Neural Computing and Applications (2023) 35:22189–22207
https://doi.org/10.1007/s00521-023-08939-6

S.I.: DEEP LEARNING IN MULTIMODAL MEDICAL IMAGING FOR CANCER DETECTION

TSP-UDANet: two-stage progressive unsupervised domain adaptation network for automated cross-modality cardiac segmentation

Yonghui Wang[1] · Yifan Zhang[1] · Lisheng Xu[1,2,3] · Shouliang Qi[1,2,3] · Yudong Yao[1,4] · Wei Qian[1,5] · Stephen E. Greenwald[6] · Lin Qi[1,2,3]

Received: 6 July 2022 / Accepted: 1 August 2023 / Published online: 11 September 2023

Abstract

Accurate segmentation of cardiac anatomy is a prerequisite for the diagnosis of cardiovascular disease. However, due to differences in imaging modalities and imaging devices, known as domain shift, the segmentation performance of deep learning models lacks reliability. In this paper, we propose a two-stage progressive unsupervised domain adaptation network (TSP-UDANet) to reduce domain shift when segmenting cardiac images from various sources. We alleviate the domain shift between the feature distribution of the source and target domains by introducing an intermediate domain as a bridge. The TSP-UDANet consists of three sub-networks: a style transfer sub-network, a segmentation sub-network, and a self-training sub-network. We conduct cooperative alignment of different domains at image level, feature level, and output level. Specifically, we transform the appearance of images across domains and enhance domain invariance by adversarial learning in multiple aspects to achieve unsupervised segmentation of the target modality. We validate the TSP-UDANet on the MMWHS (unpaired MRI and CT images), MS-CMRSeg (cross-modality MRI images), and M&Ms (cross-vendor MRI images) datasets. The experimental results demonstrate excellent segmentation performance and generalizability for unlabeled target modality images.

Keywords Cross-modality learning · Unsupervised domain adaptation · Cardiac segmentation · Intermediate domain

✉ Lin Qi
qilin@bmie.neu.edu.cn

Yonghui Wang
1971081@stu.neu.edu.cn

Yifan Zhang
2001234@stu.neu.edu.cn

Lisheng Xu
xuls@bmie.neu.edu.cn

Shouliang Qi
qisl@bmie.neu.edu.cn

Yudong Yao
yyao@bmie.neu.edu.cn

Wei Qian
wqian@bmie.neu.edu.cn

Stephen E. Greenwald
s.e.greenwald@qmul.ac.uk

[1] College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110169, China

[2] Engineering Research Center of Medical Imaging and Intelligent Analysis, Ministry of Education, Shenyang 110169, China

[3] Key Laboratory of Medical Image Computing, Ministry of Education, Northeastern University, Shenyang 110169, China

[4] Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA

[5] Department of Electric and Computer Engineering, College of Engineering, University of Texas, El Paso, USA

[6] Blizard Institute, Barts and The London School of Medicine and Dentistry, Queen Mary University of London, London E1 2AD, UK

1 Introduction

Cardiovascular disease (CVD) is a major and ever-increasing global problem. According to the World Health Organization, CVD causes about 17.9 million deaths worldwide every year [1]. Morbidity due to CVD is an equally severe challenge, with the total number of disability life years due to ischemic heart disease and stroke having reached 182 million (95% UI: 170 to 194 million) by 2019 [2]. Therefore, the prevention and diagnosis of CVD are essential to reduce social and economic burdens. In diagnosing CVD, medical image segmentation can reveal cardiac substructures, which is the premise of quantifying human cardiac anatomy and locating lesions [3]. Thus, medical image segmentation occupies an essential position in clinical practice. In recent years, significant progress has been made due to the development of deep convolutional neural networks. Traditionally and ideally, the training and testing images for a deep learning network contain the same pixel intensity distribution. A large number of accurately annotated training images ensures that the model learns sufficiently to achieve promising segmentation results for diagnostic purposes when using images from the same modality. However, in actual clinical practice, the testing images are often of different modalities, or of the same modality but from different vendors, thus giving rise to large differences between the intensity distributions of the training and testing images. Unfortunately, these differences (known as domain shift) can often lead to a significant degradation in the model performance.

In clinical practice, magnetic resonance imaging (MRI) and computed tomography (CT) images are often used to diagnose CVD. Common cardiac magnetic resonance (CMR) imaging modalities include late gadolinium enhancement (LGE), balanced steady-state free precession (bSSFP), T1 and T2 images. LGE images are commonly used for diagnosing myocardial disease, while bSSFP images show clear borders between the myocardium and the ventricles. T1 images are used to show anatomical structures, while T2 images are used to display pathological information [4]. Figure 1 demonstrates the considerable discrepancy in intensity distribution and appearance between MRI and CT cardiac images.

However, since the annotation of medical images is extremely tedious, time-consuming, and costly, it is difficult to build a multimodal medical image segmentation dataset with many pixel-level labels. To reduce the annotation burden and prevent degradation of model performance, new studies on the unsupervised domain adaptation (UDA) segmentation method are emerging [5]. The UDA method uses the richly labeled images of one modality (the source domain) to train deep convolutional neural networks for segmenting poorly labeled images of another modality (the target domain).

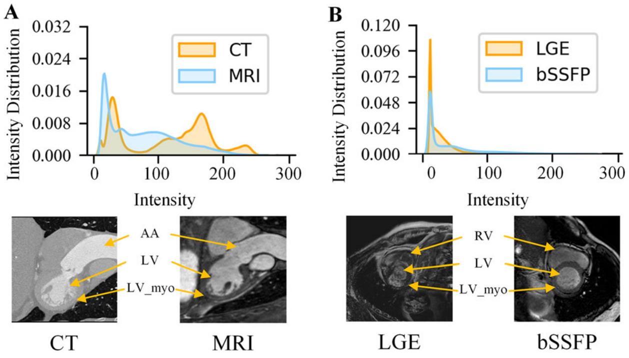

Fig. 1 Illustration of the domain shift existing between various image types. **A** Comparison of MRI and CT coronal plane cardiac images, and their pixel intensity distribution. The main cardiac substructures include the ascending aorta (AA), left ventricle (LV), and left ventricular myocardium (LV_myo). **B** Comparison of short-axis LGE and bSSFP CMR images, and their pixel intensity distribution. The main cardiac substructures include the right ventricle (RV), LV, and LV_myo

The critical issue in the UDA method for cross-modality cardiac image segmentation is to extract useful features from the source and target domains and reduce their intensity distribution discrepancy, whereas the domain-invariant features extracted by the deep learning network are implicit. To overcome the above limitation, we propose a UDA framework for cross-modality cardiac image segmentation with cooperative alignment from multiple levels. Individual alignment of feature distribution at image level, feature level, or output level is likely to lead to the loss of semantic information from the images, between the source and target domains, ultimately affecting the segmentation performance of the target domain images. The cooperative training process ensures that the network extracts more useful semantic information from multi-level feature spaces. As well as considering the significant variations between the source and target domains, we introduce an intermediate domain to gradually manage the domain shift. Thus, we employ the style transfer sub-network (CycleGAN) to effectively capture the pixel-level information of the source domain and target domain to generate fake target domain images which retain the original contents with their structural semantics unaffected. The fake target domain images act as the intermediate domain, and then we can design the following domain adaptation-based segmentation process, which contains two sub-networks to transfer the source domain label information to the target domain. The first is the segmentation sub-network (SSN) to transfer the source domain label information to the intermediate domain and generate the corresponding pseudo-labels. The second is the self-training sub-network (StSN) to transfer intermediate domain pseudo-label information to the target domain using a self-training strategy.

In summary, the main contributions of this paper are as follows:

1. We propose a two-stage progressive UDA network (TSP-UDANet) for cross-modality cardiac image segmentation based on generative adversarial learning. The TSP-UDANet includes a style transfer sub-network, a segmentation sub-network, and a self-training sub-network, that aligns the source and target domains at image level, feature level, and output level, respectively.
2. We introduce an intermediate domain as a bridge between the source and target domains. The intermediate domain is trained in an adversarial manner in the segmentation sub-network and the self-training sub-network with the source and target domains, respectively, to progressively reduce the discrepancy in feature distribution between the source and target domains.
3. In the self-training sub-network (StSN), we introduce a self-training strategy into the self-training sub-network. This strategy combines the labeled and unlabeled data to expand the total amount of data available for network training, hence improving the performance of the UDA segmentation.

To validate the generalization performance of the TSP-UDANet, we have conducted extensive experiments on three different cross-modality multi-objective medical image segmentation tasks, including the MMWHS, MS-CMRSeg, and M&Ms datasets. The results of the experiments on the three datasets demonstrate the effectiveness of the TSP-UDANet and its further application potential in various tasks, e.g., the detection and segmentation of tumors in medical images.

The remainder of this paper is organized as follows: Section 2 presents related works from the literature. Section 3 gives the details of the TSP-UDANet, including a method overview, the style transfer sub-network, segmentation sub-network, self-training sub-network, network configurations, and implementation details. Section 4 describes the design of the experiments. Section 5 presents the experimental results. Section 6 introduces the ablation analysis and Section 7 discusses the performance of the TSP-UDANet. Finally, our conclusions and suggestions for future work are offered in Section 8.

2 Related work

When analyzing images from one modality, exploiting labeled images from another is challenging due to the significant domain shift caused by the obvious differences in image properties between them. Specifically, in an unsupervised cross-modality cardiac segmentation task, the main idea is to extract domain-invariant features from the source and target domains, and to transfer label information from the source domain to the target domain. To better transfer knowledge learned from a source domain with rich labels to a target domain without labels, the UDA segmentation method has attracted recent attention [6, 7]. This approach uses images from the source domain with labels to train a model and applies it to the segmentation of target domain images, which are generally of another modality or images of the same modality but derived from machines made by different vendors. Current cross-modality UDA segmentation methods typically include two strategies. The first is to train the segmentation network with the labeled source domain images and then to use some of the target domain images for fine-tuning [8] or directly, for segmenting the target domain images. The second is to minimize the discrepancies in feature distribution between the source and target domains and to align latent features from the image level, the feature level, and the output level.

2.1 Image-level alignment

The goal of image-level alignment is to minimize the differences in the distribution of pixel intensities between the source and target domains. This ensures that the knowledge gained from the source domain can be effectively transferred to the target domain, thus improving the segmentation performance of the target domain. Image-level alignment is usually achieved in one of two ways. One is to extract domain-invariant features at the input level of the segmentation network [9, 10]. Here, the source domain and target domain images can share the feature extraction part of the segmentation network to learn image-level features, such as grayscale distribution and texture information. The other is the style transfer method for cross-modality images. In this case, the mapping relationship between the source domain and target domain is learned, and the generated cross-modality images are sent to the segmentation network for segmentation [11, 12]. Traditional cross-domain image segmentation methods require a large number of paired training images. However, such paired images are usually difficult or even impossible to obtain. Zhu et al. [13] proposed a cycle generative adversarial network (CycleGAN) to generate fake target domain and fake source domain images without the need for paired training images. Following the success of CycleGAN in the image-to-image translation task, researchers have converted the appearance of images from different modalities by translating the image style from the source (target) domain to the target (source) domain. Jiang et al. [11] proposed an unsupervised cross-modality domain adaptation network for lung cancer region segmentation by transforming the CT image style into that of MRI images. Chen et al. [12] proposed a semantics-aware generative adversarial network (SeUDA) to align the image-level features of different X-ray datasets for left/right lung segmentation.

2.2 Feature-level alignment

Feature-level alignment aims to adjust the features of the source and target domain data so that they have a consistent representation in feature space. Feature-level alignment is primarily used to reduce domain shift in higher dimensional feature spaces by minimizing the distribution discrepancy of feature maps extracted from the source and target domains. This process includes minimizing the maximum mean discrepancy (MMD) [14, 15], the loss of Wasserstein generative adversarial network (WGAN) [16], and the distribution distance in unique feature spaces [17]. Some studies have introduced the GAN into feature-level alignment, where adversarial training of generators and discriminators makes the generators focus on the common features of the target and source domains [18–21]. Kamnitsas et al. [22] proposed learning domain-invariant features for brain lesion segmentation with an adversarial network, and designed a multi-connected domain discriminator that predicts the input image domain. Jain et al. [23] employed an adversarial learning scheme to adapt knowledge from PV phase images to ART phase images for detecting liver tumors.

2.3 Output-level alignment

Output-level alignment is primarily used to extract the domain-invariant features in semantic prediction space. The output-level alignment can make the segmentation results of the source and target domains semantically consistent, so as to improve the segmentation performance in the target domain. For output-level alignment, most methods are based on the GAN [24–26], where the output-level features obtained from the generator are fed to the discriminator for generative adversarial training. Panfilov et al. [24] proposed a two-stage network for unsupervised domain adaptation by generative adversarial learning in multi-level feature spaces. Their methods achieved unsupervised segmentation of MRI images from different scanners. Yang et al. [25] proposed a self-attentive GAN that forces the feature maps generated by the generator between the source and target domains to be indistinguishable at the output level.

