Unlearning Scanner Bias for MRI Harmonisation in Medical Image Segmentation

Nicola K. Dinsdale[1](✉), Mark Jenkinson[1,2,3], and Ana I. L. Namburete[4]

[1] Wellcome Centre for Integrative Neuroimaging, FMRIB,
University of Oxford, Oxford, UK
nicola.dinsdale@dtc.ox.ac.uk

[2] Australian Institute for Machine Learning (AIML), Department of Computer Science, University of Adelaide, Adelaide, Australia

[3] South Australian Health and Medical Research Institute (SAHMRI),
North Terrace, Adelaide, Australia

[4] Institute of Biomedical Engineering, University of Oxford, Oxford, UK

Abstract. The combination of datasets is vital for providing increased statistical power, and is especially important for neurological conditions where limited data is available. However, our ability to combine datasets is limited by the addition of variance caused by factors such as differences in acquisition protocol and hardware. We aim to create scanner-invariant features using an iterative training scheme based on domain adaptation techniques, whilst simultaneously completing the desired segmentation task. We demonstrate the technique using an encoder-decoder architecture similar to the U-Net but expect that the proposed training scheme would be applicable to any feedforward network and task. We show that the network can be used to harmonise two datasets and also show that the network is applicable in the common scenario of limited available training data, meaning that the network should be applicable for real-world segmentation problems.

Keywords: Harmonisation · Joint domain adaptation · MRI

1 Introduction

Although a few large neuroimaging projects new exist, such as the UK Biobank [12], the majority of dataset sizes remain small; additionally, those with expert manual segmentations are even more limited due to its time consuming nature. Therefore, to achieve increased statistical power, it is vital to be able to combine data from multiple sites and scanners. This, however, leads to an increase in variance and bias in the data, driven by differences in acquisition protocol and hardware [4]. Data harmonisation is therefore required to allow joint unbiased analysis of data collected from different scanners at different sites.

B. W. Papież et al. (Eds.): MIUA 2020, CCIS 1248, pp. 15–25, 2020.
https://doi.org/10.1007/978-3-030-52791-4_2

ComBat [4] is a popular harmonisation method, which performs post-hoc normalisation using a linear model, to allow the image-derived values to be comparable between sites. This has then been extended in several ways including in [10] which incorporates a nonlinear model, and in [13] where the model is adapted explicitly to encode bias caused by nonbiological variance in the model. The majority of the other MRI harmonisation methods focus on image generation: given an image from a scanner they generate a set of images that appear to have been acquired from another scanner. Recent studies have used deep learning methods for this. U-Net style networks have been used such as [3], which learns features at different levels of abstraction to recreate images given paired training data. CycleGANS [17] have also been used, for example as in [16], transforming data between scanners in a cycle-consistent manner.

Rather than harmonising images, we propose to use a joint domain adaptation framework to harmonise the features extracted by a deep learning network where we consider each scanner to be a separate domain. If we consider the case where we have a source domain D_s and a target domain D_t with the same learning task T, then the success of the domain adaptation depends on the existence of a similarity between the two domains [14]. For harmonisation, we are considering the case where $D_s \neq D_t$ or in other words that the data have been collected on distinct scanners. One of the most successful methods for domain adaptation is DANN [6], which uses a gradient reversal layer [5] to allow adversarial training of a discriminator. This creates a feature representation that is discriminative for the main task but indiscriminate as to the domain from which the data originates. There is, however, little exploration of the effect of this domain adaptation on the network performance on the *source domain data*, whereas for successful data harmonisation it is vital that we create a network that performs well across all the source and target datasets.

