Unsupervised Domain Adaptation in Brain Lesion Segmentation with Adversarial Networks

Konstantinos Kamnitsas[1,4(✉)], Christian Baumgartner[1], Christian Ledig[1], Virginia Newcombe[2,3], Joanna Simpson[2], Andrew Kane[2], David Menon[2,3], Aditya Nori[4], Antonio Criminisi[4], Daniel Rueckert[1], and Ben Glocker[1]

[1] Biomedical Image Analysis Group, Imperial College London, London, UK
konstantinos.kamnitsas12@imperial.ac.uk
[2] Division of Anaesthesia, Department of Medicine, Cambridge University, Cambridge, UK
[3] Wolfson Brain Imaging Centre, Cambridge University, Cambridge, UK
[4] Microsoft Research Cambridge, Cambridge, UK

Abstract. Significant advances have been made towards building accurate automatic segmentation systems for a variety of biomedical applications using machine learning. However, the performance of these systems often degrades when they are applied on new data that differ from the training data, for example, due to variations in imaging protocols. Manually annotating new data for each test domain is not a feasible solution. In this work we investigate unsupervised domain adaptation using adversarial neural networks to train a segmentation method which is more robust to differences in the input data, and which does not require any annotations on the test domain. Specifically, we derive domain-invariant features by learning to counter an adversarial network, which attempts to classify the domain of the input data by observing the activations of the segmentation network. Furthermore, we propose a multi-connected domain discriminator for improved adversarial training. Our system is evaluated using two MR databases of subjects with traumatic brain injuries, acquired using different scanners and imaging protocols. Using our unsupervised approach, we obtain segmentation accuracies which are close to the upper bound of supervised domain adaptation.

1 Introduction

Great advancements have been achieved in machine learning, particularly with supervised learning algorithms, reaching human-level performance on applications that a few years ago would be considered extremely challenging. However, a common assumption in machine learning is that training and test data are drawn from the same probability distribution [19]. Methods are trained on data from a *source domain* $D_S = \{\mathcal{X}_S, P(X_S)\}$, where $\mathcal{X}_S$ is a feature space,

K. Kamnitsas—Part of this work was carried on when KK was an intern at Microsoft Research.

M. Niethammer et al. (Eds.): IPMI 2017, LNCS 10265, pp. 597–609, 2017.
DOI: 10.1007/978-3-319-59050-9_47

$X_S = \{x_{S1}, \ldots, x_{Sn}\}, x_{Si} \in \mathcal{X}_S$ the data and $P(X_S)$ the marginal distribution that their features follow. In an image segmentation problem, for example, X_S could be samples (voxels or patches) from multi-spectral MR scans, $\mathcal{X}_S$ is the feature space defined by the available MR sequences and $P(X_S)$ is the distribution of intensities in the sequences. In the developing stage of a supervised algorithm, given corresponding ground truth labels $Y_S = \{y_{S1}, \ldots, y_{Sn}\}, y_{Si} \in \mathcal{Y}_S$, such as segmentation masks, where $\mathcal{Y}_S$ the label space, a predictive function $f_S(x) = P_S(y|x)$ is learnt via training and configuration of hyper-parameters on the data (X_S, Y_S). $f_S(\cdot)$ tries to approximate the optimal function $f'_S(x), x \in \mathcal{X}_S$ that generated Y_S. At the time of deployment, however, these methods often under-perform or fail if the testing data come from a different *target domain* $D_T = \{\mathcal{X}_T, P(X_T)\}$, with $\mathcal{X}_T \neq \mathcal{X}_S$ and/or $P(X_T) \neq P(X_S)$. This is because the optimal predictive function $f'_T(x), x \in \mathcal{X}_T$ for D_T may differ from $f'_S(\cdot)$, and so the learnt $f_S(\cdot)$ will not perform well on D_T. The above scenario is common in biomedical applications due to variations in image acquisition, in particular, in multi-center studies. Training and testing data may differ in contrast, resolution, noise levels ($P(X_T) \neq P(X_S)$) or even type of sequences ($\mathcal{X}_T \neq \mathcal{X}_S$). Despite the rapid advancements in representation learning, this issue has been shown to affect even the latest models [18]. Generating labelled databases is time consuming and often expensive, and assuming annotations for training are available for each new domain is neither realistic nor scalable. Instead, it is desired to develop methods that can learn from existing databases and generalize well or adapt to the target domain without the need for additional training data.