When facing the challenges of severe domain shift in cross-modality medical image segmentation, the approaches which use image-level, feature-level, or output-level alignment alone are often not sufficient. Thus, multi-level alignment methods should be beneficial for extracting domain-invariant features.

3 Methods

3.1 Overview

This work aims to train a UDA segmentation network for segmenting target domain images where pixel-level annotations are unavailable. We achieve UDA segmentation of unlabeled target domain images by introducing an adversarial training strategy to the segmentation network. Due to different imaging techniques or imaging parameters, there are significant style discrepancies in the appearance of the source and target domain images, thus we introduce an intermediate domain as a bridge to transfer the label information from the source domain to the target domain. Figure 2 shows the framework of the TSP-UDANet, which consists of three sub-networks: the style transfer sub-network (CycleGAN), the segmentation sub-network (SSN), and the self-training sub-network (StSN). Table 1 summarizes the symbols used in the following sections.

3.2 Style transfer sub-network

To reduce the visual difference and the effect of domain shift between the source and target domains, we use image-level alignment to transform the style of the source domain images to the style of the target domain images. The style transfer sub-network for image-level alignment in our framework borrows ideas from the CycleGAN [13], which consists of a source domain generator ($G_{\mathrm{t}\rightarrow\mathrm{s}}$), target domain generator ($G_{\mathrm{s}\rightarrow\mathrm{t}}$), source domain discriminator (D_0), and target domain discriminator (D_1). The generators are used for image reconstruction and generating fake images. The target domain generator ($G_{\mathrm{s}\rightarrow\mathrm{t}}$) is used to transfer the source domain (X^{s}) style to that of the target domain (X^{t}), while the source domain generator ($G_{\mathrm{t}\rightarrow\mathrm{s}}$) is used to transfer the target domain (X^{t}) style to that of the source domain (X^{s}).

During the training of the CycleGAN, the source domain images (x^{s}) are fed into $G_{\mathrm{s}\rightarrow\mathrm{t}}$ to generate the fake target domain images ($x^{\mathrm{s}\rightarrow\mathrm{t}}$), then these fake target domain images ($x^{\mathrm{s}\rightarrow\mathrm{t}}$) are put into $G_{\mathrm{t}\rightarrow\mathrm{s}}$ to generate the reconstructed source domain images ($x^{\mathrm{s}\rightarrow\mathrm{t}\rightarrow\mathrm{s}} = G_{\mathrm{t}\rightarrow\mathrm{s}}(G_{\mathrm{s}\rightarrow\mathrm{t}}(x^{\mathrm{s}}))$.

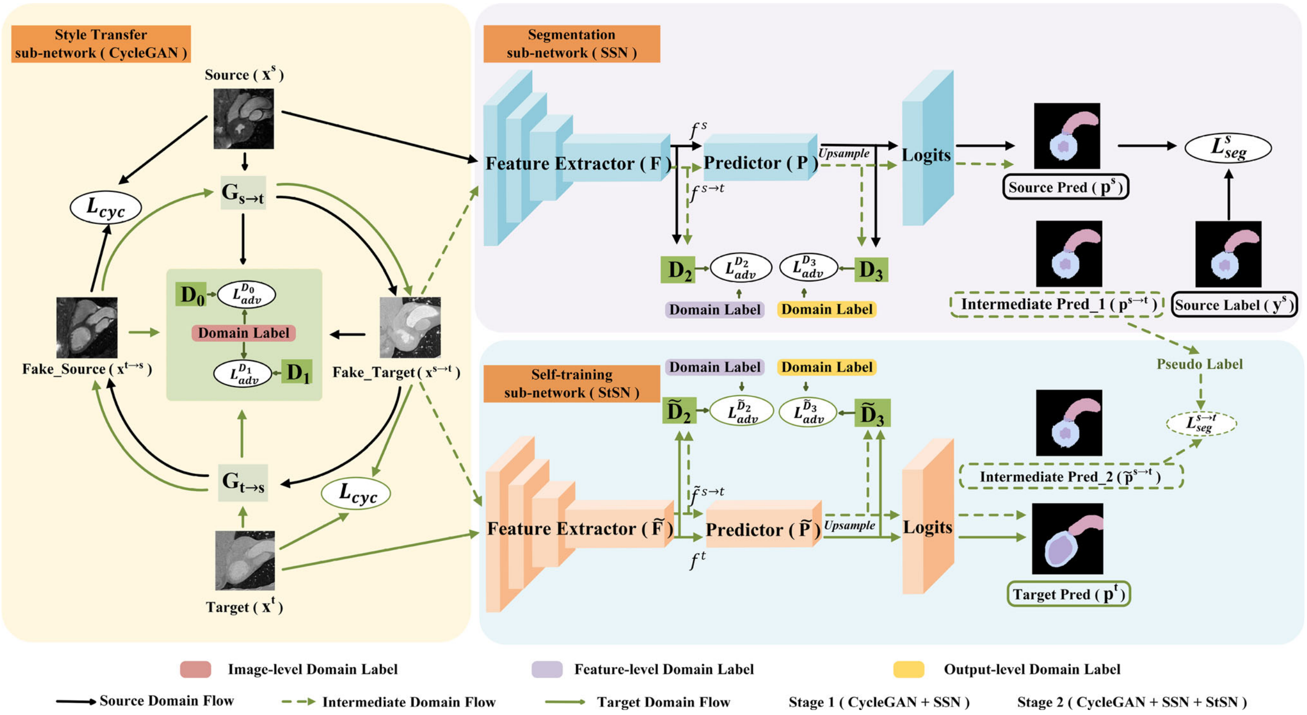

Fig. 2 The framework of the TSP-UDANet for cross-modality cardiac segmentation, including the style transfer sub-network (CycleGAN), the segmentation sub-network (SSN), and the self-training sub-network (StSN). D_0 and D_1 are used for image-level adversarial training, D_2 and $\widetilde{D}_2$ are used for feature-level adversarial training, D_3 and $\widetilde{D}_3$ are used for output-level adversarial training. Upsample scales the predicted images to the raw image size using bilinear interpolation

Table 1 Summary of symbols

Symbol	Notation
X^s, $X^{s\to t}$,X^t	Image sets of source, intermediate and target domains
x^s,x^t	Image samples of source and target domains
$x^{s\to t}$, $x^{t\to s}$	Image samples of fake target and fake source domains
$x^{s\to t\to s}$, $x^{t\to s\to t}$	Cycle reconstructed image samples
Y^s, y^s	Source domain annotation sets and label samples
Y^t, y^t	Target domain annotation sets and label samples
f^s, $f^{s\to t}$	Feature maps of the source and intermediate domains in SSN
$\widetilde{f}^{s\to t}$, f^t	Feature maps of the intermediate and target domains in StSN
p^s, $p^{s\to t}$	Prediction of the source and intermediate domains in SSN
$\widetilde{p}^{s\to t}$, p^t	Prediction of the intermediate and target domains in StSN
$G_{s\to t}$, $G_{t\to s}$	Generators of style transfer sub-network
G_{seg}, $\widetilde{G}_{seg}$	Generator, G_{seg} used in SSN, $\widetilde{G}_{\text{seg}}$ used in StSN
D_i, $\widetilde{D}_i$	Discriminator, including image level, feature level, and output level. $i \in \{0, 1\}$ in the style transfer sub-network, while $i \in \{2, 3\}$ in the segmentation sub-network and self-training sub-network
F, $\widetilde{F}$	Feature extractor, F used in SSN, $\widetilde{F}$ used in StSN
P, $\widetilde{P}$	Predictor, P used in SSN, $\widetilde{P}$ used in StSN

$\widetilde{*}$ Represents the module and feature maps used in StSN

Similarly, the target domain images (x^{t}) pass through the generators $G_{\mathrm{t}\to\mathrm{s}}$ and $G_{\mathrm{s}\to\mathrm{t}}$ in turn, to generate the reconstructed target domain images ($x^{\mathrm{t}\to\mathrm{s}\to\mathrm{t}} = G_{\mathrm{s}\to\mathrm{t}}(G_{\mathrm{t}\to\mathrm{s}}(x^{\mathrm{t}}))$). The source and target domain images share source and target domain generators. In the CycleGAN, the cyclic structure enables bidirectional style transfer between the source and target domain images. The loss function used for the cycle reconstruction is:

$$\begin{aligned} L_{\mathrm{cyc}}(G_{\mathrm{s}\to\mathrm{t}}, G_{\mathrm{t}\to\mathrm{s}}) = {} & E_{x^{\mathrm{t}} \sim X^{\mathrm{t}}}\left[\left|G_{\mathrm{s}\to\mathrm{t}}(G_{\mathrm{t}\to\mathrm{s}}(x^{\mathrm{t}})) - x^{\mathrm{t}}\right|\right] \\ & + E_{x^{\mathrm{s}} \sim X^{\mathrm{s}}}\left[\left|G_{\mathrm{t}\to\mathrm{s}}(G_{\mathrm{s}\to\mathrm{t}}(x^{\mathrm{s}})) - x^{\mathrm{s}}\right|\right] \end{aligned} \tag{1}$$

where the cycle consistency loss L_{cyc} ensures that the reconstructed images preserve the contents of the real images.

In contrast to the GAN, the CycleGAN performs bidirectional generation for the source and target domains. The target domain generator ($G_{\mathrm{s}\to\mathrm{t}}$) generates fake target domain images ($x^{\mathrm{s}\to\mathrm{t}}$) and the source domain generator ($G_{\mathrm{t}\to\mathrm{s}}$) generates fake source images ($x^{\mathrm{t}\to\mathrm{s}}$). Optimization of $G_{\mathrm{s}\to\mathrm{t}}$ and $G_{\mathrm{t}\to\mathrm{s}}$ relies on the generator loss:

$$\begin{aligned} L^{G}_{\mathrm{adv}}(G_{\mathrm{s}\to\mathrm{t}}, G_{\mathrm{t}\to\mathrm{s}}) = {} & E_{x^{\mathrm{s}} \sim X^{\mathrm{s}}}\left[(D_1(G_{\mathrm{s}\to\mathrm{t}}(x^{\mathrm{s}})) - 1)^2\right] \\ & + E_{x^{\mathrm{t}} \sim X^{\mathrm{t}}}\left[(D_0(G_{\mathrm{t}\to\mathrm{s}}(x^{\mathrm{t}})) - 1)^2\right] \\ & + \lambda \cdot L_{\mathrm{cyc}}(G_{\mathrm{s}\to\mathrm{t}}, G_{\mathrm{t}\to\mathrm{s}}) \end{aligned} \tag{2}$$

where λ is the weighting of the cycle consistency loss (L_{cyc}) in the generator loss (L^{G}_{adv}).

The discriminator D_0 is used to determine whether the image input to D_0 is a fake source image ($x^{\mathrm{t}\to\mathrm{s}}$) or a real source image (x^{s}). D_1 is used to determine whether the image fed to D_1 is the fake target domain image ($x^{\mathrm{s}\to\mathrm{t}}$) or the real target image (x^{t}). Optimization of D_0 and D_1 relies on the discriminator loss:

$$\begin{aligned} & L^{D}_{\mathrm{adv}}(D_0, D_1) \\ & = E_{x^{\mathrm{t}} \sim X^{\mathrm{t}}}\left[(D_1(x^{\mathrm{t}}) - 1)^2 + (D_0(G_{\mathrm{t}\to\mathrm{s}}(x^{\mathrm{t}})) - 0)^2\right] \\ & \quad + E_{x^{\mathrm{s}} \sim X^{\mathrm{s}}}\left[(D_1(G_{\mathrm{s}\to\mathrm{t}}(x^{\mathrm{s}})) - 0)^2 + (D_0(x^{\mathrm{s}}) - 1)^2\right] \end{aligned} \tag{3}$$

where $1 \in \mathcal{R}^{H/8 \times W/8 \times 1}$ represents the real images, while $0 \in \mathcal{R}^{H/8 \times W/8 \times 1}$ represents the fake images. The generators and discriminators are alternately optimized to generate fake images that can confuse the discriminators. The fake target domain images are used in the intermediate domain for the SSN and StSN. The training process of CycleGAN is summarized in Algorithm 1.

