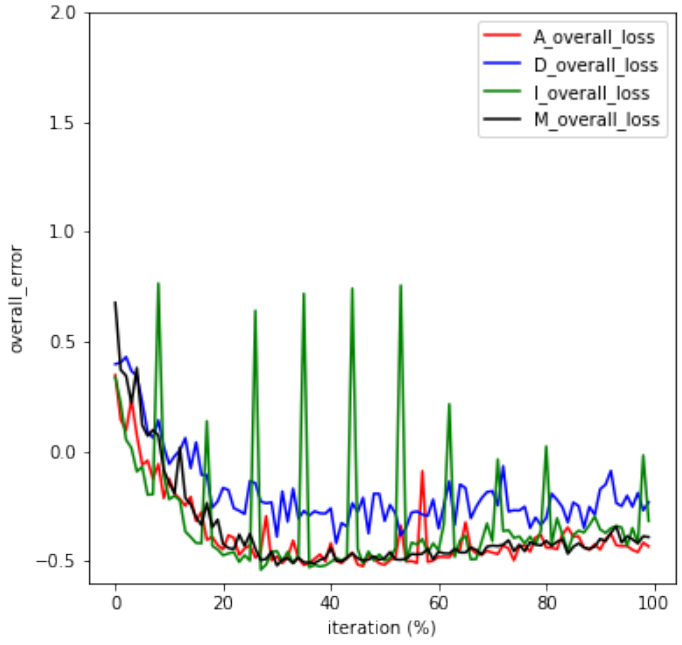


Figure 5.8: Feature Extractor loss

From the feature extractor overall loss curve plot 5.8, domains A and M have relatively stable loss curves. The loss curve for domain D contains small fluctuation, while domain I has periodically big jumps. Except for the reasons introduced previously, the small dataset size for the domain I and dirty synthetic data may be another problem. Compared with

other domains (A - 3662, D - 12522, M - 2944), domain I only contains 512 images. The hyperparameters like batch size, learning rate suited to other domains may be unfit for the training of source model in domain I. Furthermore, during training, the data loader is set to not be shuffled for each epoch. Since synthetic data may contain some dirty data, then the network will incorrectly predict the dirty data every epoch when it processes this batch data, resulting in a sudden increase in the loss and a large oscillation in the period.

Training Source Models

Source	Training Accuracy	Training val1	Training val 2	Training val 3
A	99.73%	A-D: 70.48%	A-I: 85.08%	A-M: 75.31%
D	95.61%	D-A: 79.46%	D-I: 83.14%	D-M: 66.68%
I	100.00%	I-A: 82.85%	I-D: 68.50%	I-M: 72.66%
M	100.00%	M-A: 78.54%	M-D: 69.16%	M-I: 83.91%

Table 5.13: Training performance (ours) (*diabetic retinopathy*)

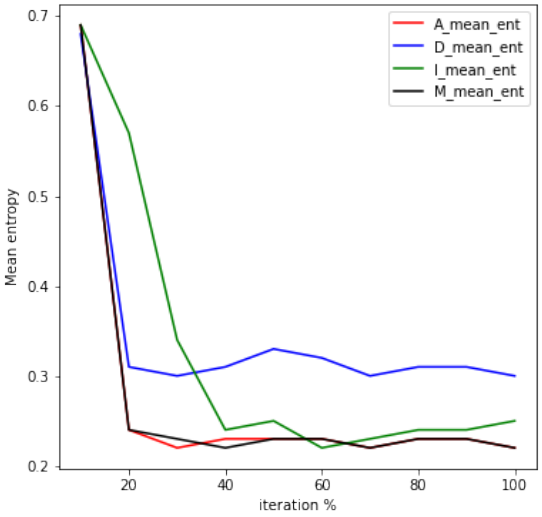


Figure 5.9: Mean entropy (*diabetic retinopathy*)

