Transfer learning improves supervised image segmentation across imaging protocols

van Opbroek, Annegreet; Ikram, M. Arfan; Vernooij, Meike W.; de Bruijne, Marleen

Published in:
IEEE transactions on medical imaging

DOI:
10.1109/TMI.2014.2366792

Publication date:
2015

Document version
Peer reviewed version

Citation for published version (APA):
van Opbroek, A., Ikram, M. A., Vernooij, M. W., & de Bruijne, M. (2015). Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging*, *34*(5), 1018-1030. https://doi.org/10.1109/TMI.2014.2366792

Transfer Learning Improves Supervised Image Segmentation Across Imaging Protocols

Annegreet van Opbroek, M. Arfan Ikram, Meike W. Vernooij, Marleen de Bruijne

***Abstract*—The variation between images obtained with different scanners or different imaging protocols presents a major challenge in automatic segmentation of biomedical images. This variation especially hampers the application of otherwise successful supervised-learning techniques which, in order to perform well, often require a large amount of labeled training data that is exactly representative of the target data.**

We therefore propose to use transfer learning for image segmentation. Transfer-learning techniques can cope with differences in distributions between training and target data, and therefore may improve performance over supervised learning for segmentation across scanners and scan protocols. We present four transfer classifiers that can train a classification scheme with only a small amount of representative training data, in addition to a larger amount of other training data with slightly different characteristics. The performance of the four transfer classifiers was compared to that of standard supervised classification on two MRI brain-segmentation tasks with multi-site data: white matter, gray matter, and CSF segmentation; and white-matter-/MS-lesion segmentation.

The experiments showed that when there is only a small amount of representative training data available, transfer learning can greatly outperform common supervised-learning approaches, minimizing classification errors by up to 60%.

I. Introduction

Segmentation of biomedical images plays a crucial role in many medical imaging applications, forming an important step in enabling quantification in medical research and clinical practice. Since manual segmentation is very time consuming and prone to intra- and inter-observer variations, a variety of techniques have been developed to perform segmentation automatically.

Many successful approaches to automatic segmentation rely on voxelwise classification by supervised-learning techniques. In supervised learning (manually) labeled training data is used to train a classification scheme for the target data. First, features are extracted from the training and target data, after which a classifier is trained. This classifier can then be used to segment the target data into the different tissue classes, based on the extracted features.

Examples of successful voxelwise-classification methods can, among many other applications, be found in brain-tissue segmentation, lesion segmentation, cartilage segmentation, and plaque segmentation. Anbeek et al. [1] performed brain-tissue segmentation by a kNN classifier with intensity and spatial features. The same classification framework was also used for segmentation of white-matter lesions [2]. Geremia et al. [14] performed MS-lesion segmentation with a spatial decision forest classifier on local and context features. Here, local features consisted of voxel intensities, while context features consisted of mean intensities of a three-dimensional box around the voxel. Folkesson et al. [12] performed knee-cartilage segmentation with a kNN classifier with intensity and spatial features, as well as intensity after convolution with a Gaussian, and first-, second-, and third-order derivative features. Liu et al. [18] performed plaque-component segmentation by first performing a voxelwise classification with a Parzen classifier on features like intensity and distance to the lumen. Next, the region boundaries were determined with an active-contour model in order to eliminate isolated voxels.

In order for supervised-learning algorithms to perform well, the used training data needs to be representative of the target data. However, in medical image segmentation a sufficient amount of exactly representative manually labeled training data is often not available because of between-patient variability or because images are acquired with different scanners and/or different scan protocols.

We propose to perform segmentation through a different type of machine learning, called *transfer learning*. Transfer-learning algorithms exploit similarities between different classification problems or datasets to facilitate the construction of a new classification model. They possess the ability of supervised-learning algorithms to capture class-specific knowledge in the training phase without requiring exactly representative training data. Except for a preliminary study presented in [35], to the best of our knowledge transfer learning has not yet been applied to medical image segmentation.

The purpose of our study was to investigate whether transfer-learning techniques can improve upon regular supervised segmentation of images obtained with different scan protocols. We compare the performance of four transfer classifiers with that of standard supervised-learning classifiers. All four transfer classifiers use training data from sources other than the target source, which was acquired with different scan protocols and at different scanners, as well as a small amount of representative training data from the target source acquired with the same protocol as the target data. We performed experiments on voxelwise MRI brain-tissue segmentation and

A. van Opbroek is with the Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology of the Erasmus MC - University Medical Center Rotterdam, 3000 CA Rotterdam, The Netherlands (e-mail: a.vanopbroek@erasmusmc.nl).