Algorithm 1 Training process of CycleGAN

```
Input: The samples of source domain images, $x^s$
       The samples of target domain images, $x^t$
       Source and target domain generators, ($G_{t\to s}$ and $G_{s\to t}$)
       Source and target domain discriminators, ($D_0$ and $D_1$)
Initialize module $G_{s\to t}$, $G_{t\to s}$, $D_0$, and $D_1$, $epoch = 1500$, $batch\_size = 4$.
For $n = 1$ to $epoch$ do
    For $i = 1$ to $len(X^s)/batch\_size$ do
        Get source and target domain images:
            $G_{s\to t} \leftarrow x^s$, $G_{t\to s} \leftarrow x^t$
        Update generators based on:
            $L^G_{adv}(G_{s\to t}, G_{t\to s})$
        Generate fake images:
            $x^{s\to t} \leftarrow G_{s\to t}(x^s)$, $x^{t\to s} \leftarrow G_{t\to s}(x^t)$
        Get fake and real images:
            $D_0 \leftarrow (x^{t\to s}, x^s)$, $D_1 \leftarrow (x^{s\to t}, x^t)$
        Update discriminators based on:
            $L^D_{adv}(D_0, D_1)$
    end for
end for
return $G_{s\to t}$, $G_{t\to s}$, $D_0$, $D_1$
```

3.3 Segmentation sub-network (SSN)

Due to the different principles and parameter values of image acquisition by differing modalities, there are disparities in the feature distribution between the source and target domains. To better extract the domain-invariant features of the source and target domains, we use SSN to transfer the label information of the source domain to the intermediate domain. The SSN is a two-level generative adversarial network that includes feature-level and output-level alignment of the source and intermediate domains. Due to the large discrepancies in the distribution of features between the source and target domain images, we introduce an intermediate domain, which consists of the generated fake target domain images with the source domain style, so the SSN can better learn the label information from the source domain. The SSN transfers the source domain label information to the intermediate domain and generates the segmentation results of the intermediate domain images as the pseudo-labels in the StSN training process. The SSN is a generative adversarial network consisting of a generator (G_{seg}), a feature-level discriminator (D_2) and an output-level discriminator (D_3), where G_{seg} is composed of the feature extractor (F) and the class predictor (P). F uses the modified Resnet101 [27]. P is the Atrous Spatial Pyramid Pooling (ASPP) module [28] which uses multi-scale convolution to extract multi-level semantic features for pixel classification. Generative adversarial learning aligns the feature distribution either at the feature level or output level to reduce the domain shift between the source and intermediate domains.

During the training of the SSN, F extracts the source domain feature maps (f^{s}) from the source domain images (x^{s}), where $f^{\mathrm{s}} = F(x^{\mathrm{s}})$. P takes f^{s} as input and upsamples to produce the source domain pixel-level prediction output ($p^{\mathrm{s}}_{i,c}$), where $p^{\mathrm{s}}_{i,c} = Up(P(f^{\mathrm{s}}))$. The operator Up is a bilinear interpolation algorithm that upsamples the output feature maps to the size of the raw image. We use $p^{\mathrm{s}}_{i,c}$ and one-hot source domain ground truths ($y^{\mathrm{s}}_{i,c}$) to compute $L^{\mathrm{s}}_{\mathrm{seg}}$ and optimize G_{seg}. During the training of the SSN, the source domain image segmentation is supervised, and G_{seg} applies the source domain pixel-level label information to the intermediate domain. Source domain supervised image segmentation loss $L^{\mathrm{s}}_{\mathrm{seg}}(G_{\mathrm{seg}})$ is:

$$L^{\mathrm{s}}_{\mathrm{seg}}\left(G_{\mathrm{seg}}\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y^{\mathrm{s}}_{i,c}\cdot log(p^{\mathrm{s}}_{i,c}) \tag{4}$$

where c is the class index, C is the total number of classes (determined by the number of segmented objects in the different datasets), and N is the number of samples in a batch.

We feed the fake target domain images ($x^{\mathrm{s}\to\mathrm{t}}$) as an intermediate domain into F and output the intermediate domain feature maps ($f^{\mathrm{s}\to\mathrm{t}}$), where $f^{\mathrm{s}\to\mathrm{t}} = F(x^{\mathrm{s}\to\mathrm{t}})$. P takes $f^{\mathrm{s}\to\mathrm{t}}$ as an input feature map which is then upsampled to form the intermediate domain pixel-level prediction results ($p^{\mathrm{s}\to\mathrm{t}}_{i,c}$), where $p^{\mathrm{s}\to\mathrm{t}}_{i,c} = Up(P(f^{\mathrm{s}\to\mathrm{t}}))$. $p^{\mathrm{s}\to\mathrm{t}}_{i,c}$ are used as the pseudo-labels for the intermediate domain images in StSN. The unsupervised domain adaptation is conducted by alternately optimizing the generator (G_{seg}) and discriminators (D_2 and D_3). The adversarial loss $L^{G_{\mathrm{seg}}}_{\mathrm{adv}}$ is used to confuse D_2 and D_3 to align the feature distribution of $f^{\mathrm{s}\to\mathrm{t}}$ and f^{s} at feature level, and the output distribution of $p^{\mathrm{s}\to\mathrm{t}}_{i,c}$ and $p^{\mathrm{s}}_{i,c}$ at output level. The adversarial loss $L^{G_{\mathrm{seg}}}_{\mathrm{adv}}(G_{\mathrm{seg}})$ is:

$$\begin{aligned} L^{G_{\mathrm{seg}}}_{\mathrm{adv}}\left(G_{\mathrm{seg}}\right) = E_{x^{\mathrm{s}\to\mathrm{t}}\sim X^{\mathrm{s}\to\mathrm{t}}}\Big[&(D_2(F(x^{\mathrm{s}\to\mathrm{t}}))-0)^2 \\ &+(D_3(G_{\mathrm{seg}}(x^{\mathrm{s}\to\mathrm{t}}))-0)^2\Big] \end{aligned} \tag{5}$$

where G_{seg} extracts the domain-invariant features of the source and intermediate domains. Finally, D_2 and D_3 are used to distinguish the features of the source and intermediate domains. The adversarial loss $L^{D}_{\mathrm{adv}}(D_2, D_3)$ is:

$$\begin{aligned} &L^{D}_{\mathrm{adv}}(D_2, D_3) \\ &= E_{x^{\mathrm{s}}\sim X^{\mathrm{s}}}\left[(D_2(F(x^{\mathrm{s}}))-0)^2 + \left(D_3\left(G_{\mathrm{seg}}(x^{\mathrm{s}})\right)-0\right)^2\right] \\ &\quad + E_{x^{\mathrm{s}\to\mathrm{t}}\sim X^{\mathrm{s}\to\mathrm{t}}}\left[(D_2(F(x^{\mathrm{s}\to\mathrm{t}}))-1)^2+\left(D_3\left(G_{\mathrm{seg}}(x^{\mathrm{s}\to\mathrm{t}})\right)-1\right)^2\right] \end{aligned} \tag{6}$$

where $1\in\mathcal{R}^{H/8\times W/8\times 1}$ and $0\in\mathcal{R}^{H/8\times W/8\times 1}$ represent the intermediate and source domains, respectively. The training process of the SSN is summarized in Algorithm 2.

Algorithm 2 Training process of SSN

```
Input: The samples of source domain images and labels, $x^s, y^s$
       The samples of intermediate domain images, $x^{s\to t}$
       Feature extractor $F$, class predictor, $P$
       Feature-level and output-level discriminators, ($D_2$ and $D_3$)
Initialize the module $F$, $P$, $D_2$, and $D_3$, epoch=2000, batch_size=4.
For $n \leftarrow 0$ to epoch do
    For $i \leftarrow 0$ to $len(X^s)/batch\_size$ do
        Get source domain images and labels:
            $P, F \leftarrow (x^s, y^s)$
        Output source domain feature maps:
            $f^s \leftarrow F(x^s)$, $p^s_{i,c} \leftarrow G_{seg}(x^s)$
        Update $F$ and $P$ based on:
            $L^s_{seg}(G_{seg})$
        Get intermediate domain images:
            $P, F \leftarrow x^{s\to t}$
        Output intermediate domain feature maps:
            $f^{s\to t} \leftarrow F(x^{s\to t})$, $p^{s\to t}_{i,c} \leftarrow G_{seg}(x^{s\to t})$
        Update $F$ and $P$ based on:
            $L^{G_{seg}}_{adv}(G_{seg})$
        Get feature maps to discriminators:
            $D_2 \leftarrow (f^s, f^{s\to t})$, $D_3 \leftarrow (p^s_{i,c}, p^{s\to t}_{i,c})$
        Update $D_2$ and $D_3$ based on:
            $L^D_{adv}(D_2, D_3)$
    end for
end for
return $F$, $P$, $D_2$, $D_3$
```

3.4 Self-training sub-network (StSN)

With the abovementioned adversarial training of the source and intermediate domains in SSN, we have obtained good segmentation performance using the fake target domain images ($x^{\mathrm{s}\to\mathrm{t}}$) as the intermediate domain. Unfortunately, this is still insufficient to achieve the desired performance when domain shift is severe. Therefore, we introduce a self-training strategy and use the intermediate domain and target domain to train the StSN to transfer the label information of the intermediate domain to the target domain for further improving the image segmentation results of the target domain. It is worth noting that the network structure of the StSN is identical to that of the SSN, the only difference being the images fed into the network during the training process. In the SSN, we use the source and intermediate domains for generative adversarial training, while in the StSN, we use the intermediate and target domains. The prediction results of the intermediate domain images ($x^{\mathrm{s}\to\mathrm{t}}$) in the SSN act as pseudo-labels ($p_{i,c}^{\mathrm{s}\to\mathrm{t}}$) in the StSN.

Firstly, the outputs ($\widetilde{p}_{i,c}^{\mathrm{s}\to\mathrm{t}}$) of the generator ($\widetilde{G}_{\mathrm{seg}}$) are used to compute the segmentation loss ($L_{\mathrm{seg}}^{\mathrm{s}\to\mathrm{t}}$) under the supervision of the one-hot pseudo-labels ($p_{i,c}^{\mathrm{s}\to\mathrm{t}}$). The intermediate domain segmentation loss ($L_{\mathrm{seg}}^{\mathrm{s}\to\mathrm{t}}(\widetilde{G}_{\mathrm{seg}})$) is:

$$L_{\mathrm{seg}}^{\mathrm{s}\to\mathrm{t}}(\widetilde{G}_{\mathrm{seg}}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} p_{i,c}^{\mathrm{s}\to\mathrm{t}} \cdot \log(\widetilde{p}_{i,c}^{\mathrm{s}\to\mathrm{t}}) \tag{7}$$

where c is the class index, C is the total number of classes (determined by the number of segmented objects in different datasets), and N is the number of samples in a batch in the training process.

Secondly, the introduction of the intermediate domain progressively transfers the label information of the intermediate domain to the target domain. The adversarial loss ($L_{\mathrm{adv}}^{\widetilde{G}_{\mathrm{seg}}}\left(\widetilde{G}_{\mathrm{seg}}\right)$) is:

$$\begin{aligned} L_{\mathrm{adv}}^{\widetilde{G}_{\mathrm{seg}}}\left(\widetilde{G}_{\mathrm{seg}}\right) = E_{x^{\mathrm{t}}\sim X^{\mathrm{t}}}\Big[\left(\widetilde{D}_2\left(\widetilde{F}(x^{\mathrm{t}})\right) - 0\right)^2 \\ + \left(\widetilde{D}_3\left(\widetilde{G}_{\mathrm{seg}}(x^{\mathrm{t}})\right) - 0\right)^2\Big] \end{aligned} \tag{8}$$

Finally, the feature-level discriminator ($\widetilde{D}_2$) and the output-level discriminator ($\widetilde{D}_3$) are optimized by $L_{\mathrm{adv}}^{\widetilde{D}}$. $\widetilde{D}_2$ and $\widetilde{D}_3$ are used to distinguish features from different domains and train them adversarially with the generator ($\widetilde{G}_{\mathrm{seg}}$). The adversarial loss $L_{\mathrm{adv}}^{\widetilde{D}}(\widetilde{D}_2, \widetilde{D}_3)$ is:

$$\begin{aligned} & L_{\mathrm{adv}}^{\tilde{D}}(\tilde{D}_2, \tilde{D}_3) \\ & = E_{x^{\mathrm{t}}\sim X^{\mathrm{t}}}\left[\left(\tilde{D}_2(\tilde{F}(x^{\mathrm{t}})) - 1\right)^2 + \left(\tilde{D}_3(\tilde{G}_{\mathrm{seg}}(x^{\mathrm{t}})) - 1\right)^2\right] \\ & \quad + E_{x^{\mathrm{s}\to\mathrm{t}}\sim X^{\mathrm{s}\to\mathrm{t}}}\left[\left(\tilde{D}_2(\tilde{F}(x^{\mathrm{s}\to\mathrm{t}})) - 0\right)^2 + \left(\tilde{D}_3(\tilde{G}_{\mathrm{seg}}(x^{\mathrm{s}\to\mathrm{t}})) - 0\right)^2\right] \end{aligned} \tag{9}$$

where $1 \in \mathcal{R}^{H/8\times W/8\times 1}$ and $0 \in \mathcal{R}^{H/8\times W/8\times 1}$ represent the target and intermediate domains. The potential feature distributions between the intermediate and the target domains are aligned by optimizing the adversarial loss ($L_{\mathrm{adv}}^{\widetilde{G}_{\mathrm{seg}}}$) and discriminator loss ($L_{\mathrm{adv}}^{\widetilde{D}}$). The training process of the StSN is summarized in Algorithm 3.