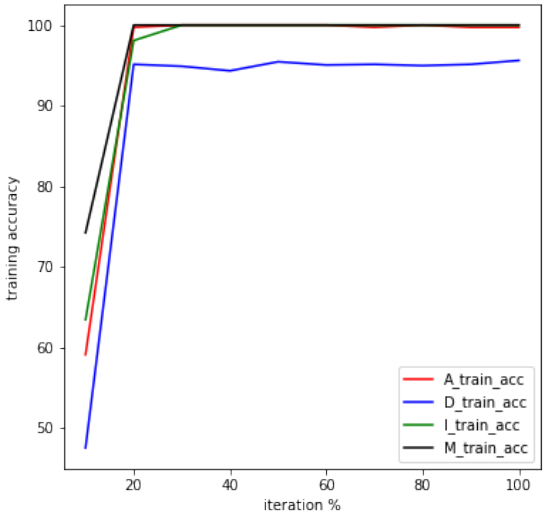


Figure 5.10: Training accuracy (*diabetic retinopathy*)

Compared with the baseline training process 5.4, our model has higher training accuracy on this dataset, while the accuracy of training validation is also similar. This can demonstrate that the information intersection within domains in the diabetic retinopathy dataset is relatively strong. Therefore, after the feature extractor maps the validation target dataset to the source domain-specific latent feature space, those features can also play an important role in the label classification of the other domains.

Regarding the mean entropy plot 5.9 and training accuracy plot 5.10, all domain models have a quick decrease in the mean entropy (quick increase in the training accuracy) and stable convergence after about 20% of the total iterations. Even though some of the training accuracy scores are close to 100%, however, the validation result does not drop a lot, indicating that the models have good fitting on the source domains and have high confidence in the prediction results.

However, especially for small domain I, setting the learning rate as the others may cause the risk of overfitting. While for larger domain like D, such a fixed learning rate may also cause underfitting and make the loss curve stuck at some local minima. So in the future study, we may treat the epoch number as a hyperparameter and fine-tune it for each domain.

Adaptation

Source	Adapting Accuracy	Improvement from DECISION Adaptation
D + I + M → A	90.44%	+0.95%
A + I + M → D	78.52%	+2.20%
A + D + M → I	88.57%	+2.72%
A + D + I → M	79.11%	+8.12%
Average	84.16%	+3.50%

Table 5.14: Adapting performance (ours) (*diabetic retinopathy*)

From Table 5.14, the model has an average accuracy of 84.16% during the adaptation process, which also has an average 3.50% improvement compared with adaptation accuracy of DECISION. This shows that the improved feature extractor model has good generalization for this dataset. Among all sources, target domain M has the most improvement compared to baseline (increase by 8.12%) while target domain A has the best performance but least improvement compared to baseline (increase by 0.95%).

Distillation

Source	Distill Accuracy	Increase from Adaptation	Increase from DECISION
D + I + M → A	91.04%	+0.60%	+1.55%
A + I + M → D	78.76%	+0.24%	+2.14%
A + D + M → I	88.95%	+0.38%	+2.32%
A + D + I → M	79.21%	+0.10%	+7.98%
Average	84.49%	+0.33%	+3.50%

Table 5.15: Distil performance comparison (*diabetic retinopathy*)

From Table 5.15, the fine-tuning of source models' weights and combining them into a single target model has an average accuracy of 84.49%. Even though the distillation step doesn't improve much from the adaptation process (only increase by 0.33%), according to the DECISION baseline, the average boost is around 3.50%.

Source	Distill Accuracy	Precision	Recall (Sensitivity)	F1 score
D + I + M → A	91.04%	96.84%	83.67%	89.78%
A + I + M → D	78.76%	70.97%	79.54%	75.01%
A + D + M → I	88.95%	88.24%	93.75%	90.91%
A + D + I → M	79.21%	60.86%	71.10%	65.58%
Average	84.49%	79.22%	82.01%	80.59%

Table 5.16: Distil performance (ours) (with lr: 1e-2) (*diabetic retinopathy*)

When breaking down the results into different metrics, the target model in domain A and I have better generalization performance than the other domains, with average accuracy and f1 score up to around 90%. However, although domains D and M have plausible accuracy rates, their f1 scores are relatively lower. The main cause for their lower f1 score is due to the precision rate. Especially for domain D, its precision rate of only 60% which reduces the overall f1 score significantly to 65.58%.