M.A. Ikram and M.W. Vernooij are with the Departments of Radiology and Epidemiology of the Erasmus MC - University Medical Center Rotterdam.

M. de Bruijne is with the Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology of the Erasmus MC - University Medical Center Rotterdam, and also with the Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen, Denmark. (e-mail: marleen.debruijne@erasmusmc.nl)

white-matter-lesion segmentation.

This paper is organized as follows: first some background information on transfer learning is given in Section II. Section III describes the four transfer classifiers we used. Section IV describes the experiments. Section V-A presents the performance of the four classifiers on brain-tissue segmentation, and Section V-B on MS-lesion and white-matter-lesion segmentation. The conclusion and discussion are given in Section VI.

II. Background

Transfer learning is a relatively new form of machine learning that allows for differences between training and target domains, tasks, and distributions. This means that training and test data may follow different distributions $P(\boldsymbol{x})$, may have different labeling functions $P(y|\boldsymbol{x})$, may have different features, and may even consist of different classes. In the transfer-learning literature data that follows the same distribution, has the same labeling function, and the same features is often referred to as data that comes from the same *source*. The goal of transfer learning is to learn a classification algorithm for the target data, that benefits from already available data that originates from different sources, i.e. data that is somehow similar, but not necessarily exactly representative for the target data. This approach is opposed to that of traditional supervised-learning algorithms, which assume that training and target data come from the same source.

Pan and Yang [22] provide an overview of the transfer-learning literature, where they distinguish between three types of transfer learning: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. In this paper we are dealing with inductive transfer learning, where the training and target data may have different labeling functions $P(y|\boldsymbol{x})$, as well as different features and/or prior distributions $P(\boldsymbol{x})$. We assume that a small number of labeled training samples from the target source is available, the so-called *same-distribution training data*, and aim to transfer knowledge from a much larger amount of labeled training data that is available from sources other than the target data, the so-called *different-distribution training data*. Inductive transfer learning assumes that even though labeling functions vary between training and target sources, they are still somewhat similar, in such a way that different-distribution sources give some extra information in areas of the feature space where same-distribution training data is scarce.

We present four transfer classifiers that use this same- and different-distribution training data, all based on support vector machine (SVM) classification. Three of the four classifiers use sample weighting. First of all, the Weighted SVM [41], in which both same- and different-distribution training samples are used for training, the latter with a lower weight than the former. Secondly, the Reweighted SVM, which we proposed in [35], which is an extension to the Weighted SVM where iteratively the weights of misclassified different-distribution training samples are reduced. And thirdly, TrAdaBoost [7], which builds a boosting classifier for transfer learning by iteratively increasing the weights of misclassified same-distribution samples while reducing the weights of misclassified different-distribution samples. Removing misleading different-distribution samples is considered a common approach in transfer learning [17]. The fourth transfer classifier presented in this paper, Adaptive SVM [42], is not based on sample weighting. The Adaptive SVM trains an SVM on the same-distribution samples only, with the restriction that the resulting classifier should be close to an SVM on the different-distribution samples. The next section will discuss the four transfer classifiers in detail.

III. Methods

Let $\boldsymbol{x}_i \in \mathbb{R}^n$ denote a training sample i which is a vector containing a value for each of the n features. We assume to have a total of N_s same-distribution training samples $\boldsymbol{x}_i^s$ ($i = 1, 2, \ldots, N_s$) with their corresponding labels y_i^s. The total of all same-distribution training samples is denoted by $T_s = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{N_s}$. In a similar way, the different-distribution training samples are denoted by $T_d = \{\boldsymbol{x}_i^d, y_i^d\}_{i=1}^{N_d}$, so that there is a total training set $T = T_s \cup T_d$ of size $N = N_s + N_d$. For the moment we assume $y_i^s, y_i^d \in \{1, -1\}\ \forall i$, but all the presented algorithms can easily be adapted to more than two classes by one-vs-one or one-vs-all classification.