Algorithm 3 Training process of StSN

```
Input: The samples of intermediate domain images and pseudo-labels, $x^{s\to t}, p^{s\to t}$
       The samples of target domain images, $x^t$
       Feature extractor $\tilde{F}$, class predictor, $\tilde{P}$
       Feature-level and output-level discriminators, ($\widetilde{D}_2$ and $\widetilde{D}_3$)
Initialize the module $\tilde{F}$, $\tilde{P}$, $\widetilde{D}_2$, and $\widetilde{D}_3$, $epoch$ =1000, $batch\_size$ = 4.
For $n$=1 to $epoch$ do
    For $i$ = 1 to $len(X^{s\to t})/batch\_size$ do
        Get intermediate domain images and labels:
            $\tilde{P}, \tilde{F} \leftarrow (x^{s\to t}, p_{i,c}^{s\to t})$
        Output intermediate domain feature maps:
            $\tilde{f}^{s\to t} \leftarrow \tilde{F}(x^{s\to t})$, $\tilde{p}_{i,c}^{s\to t} \leftarrow \tilde{G}_{seg}(x^{s\to t})$
        Update $\tilde{F}$ and $\tilde{P}$ based on:
            $L_{seg}^{s\to t}(\tilde{G}_{seg})$
        Get target domain images:
            $\tilde{P}, \tilde{F} \leftarrow x^t$
        Output target domain feature maps:
            $\tilde{f}^t \leftarrow \tilde{F}(x^t)$, $\tilde{p}_{i,c}^t \leftarrow \tilde{G}_{seg}(x^t)$
        Update $\tilde{F}$ and $\tilde{P}$ based on:
            $L_{adv}^{\tilde{G}_{seg}}(\tilde{G}_{seg})$
        Get feature maps to discriminators:
            $\widetilde{D}_2 \leftarrow (\tilde{f}^{s\to t}, \tilde{f}^t)$, $\widetilde{D}_3 \leftarrow (\tilde{p}_{i,c}^{s\to t}, \tilde{p}_{i,c}^t)$
        Update $\widetilde{D}_2$ and $\widetilde{D}_3$ based on:
            $L_{adv}^{\widetilde{D}}(\widetilde{D}_2, \widetilde{D}_3)$
    end for
end for
return $\tilde{F}$, $\tilde{P}$, $\widetilde{D}_2$, $\widetilde{D}_3$
```

3.5 Network configurations

In the style transfer sub-network, the generators ($G_{\mathrm{s}\to\mathrm{t}}$ and $G_{\mathrm{t}\to\mathrm{s}}$) have the same structure (Fig. 3A), and the discriminators (D_0 and D_1) also have the same structure (Fig. 3B).

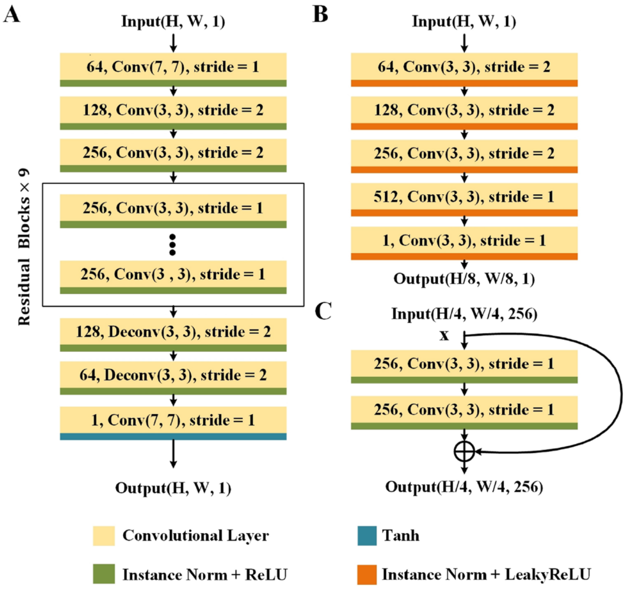
A
Input(H, W, 1)
64, Conv(7, 7), stride = 1
128, Conv(3, 3), stride = 2
256, Conv(3, 3), stride = 2
Residual Blocks × 9
256, Conv(3, 3), stride = 1
256, Conv(3 , 3), stride = 1
128, Deconv(3, 3), stride = 2
64, Deconv(3, 3), stride = 2
1, Conv(7, 7), stride = 1
Output(H, W, 1)
B
Input(H, W, 1)
64, Conv(3, 3), stride = 2
128, Conv(3, 3), stride = 2
256, Conv(3, 3), stride = 2
512, Conv(3, 3), stride = 1
1, Conv(3, 3), stride = 1
Output(H/8, W/8, 1)
C
Input(H/4, W/4, 256)
x
256, Conv(3, 3), stride = 1
256, Conv(3, 3), stride = 1
Output(H/4, W/4, 256)
Convolutional Layer
Tanh
Instance Norm + ReLU
Instance Norm + LeakyReLU

Fig. 3 Details of the generators ($G_{\mathrm{s}\to\mathrm{t}}$ and $G_{\mathrm{t}\to\mathrm{s}}$) and the discriminators (D_0 and D_1) in the style transfer sub-network. **A** Structure of generators ($G_{\mathrm{s}\to\mathrm{t}}$ and $G_{\mathrm{t}\to\mathrm{s}}$), **B** Structure of discriminators (D_0 and D_1), and **C** Residual block used in the generators ($G_{\mathrm{s}\to\mathrm{t}}$ and $G_{\mathrm{t}\to\mathrm{s}}$). W and H are the width and height of the raw image

The SSN and StSN have the same structure, which includes a feature extractor, class predictor, feature-level discriminator, and output-level discriminator. The feature extractors (F and $\widetilde{F}$) are based on the basic ResNet101 [27], without the redundant fully connected layer. Introducing the residual structure of ResNet101 into the feature extractors (F and $\widetilde{F}$) can alleviate the problem of gradient disappearance. The feature extractors (F and $\widetilde{F}$) generate feature maps of size $H/8 \times W/8 \times 2048$, which are fed into the class predictors (P and $\widetilde{P}$), respectively. The class predictors (P and $\widetilde{P}$) adopt the classical ASPP [28]. The ASPP uses convolutional kernels with different dilation rates sampled in parallel to extract multi-scale cardiac context information, as shown in Fig. 4. The size of the feature maps generated by P and $\widetilde{P}$ is $H/8 \times W/8 \times C$, where C is the number of classes, including the number of foreground and background classes. The feature maps generated by P and $\widetilde{P}$ are scaled to the raw image size of $H \times W \times C$ by a bilinear interpolation algorithm.

Figure 5 shows the structure of the feature-level discriminators (D_2 and $\widetilde{D}_2$) and output-level discriminators (D_3 and $\widetilde{D}_3$).

3.6 Training strategy

The training of the TSP-UDANet is divided into two stages: Stage 1 (CycleGAN + SSN) and Stage 2 (CycleGAN + SSN + StSN). In Stage 1, we implement the image-level alignment by using the CycleGAN, which generates fake target domain images to serve as intermediate domain images. Then we train the SSN using the intermediate and source domain images, and the source domain label information is transferred to the intermediate domain. The trained SSN can segment the intermediate domain images, and the segmentation results can be used as pseudo-labels. In Stage 2, we train the StSN using the intermediate domain and the target domain images to transfer the intermediate domain pseudo-label information to the target domain. The trained StSN can then segment the target domain testing images. In the two stages, the source domain label information is progressively transferred to the target domain by using the multi-level adversarial training of the SSN and StSN. Thus, the intermediate domain acts as a bridge between the source and target domains to transfer domain-invariant features between them. In the testing process, the StSN trained in Stage 2 is used as the final segmentation network.

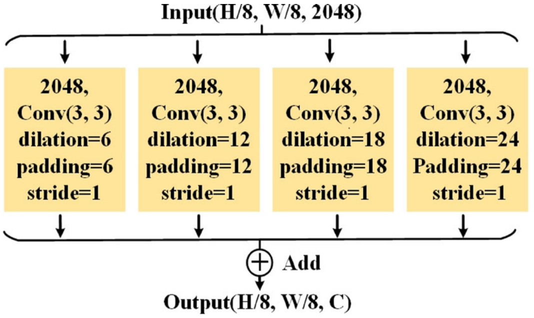
Input(H/8, W/8, 2048)
2048, Conv(3, 3) dilation=6 padding=6 stride=1
2048, Conv(3, 3) dilation=12 padding=12 stride=1
2048, Conv(3, 3) dilation=18 padding=18 stride=1
2048, Conv(3, 3) dilation=24 Padding=24 stride=1
Add
Output(H/8, W/8, C)

Fig. 4 The class predictor is the ASPP. The dilation rates of the convolution kernels are 6, 12, 18, and 24 in order. W and H are the width and height of the raw image. C is the number of classes

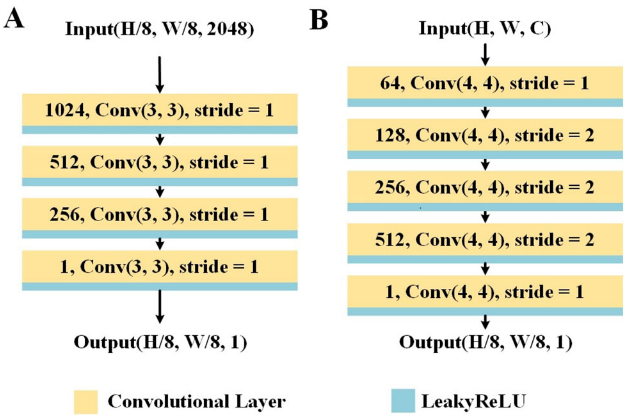
A
Input(H/8, W/8, 2048)
1024, Conv(3, 3), stride = 1
512, Conv(3, 3), stride = 1
256, Conv(3, 3), stride = 1
1, Conv(3, 3), stride = 1
Output(H/8, W/8, 1)
B
Input(H, W, C)
64, Conv(4, 4), stride = 1
128, Conv(4, 4), stride = 2
256, Conv(4, 4), stride = 2
512, Conv(4, 4), stride = 2
1, Conv(4, 4), stride = 1
Output(H/8, W/8, 1)
Convolutional Layer
LeakyReLU

Fig. 5 Details of D_2, $\widetilde{D}_2$, D_3, and $\widetilde{D}_3$ in the SSN and StSN. **A** Structure of the feature-level discriminators (D_2 and $\widetilde{D}_2$), **B** Structure of output-level discriminators (D_3 and $\widetilde{D}_3$). W and H are the width and height of the raw image. C is the number of classes

3.7 Implementation details

We implemented our framework in Pytorch (Version1.7.0). Each sub-network was trained on a computer fitted with an NVIDIA Quadro RTX 5000 and Intel® Xeon® W-2133 CPU. For the CycleGAN, the batch size was 4, and the generators ($G_{s \to t}$ and $G_{t \to s}$) and discriminators (D_0 and D_1) all used the Adam optimizer [29] with a learning rate of 2.0×10^{-4}. The weight λ of cycle consistency loss (L_{cyc}) in the generative adversarial loss (L^{G}_{adv}) was set to 0.8 for the MMWHS dataset and 1.0 for the MS-CMRSeg and the M&Ms datasets. The CycleGAN was trained to generate fake target domain images as the intermediate domain in the TSP-UDANet. Algorithm 1 outlines the CycleGAN training process.

The SSN uses a labeled source domain and an intermediate domain for adversarial training. After training, the SSN implements the initial segmentation of the intermediate domain images, the results of which are used as the pseudo-labels of the intermediate domain images. The StSN uses the pseudo-labeled intermediate domain and the target domain to carry out adversarial training, and the trained StSN achieves an accurate segmentation of the target domain images. The SSN and the StSN undergo the same training process. The generators (G_{seg}, and $\widetilde{G}_{\mathrm{seg}}$) use the stochastic gradient descent (SGD) optimizer [30] with a learning rate of 2.0×10^{-4}, the momentum is set to 0.9, and the decay rate, to 5.0×10^{-4}. The discriminators (D_2, and D_3) use the Adam optimizer with a learning rate of 1.0×10^{-4}. The discriminators ($\widetilde{D}_2$, and $\widetilde{D}_3$) also use the Adam optimizer with the same learning rate. Algorithms 2 and 3 outline the training processes of the SSN and the StSN, respectively.