Transfer learning (TL) [14] investigates development of predictive models by leveraging knowledge from potentially different but related domains and tasks. Even between tasks where label spaces $\mathcal{Y}_S$ and $\mathcal{Y}_T$ differ, TL can take advantage of similarities in the underlying structure of the mappings $f_S : \mathcal{X}_S \mapsto \mathcal{Y}_S$ and $f_T : \mathcal{X}_T \mapsto \mathcal{Y}_T$. A subclass of TL is *multi-task learning*, where a model is trained on multiple related tasks *simultaneously*. Most related to our work, *domain adaptation* (DA) is the subclass of TL that assumes $\mathcal{Y}_S = \mathcal{Y}_T$ and only the domains differ. It explores learning a function $f_a(\cdot)$ that performs well on both domains, under the basic assumption that such a function exists [1].

In this work we investigate *unsupervised domain adaptation* (UDA) [7]. In this setting we assume the availability of a labeled database $S = (X_S, Y_S)$ from source domain D_S, along with an *unlabeled* database $T = (X_T)$ from a different but related target domain D_T. We wish to model the unknown optimal function $f'_T(\cdot)$ for labelling X_T. However since no labels are available for D_T, $f'_T(\cdot)$ cannot be learnt. This is in contrast to supervised DA, which requires at least some labelled data for D_T. Instead, we try to learn a representation $h_a(x)$ that maps X_S and X_T to a feature space that is invariant to differences between the two domains, as well as a function $f_{ah}(\cdot)$ learnt using data $\{X_S, Y_S, X_T\}$, such that $f_a(x) = f_{ah}(h_a(x))$ approximates $f'_S(\cdot)$ and is closer to $f'_T(\cdot)$ than any function $f_S(\cdot)$ that can be learnt using only the source data (X_S, Y_S).

Contributions: In this work we develop a domain adaptation method based on adversarial neural networks [4,5]. We propose the adversarial training of a

segmenter and a domain-classifier, which aims to make the representation learnt by the segmenter invariant to domain-specific factors. We describe and analyse the development of domain-adversarial networks for the purpose of segmentation, which to the best of our knowledge has not been previously performed. We investigate the adaptation of layers at various depths and propose multi-connected adversarial networks, which we show improve domain adaptation. We employ our system for the segmentation of traumatic brain injuries (TBI), investigating adaptation between databases acquired using two different scanners with difference in the available MR sequences. We show that without utilizing any labels in the target domain, our method closes the performance gap with respect to supervised learning with target labels to a large extent.

Related Work: TL and DA have attracted significant interest over the years. Comprehensive reviews of early works can be found in [1,7,14]. Popularity of TL increased with the wide adoption of neural networks when their features were found to be effective when transferred across tasks. For example, features learnt from natural images were used off-the-shelf for detecting peri-fissural nodules [3]. More commonly, TL is performed via pre-training on a source task, followed by fine-tuning for the target task via supervised training [16]. A representative example of TL via multi-task learning was presented in [12]. A network was trained simultaneously for segmentation of brain tissue, pectoral muscle and coronary arteries. These experiments show that much of a network's capacity can be shared between a variety of tasks. Note, all of the above require labels in D_T.

In contrast, DA explores the case where label spaces (Y_S, Y_T) are the same and little or no labelled data is available in D_T. In [13] the authors explored supervised DA with SVM-based adaptive classifiers in the scenario where source and target data are acquired with different protocols. This method, however, requires labelled target data. Unsupervised DA was tackled in [6] via instance weighting, but this relies on strong assumptions about the data distributions. [2] performed UDA with boosted decision stumps with a search for visual correspondences between source and target samples. This is not as flexible as our approach nor scales well to large databases. The authors in [2] question the feasibility of DA with neural networks on 3D data due to memory requirements. Here, we show that using adversarial 3D networks is indeed a viable approach.