We compared the performance of four transfer classifiers with the performance of the traditional SVM classifier. The traditional, soft-margin SVM by Cortes and Vapnik [6] constructs a linear decision function $f(\boldsymbol{x}) = \boldsymbol{v} \cdot \boldsymbol{x} + v_0$, where $\boldsymbol{v}$ and v_0 are model parameters that have to be optimized from the data by minimizing the SVM optimization criterion:

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \frac{1}{2}\|\boldsymbol{v}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{v}^T\boldsymbol{x}_i + v_0) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall\, \boldsymbol{x}_i \ . \end{aligned} \tag{1}$$

In this optimization the term $\|\boldsymbol{v}\|^2$ maximizes the margin around the decision function, and $C\sum_{i=1}^{N}\xi_i$ minimizes the number of samples that are either misclassified or lie within the margin. C is the SVM parameter to trade off between maximizing the margin and minimizing $\sum_i \xi_i$, where a sample $\boldsymbol{x}_i$ receives a value $\xi_i > 1$ if it is misclassified, a value $0 < \xi_i \leq 1$ if it is correctly classified but lies within the margin, and a value $\xi_i = 0$ otherwise.

The original soft-margin SVM presented above can only produce linear decision functions. By using kernel learning one can obtain a non-linear decision function [26]. In kernel SVM a map ϕ is created that maps every sample $\boldsymbol{x}_i$ into a (possibly high-dimensional) feature space $\phi(\boldsymbol{x}_i)$, where an SVM decision function $f(\boldsymbol{x}) = \boldsymbol{v} \cdot \phi(\boldsymbol{x}) + v_0$ can be calculated. This results in a decision function that is linear in the new feature space $\phi(\boldsymbol{x})$, but depending on the mapping ϕ can be non-linear in the original feature space. Explicitly calculating $\phi(\boldsymbol{x})$ however could be very expensive. Luckily, the resulting decision function $f(\boldsymbol{x}) = \boldsymbol{v} \cdot \phi(\boldsymbol{x}) + v_0$ can be calculated without explicitly calculating the feature space $\phi(\boldsymbol{x})$, by use of a kernel matrix. This kernel matrix $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ gives the inner product between every combination of samples in the feature space $\phi(\boldsymbol{x})$. The decision function $f(\boldsymbol{x}) = \boldsymbol{v} \cdot \phi(\boldsymbol{x}) + v_0$ can be calculated entirely by means of inner products of samples in $\phi(\boldsymbol{x})$. This means that only the kernel matrix K needs to be calculated in order to obtain a non-linear decision function, and the accompanying mapping ϕ need not be calculated.

A. Weighted SVM

Sample weighting can be incorporated in the original SVM definition by assigning a weight $w_i \geq 0$ to every training sample $\boldsymbol{x}_i$, which indicates the importance of the sample. The sum of all weights, $|\boldsymbol{w}|$ should equal the total number of training samples, N. Incorporating sample weights in the SVM objective function results in the following objective function [4]

$$\min_{\boldsymbol{v}} \frac{1}{2} \|\boldsymbol{v}\|^2 + C \sum_{i=1}^{N} w_i \xi_i. \tag{2}$$

The constraints remain the same as in the traditional SVM.

Now, one way to perform transfer learning is by training a classifier on T where T_s samples receive a weight of one and T_d samples receive a weight of R_W, as is also done in the transfer SVM classifier presented by Wu and Dietterich [41]. This results in the following SVM objective function:

$$\min_{\boldsymbol{v}} \frac{1}{2} \|\boldsymbol{v}\|^2 + C R_W \sum_{i:\boldsymbol{x}_i \in T_d} \xi_i + C \sum_{i:\boldsymbol{x}_i \in T_s} \xi_i. \tag{3}$$

In our experiments R_W was determined with cross validation as described in Sect. IV-A.

We will refer to this method as the Weighted SVM (WSVM).

B. Reweighted SVM

The second transfer classifier we studied is a transfer SVM we presented in a preliminary workshop paper [35]. This algorithm is an adaptation of the Weighted SVM that performs N_{it} iterations in which the weights of misclassified T_d samples are decreased in order to reduce the influence of T_d samples that contradict the rest of the data. This algorithm is a hybrid between the WSVM and TrAdaBoost, which is described in the next subsection.