In [7] a method is proposed for simultaneous domain and task adaptation. Similarly to DANN, domain adaptation is completed adversarially but, rather than using a gradient reversal layer to update the domain predictor in opposition to the task, they use an iterative training scheme, iterating between learning the best domain classifier for a given feature representation and minimising a confusion loss that aims to force the domain predictions to become closer to a uniform distribution. In this way, the network maximally confuses the domain classifier [7]. Compared to DANN-style networks, this method is better at ensuring that we achieve a classifier which is equally uninformative across the domains [2] because of the confusion loss, which is highly desirable for the harmonisation scenario. This learning scheme is applied in [2] where the iterative unlearning creates classifiers that are blind to spurious variations in the data - variations which are not directly related to the task of interest - but only aims to do this for a single dataset, whereas for harmonization it is vital to be able to consider larger numbers of datasets. Together, these form the inspiration for this work.

In this work, we apply a framework similar to that introduced in [7] for harmonisation within the setting of image segmentation. We do this by posing the problem as a joint domain adaptation problem. We aim to create a feature representation that is invariant to the scanner from which the data were acquired and show that this network is still able to segment the images successfully. We also explore the effect of our training scheme when very small amounts of labelled data are available, as this is a very realistic scenario for segmentation tasks. We show that scanner information can successfully be ‘unlearned’, and thus is not used to create the final segmentations, allowing us to harmonise data for the segmentation task.

2 Method

2.1 Standard Supervised Training

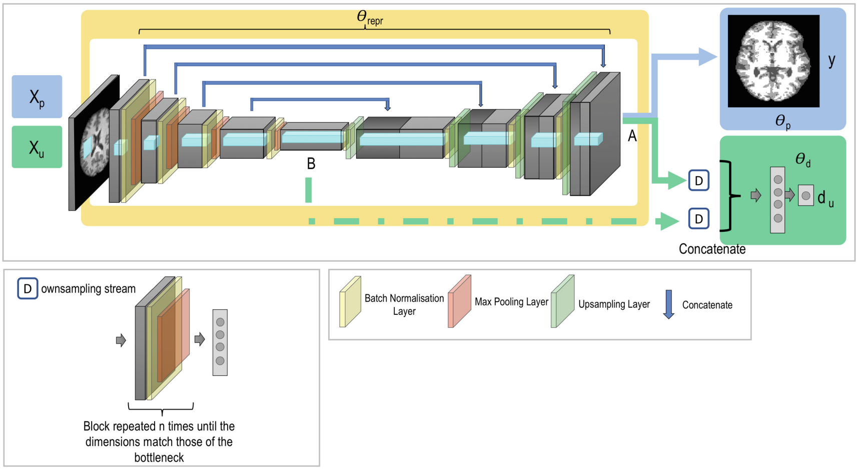


Fig. 1. Network architecture for segmentation task. $\boldsymbol{X}_p$ and $\boldsymbol{X}_u$ represent the input data for the network, where $\boldsymbol{X}_p$ is the input data used for training the main task and $\boldsymbol{X}_u$ is the input data for the unlearning iterations. These can either be the same data, subsets of each other, or different datasets, dependent on the labels that are available. For $\boldsymbol{X}_p$ the labels $\boldsymbol{y}_p$ are the main task labels - the segmentation labels - and for $\boldsymbol{X}_u$ the labels are the domain labels $\boldsymbol{d}_u$. $\boldsymbol{\Theta}_{repr}$ are the parameters of the convolutional layers in the encoder and decoder which form the U-Net architecture, $\boldsymbol{\Theta}_p$ are the parameters of the final convolutional layers that produce the segmentation, and $\boldsymbol{\Theta}_d$ are the parameters of the domain predictor layers. Unlearning is completed either from location A, location B or locations A and B in combination - that is the domain predictor is attached in these locations. If unlearning is completed from A and B together, the first fully connected layers are concatenated to produce a single feature representation.

Consider the training regime for which we have segmentation labels available for the data from all scanners and that the segmentation tasks are all the same.

In this case X_p and X_u form a singular dataset X which can be used to evaluate all the loss functions used in training.