4 Experiments

In this section, we describe the assessment of the effectiveness of our method under various conditions. These include MRI and CT cardiac images, bSSFP and LGE MRI images, and multi-disease MRI images from different centers and device manufacturers.

4.1 Datasets

To validate the segmentation performance of the TSP-UDANet for the segmentation of cardiac substructures from multimodal medical images, we performed experiments on three datasets: the cross-modality Multi-Modality Whole Heart Segmentation Challenge (MMWHS) dataset [31], the Multi-sequence Cardiac MR Segmentation Challenge (MS-CMRSeg) dataset [32], and the Multi-Center, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge (M&Ms) dataset [33]. We normalized the image slices of the three cardiac datasets and performed data augmentation by rotation, mirroring, and affine transformations to reduce overfitting.

The MMWHS dataset contains unpaired MRI images of 20 subjects and CT images of 20 subjects. The labels include four cardiac substructures: left ventricular myocardium (LV_myo), left atrium (LA), left ventricle (LV), and ascending aorta (AA). In the MRI → CT adaptation, the source domain is MRI, and the target domain is CT; whereas in the CT → MRI adaptation, the source and target domains are reversed. For the MRI and CT images, we randomly selected 80% of the subjects as the training set and the remaining 20% as the testing set. We resampled the raw images to the same in-plane resolution of 1.0 × 1.0 mm. We used 2D slices to train our framework and cropped all images at an ROI of 256 × 256 pixels, centered on the cardiac area. The size of the ROI was sufficient to contain the entirety of the cardiac substructures to be segmented. There were 70 to 100 slices per subject in the MRI image stacks, with 200 to 250 slices per subject for the CT images.

The MS-CMRSeg dataset consists of CMR images in three modalities: LGE, bSSFP, and T2. In the cross-modality UDA experiments, since the number of T2 images was relatively small, we chose to use only the bSSFP images as the source domain and the LGE images as the target domain. There were bSSFP images of 35 subjects and LGE images of 40 subjects. Segmentation objectives included the LV, LV_myo, and right ventricle (RV). There were 8 to 12 slices per subject in the bSSFP images, with 10 to 18 slices per subject for the LGE images. All images were resampled to the same in-plane resolution of 1.25 × 1.25 mm and cropped at an ROI of 224 × 224 pixels, centered on the cardiac area. We used the labeled bSSFP image as the source domain to segment the LGE images.

The M&Ms dataset consists of patients with hypertrophic cardiomyopathy, dilated cardiomyopathy, and healthy subjects. All subjects were scanned at clinical centers in three countries (Spain, Germany, and Canada) using a MR scanner from one of 4 vendors (Siemens, General Electric, Philips, and Canon). The training dataset contains labeled images of 150 subjects from two different MRI vendors (Siemens and Philips). The labeled areas included the LV, LV_myo, and RV. The testing set images were from one of four MR scanner vendors (Siemens, General Electric, Philips, and Canon), including 160 subjects with 10 to 20 slices per subject. Since a significant cross-scanner performance drop was observed on the M&Ms dataset [33], in this study, the M&Ms challenge was chosen as a cross-modality cardiac segmentation task. We chose the training set of the M&Ms dataset as the source domain and the testing set of the M&Ms dataset as the target domain to further validate the generalizability of the TSP-UDANet. All images were aligned and resampled to 1.25 × 1.25 mm and cropped at an ROI of 224 × 224 pixels, centered on the cardiac area.

4.2 Evaluation metrics

We used three evaluation metrics: the dice similarity coefficient (Dice) [34], the average surface distance (ASD) [34], and the Hausdorff distance (HD) [35]. The Dice is used mainly to calculate the similarity between a 3D prediction and the ground truth. The higher the Dice score, the better the segmentation performance. The ASD is used to calculate the average distance between the surface of the 3D prediction and the ground truth, and the HD is the maximum distance from one group to the nearest point in another group, the groups being the 3D prediction and ground truth. In image segmentation, lower ASDs and HDs indicate better segmentation performance. To allow comparison with other studies using the same datasets, we selected Dice and ASD as the evaluation metrics for the

MMWHS dataset and the Dice and HD for the MS-CMRSeg and M&Ms datasets.

5 Results

This section shows the results of applying the TSP-UDA-Net for cardiac segmentation to the three cardiac datasets. We compared our approach with several recently developed methods to explore the segmentation performance of the two-stage multi-level generative adversarial network.

5.1 MMWHS dataset

We evaluated the MRI and CT image cross-modality unsupervised cardiac segmentation in two directions using the MMWHS dataset, namely from MRI to CT images (MRI → CT) and CT to MRI images (CT → MRI). In the MRI → CT adaptation, we used labeled MRI and unlabeled CT images to train our TSP-UDANet, and in the CT → MRI adaptation, we used labeled CT and unlabeled MRI images.

Table 2 shows the performance of our TSP-UDANet on the MMWHS dataset. In the MRI → CT adaptation, we achieved a mean Dice score of 77.1% and a mean ASD of 7.9 mm for the four cardiac substructures. In the CT → MRI adaptation, we achieved a mean Dice score of 69.0% and a mean ASD of 7.2 mm. The segmentation performance of CT → MRI was worse than that of MRI → CT because of the few MRI training images or inherent MRI image characteristics (i.e. their limited contrast) [34]. Figure 6 is a visualization of the segmentation results from the MMWHS dataset. It shows that our method can accurately segment the four cardiac substructures when compared to the ground truth.

5.2 MS-CMRSeg dataset

We used the MS-CMRSeg dataset to validate the generalizability of our TSP-UDANet and found that it achieved precise segmentation of the cardiac substructures, including the LV, LV_myo, and RV. The task of the MS-CMRSeg challenge was to train the segmentation network using labeled bSSFP images for the segmentation of LGE images. Thus, we validated the TSP-UDANet using bSSFP images with labels and LGE images without labels, as required by the MS-CMRSeg segmentation challenge.

As shown in Table 3, we achieved a mean Dice score of 87.5% and a mean HD of 8.2 mm on unsupervised segmentation of LGE images. Figure 7 is a visualization of the segmentation results on the MS-CMRSeg dataset. We can clearly see the changes in the cardiac slices and the segmentation results of the TSP-UDANet from the base to the apex.

Table 2 Results of MMWHS (MRI → CT) adaptation segmentation

	DICE (%)					ASD (mm)				
	AA	LA	LV	LV_myo	Mean	AA	LA	LV	LV_myo	Mean
AdaOutput [26]*	65.2	76.6	54.4	43.6	59.9	17.9	**5.5**	5.9	8.9	9.6
CycleGAN [13]*	73.8	75.7	52.3	28.7	57.6	11.5	13.6	9.2	8.8	10.8
PnP-AdaNet [16]*	74.0	68.9	61.9	50.8	63.9	12.8	6.3	17.4	14.7	12.8
CyCADA [36]*	72.9	77.0	62.4	45.3	64.4	9.6	8.0	9.6	10.5	9.4
SIFA [34]	***81.3***	***79.5***	***73.8***	61.6	74.1	***7.9***	6.2	***5.5***	8.5	***7.0***
ARL-GAN [37]	71.3	**80.6**	69.5	**81.6**	***75.7***	**6.3**	***5.9***	6.7	***6.5***	**6.4**
TSP-UDANet (Ours)	**82.4**	73.7	**87.4**	***65.0***	**77.1**	11.9	10.1	**4.9**	**4.7**	7.9

Results of MMWHS (CT → MRI) adaptation segmentation

	DICE (%)					ASD (mm)				
	AA	LA	LV	LV_myo	mean	AA	LA	LV	LV_myo	mean
AdaOutput [26]*	60.8	39.8	71.5	35.5	51.9	**5.7**	8.0	4.6	***4.6***	**5.7**
CycleGAN [13]*	64.3	30.7	65.0	43.0	50.7	***5.8***	9.8	6.0	5.0	6.6
PnP-AdaNet [16]*	43.7	47.0	***77.7***	***48.6***	54.3	11.4	14.5	***4.5***	5.3	8.9
CyCADA [36]*	60.5	44.0	77.6	47.9	57.5	7.7	13.9	4.8	5.2	7.9
SIFA [34]	***65.3***	***62.3***	**78.9**	47.3	***63.4***	7.3	***7.4***	**3.8**	**4.4**	***5.7***
TSP-UDANet (Ours)	**70.3**	**78.3**	70.4	**57.0**	**69.0**	9.5	**6.5**	7.1	5.7	7.2

Bold indicates the best scores, and bolditalic, the second-best scores

*Denotes the results reported by SIFA [34]

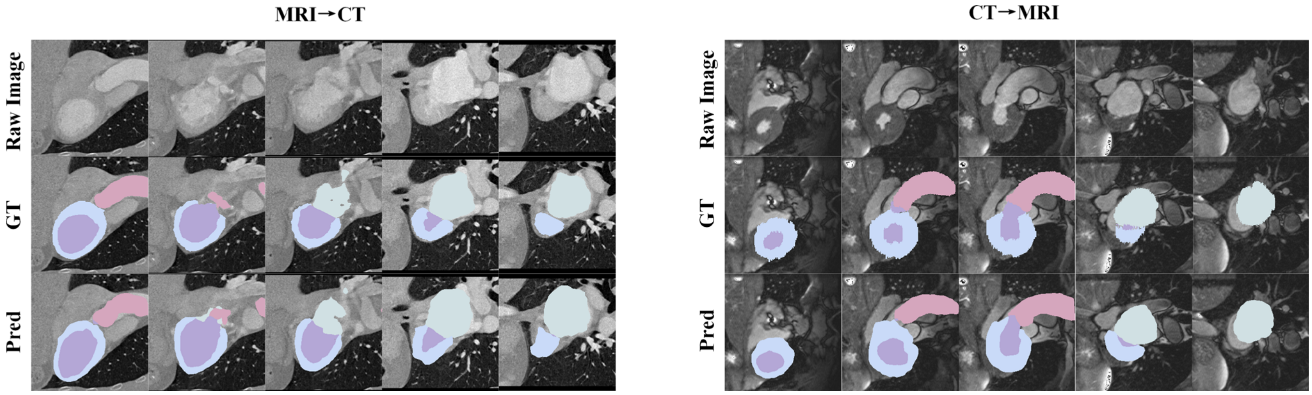

Fig. 6 Visualization of the results of our method from a representative subject in the MMWHS testing set. The uppermost row is the raw images, the middle row is the ground truth (GT), and the bottom row shows the predicted result (Pred). The images in the left panel were taken from MRI → CT, and those on the right from CT → MRI. The cardiac substructures AA, LA, LV, and LV_myo are shaded in pink, pale grey, light purple, and pale blue, respectively

Table 3 MS-CMRSeg segmentation results (Cardiac bSSFP → Cardiac LGE)

	DICE (%)				HD (mm)			
	LV	LV_myo	RV	Mean	LV	LV_myo	RV	Mean
Tao et al. [38]	84.7	68.6	77.6	77.0	17.9	21.9	17.4	19.1
Vesal et al. [35]	***91.2***	78.8	83.2	84.4	10.8	12.5	17.1	13.4
Wang et al. [21]	89.3	80.1	87.1	85.5	15.7	13.5	15.2	14.8
Vesal et al. [20]	90.9	79.4	***87.8***	86.0	**7.6**	***9.3***	***8.4***	***8.4***
Chen et al. [39]	**91.9**	**82.6**	87.5	***87.3***	12.4	10.2	15.3	12.7
TSP-UDANet (Ours)	90.8	***81.0***	**90.6**	**87.5**	***8.2***	**8.4**	**8.1**	**8.2**

Bold indicates the best scores, and bolditalic, the second-best scores

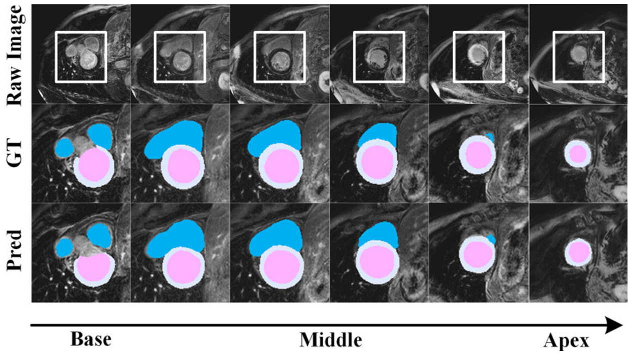

Fig. 7 Visualization results of our method from a representative subject (Pat_40) with median Dice score in the MS-CMRSeg testing dataset. The leftmost images in each row are from the base of the heart, moving to the right are slices progressing towards the apex. The uppermost row is the raw LGE images, the middle row is the ground truth (GT) images, and the bottom row shows the predicted result (Pred). The LV, RV, and LV_myo are shown in pink, cyan and gray, respectively. Note that the sub-figures of the second and third rows are zoomed and cropped for improved clarity

5.3 M&Ms dataset

On the M&Ms dataset, the cardiac images were acquired from 4 different MR scanners, where the training images in the source domain were from Siemens and Philips, and the testing images in the target domain were from Siemens, Philips, General Electric, and Canon. In this experiment, the target domain labels were used for evaluation only, without being used in the training process.