Regarding the Figure of accuracy 5.11, precision 5.12 and recall 5.13 for all domains, the overall trends are also reasonable, with all models almost converging after the first few epochs and then stays stable. However, during the distillation process of target domain I, there is a rapid rise in the precision rate and a drop in the accuracy and recall rate in epoch 2. After reviewing the log, during epoch 2, the model predicts more negative cases which contribute to a higher FN rate and lower TP rates and pull down the recall rate.

Epoch	TP	FP	TN	FN
1	279	18	175	44
2	235	4	189	88
3	285	16	177	38

Table 5.17: Domain I distillation log for first 3 epochs (*diabetic retinopathy*)

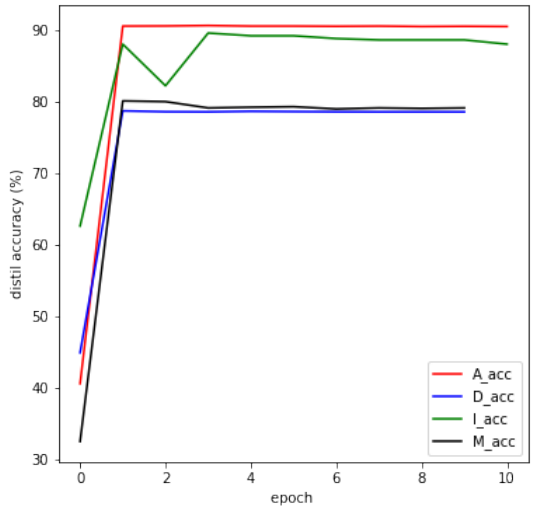


Figure 5.11: Distillation accuracy

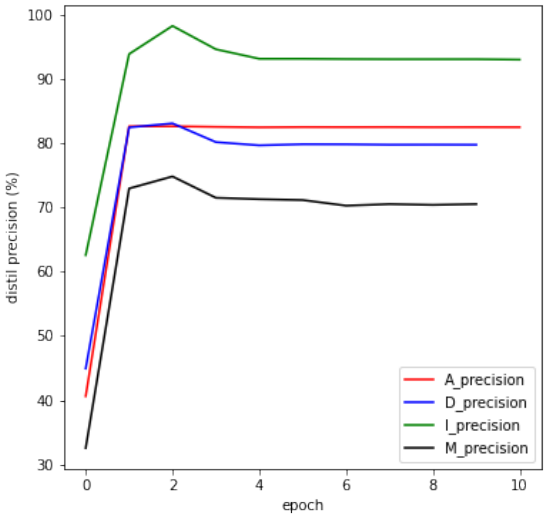


Figure 5.12: Precision

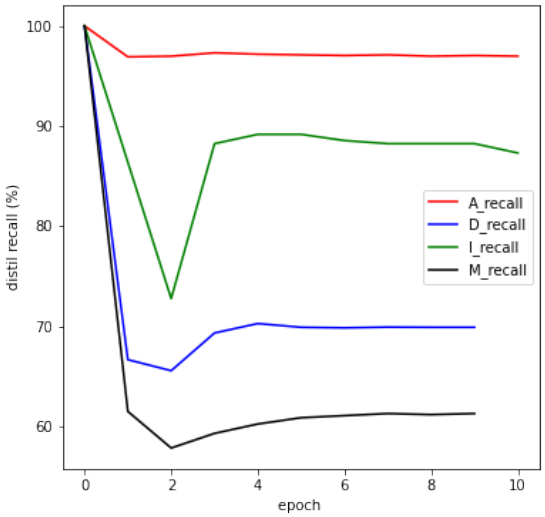


Figure 5.13: Recall

5.3 Ablation study

For ablation study, I also construct the feature extractor without domain classifier and gradient reversal layer. And then we want to test the contribution of the domain classifier ($\mathcal{L}_{domain}$) towards the final accuracy.