The aim of the 2D network shown in Fig. 1 is to find a representation $\boldsymbol{\Theta}_{repr}$ that maximises the performance on the primary segmentation task while minimising the performance of a discriminator, which aims to predict the site of origin of the data. Although in this work we focus on segmentation, the training scheme should generalise to any feedforward architecture and task. $\boldsymbol{\Theta}_{repr}$ represents the features of the encoder-decoder network which are shared between the two output branches. $\boldsymbol{\Theta}_p$ are then the parameters for the primary segmentation task and $\boldsymbol{\Theta}_d$ are the parameters associated with the domain predictor branch. We consider trying to segment two datasets, each with input images $\boldsymbol{X} \in \mathbb{R}^{W \times H \times D \times 1}$ and task labels $\boldsymbol{y} \in \mathbb{R}^{W \times H \times D \times C}$ where C is the number of labels, with different domains d, representing scans acquired from two different distinct scanners.

To train the network, three loss functions are minimised iteratively. The first loss is the loss function for the main task and is conditioned on each domain - that is, evaluated separately for the data from each scanner:

$$L_p(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{d}; \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p) = \sum_{n=1}^{N} \frac{1}{S_n} \sum_{j=1}^{S_n} L_n(\boldsymbol{y}_{j,n}, \hat{\boldsymbol{y}}_{j,n}) \tag{1}$$

where N is the number of domains and S_n is the number of subjects from domain n such that $\boldsymbol{y}_{j,n}$ is the true label for the j^{th} subject from the n^{th} domain. This loss takes the form of the Sorensen-Dice loss function. The loss is calculated for each domain in turn, preventing the performance being driven by one dataset. This is especially vital if one dataset is significantly larger than the other. The domain information is then unlearned using a combination of two loss functions, which work in opposition to each other. The first is the domain loss, which is simply the categorical cross-entropy:

$$L_d(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_{repr}; \boldsymbol{\Theta}_d) = -\sum_{n=1}^{N} \mathbb{1}[d = n] log(p_n) \tag{2}$$

which assesses how much information remains in Θ_{repr} about the domains. p_n are the softmax outputs of the domain classifier and also used by the confusion loss to remove information. This is done by penalising deviations from a uniform distribution:

$$L_{conf}(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_d; \boldsymbol{\Theta}_{repr}) = -\sum_{n=1}^{N} \frac{1}{N} log(p_n) \tag{3}$$

Therefore, the overall method minimises the total loss function:

$$\begin{aligned} L(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{d}, \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_d) = & \; L_p(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{d}; \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p) \\ & + \alpha L_d(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_{repr}; \boldsymbol{\Theta}_d) \\ & + \beta L_{conf}(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_d; \boldsymbol{\Theta}_{repr}) \end{aligned} \tag{4}$$

where α and β represent weights of the relative contributions of the different loss functions. Equations (2) and (3) cannot be optimised in a single step because they act in direct opposition to each other: therefore, we update the loss functions iteratively. This results in three forward passes per batch.

$$\begin{aligned} L(\boldsymbol{X}_p, \boldsymbol{X}_u, \boldsymbol{y}_p, \boldsymbol{d}_p, \boldsymbol{d}_u, \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_d) &= L_p(\boldsymbol{X}_p, \boldsymbol{y}_p, \boldsymbol{d}_p; \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p) \\ &+ \alpha L_d(\boldsymbol{X}_u, \boldsymbol{d}_u, \boldsymbol{\Theta}_{repr}; \boldsymbol{\Theta}_d) \\ &+ \beta L_{conf}(\boldsymbol{X}_u, \boldsymbol{d}_u, \boldsymbol{\Theta}_d; \boldsymbol{\Theta}_{repr}) \end{aligned} \tag{5}$$

2.2 Semi Supervised Learning

We here consider the case where we have limited access to manual annotations for one of the scanners; this is a very likely scenario for segmentation, where manual labels are expensive to obtain. In addition to the small set of labelled data points, we assume access to more unlabelled examples which can be used for the domain unlearning. In most cases, domain labels are trivial and nearly always available.