As shown in Table 4, the TSP-UDANet achieved a mean Dice score of 85.2% and a mean HD of 13.2 mm. The Dice scores of TSP-UDANet were 90.1% (LV), 79.5% (LV-myo), and 85.2% (RV), respectively. The HD values are respectively 11.8 mm (LV), 8.7 mm (LV_myo), and 19.1 mm (RV). Figure 8 shows the segmentation results on the M&Ms dataset. The sizes of the three target structures (including LV, LV_myo, and RV) vary greatly from the base to the apex of the heart, but TSP-UDANet can locate and segment the target structures well.

5.4 Comparison with other methods

To demonstrate the effectiveness of our proposed UDA method on multi-modality data, we compared our TSP-UDANet with other state-of-the-art (SOTA) unsupervised learning methods. For a fair comparison, we selected the methods developed on each of the three datasets (including

Table 4 M&Ms segmentation results

	DICE (%)				HD (mm)			
	LV	LV_myo	RV	Mean	LV	LV_myo	RV	Mean
Li et al. [43]	76.7	71.6	63.6	70.6	20.1	33.0	50.7	34.6
Carscadden et al. [44]	88.2	79.3	76.2	81.2	13.7	16.2	31.9	20.6
Scannell et al. [45]	88.0	***80.0***	84.0	84.0	14.5	17.3	***17.5***	16.4
Full et al. [46]	**91.2**	**85.3**	**88.5**	**88.3**	**9.1**	***11.7***	**12.3**	**11.0**
TSP-UDANet (Ours)	***90.1***	79.5	***86.0***	***85.2***	***11.8***	**8.7**	19.1	***13.2***

Bold indicate the best scores, and bolditalic, the second-best score

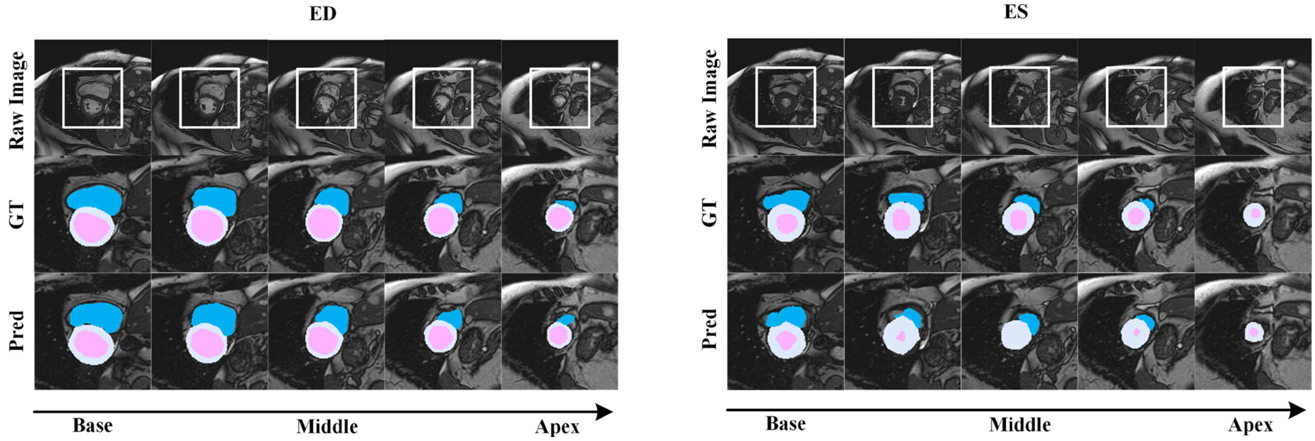

Fig. 8 Visualization of the results of our method from a representative subject (Pat_E5J6L2) with median Dice in the M&Ms testing set. The uppermost row is the raw images, the middle row is the ground truth (GT), and the bottom row shows the predicted result (Pred). The images in the left panel were taken at end diastole (ED) and those on the right, at end systole (ES). In both panels, the images in each column are, from the left to right, the base, middle, and apex slice samples, respectively. The LV, RV, and LV_myo are shown in pink, cyan and gray, respectively. Note that the sub-figures of the second and third rows are zoomed and cropped for improved clarity

the MMWHS dataset, MS-CMRSeg dataset, and M&Ms dataset) for comparison and have cited the results from the original papers.

5.4.1 MMWHS dataset

In Table 2, we compare the performance of our method and other SOTA methods on the MMWHS dataset, including AdaOutput [26], CycleGAN [13], PnP-AdaNet [16], CyCADA [36], and SIFA [34]. In both the MRI → CT and the CT → MRI adaptations, our method performed well as measured by the Dice and ASD. In more detail, the mean Dice score of our method was higher than that of CyCADA [36] by 12.7% (MRI → CT) and 11.5% (CT → MRI), and the mean ASD of our method was lower than that of CyCADA [36] by 1.5mm (MRI → CT) and 0.7mm (CT → MRI). When compared with SIFA [34], our method improved the mean Dice score by 3.0% (MRI → CT) and 5.6% (CT → MRI). Our method also showed the best performance in the ASD on LV and LV_myo (MRI → CT), and on LA (CT → MRI). These results demonstrate the effectiveness of the TSP-UDANet for cross-modality cardiac image segmentation. Table 2 also shows that the mean Dice score was not consistent with the mean ASD in the MRI → CT and CT → MRI adaptation tasks, because of the unsuccessful segmentation results in the slices at the base and apex of the heart [40]. Furthermore, the Dice score was sensitive to internal filling of the mask, while the ASD was sensitive to segmented edges [41].

In this study, the TSP-UDANet combines image-level, feature-level, and output-level alignments to segment cross-modality cardiac images and achieved the best mean Dice score in the MRI → CT and CT → MRI adaptation. Among the other approaches we tested, PnP-AdaNet [16], an extended network only aligns the feature distribution between the source and target domains in the output-level feature space. AdaOutput [26] conducts adversarial training only at the output level in the source and target domains, so the segmentation performance was poor. SIFA [34] introduced feature-level and image-level alignments, and achieved the second-best segmentation result for LA in both directions in bidirectional domain adaptation. ARL_GAN [37] employs image-level alignment and then uses the generated images to train a single-level generative adversarial segmentation network. In the MRI → CT adaptation, the Dice scores of AA and LV obtained by ARL_GAN were 11.1% and 17.9% lower than that of the TSP-UDANet. Furthermore, ARL_GAN [37] only operated in one direction (MRI → CT).

5.4.2 MS-CMRSeg dataset

In Table 3, we compare the performance of our method and other SOTA methods on the MS-CMRSeg dataset, including Tao et al. [38], Vesal et al. [35], Wang et al. [21], Vesal et al. [20], and Chen et al. [39]. We achieved the best mean Dice score compared with other methods. Our mean Dice score was 0.2% higher than that of Chen et al. [39], who achieved a value of 87.3%. Furthermore, we obtained the best Dice score for the RV with a value of 90.6%, which was 3.1% higher than that of Chen et al. [39] and 2.8% higher than that of Vesal et al. [20]. The HD scores of our method for the LV_myo and RV were 8.4 mm and 8.1 mm, both of which are better than the other methods.

In our model, we use CycleGAN to generate pseudo-LGE images as intermediate domain images and use a self-training strategy to bridge the SSN and StSN. Furthermore, our proposed backbone combines basic Resnet101 and ASPP, and works well on the image segmentation tasks. Vesal et al. [20] achieved the second-best segmentation results for the RV. They used entropy minimization and point-cloud shape adaptation to extract domain-invariant features from cross-modality cardiac images. Vesal et al. [35] and Wang et al. [21] achieved poor segmentation results for LV, LV_myo, and RV. Vesal et al. [35] trained a U-net [42] using labeled bSSFP images, and fine-tuned the trained network using LGE images. Wang et al. [21] used a two-channel U-net [42], which only used feature-level alignment to extract image features separately from the source and target domains. Compared with our method, Chen et al. [39] achieved a lower mean Dice score.

5.4.3 M&Ms dataset

In Table 4, we compare the performance of our method and other SOTA methods on the M&Ms dataset, including Li et al. [43], Carscadden et al. [44], Scannell et al. [45], and Full et al. [46]. We achieved the best HD for LV_myo in all methods, namely 8.7 mm. Full et al. [46] used a supervised learning method based on nnU-Net [47] and achieved a mean Dice score of 88.3% and a mean HD of 11.0 mm. Our method achieved Dice scores for LV and RV, which were 2.1% and 2.0% higher than those of Scannell et al. [45]. In the TSP-UDANet, the ASPP acted as a class predictor after the Resnet101 to fuse the multi-scale cardiac features. The mean Dice score of our result was 4.0% higher than that obtained by Carscadden et al. [44], who only used Resnet101 as the segmentation network, whereas our segmentation sub-network (SSN) can be used as a general backbone for image segmentation. Li et al. [43] proposed a cascaded encoding–decoding network as the backbone and achieved a mean Dice score of 70.6%, showing that it is not a good strategy to use a single network for both segmentation and style transfer tasks. Our mean Dice score was 14.6% higher than that of Li et al. [43]. Scannell et al. [45] used a traditional GAN as the backbone, and the generator used the U-net. Compared with the traditional GAN of [45], we added a feature-level discriminator to learn more useful features. When compared with the results of [45], our method improved the Dice score by 2.1% and 2.0% for the LV and RV, respectively. As a supervised training approach, Full et al. [46] used an ensemble of five 2D and five 3D nnU-Net and achieved better segmentation performance on the M&Ms cardiac dataset, using a variety of intensity-based data augmentation methods (i.e., noise addition, brightness modification and contrast modification). These data augmentation techniques are specifically designed for the M&Ms dataset due to the variety of imaging protocols and MRI vendors [33]; whereas our method employed a domain adaptation strategy to achieve good cardiac segmentation, which is less dependent on the specific vendors.

6 Ablation analysis

We performed an ablation analysis to demonstrate the effect of introducing the intermediate domain and the multi-level generative adversarial approach for UDA cross-modality cardiac segmentation. In the MMWHS dataset, the discrepancy in appearance between the CT and MRI cardiac images is evident, which further illustrates the superiority of TSP-UDANet.

In the ablation analysis, we compared the segmentation results of SSN (w/o CycleGAN), Stage 1 (CycleGAN + SSN), and Stage 2 (CycleGAN + SSN + StSN) in the MMWHS dataset to verify the impact of the key components, as shown in Table 5. In SSN (w/o CycleGAN), we used the source and target domains to train the SSN for feature-level and output-level alignment. The trained SSN segmented the testing target domain images in the testing process. In Stage 1, we introduced image-level alignment using CycleGAN, where the generated fake target domain images acted as intermediate domain images.

We used the source and intermediate domains to train the SSN. During the testing process, the testing target domain images were segmented by the trained SSN. In Stage 2, the fake target domain images were used as the intermediate domain to connect the source and target domains. We used the intermediate and target domains to

Table 5 Results of the MMWHS (MRI → CT) segmentation in the ablation experiment

	DICE (%)					ASD (mm)				
	AA	LA	LV	LV_myo	Mean	AA	LA	LV	LV_myo	Mean
SSN (w/o CycleGAN)	79.0	**78.3**	84.6	53.4	73.8	19.2	19.7	***5.1***	**4.6**	12.2
Stage 1 (CycleGAN + SSN)	**82.7**	72.2	***87.0***	***62.5***	***76.1***	***17.0***	***13.6***	***5.1***	5.9	***10.4***
Stage 2 (CycleGAN + SSN + StSN)	***82.4***	***73.7***	**87.4**	**65.0**	**77.1**	**11.9**	**10.1**	**4.9**	***4.7***	**7.9**

Results of the MMWHS (CT → MRI) segmentation in the ablation experiment

	DICE (%)					ASD (mm)				
	AA	LA	LV	LV_myo	Mean	AA	LA	LV	LV_myo	Mean
SSN (w/o CycleGAN)	47.2	68.3	56.4	22.9	48.7	27.4	9.1	19.8	20.6	19.2
Stage 1 (CycleGAN + SSN)	***70.2***	***78.0***	***69.9***	***54.1***	***68.0***	**9.4**	**6.4**	***7.6***	***6.3***	***7.4***
Stage 2 (CycleGAN + SSN + StSN)	**70.3**	**78.3**	**70.4**	**57.0**	**69.0**	***9.5***	***6.5***	**7.1**	**5.7**	**7.2**

Bold indicates the best score, and bolditalic is the second-best score

train the StSN, where the intermediate domain images were matched with pseudo-labels generated by the SSN. During the testing process, the testing images of the target domain were segmented with the trained StSN.