2 Unsupervised Domain Adaptation with Adversarial Nets

The accuracy of a binary classifier that distinguishes between samples from two domains can serve as a proxy of the divergence of distributions $P(X_S)$ and $P(X_T)$, which otherwise is not straightforward to compute. This idea was first introduced in [1]. Inspired by this, the authors of [4] presented a method for simultaneously learning a domain-invariant representation and a task-related classifier by a single neural network. This is done by minimizing the accuracy of

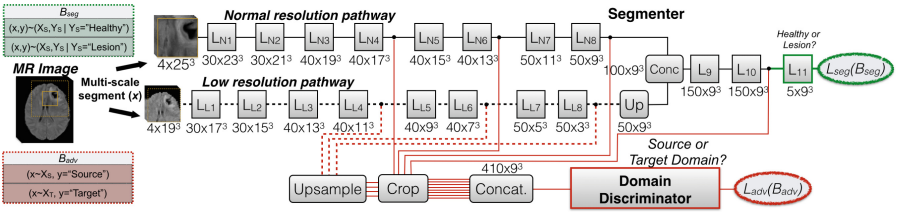

Fig. 1. Proposed multi-connected adversarial networks. Segmenter: we use the 3D CNN architecture presented in [8]. Dashed lines denote low resolution features. Input samples are multi-modal, although not depicted. Discriminator: We use a second 3D CNN for classifying the domain of input x, by processing activations at multiple layers of the segmenter. Red lines show the path of the adversarial gradients, from L_{adv} back to the segmenter. See text for details on architecture. (Color figure online)

an auxiliary network, a domain-discriminator, that processes a hidden representation of the main network and tries to classify the domain of the input sample. This approach formed the basis of our work. We below describe its extension for segmentation and our proposed multi-connected system.

2.1 Segmentation System with Domain Discriminator

Segmenter: At the core of our system is a fully convolutional neural network (CNN) for image segmentation [10]. Given an input x of arbitrary size, which can be a whole image or a sub-segment, this type of network predicts labels for multiple voxels in x, one for each stride of the network's receptive field over the input. The parameters of the network θ_{seg} are learnt by iteratively minimizing a segmentation loss $\mathcal{L}_{seg}$ using stochastic gradient descent (SGD). The loss is commonly the cross-entropy of the predictions on a training batch $B_{seg} = \{(x_1, y_1), \ldots, (x_{N_{seg}}, y_{N_{seg}})\}$ of N_{seg} samples. In our settings, (x_i, y_i) are sampled from the source database $S = (X_S, Y_S)$, for which labels Y_S are available. We borrowed the 3D multi-scale CNN architecture from [8], the segmenter depicted in Fig. 1, and adopt the same configuration for all meta-parameters.

Domain Discriminator: When processing an input x, the activations of any feature map (FM) in the segmenter encode a hidden representation $h(x)$. If samples come from different distributions $P(X_S) \neq P(X_T)$, e.g. due to different domains, and the filters of the segmenter are not invariant to the domain-specific variations, the distributions of the corresponding activations will differ as well, $P(h(X_S)) \neq P(h(X_T))$. This is expected when the segmenter is trained only on samples from S where learnt features will be specific to the source domain. Similar to [4], we choose a certain representation $h_a(x)$ from the segmenter and use a second network as a domain-classifier that takes $h_a(x)$ as input and tries to classify whether it comes from $P(h_a(X_S))$ or $P(h_a(X_T))$. This is equivalent to classifying the domain of x. Classification accuracy serves as an indication of how source-specific the representation $h_a(\cdot)$ is. The architecture we use for a domain

classifier is a 3D CNN with five layers. The first four have 100 kernels of size 3^3. The last classification layer uses 1^3 kernels. This architecture has a receptive field of 9^3 with respect to its input $h_a(\cdot)$ and was chosen for compatibility with the size of feature maps in the 3 last layers of the segmenter.

We train this domain-discriminator simultaneously with the segmenter. For this, we form a second training batch $B_{adv} = \{(x_1, y_1^d), \ldots, (x_{N_{adv}}, y_{N_{adv}}^d)\}$. Equal number of samples x_i are extracted from X_S and X_T, so there is no bias towards either. y_i^d is a label that encodes the domain of x_i, used as the training target. B_{adv} is processed by the segmenter, at the same time with B_{seg} or interleaved to lower memory requirements, computing activations $h_a(x) \forall x \in B_{adv}$. These activations are then processed by the discriminator, which classifies the domain of each sample in B_{adv}. The discriminator's classification loss $\mathcal{L}_{adv}$ is minimized through optimization of the parameters θ_{adv}.

A complication arises for the joint training. The samples from S are shared in an SGD iteration for the two losses in the algorithm of [4]. However, many segmentation methods use weighted sampling in order to mitigate class-imbalance, for example by oversampling rare classes [8,12]. Such sampling requires segmentation masks that are not available for T whose samples are extracted randomly. In this case, the discriminator should not compare those against non-randomly extracted samples from S, as it could easily associate activations for the overweighted classes with domain S and fail to learn useful domain-discriminative features. Hence, we resort to forming entirely separate batches. B_{adv} is formed of 20 image segments, randomly extracted from images in S and T. As done in [8], weighted sampling is used for extracting 10 segments from S to form B_{seg}. This ensures countering of class-imbalance for the segmenter, while being unbiased on the samples used for the discriminator.