Dataset	Domain	Baseline	Model (no $\mathcal{L}_{domain}$)	Model (with $\mathcal{L}_{domain}$)
Chest Lesions	M+R → C	76.62%	73.50%	**86.34%**
	C+R → M	70.57%	71.02%	**72.95%**
	C+M → R	68.78%	67.98%	**75.49%**
	average	71.99%	70.83%	**78.26%**
Diabetic Retinopathy	D+I+M → A	89.49%	89.71%	**91.04%**
	A+I+M → D	76.62%	78.28%	**78.76%**
	A+D+M → I	86.63%	87.79%	**88.95%**
	A+D+I → M	71.23%	71.40%	**79.21%**
	average	81.00%	81.79%	**84.49%**

Table 5.18: Ablation study on L_{domain}

Based on Table 5.19, the overall performance of the model without the classifier is not much different from the baseline, while the addition of a domain reversal layer significantly improves the adaptation accuracy of the model, with medical dataset *chest lesions* increasing by 6.27% and medical dataset *diabetic retinopathy* increasing by 3.49%.

5.4 Hyperparameter Analysis

In this section, two hyper-parameters (learning rate and batch size) will be analysed. The learning rate is tuned during the distillation process with values 1e-2, 1e-3 and 1e-4. The batch size is tuned during the adaptation process with values 32 and 64.

5.4.1 Learning Rate

Dataset	Domain	lr = 1e-2	lr = 1e-3	lr = 1e-4
Chest Lesions	M+R → C	**86.34%**	84.44%	85.43%
	C+R → M	**72.95%**	72.69%	72.80%
	C+M → R	**75.49%**	75.04%	74.87%
Diabetic Retinopathy	D+I+M → A	**91.04%**	90.58%	90.20%
	A+I+M → D	**78.76%**	78.57%	78.65%
	A+D+M → I	88.95%	**89.53%**	88.95%
	A+D+I → M	**79.21%**	79.08%	78.94%

Table 5.19: Distillation learning rates versus final accuracy

The distillation step with learning rate 1e-2 gives the highest accuracy for most of the domains. However, domain I in the diabetic retinopathy dataset gets the highest distillation accuracy when the learning rate equals 1e-3. Even though the changes between different learning rates didn't vary much, the possible reason behind this is that since domain I contain much less data than the other domains, so smaller learning rate (1e-3) is preferable.

5.4.2 Batch Size

Dataset	Domain	batch size = 32	batch size = 64
Chest Lesions	M+R → C	83.28%	**83.38%**
	C+R → M	**72.95%**	72.28%
	C+M → R	**75.49%**	74.45%
Diabetic Retinopathy	D+I+M → A	90.44%	**90.47%**
	A+I+M → D	78.52%	**78.64%**
	A+D+M → I	**88.57%**	87.98%
	A+D+I → M	**79.11%**	79.01%

Table 5.20: Adaptation learning rates versus adaptation accuracy

Based on the current result 5.20, batch 32 and batch 64 don't have much influence towards the adaptation accuracy. For better analysis of the influence of batch size, we can change to 128 or 256 and apply it not only to one adaptation step but through all training processes.

Chapter 6

Conclusion and Future Work

In this thesis, I propose a novel multiple-source disentanglement-based domain adaptation model without access to the source data. Our whole model contains three steps: synthetic data generation, feature extractor and source model training, adaptation and single target model distillation. The model first applies Fourier domain adaptation to generate synthetic source data and perform data augmentation. Then, with real and synthetic data fed into training the feature extractor network, the combination of domain-specific label classifier and domain classifier (with gradient reversal layer) will help the source model to extract features that are significant to label classification but also domain-invariant. Moreover, during adaptation, our model also applies the idea of pseudo-labelling with information maximisation so each of the source models can better adapt to the target domain. Finally, we refer to the idea from the baseline model DECISION and find the optimally weighted combination of source models into a single target model.