In Stage 1, we introduced the CycleGAN for image-level alignment and the SSN for aligning the feature distribution between the source and intermediate domains at the feature level and the output level. In the MRI → CT adaptation, Stage 1 outperformed the SSN (w/o CycleGAN), where the mean Dice score was 2.3% higher than that of the SSN (w/o CycleGAN), and the mean ASD was 1.8 mm lower. In the CT → MRI adaptation, the segmentation results of Stage 1 outperformed the SSN (w/o CycleGAN) for all cardiac substructures assessed. The mean Dice score of Stage 1 was 19.3% higher than that of the SSN (w/o CycleGAN), and a mean ASD was 11.8 mm lower. The increase in segmentation accuracy demonstrated that image-level feature alignment is effective for target domain image segmentation.

Unlike the SSN (w/o CycleGAN), Stage 2 used fake target domain images as an intermediate domain to bridge the source and target domains. We took the result of the StSN as the final segmentation result wherein the mean Dice score was 3.3% higher, and the mean ASD was 4.3 mm lower than those of the SSN (w/o CycleGAN) in the MRI → CT adaptation. In the CT → MRI adaptation, the mean Dice score of the StSN was 20.3% higher, and the mean ASD was 12.0 mm lower than those of the SSN (w/o CycleGAN). In Stage 2, the source domain label information was progressively transferred to the target domain through the intermediate domain, which reduced the domain shift and improved the segmentation performance of the target domain images. As shown in Fig. 9, the segmentation results produced by Stage 2 were closer to the ground truth than that of Stage 1.

7 Discussion

In this paper, we focus on the UDA problem for cross-modality cardiac segmentation. We present a novel framework, TSP-UDANet, which can effectively extract and align the cardiac domain-invariant features from multiple levels in cardiac domain adaptation segmentation tasks. We conduct generative adversarial training at three levels, namely image level, feature level, and output level, to achieve cross-modality cardiac segmentation on the MMWHS dataset. The results of TSP-UDANet are compared with other methods in Table 2, in which it can be seen that the results of the cooperative adversarial learning at the three levels are better than the results of individual image-level or feature-level alignment. The network can extract more semantic features in cooperative adversarial learning and achieve better alignment of image features from different modalities. For example, PnP-AdaNet [16], CycleGAN [13], and ARL_GAN [37] do not incorporate cooperative adversarial learning methods with feature-level and image-level alignment. AdaOutput [26] uses output-level semantic space alignment, which is less effective than CyCADA [36]. CyCADA [36] and SIFA [34] only use image-level and feature-level alignments. Our method outperforms CyCADA [36] and SIFA [34], demonstrating the effectiveness of generative adversarial training through image-level, feature-level, and output-level alignments.

We introduce the intermediate domain to bridge the source and target domains and reduce the domain shift between them for two reasons: one is the failure of

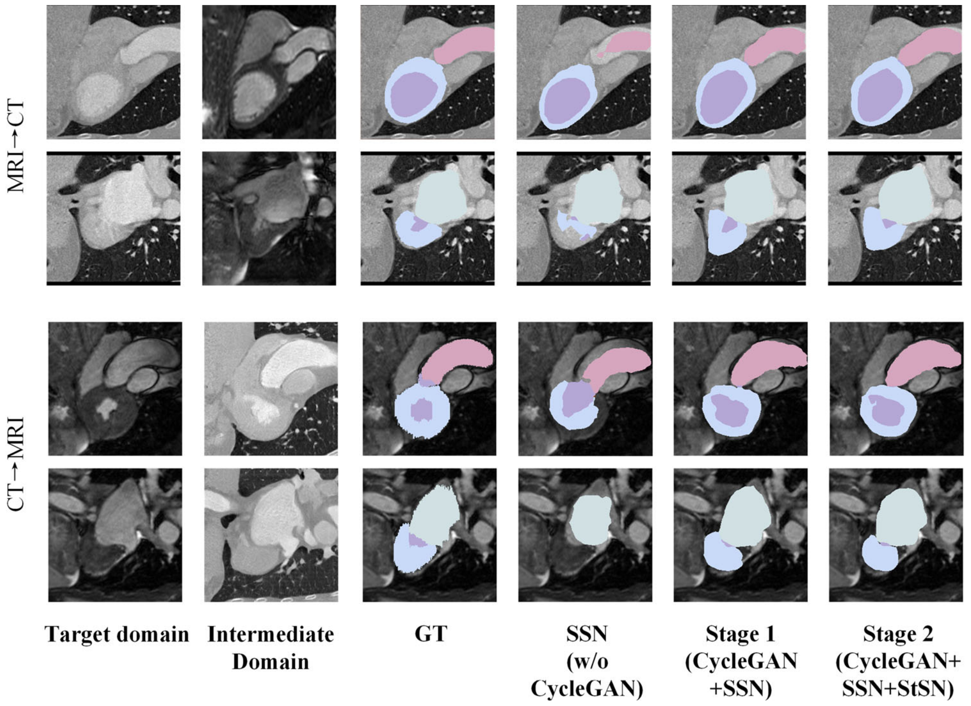

Fig. 9 Visual comparison of the 2D slice results of the proposed method from a representative case with the median Dice for the MMWHS testing set in the ablation experiments. From left to right are the target domain image (column 1), the intermediate domain image generated by CycleGAN (column 2), the target domain ground truth (column 3), the segmentation results of SSN (w/o CycleGAN) and Stage 1 (columns 4–5), and the final segmentation results of Stage 2 (column 6). The cardiac substructures AA, LA, LV, and LV_myo are shaded in pink, pale grey, light purple, and pale blue, respectively. The first and second rows are MRI → CT adaptation examples, with only AA, LV, and LV_myo in the first row, and LA, LV, and LV_myo in the second row. The third and fourth rows are CT → MRI adaptation examples, with only AA, LV, and LV_myo in the third row, and LA, LV, and LV_myo in the fourth row

adversarial training in the SSN (w/o CycleGAN), and the other is the discrepancy in image appearance between the source and target domains. Obviously, as shown by the ablation experiment, the discrepancies in the appearance of images obtained by different modalities substantially impact cardiac segmentation results. We attempted to use source and target domain adversarial training of the SSN (w/o CycleGAN), but the segmentation results were unsuccessful, as shown in Table 5. The main reason is the large discrepancy in image intensity distribution between the source and target domain images. Images with a similar style are more likely to perform better in UDA segmentation, so we used CycleGAN to generate fake target domain images as the intermediate domain. The intermediate domain divides the label information transfer process between the source and target domains into two parts: SSN and StSN. The SSN segments the intermediate domain images, and then the segmentation results, as the pseudo-labels, are used for the training of the StSN. The trained StSN segments the target domain images. This achieves a two-stage progressive UDA cross-modality cardiac segmentation. Figure 10 shows the mean Dice statistics on the MMWHS dataset in the form of box plots, where Fig. 10A is the MRI → CT adaptation, and Fig. 10B is the CT → MRI adaptation. It can be seen that the mean Dice score increases after introducing the intermediate domain.

To verify the generalizability of the TSP-UDANet, we conducted experiments on the MMWHS, MS-CMRSeg, and M&Ms datasets. In the MMWHS dataset, the mean Dice scores obtained by the TSP-UDANet were 3% and 5.6% higher than those of SIFA [34] in both MRI → CT and CT → MRI adaptations. In the MS-CMRSeg dataset, TSP-UDANet achieved the best mean Dice score, being 0.2% higher than the result of Chen et al. [39], and TSP-UDANet achieved the best HDs in the LV_myo and RV. In the M&Ms dataset, TSP-UDANet achieved the best HD for LV_myo. Taken together, these results show that the TSP-UDANet is a generalizable method for UDA cross-modality segmentation. This is mainly because the TSP-UDANet can effectively reduce the domain shift using style-transferred images as the intermediate domain. Moreover, the segmentation networks employ multiple discriminators for adversarial training to extract domain-invariant features, which enables the generator to better align the feature distribution between different domains at multiple levels. The TSP-UDANet uses the classical Resnet101 and ASPP for the segmentation backbone, which can serve as a general network configuration in the task of

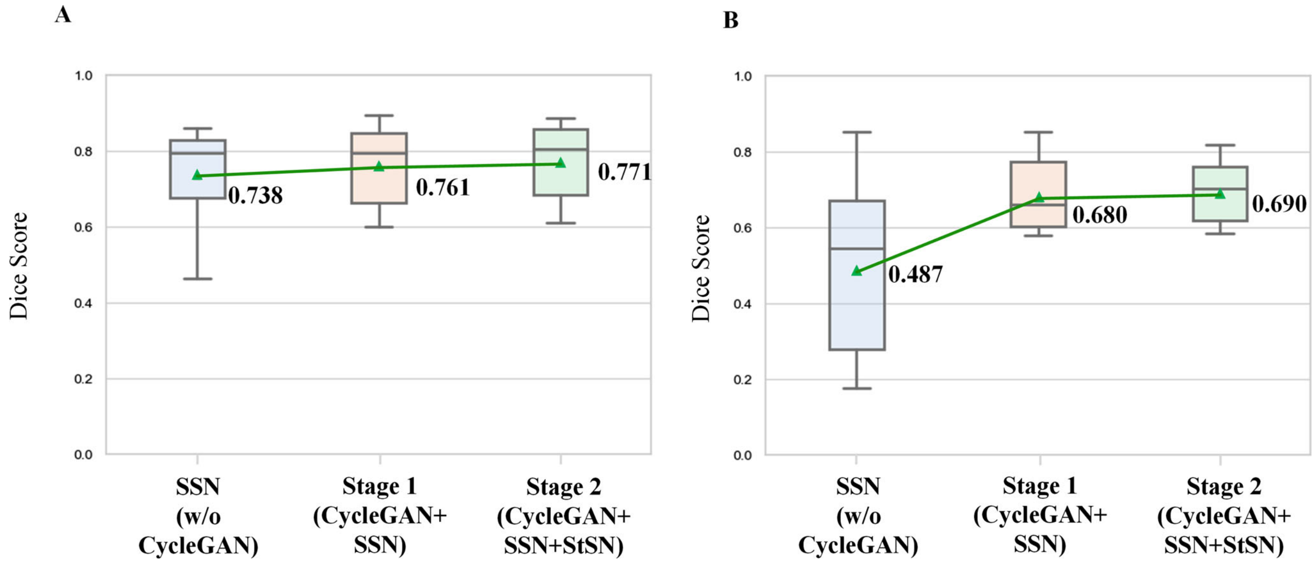

Fig. 10 Segmentation results of each ablation experiment using the proposed method, showing the change in Dice score at each stage of the progressive unsupervised cross-modality adaptation segmentation process. The vertical coordinate represents the mean Dice score of all segmented objects from the MMWHS testing set. **A** MRI → CT adaptation segmentation, **B** CT → MRI adaptation segmentation. The green triangles represent the mean values for each stage. Upper and lower rectangle boundaries indicate the interquartile range; middle horizontal lines are median values and whiskers indicate the full range of the data

image segmentation. The Resnet101 can provide enough depth for feature extraction, and its residual structure can effectively prevent gradient disappearance. The ASPP can adapt to multi-scale contextual features because it has multi-scale receptive fields.

Our TSP-UDANet achieves good performance on three cross-modality cardiac datasets, but there are still some limitations. Figure 11 shows the visualization results of the TSP-UDANet on the MS-CMRSeg and M&Ms datasets. The segmentation results for LV_myo and LV in the apical region are weaker than those in the basal region. Some substructures in the region of the cardiac apex are very small and occupy fewer pixels in each image slice, which makes the extraction of meaningful 2D anatomical features very challenging. In future, we will explore the possibility of introducing 3D anatomical information to tackle the difficulties in segmenting small objects.