Domain Adaptation via Adversarial Training: We aim at adapting the representation $h_a(\cdot)$ to become invariant to variations between S and T. To this end, we expose the accuracy of the domain-discriminator to the segmenter and let it alter its parameters such that its FMs that comprise $h_a(\cdot)$ do not contain cues about the input domain. This is done by incorporating the domain-discriminator's loss $\mathcal{L}_{adv}$ into the training objective of the segmenter, which now aims to simultaneously maximize the domain classification loss and minimize the segmentation loss $\mathcal{L}_{seg}$, or:

$$\mathcal{L}_{segAdv}(\theta_{seg}) = \mathcal{L}_{seg}(\theta_{seg}) - \alpha \mathcal{L}_{adv}(\theta_{seg}) \tag{1}$$

α is a positive weight that defines the relative importance of the domain adaptation task for the segmenter. This optimization is possible with regular SGD, as the adversarial networks are interconnected and gradients of $\mathcal{L}_{adv}$ can propagate back through the discriminator and into the segmenter. This process was implemented in [4] via a custom *gradient-reversal layer*, which is not needed if the optimization is formulated as in Eq. (1), as also noted by the authors.

2.2 Multi-connected Adversarial Networks

A natural question to arise concerns which layer(s) of the segmenter should be adapted. In [17], the authors investigated which of the last three fully connected layers of an AlexNet leads to better accuracy when adapted, concluding it is the last hidden layer that is optimal in their settings. Earlier layers are commonly not adapted as their features are considered rather generic and transferable across related tasks [4,11].

We argue that adapting only the last layers might not be ideal, especially for the case of segmentation. The accuracy of classification networks depends mostly on high-level patterns. For precise segmentation, however, fine patterns such as detailed texture and small contrast variations are likely to be important. These fine patterns are extracted in early layers and are more susceptible to image-quality variations between domains. Adapting top layers makes them invariant to such variations, but it is still a loss of capacity if such features have been already extracted by early layers, which may not be well adapted by the weakened adversarial gradients that reach them. On the other hand, if only early layers are adapted, assuming that the adaptation is not ideal and the features not entirely free of factors of variation between the two domains, the network could recover source-specific patterns at greater depth. For these reasons we propose an architecture where the domain discriminator is connected at multiple layers of the segmenter. First, this removes source-specific patterns early on but also disallows their recovery at deeper layers. Furthermore, the discriminator is enabled to process a large variety of features for discriminating between the domains, increasing its performance and thus the quality of the gradients for the domain adaptation. Finally, by seeing the whole adversarial network as an auxiliary cost function for the segmenter, this type of connections can be compared with deep-supervision [9], which allows better flow of the gradients incoming from $\mathcal{L}_{adv}$ throughout the segmenter and as such can improve learning of quality features. Our main results are based on feeding input $h_{in}(\cdot)$ to the discriminator from FMs of layers 4,6 and 8 of both high and low resolution pathways, as well as the 10-th hidden layer of the segmenter (cf. Fig. 1). After the FMs of the low resolution pathway are upsampled, all FMs are cropped to match the size of the deepest layer and concatenated. A detailed analysis of the effect of adapting different layers is presented in Sect. 3.4.

3 Experiments

3.1 Material

We make use of two databases with multi-spectral MR brain scans of patients with moderate to severe TBI, acquired within the first week of injury. The first database consists of 61 subjects, imaged on a 3-T Siemens Magnetom TIM Trio. The MR sequences are isotropic MPRAGE ($1\,\text{mm}^3$), axial FLAIR, T2 and Proton Density (PD) ($0.7 \times 0.7 \times 5\,\text{mm}$), and Gradient-Echo (GE) ($0.86 \times 0.86 \times 5\,\text{mm}$). The second database consists of 41 subjects, imaged on

a 3-T Siemens Magnetom Verio. This database includes MPRAGE, FLAIR, T2 and PD sequences, acquired at the same resolution as in the first database. The important difference is that instead of GE, a Susceptibility Weighted Image (SWI) is available ($0.7 \times 0.7 \times 5\,\text{mm}$). On both databases, all visible lesions were manually annotated on the FLAIR and GE/SWI by clinical experts. We merge them into a single lesion mask, as we here focus on binary segmentation of abnormalities within the brain tissue. Extra-cerebral pathologies are treated as background. All images are skull-stripped, resampled to isotropic $1\,\text{mm}^3$ and affinely registered to MNI space. Image intensities under the brain masks are normalized to zero-mean and unit-variance, after windowing the lowest and top 2% of the intensity histograms.