No changes to the architecture are necessary: rather, we simply evaluate the equations for different subsets of the data. Equation (1) is now only evaluated for $\boldsymbol{X}_p$ and $\boldsymbol{y}_p$ where these are the data points for which we have main task labels. Equations (2) and (3) can still be evaluated for the full dataset and so the overall method minimises:

$$\begin{aligned} L(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{d}, \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_d) &= L_p(\boldsymbol{X}_p, \boldsymbol{y}_p, \boldsymbol{d}; \boldsymbol{\Theta}_{repr}, \boldsymbol{\Theta}_p) \\ &+ \alpha L_d(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_{repr}; \boldsymbol{\Theta}_d) \\ &+ \beta L_{conf}(\boldsymbol{X}, \boldsymbol{d}, \boldsymbol{\Theta}_d; \boldsymbol{\Theta}_{repr}) \end{aligned} \tag{6}$$

where $\boldsymbol{X}_p$ and $\boldsymbol{y}_p$ are subsets of $\boldsymbol{X}$ and $\boldsymbol{y}$.

2.3 The Location of the Domain Predictor

In addition to the effect of available data, we also need to consider the effect of the location of the domain predictor. It was hypothesised that the domain predictor at least needed to be connected after the last cross connection, otherwise the network might be able to learn features where domain information is present again, due to the skip connections from before the unlearning was completed. In [8] the effect of the location of domain adaptation is explored with relation to the quality of the segmentation. They suggest that by having the domain adaptation too early in the network, if the domain information is not entirely unlearned, the network is able to learn how to use the remaining information by the end of the network. Conversely, they argue that adapting layers to make them invariant to variations may lead to a reduction in performance because of the constraints on the features which are being learnt. For segmentation, later layers learn fine details which are vital to the performance of the segmentation and so it may be detrimental to performance to constrain these features too strongly.

We therefore compare the performance of the network and the degree of unlearning achieved by unlearning just after the final cross connection, at the bottleneck, and a combination of the two, as shown in Fig. 1. The domain predictor takes the form of a chain of convolutional blocks and max pooling layers until the dimensions match those of the bottleneck layer, at which point they are then connected to two fully connected layers to produce the final prediction. In the case of unlearning from two locations, the first fully connected layers are concatenated to produce a single shared set of features and a single domain prediction.

3 Experimental Setup

For the experiments in this work, T1 weighted MRI scans from two datasets were used. The first dataset was the UK Biobank dataset [12] which had been processed using the UK Biobank Pipeline [1] (2095 training, 937 testing); the other dataset comprised of the healthy subjects from the OASIS dataset [9] (813 training, 217 testing), at multiple time points, split into training and test sets at subject level. The input images for both of the datasets were resized to $128 \times 128 \times 128$, normalised to have zero mean and unit standard deviation, and then split into slices in the third dimension. The labels were obtained using FSL FAST [15] as a proxy for manual annotations and were converted into one-hot labels.

The network was implemented using Python 3.6 and PyTorch (1.0.1) and is based on the U-Net architecture [11]. The training regime should be applicable to any feedfoward network but we chose to investigate use with the U-Net as it is the most frequently used for medical image segmentation and has the added complication of the cross connections. The network has four downsampling and upsampling layers with each layer being formed of a convolutional layer, a ReLU activation function and a batch normalisation layer with the number of convolutions increasing as 8f where f is the depth. A batch size of 5 was used throughout, with each batch constrained to contain at least one example from each dataset, increasing training stability. To achieve this, the smaller dataset was oversampled. The parameters α and β were set experimentally using ten fold cross validation for the different experiments and took values of between 1 and 20.