8 Conclusion

In this paper, we have proposed a two-stage progressive UDA network for segmenting multi-modality cardiac images. The network is trained from multi-level feature spaces at the image level, feature level, and output level. We introduce an intermediate domain linking the source and target domains. An improved self-training process is used in Stage 2 to progressively reduce the domain shift between the different domains and to extract domain-invariant features. We have validated the method using

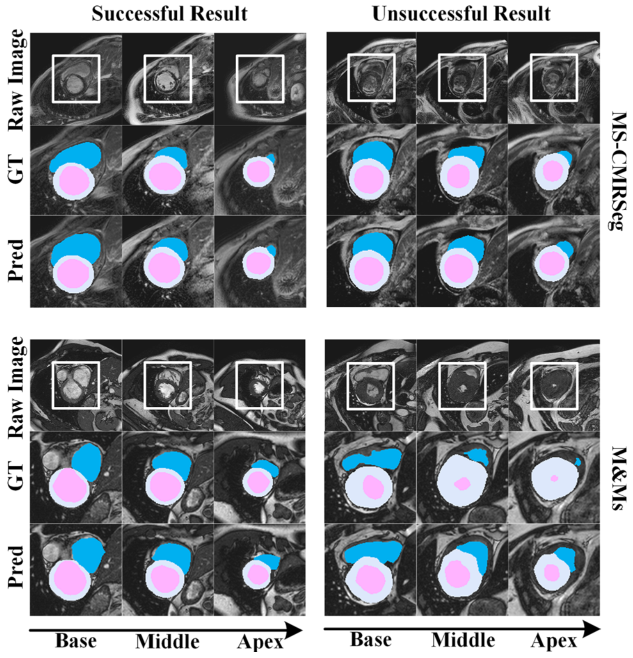

Fig. 11 Visualization of two successful subjects (Pat_6 and Pat_A2H5K9, upper and lower left-hand panels) and two unsuccessful subjects (Pat_8 and Pat_ A8C5E9, right-hand panels) (n.b, successful and unsuccessful refers to the subjects with the highest and the lowest Dice score in the MS-CMRSeg and M&Ms testing set). The upper three rows show the segmentation results from the MS-CMRSeg dataset, and lower three rows show the segmentation results from the M&Ms dataset. The left three columns show successful segmentation results, and the right three columns show unsuccessful segmentation results. LV, RV, and LV_myo are shown in pink, cyan and gray. Note that the sub-figures of the second, third, fifth, and sixth rows are zoomed and cropped for improved clarity

unpaired cardiac MRI and CT images, LGE and bSSFP images, and CMR images acquired with devices manufactured by multiple vendors. Compared with existing methods, our approach achieves good segmentation performance for a variety of source images and has good generalizability making it possible to apply the UDA network to the segmentation of other medical images. In future, to demonstrate the generalizability and robustness of our method, we will explore its application in other areas beyond cardiac segmentation in multimodal images, for instance the segmentation of solid tumors.

Acknowledgements This work is supported by the National Key Research and Development Program of China (2020YFC2004400), the National Natural Science Foundation of China (No. 61773110), the Fundamental Research Funds for the Central Universities (No. N2119008), Natural Science Foundation of LiaoNing Province (General Program) (No.2021-MS-087) and Open Grant by National Health Commission Key Laboratory of Assisted Circulation (Sun Yat-sen University) (No. cvclab201901). The authors would also like to thank the editor and reviewers for their valuable advice which has helped to improve the article.

Data availability The data that support the findings of this study are publicly available in the following links: MMWHS https://zmiclab.github.io/projects/mmwhs/. MS-CMRSeg https://zmiclab.github.io/projects/mscmrseg19/. M&Ms https://www.ub.edu/mnms/.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. World Health Organization (2019) Cardiovascular diseases (CVDs). Available from: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds). Accessed 2022
2. Roth GA, Mensah GA, Johnson CO, Addolorato G, Ammirati E, Baddour LM, Barengo NC, Beaton AZ, Benjamin EJ, Benziger CP (2020) Global burden of cardiovascular diseases and risk factors, 1990–2019: update from the GBD 2019 study. J Am Coll Cardiol 76(25):2982–3021
3. Zhuang X, Li L, Payer C, Štern D, Urschler M, Heinrich MP, Oster J, Wang C, Smedby Ö, Bian C (2019) Evaluation of algorithms for multi-modality whole heart segmentation: an open-access grand challenge. Med Image Anal 58:101537
4. Leiner T, Rueckert D, Suinesiaputra A, Baeßler B, Nezafat R, Išgum I, Young AA (2019) Machine learning in cardiovascular magnetic resonance: basic concepts and applications. J Cardiovasc Magn Reson 21(1):1–14
5. Hoffman J, Wang D, Yu F, Darrell T (2016) FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. http://arxiv.org/abs/1612.02649
6. Dong N, Kampffmeyer M, Liang X, Wang Z, Dai W, Xing E (2018) Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. Paper presented at the international conference on medical image computing and computer-assisted intervention, Granada, Spain, pp 544–552
7. Ouyang C, Kamnitsas K, Biffi C, Duan J, Rueckert D (2019) Data efficient unsupervised domain adaptation for cross-modality image segmentation. Paper presented at the international conference on medical image computing and computer-assisted intervention, Shenzhen, China, pp 669–677
8. Liu Y, Wang W, Wang K, Ye C, Luo G (2019) An automatic cardiac segmentation framework based on multi-sequence MR image. Paper presented at the international workshop on statistical atlases and computational models of the heart, Shenzhen, China, pp 220–227
9. Valindria VVPN, Rajchl M, Lavdas I, Aboagye EO, Rockall AG, Rueckert D, Glocker B (2018) Multi-modal learning from unpaired images: Application to multi-organ segmentation in CT and MRI. Paper presented at the 2018 IEEE winter conference on applications of computer vision (WACV), LakeTahoe, NV, USA, pp 547–556
10. Dou Q, Liu Q, Heng PA, Glocker B (2020) Unpaired multi-modal segmentation via knowledge distillation. IEEE Trans Med Imaging 39:2415–2425
11. Jiang J, Hu YC, Tyagi N, Zhang P, Rimner A, Mageras GS, Deasy JO, Veeraraghavan H (2018) Tumor-aware, adversarial domain adaptation from CT to MRI for lung cancer segmentation. Paper presented at the international conference on medical image computing and computer-assisted intervention, Granada, Spain, pp 777–785
12. Chen C, Dou Q, Chen H, Heng PA (2018) Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. Paper presented at the international workshop on machine learning in medical imaging, Granada, Spain, pp 143–151
13. Zhu J Y, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. Paper presented at the proceedings of the IEEE international conference on computer vision, Venice, Italy, pp 2223–2232
14. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: Maximizing for domain invariance. http://arxiv.org/abs/1412.3474
15. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. Paper presented at the international conference on machine learning, Lille, France, pp 97–105
16. Dou Q, Ouyang C, Chen C, Chen H, Glocker B, Zhuang X, Heng PA (2019) PnP-AdaNet: Plug-and-Play Adversarial domain adaptation Network with a benchmark at cross-modality cardiac segmentation. IEEE Access 7:99065–99076
17. Borgwardt KM, Gretton A, Rasch MJ, Kriegel HP, Schölkopf B, Smola AJ (2006) Integrating structured biological data by kernel maximum mean discrepancy. Bioinformatics 22(14):e49–e57
18. Dou Q, Ouyang C, Chen C, Chen H, Heng PA (2018) Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. http://arxiv.org/abs/1804.10916.
19. Chen Y, Li W, Sakaridis C, Dai D, Van Gool L (2018) Domain adaptive faster R-CNN for object detection in the wild. Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, pp 3339–3348
20. Vesal S, Gu M, Kosti R, Maier A, Ravikumar N (2021) Adapt everywhere: unsupervised adaptation of point-clouds and entropy minimisation for multi-modal cardiac image segmentation. IEEE Trans Med Imaging 40(7):1838–1851
21. Wang J, Huang H, Chen C, Ma W, Huang Y, Ding X (2019) Multi-sequence cardiac MR segmentation with adversarial domain adaptation network. Paper presented at the international workshop on statistical atlases and computational models of the heart, Shenzhen, China, pp 254–262

22. Kamnitsas K, Baumgartner C, Ledig C, Newcombe V, Simpson J, Kane A, Menon D, Nori A, Criminisi A, Rueckert D (2017) Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. Paper presented at the international conference on information processing in medical imaging, pp 597–609
23. Jain RK, Sato T, Watasue T, Nakagawa T, Iwamoto Y, Han X, Lin L, Hu H, Ruan X, Chen YW (2022) Unsupervised domain adaptation using adversarial learning and maximum square loss for liver tumors detection in multi-phase CT images. Paper presented at the 2022 44th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 1536–1539
24. Panfilov E, Tiulpin A, Klein S, Nieminen M T, Saarakkala S (2019) Improving robustness of deep learning based knee mri segmentation: Mixup and adversarial domain adaptation. Paper presented at the proceedings of the IEEE/CVF international conference on computer vision workshops, Seoul Korea (South), pp 450–459
25. Yang J, An W, Yan C, Zhao P, Huang J (2021) Context-aware domain adaptation in semantic segmentation. Paper presented at the proceedings of the IEEE/CVF winter conference on applications of computer vision, *V*aikoloa, HI, USA, pp 514–524
26. Tsai YH, Hung WC, Schulter S, Sohn K, Yang MH, Chandraker M (2018) Learning to adapt structured output space for semantic segmentation. Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA, pp 7472–7481
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. Paper presented at the proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, pp 770–778
28. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal 40(4):834–848
29. Kingma DP, Ba J (2014). Adam: a method for stochastic optimization. http://arxiv.org/abs/1412.6980
30. Cherry JM, Adler C, Ball C, Chervitz SA, Dwight SS, Hester ET, Jia Y, Juvik G, Roe T, Schroeder M (1998) SGD: saccharomyces genome database. Nucleic Acids Res 26(1):73–79
31. Zhuang X (2018) Multivariate mixture model for myocardial segmentation combining multi-source images. IEEE Trans Pattern Anal Mach Intell 41(12):2933–2946
32. Zhuang X, Shen J (2016) Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI. Med Image Anal 31:77–87
33. Campello VM, Gkontra P, Izquierdo C, Martín-Isla C, Sojoudi A, Full PM, Maier-Hein K, Zhang Y, He Z, Ma J (2021) Multi-centre, multi-vendor and multi-disease cardiac segmentation: the M&Ms challenge. IEEE Trans Med Imaging 40(12):3543–3554
34. Chen C, Dou Q, Chen H, Qin J, Heng P-A (2019) Synergistic image and feature adaptation: towards cross-modality domain adaptation for medical image segmentation. Paper presented at the proceedings of the AAAI conference on artificial intelligence, vol 33, no 01, pp 865–872
35. Vesal S, Ravikumar N, Maier A (2019) Automated multi-sequence cardiac MRI segmentation using supervised domain adaptation. Paper presented at the international workshop on statistical atlases and computational models of the heart, Shenzhen, China, pp 300–308
36. Hoffman J, Tzeng E, Park T, Zhu JY, Isola P, Saenko K, Efros A, Darrell T (2018) CyCADA: cycle-consistent adversarial domain adaptation. Paper presented at the international conference on machine learning, Vienna, Austria, pp 1989–1998
37. Chen X, Lian C, Wang L, Deng H, Kuang T, Fung S, Gateno J, Yap PT, Xia JJ, Shen D (2020) Anatomy-regularized representation learning for cross-modality medical image segmentation. IEEE Trans Med Imaging 40(1):274–285
38. Tao X, Wei H, Xue W, Ni D (2019) Segmentation of multimodal myocardial images using shape-transfer GAN. Paper presented at the international workshop on statistical atlases and computational models of the heart, Shenzhen, China, pp 271–279
39. Chen C, Ouyang C, Tarroni G, Schlemper J, Qiu H, Bai W, Rueckert D (2019) Unsupervised multi-modal style transfer for cardiac MR segmentation. Paper presented at the international workshop on statistical atlases and computational models of the heart, Shenzhen, China, pp 209–219
40. Wu F, Zhuang X (2020) CF distance: a new domain discrepancy metric and application to explicit domain adaptation for cross-modality cardiac image segmentation. IEEE Trans Med Imaging 39(12):4274–4285
41. Li H et al (2021) 3D IFPN: improved feature pyramid network for automatic segmentation of gastric Tumor. Front Oncol 11:1654
42. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. Paper presented at the international conference on medical image computing and computer-assisted intervention, Munich, Germany, pp 234–241
43. Li L, Zimmer VA, Ding W, Wu F, Huang L, Schnabel JA, Zhuang X (2020). Random style transfer based domain generalization networks integrating shape and spatial information. Paper presented at the international workshop on statistical atlases and computational models of the heart, Lima, Peru, pp 208–218
44. Carscadden A, Noga M, Punithakumar K (2020) A deep convolutional neural network approach for the segmentation of cardiac structures from MRI sequences. Paper presented at the international workshop on statistical atlases and computational models of the heart, Lima, Peru, pp 250–258
45. Scannell CM, Chiribiri A, Veta M (2020) Domain-adversarial learning for multi-centre, multi-vendor, and multi-disease cardiac mr image segmentation. Paper presented at the international workshop on statistical atlases and computational models of the heart, Lima, Peru, pp 228–237
46. Full PM, Isensee F, Jäger PF, Maier Hein K (2020) Studying robustness of semantic segmentation under domain shift in cardiac MRI. Paper presented at the international workshop on statistical atlases and computational models of the heart, Lima, Peru, pp 238–249
47. Isensee F, Jaeger PF, Kohl SA, Petersen J, Hein KHM (2021) nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat Methods 18(2):203–211

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.