Source (S) and Target (T) Databases: GE and SWI are commonly used in TBI studies due to their great sensitivity to haemorrhages. They enable detection of lesions invisible in other sequences, such as micro-bleeds. SWI is actually a type of GE that offers greater sensitivity and image quality [15]. See Fig. 2 for visual examples. For the purpose of this study, the first database, with GE available, is considered the *source* database S used to train the segmenter in a supervised manner. The second database, with SWI available, is considered the *target* database T on which we aim to successfully apply the trained segmenter. This corresponds to a typical scenario where a training database is generated on data coming from one clinical site, and new test data coming from another site with varying protocol. Motivated by their common property of being sensitive to blood and thus providing similar information for TBI segmentation, we consider GE and SWI as interchangeable for the same input channel to our system, unless stated otherwise. However the difference in appearance of GE and SWI images (cf. Fig. 2) contributes the largest variation between distributions $P(X_S)$ and $P(X_T)$. Further variations may be present due to the different scanners used for acquiring S and T. Using our method, we aim to learn features invariant to these domain differences without the need for any annotations on the target domain.

3.2 Configuration of the Training Schedule

A complication of adversarial training concerns the training schedule of the two connected networks, which influences the way they interact. The strength with which the segmenter is adapting its features in order to counter the domain-discriminator is controlled by the parameter α (cf. Eq. (1)). We set $\alpha = 0$ for the first $e_1 = 10$ epochs and let both networks learn independently. This allows the segmenter to initially learn features for the segmentation of S without being influenced by noisy adversarial gradients from an initially poorly performing domain-discriminator. After epochs e_1, when the discriminator's performance has increased, we start countering it to learn domain invariant features with the segmenter. For this, we increase α according to the linear schedule $\alpha = \alpha_{max} \frac{e_{curr} - e_1}{e_2 - e_1}$, where $e_2 = 35$ and α_{max} is the maximum weighting, so α equals α_{max} after epoch e_2. Finally, at epoch 43 we start refining the segmenter's features by gradually lowering its learning rate. The discriminator is optimized

with constant learning rate 0.001. In the following, $\alpha_{max} = 0.05$ is used. In Sect. 3.4 we present a sensitivity analysis showing robust behaviour across a range of values for α_{max}. e_1, e_2 and the total duration of this piecewise linear schedule were determined empirically for satisfactory convergence without prolonging training time. Optimal settings are not fully explored yet and may vary between applications and the relative difficulty of each network's specific task.

3.3 Evaluation

We performed multiple experiments to obtain upper and lower bounds of baseline accuracy on the challenging task of TBI segmentation. We discuss experiments below, summarize results in Table 1 and give examples of segmentations in Fig. 2.

Table 1. Comparison of our method's performance on T with several baselines. Our system significantly closes the gap between the lower bound, when the segmenter is trained on S only, and the upper bound, when the segmenter is also trained with labelled data from T. Values are given in format *mean* (*std*).

	DSC	Recall	Precision
Train on S	15.7(13.5)	80.4(12.3)	09.5(09.0)
Train on S (No GE/SWI)	59.7(22.1)	55.7(22.6)	69.7(21.5)
Train on S → UDA to T (ours)	62.7(19.8)	58.9(21.2)	71.6(18.4)
Train on T	63.5(20.2)	60.6(21.1)	71.5(19.8)
Train on S+T	66.5(17.7)	66.6(19.1)	69.4(19.0)
Train on S+T (GE/SWI diff chan.)	64.7(19.2)	65.7(20.2)	67.0(20.8)

Train on S, Test on T: We perform standard supervised training of the segmenter on S without adaptation. To segment T, motivated by the similarity between GE and SWI sequences, at test time we use SWI in the channel used for GE during training. Even though these sequences can serve similar purposes in the analysis of TBI by radiologists, this approach totally fails, proving them not directly interchangeable as input to a CNN.