6.1 Limitation

As we explained in the previous analysis, fluctuation exists for some specific domains' loss curves during the iterations. The possible reason is that we fix the hyperparameters to be the same for all domains. However, due to the size difference, hyperparameters that are suitable to one domain may cause another small-size domain to overfit. It is necessary to fine-tune those hyperparameters for each domain.

Also, the current Fourier-based image synthesis method may generate some noise and cover the critical pattern for medical images. This noisy data can negatively affect the model fit and generalization performance so we need to revise those synthetic image generation methods and add constraints.

6.2 Future Work

Some improvements are considered in future work:

1. Firstly, our current method only applies the Fourier domain adaptation method for synthetic data generation. More image synthesis methods should be implemented and compared, such as generative adversarial-related methods and other unsupervised methods.

2. Likewise, because the screening of medical images relies more on the local characteristics of the organs, such as the thickness of blood vessels, transparency of the lungs, etc. Therefore, we need to be more restrictive in generating images to avoid the mixing of noisy images to affect the training effect.

3. The disentanglement-based structure should not only be employed during model training but also be considered during target domain adaptation. With high-quality target synthetic data incorporated into the adaptation step, we can then fine-tune the source model based on those two types of target domain data and map the whole feature space from more source domain-related to more target domain-related.

4. Finally, for the distillation step, instead of finding a linear combination of all source models, we can also apply some model ensemble techniques, like boosting or bagging.

Bibliography

[1] Adler-Milstein, J., Jha, A.K., 2012. Sharing clinical data electronically: a critical challenge for fixing the health care system. JAMA 307, 1695–1696.

[2] Ahmed, S.M., Raychaudhuri, D.S., Paul, S., Oymak, S., Roy-Chowdhury, A.K., 2021. Unsupervised multi-source domain adaptation without access to source data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10103–10112.

[3] ARTLab, . Transfer learning as a tool for efficient machine learning. URL: https://engineering.purdue.edu/artlab/?page_id=601.

[4] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D., 2016. Domain separation networks. Advances in neural information processing systems 29.

[5] Caron, M., Bojanowski, P., Joulin, A., Douze, M., 2018. Deep clustering for unsupervised learning of visual features, in: Proceedings of the European conference on computer vision (ECCV), pp. 132–149.

[6] Chapelle, O., Scholkopf, B., Zien, A., 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. IEEE Transactions on Neural Networks 20, 542–542.

[7] Choudhary, A., Tong, L., Zhu, Y., Wang, M.D., 2020. Advancing medical imaging informatics by deep learning-based domain adaptation. Yearbook of medical informatics 29, 129–138.

[8] Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., et al., 2014. Feedback on a publicly distributed image database: the messidor database. Image Analysis & Stereology 33, 231–234.

[9] Ganin, Y., Lempitsky, V., 2015. Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR. pp. 1180–1189.

[10] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial networks. Communications of the ACM 63, 139–144.

[11] Guan, H., Liu, M., 2021. Domain adaptation for medical image analysis: a survey. IEEE Transactions on Biomedical Engineering 69, 1173–1185.

[12] Guo, J., Shah, D.J., Barzilay, R., 2018. Multi-source domain adaptation with mixture of experts, in: EMNLP.

[13] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[14] Karani, N., Chaitanya, K., Baumgartner, C., Konukoglu, E., 2018. A lifelong learning approach to brain mr segmentation across scanners and protocols, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 476–484.

[15] Khalifa, N.E.M., Loey, M., Taha, M.H.N., Mohamed, H.N.E.T., 2019. Deep transfer learning models for medical diabetic retinopathy detection. Acta Informatica Medica 27, 327.

[16] Li, T., Gao, Y., Wang, K., Guo, S., Liu, H., Kang, H., 2019. Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening. Information Sciences 501, 511 – 522. URL: http://www.sciencedirect.com/science/article/pii/S0020025519305377, doi:https://doi.org/10.1016/j.ins.2019.06.011.