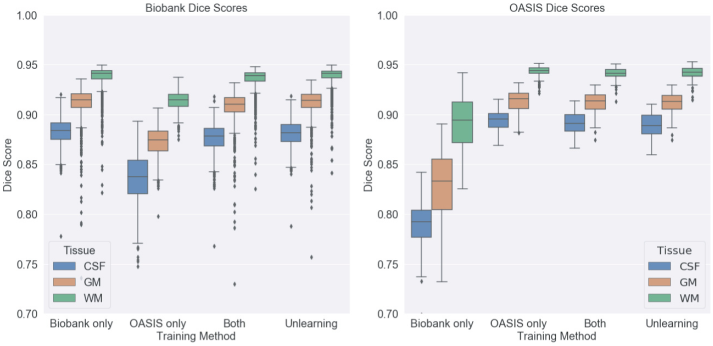
4 Results

4.1 Supervised Unlearning

We compared our method - with the domain predictor simply in location A - to standard training on both datasets individually and on the combination of the two datasets. The results can be seen in Table 1. Scanner classification accuracy was found by fixing the feature representation $\boldsymbol{\Theta}_{repr}$ and then training a domain predictor to classify the resulting features. A classifier with accuracy near random

Table 1. Dice scores comparing unlearning to training the network in different combinations on the datasets averaged across the tissue types. Scanner accuracy is the accuracy achieved by a domain predictor given the fixed feature representation at convergence. The number in brackets indicates random chance.

Training data	Biobank	OASIS	Scanner classification accuracy (%)
Biobank only	0.910 ± 0.022	0.836 ± 0.043	–
OASIS only	0.874 ± 0.032	0.917 ± 0.020	–
Both (normal training)	0.906 ± 0.024	0.915 ± 0.020	100 (50)
Both (unlearning)	0.910 ± 0.023	0.916 ± 0.021	51 (50)



(a) Biobank

(b) OASIS

Fig. 2. Dice scores for the different training methods split by tissue within dataset. CSF = Cerebral Spinal Fluid, WM = White Matter, GM = Grey Matter.

chance indicates that information about the scanner has been removed from the feature representation.

It can be seen, as would be expected, that training on both datasets gives the best overall performance for normal training compared to training on just a single dataset. It can also be seen that the performance of the network does not change significantly with the introduction of the unlearning. This is despite the fact that the domain predictor accuracy, given the frozen feature representation $\boldsymbol{\Theta}_{repr}$, has decreased from 100% before unlearning to 51% after unlearning where random chance is 50%. This shows that nearly all of the information about the scanner has been removed and the features which remain are almost entirely invariant to the scanner on which the data was acquired.

The results are broken down by tissue type in Fig. 2. It can be seen that the pattern is the same across tissue types. A representative example segmentation can be seen in Fig. 3.

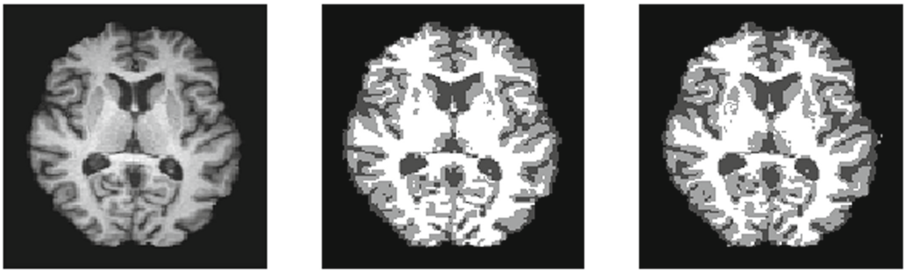


Fig. 3. Representative segmentation from the OASIS dataset. From L-R: T1 slice, FAST segmentation used as a proxy to a manual segmentation, output segmentation after unlearning.

4.2 Semi Supervised Results

To explore the effect of training the network with low numbers of labelled data points, the network was trained both with normal learning and with unlearning for increasing numbers of OASIS datapoints. It can be seen that unlearning gives large improvement in segmentation performance with low numbers of data points, not only in terms of average performance but also in terms of the consistency of the segmentation. The amount of improvement decreases as the number of training examples increases, as would be expected, but the unlearning never decreases the overall performance of the network. Therefore, it is evident that the network is effective in a likely scenario for medical image segmentation, where low amounts of labelled training data are available. The scanner classification accuracy was 100% for all cases of normal training and between 50 and 55% for the unlearning cases.