Train on S (No GE/SWI), Test on T: We repeat the previous experiment but only use the common sequences of S and T in both training and testing, neglecting GE and SWI. The experiment was repeated twice to reduce random variations between training sessions. This corresponds to a practical scenario, where we need to segment T by only using annotated training data from S, and serves as the *lower bound* of accuracy for our system.

Train on T, Test on T: We perform a 2-fold validation using supervised training on half of T and testing on the other half. We use all sequences of T. The obtained performance is similar to what was reported in [8], although on a

different database. This experiment provides another indication for the expected accuracy on this challenging segmentation task.

Train on S and T, Test on T: To obtain an *upper bound* of accuracy, we train the segmenter on all data of S and half the data of T, using their manual annotations. The same input channel is used for GE of S and SWI of T. We then test on the other half of data from T. The experiment is repeated for the other split of T. We balance the samples from the two domains in each batch B_{adv} to avoid biasing the segmenter towards S that has more subjects. With supervised training on T, the system learns to interchange GE and SWI successfully. This setting uses all available data from both domains, both images and manual annotations, and serves as an estimate of optimal, supervised transfer learning.

Train on S and T, Test on T (GE/SWI in Different Channels): We perform a sanity check that using GE and SWI in the same input channel is reasonable. We repeat the previous experiment but using a CNN with six channels, with separate ones for GE and SWI. The channel is filled with -4 when the sequence is not available, which corresponds to a very low value after our intensity normalization. From this the CNN learns when the sequence is missing and we found this to behave better than common zero-filling. The segmenter performs better than supervised training on T only. This indicates that information from both domains is used. However, knowledge transfer is not as strong as when GE and SWI, which share much information, are used in the same channel.

Proposed Unsupervised Domain Adaptation: We train the segmenter on all data of S and adapt the domains using half the subjects of T, but no labels. GE and SWI share the same input channel. We test accuracy on the other half of T. The experiment is repeated for the other fold. Our method learns filters invariant to the two imaging protocols and transfers knowledge from S to T, allowing the system to segment haemorrhages only visible on SWI without ever seeing a manual annotation from T (Fig. 2). This improves by 3% DSC over the non-adapted segmenter that uses only information from S and the common sequences, covering 44% of the difference between this practical lower bound and the upper bound achieved by supervised training with labels from both domains.

3.4 Analysis of the System

Effect of Adapting Layers at Different Depths: To investigate how depth of adapted layers affects our system, we repeat the experiment with domain adaptation from S to T, changing the layers at which the adversarial networks are connected. Results are shown on Fig. 3 and Table 2. Note that connections are added to both pathways of the segmenter at the same depth (for example, $L4$ means connections to the 4th layers of both pathways). Adapting shallow layers tends towards over-segmentation (increased recall but lower precision). It has been noticed that severe over-segmentation occurs without adaptation (Fig. 2). These observations indicate that source-specific features are possibly recovered between the adapted and the classification layer. Comparing $L2$ and

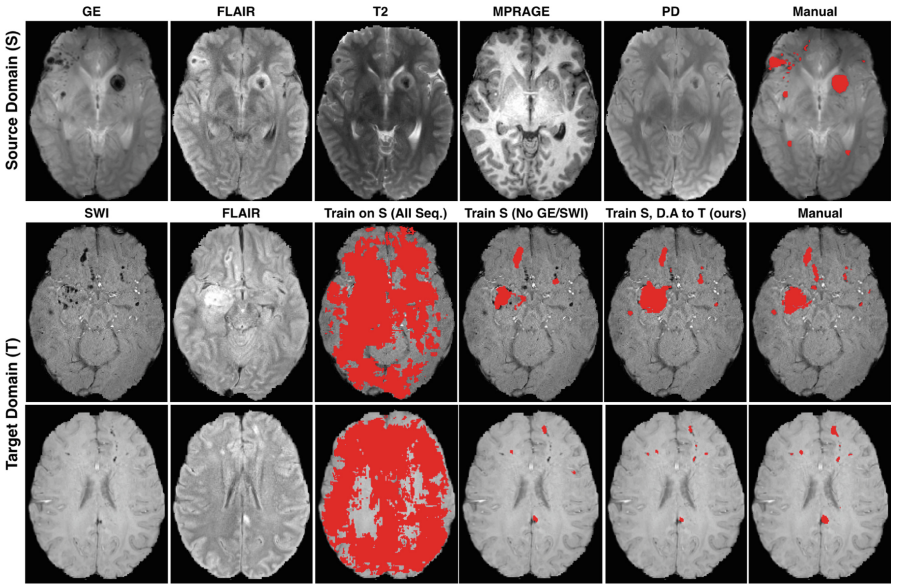

Fig. 2. (top) Example case from S. (middle/bottom) SWI and FLAIR of two subjects from T (T2, MPRAGE, PD also used but not shown). Notice that only GE and SWI show certain lesions, such as micro-bleeds. However, brain tissue appears differently in GE and SWI. Consequently, a model trained on S fails on T when SWI is naively used in place of GE (3rd col.). A model trained using only the four common sequences misses lesions visible only on SWI (4th col.). Our method mitigates these problems by learning features invariant to the imaging protocol (5th col.).