[17] Liang, J., Hu, D., Feng, J., 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in: International Conference on Machine Learning, PMLR. pp. 6028–6039.

[18] Liu, Y., Jun, E., Li, Q., Heer, J., 2019. Latent space cartography: Visual analysis of vector space embeddings, in: Computer Graphics Forum, Wiley Online Library. pp. 67–78.

[19] Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.

[20] Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B., 2019. Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1406–1415.

[21] Perrot, M., Habrard, A., 2015. A theoretical analysis of metric hypothesis transfer learning, in: International Conference on Machine Learning, PMLR. pp. 1708–1717.

[22] Pooch, E.H., Ballester, P.L., Barros, R.C., 2019. Can we trust deep learning models diagnosis? the impact of domain shift in chest radiograph classification. arXiv preprint arXiv:1909.01940 .

[23] Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabuddhe, V., Meriaudeau, F., 2018. Indian diabetic retinopathy image dataset (idrid): a database for diabetic retinopathy screening research. Data 3, 25.

[24] Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D., 2008. Dataset shift in machine learning. Mit Press.

[25] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al., 2017. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 .

[26] Sachdeva, M.M., 2022. Diabetic retinopathy. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/diabetes/diabetic-retinopathy.

[27] Shan, X., Wen, Y., Li, Q., Lu, Y., Cai, H., 2021. A coherent cooperative learning framework based on transfer learning for unsupervised cross-domain classification, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 96–106.

[28] Sun, B., Saenko, K., 2016. Deep coral: Correlation alignment for deep domain adaptation, in: European conference on computer vision, Springer. pp. 443–450.

[29] Tsirikoglou, A., Eilertsen, G., Unger, J., 2020. A survey of image synthesis methods for visual machine learning, in: Computer Graphics Forum, Wiley Online Library. pp. 426–451.

[30] Wachinger, C., Reuter, M., Initiative, A.D.N., et al., 2016. Domain adaptation for alzheimer's disease diagnostics. Neuroimage 139, 470–479.

[31] Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. Neurocomputing 312, 135–153.

[32] Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. Journal of Big data 3, 1–40.

[33] Wilson, G., Cook, D.J., 2020. A survey of unsupervised deep domain adaptation. ACM Transactions on Intelligent Systems and Technology (TIST) 11, 1–46.

[34] Wu, X., Xu, K., Hall, P., 2017. A survey of image synthesis and editing with generative adversarial networks. Tsinghua Science and Technology 22, 660–674.

[35] Xu, Q., Zhang, R., Zhang, Y., Wang, Y., Tian, Q., 2021. A fourier-based framework for domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14383–14392.

[36] Yang, S., Wang, Y., van de Weijer, J., Herranz, L., Jui, S., 2021. Generalized source-free domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8978–8987.

[37] Yang, Y., Soatto, S., 2020. Fda: Fourier domain adaptation for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4085–4095.

[38] Zhang, G., Sun, B., Zhang, Z., Pan, J., Yang, W., Liu, Y., 2022. Multi-model domain adaptation for diabetic retinopathy classification. Frontiers in Physiology , 1071.

[39] Zhang, J., Liu, M., Pan, Y., Shen, D., 2019. Unsupervised conditional consensus adversarial network for brain disease identification with structural mri, in: International Workshop on Machine Learning in Medical Imaging, Springer. pp. 391–399.

[40] Zhao, S., Li, B., Xu, P., Keutzer, K., 2020. Multi-source domain adaptation in the deep learning era: A systematic survey. arXiv preprint arXiv:2002.12169 .

[41] Zhu, X., Thung, K.H., Adeli, E., Zhang, Y., Shen, D., 2017. Maximum mean discrepancy based multiple kernel learning for incomplete multimodality neuroimaging data, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 72–80.

[42] Zhu, Y., Zhuang, F., Wang, D., 2019. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5989–5996.