Considering the unsupervised case - where there are no training examples for OASIS - it can be seen that unlearning gives the biggest improvement in this scenario. This is because the method is in essence a domain adaptation approach, and so, a positive side effect of the harmonisation is that the network is able to learn features from the Biobank data which are generally useful and apply them to the OASIS data. The unlearning forces the features learnt not just to specialise to the Biobank data but to be more general to the two datasets. Therefore, the training regime could be applied for harmonisation of segmentation tasks even when there are no available labels for one of the datasets (Fig. 4).

4.3 The Effect of the Location of the Domain Predictor

The domain predictor was attached to the bottleneck (B) and after the final convolution (A). The effect of unlearning from these two locations and the combination was considered. The results can be seen in Table 2.

Firstly, it can be seen that unlearning only at the bottleneck (B) does not affect the ability of a separate domain predictor located after the final convolution, which has access to the features that are used to create the final segmentations, to predict the scanner the data came from. This is as would be

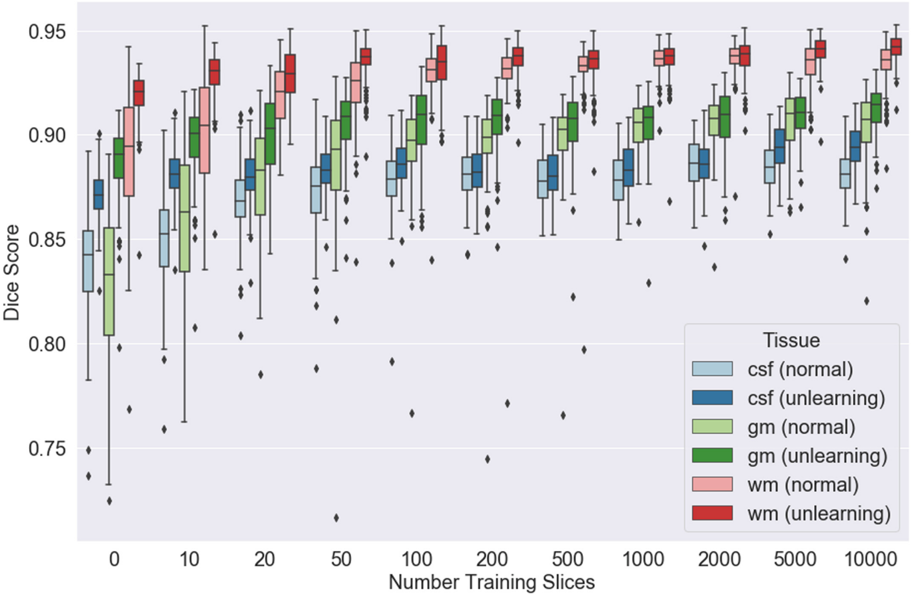


Fig. 4. Dice scores for the three different tissue types for the OASIS data with increasing numbers of OASIS training slices, comparing both normal training and unlearning. Note that for clarity of plotting the x axis is not to scale. The full Biobank dataset was used throughout.

Table 2. Dice scores comparing unlearning at different locations in the network: A) At the final convolutional layer, B) At the bottleneck, A + B) Combination of the two, as shown in Fig. 1. The scanner classification accuracy was the accuracy achieved by a separate domain predictor using the fixed feature representation at the final convolutional layer. Random chance is given in brackets.

Location of domain predictor	Biobank	OASIS	Scanner classification accuracy (%)
Final convolution (A)	0.910 ± 0.023	0.916 ± 0.021	51 (50)
Bottleneck (B)	0.871 ± 0.046	0.882 ± 0.030	100 (50)
Both (A + B)	0.903 ± 0.025	0.912 ± 0.021	55 (50)

expected because the skip connections will mean that all the domain information is still available to the domain predictor. Adding the unlearning branch to the bottleneck also has a detrimental effect on the performance of the segmenter, indicating that it constrains the features too much, so that the network is not able to perform well. It also caused the training of the network to be much less stable. The combination of the bottleneck and the final convolution (A + B) allows the network to create far more scanner-invariant features but the increased constraint on the features that the network can learn still leads to a decrease in performance. Simply unlearning at the end of the network seems to be sufficient to unlearn scanner information whilst limiting the constraint on the features that the network can learn to complete the segmentation task.