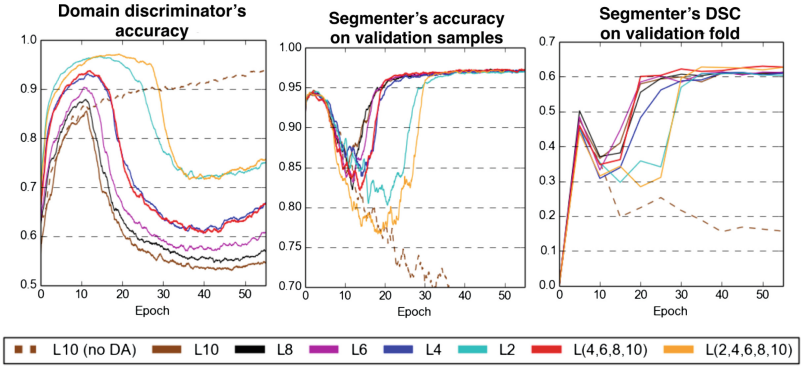

Fig. 3. Behaviour when the domain-discriminator is connected at different layers of the segmenter. Adaptation is performed after epoch 10 by linearly increasing α. Connections at earlier layers lead to higher performance of the discriminator but slower adaptation. Multiple connections increase performance. Note, features learnt at early layers during the refinement in the last stages of training seem more domain-discriminative.

Table 2. Final accuracy on T when the discriminator is connected at different depths of the segmenter. Shallow connections increase recall but significantly decrease precision. Multiple connections remove better the source-specific nuisances throughout the segmenter, closing the gap to the practical upper bound of 66.5% for UDA (Sect. 3.3) by approximately 1.5% DSC. Best configuration in bold.

	L10	L8	L6	L4	L2	**L(4,6,8,10)**	L(2,4,6,8,10)
DSC	61.3(21.0)	61.0(20.7)	61.2(19.2)	61.0(20.1)	60.4(20.2)	62.7(19.8)	62.7(19.5)
Recall	56.9(22.0)	57.3(21.6)	57.1(19.8)	59.1(20.0)	61.1(20.5)	58.9(21.2)	60.1(20.3)
Precision	71.9(20.8)	70.2(20.9)	69.9(20.8)	68.1(21.6)	64.3(21.9)	71.6(18.4)	69.8(20.0)

$L(2, 4, 6, 8, 10)$ shows that this is alleviated by multiple connections that enforce domain invariance throughout the segmenter. Since, however, behaviour of multi-connected adversarials is strongly defined by the shallowest connection, we avoid adapting the earliest layers, which offer less benefit but slow down convergence.

Effect of Adaptation's Strength via α_{max}: Here we investigate how sensitive is our method to α_{max}, which defines how strongly the segmenter counters the discriminator. Figure 4 shows that higher values lead to quicker adaptation but the accuracy is rather stable for a significant range of values $\alpha_{max} \in [0.05, 1.0]$. We note this range might differ for other applications and that smooth convergence is generally preferred for learning high quality features over steep schedules that alter the loss surface aggressively. Finally, we observe that strongly countering the discriminator does not guarantee better performance on T. A theoretical reason is that a more domain-invariant representation $h_a(x)$ likely encodes less information about x. This information loss increases the Bayes error rate and the entropy of the predictions by the learnt $f_a(x) = f_{ah}(h_a(x))$. After a certain level of invariance, this can outweigh the benefits of domain-adaptation [1,7].

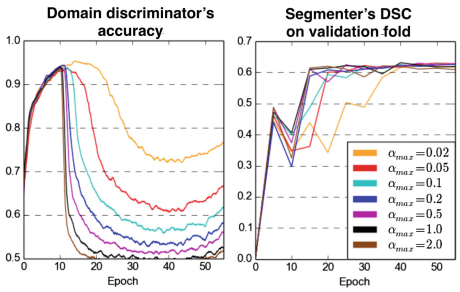

Fig. 4. The segmenter counters the domain-discriminator after epoch 10, when we linearly increase α from zero to α_{max} until epoch 35. Final accuracy on T was found rather stable for a wide range of values. Decrease greater than 1% DSC from the highest was found for values 0.02 and 2.0.

4 Conclusion

We present an unsupervised domain adaptation method for image segmentation based on adversarial training of two 3D neural networks. To the best of our knowledge this is the first work showing the plausibility and capabilities of such an approach on a biomedical imaging problem. Additionally, we propose multi-connected adversarial networks, which perform better by enabling flow of higher quality adversarial gradients throughout the adapted network. We investigate aspects of adversarial training such as the depth of the adapted layer and the strength of adaptation, providing valuable insights for development of future approaches. While unsupervised in the target domain, our method performs close to the accuracy of supervised baselines. We believe our work makes an important contribution in the context of multi-center studies where domain differences are a major limitation in current image analysis methods. Future work will investigate the capabilities of our approach to normalize different types of variations. An implementation of the proposed system will be made publicly available on https://biomedia.doc.ic.ac.uk/software/deepmedic/.

Acknowledgements. This work is supported by the EPSRC (grant No: EP/N023668/1) and partially funded by an European Union Framework Program 7 grant (CENTER-TBI; Agreement No: 60215). Part of this work was carried on when KK was an intern at Microsoft Research Cambridge. KK is also supported by the President's Ph.D. Scholarship of Imperial College London. VN is supported by an Academy of Medical Sciences/Health Foundation Clinician Scientist Fellowship. DM is supported by the Neuroscience Theme of the NIHR Cambridge Biomedical Research Centre and NIHR Senior Investigator awards. We gratefully acknowledge the support of NVIDIA Corporation with the donation of two Titan X GPUs.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. **79**(1–2), 151–175 (2010)
2. Bermúdez-Chacón, R., Becker, C., Salzmann, M., Fua, P.: Scalable unsupervised domain adaptation for electron microscopy. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 326–334. Springer, Cham (2016). doi:10.1007/978-3-319-46723-8_38
3. Ciompi, F., de Hoop, B., van Riel, S.J., Chung, K., Scholten, E.T., Oudkerk, M., de Jong, P.A., Prokop, M., van Ginneken, B.: Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. MedIA **26**(1), 195–202 (2015)
4. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(59), 1–35 (2016)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014)
6. Heimann, T., Mountney, P., John, M., Ionasec, R.: Learning without labeling: domain adaptation for ultrasound transducer localization. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8151, pp. 49–56. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40760-4_7

7. Jiang, J.: A literature survey on domain adaptation of statistical classifiers (2008). http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/da_survey.pdf
8. Kamnitsas, K., Ledig, C., Newcombe, V.F., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. MedIA **36**, 61–78 (2016)
9. Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: AISTATS, vol. 2, p. 6 (2015)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
11. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML (2015)
12. Moeskops, P., Wolterink, J.M., Velden, B.H.M., Gilhuijs, K.G.A., Leiner, T., Viergever, M.A., Išgum, I.: Deep learning for multi-task medical image segmentation in multiple modalities. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 478–486. Springer, Cham (2016). doi:10.1007/978-3-319-46723-8_55
13. van Opbroek, A., Ikram, M.A., Vernooij, M.W., De Bruijne, M.: Transfer learning improves supervised image segmentation across imaging protocols. TMI **34**(5), 1018–1030 (2015)
14. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
15. Shenton, M., Hamoda, H., Schneiderman, J., Bouix, S., Pasternak, O., Rathi, Y., Vu, M.A., Purohit, M., Helmer, K., Koerte, I., et al.: A review of magnetic resonance imaging and diffusion tensor imaging findings in mild traumatic brain injury. Brain Imaging Behav. **6**(2), 137–192 (2012)
16. Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. TMI **35**(5), 1285–1298 (2016)
17. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: maximizing for domain invariance. arXiv preprint (2014). arXiv:1412.3474
18. Ullman, S., Assif, L., Fetaya, E., Harari, D.: Atoms of recognition in human and computer vision. Proc. Nat. Acad Sci. **113**(10), 2744–2749 (2016)
19. Valiant, L.G.: A theory of the learnable. Commun. ACM **27**(11), 1134–1142 (1984). http://doi.acm.org/10.1145/1968.1972