5 Discussion

In this work, we have shown that an iterative training scheme can be used to 'unlearn' scanner information, allowing us to create features from which we have removed most scanner information whilst successfully completing the segmentation task. We have also shown that the training scheme not only works but also gives us improved performance when we have low amounts of available training data. The training regime is flexible and applicable to any feedforward network and so could be applied to many segmentation tasks, even when there is limited manual segmentation for a site. We have also shown that for the most commonly used segmentation network architecture, the domain unlearning should be completed from the end of the network, not the bottleneck as might have been expected.

Acknowledgements. ND is supported by the Engineering and Physical Sciences Research Council (EPSRC) and Medical Research Council (MRC) [grant number EP/L016052/1]. MJ is supported by the National Institute for Health Research (NIHR) Oxford Biomedical Research Centre (BRC), and this research was funded by the Wellcome Trust [215573/Z/19/Z]. The Wellcome Centre for Integrative Neuroimaging is supported by core funding from the Wellcome Trust [203139/Z/16/Z]. AN is grateful for support from the UK Royal Academy of Engineering under the Engineering for Development Research Fellowships scheme.

The computational aspects of this research were supported by the Wellcome Trust Core Award [Grant Number 203141/Z/16/Z] and the NIHR Oxford BRC. The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

References

1. Alfaro-Almagro, F., et al.: Image processing and quality control for the first 10,000 brain imaging datasets from UK biobank. bioRxiv 166, 130385, April 2017
2. Alvi, M., Zisserman, A., Nellåker, C.: Turning a blind eye: explicit removal of biases and variation from deep neural network embeddings. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11129, pp. 556–572. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11009-3_34
3. Dewey, B., et al.: DeepHarmony: a deep learning approach to contrast harmonization across scanner changes. Magn. Reson. Imaging **64**, 160–170 (2019)
4. Fortin, J.P., et al.: Harmonization of cortical thickness measurements across scanners and sites. NeuroImage **167** (2017). https://doi.org/10.1016/j.neuroimage.2017.11.024
5. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. ArXiv (2014)
6. Ganin, Y., et al.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 59:1–59:35 (2015)

7. Hoffman, J., Tzeng, E., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4068–4076 (2015)
8. Kamnitsas, K., et al.: Unsupervised domain adaptation in brain lesion segmentation with adversarial networks (12 2016)
9. Marcus, D., Wang, T., Parker, J., Csernansky, J., Morris, J., Buckner, R.: Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. J. Cogn. Neurosci. **19**, 1498–507 (2007). https://doi.org/10.1162/jocn.2007.19.9.1498
10. Pomponio, R., et al.: Harmonization of large MRI datasets for the analysis of brain imaging patterns throughout the lifespan. NeuroImage **208**, 116450 (2019). https://doi.org/10.1016/j.neuroimage.2019.116450
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
12. Sudlow, C., et al.: UK Biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age 12, e1001779, March 2015
13. Wachinger, C., Rieckmann, A., Pölsterl, S.: Detect and correct bias in multi-site neuroimaging datasets. bioRxiv, February 2020
14. Wilson, G., Cook, D.J.: A survey of unsupervised deep domain adaptation (2018)
15. Zhang, Y., Brady, M., Smith, S.: Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm. IEEE Trans. Med. Imaging **20**, 45–57 (2001). https://doi.org/10.1109/42.906424
16. Zhao, F., et al.: Harmonization of infant cortical thickness using surface-to-surface cycle-consistent adversarial networks. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11767, pp. 475–483. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32251-9_52
17. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251 (2017)