The algorithm starts by giving each sample $\boldsymbol{x}_i$ a weight

$$w_i^1 = \begin{cases} R_R & \text{for } \boldsymbol{x}_i \in T_d \\ 1 & \text{for } \boldsymbol{x}_i \in T_s \end{cases} , \tag{4}$$

where similar to R_W in the WSVM the optimal value for R_R was set with cross validation. Then a total of N_{it} iterations are performed where for each iteration $t = 1, 2, \ldots, N_{it}$ first the weights are normalized to sum up to N,

$$\boldsymbol{w}^t = N \frac{\boldsymbol{w}^t}{|\boldsymbol{w}^t|}, \tag{5}$$

a weighted SVM classifier $f^t(\boldsymbol{x})$ is calculated from T and $\boldsymbol{w}^t$, and the weights for the next iteration are determined by

$$w_i^{t+1} = \begin{cases} w_i^t & \text{for } \boldsymbol{x}_i \in T_s \\ w_i^t \beta^{\frac{1}{2}|f^t(\boldsymbol{x}_i) - y_i|} & \text{for } (\boldsymbol{x}_i, y_i) \in T_d \end{cases} . \tag{6}$$

Here $\beta = 1/(1 + \sqrt{2 \ln N_d / N_{\text{it}}})$. This value equals the value used in the TrAdaBoost algorithm, and is derived from AdaBoost [13]. The final classifier is the weighted SVM with the weights from the last iteration.

We made a small adaptation to the algorithm presented in [35] to make it more robust. A disadvantage of reducing the weights of the T_d samples is that it can dis balance the classes, since reduction of weights may happen more in one class than in the other. This is undesirable because it will change the priors of the classes, which will shift the classifier towards the class with the lowest total weight. This problem was solved by in each iteration t normalizing the weights of the different classes, so that

$$\sum_{i:y_i=1} w_i^{t+1} = \sum_{i:y_i=1} w_i^t \text{ and } \sum_{i:y_i=-1} w_i^{t+1} = \sum_{i:y_i=-1} w_i^t \tag{7}$$

The resulting algorithm will be referred to as the Reweighted SVM (RSVM).

C. Transfer AdaBoost

The third transfer classifier we studied is Transfer AdaBoost [7] (TrAdaBoost), which is based on AdaBoost [13]. Like AdaBoost, TrAdaBoost is an iterative algorithm that reduces and increases the weights of training samples according to the outcome of a classifier. The final classifier is obtained by a weighted majority vote of the resulting classifiers.

The TrAdaBoost algorithm is trained on T where each sample $\boldsymbol{x}_i$ is given an initial weight w_i^1, which in our experiments was set with cross validation. In each iteration $t = 1, 2, \ldots, N_{it}$ the weights $\boldsymbol{w}^t$ are normalized to sum up to one, and a weighted classifier $f^t(\boldsymbol{x})$ is trained. The weights for the next iteration are then determined by

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{\frac{1}{2}|f^t(\boldsymbol{x}_i) - y_i|} & \text{for } (\boldsymbol{x}_i, y_i) \in T_d \\ w_i^t \beta^{-\frac{1}{2}|f^t(\boldsymbol{x}_i) - y_i|} & \text{for } (\boldsymbol{x}_i, y_i) \in T_s \end{cases} , \tag{8}$$

for $\beta = 1/(1 + \sqrt{2 \ln N_d / N_{\text{it}}})$. Note that the weights of misclassified T_d samples are reduced by β, as in the Reweighted SVM, whereas the weights of misclassified T_s samples are increased by β, which is not the case in the Reweighted SVM, but is done in AdaBoost. After N_{it} iterations the final classification is determined by a weighted majority vote of the last $\lceil \frac{N_{it}}{2} \rceil$ classifiers $f^t(\boldsymbol{x})$:

$$f(\boldsymbol{x}) = \begin{cases} 1, & \prod_{t=\lceil \frac{N_{it}}{2} \rceil}^{N_{it}} \beta_t^{-f^t(\boldsymbol{x})} \geq 1 \\ -1, & \text{otherwise} \end{cases} , \tag{9}$$

where $\beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$, with ϵ_t the error of $f^t(\boldsymbol{x})$ on the T_s samples multiplied by the weight of each T_s sample:

$$\epsilon_t = \sum_{i:(\boldsymbol{x}_i, y_i) \in T_s} \frac{\frac{w_i^t}{2} |f^t(\boldsymbol{x}_i) - y_i|}{\sum_{i:(\boldsymbol{x}_i, y_i) \in T_s} w_i^t}. \tag{10}$$

This leads to a final classifier $f(\boldsymbol{x})$ in which the intermediate classifiers $f^t(\boldsymbol{x})$ that have a good performance on the T_s samples are given a large weight.

D. Adaptive SVM

The fourth transfer classifier is based on a different approach than the previous three. Instead of adding the T_d samples as training samples, one could also train a separate classifier on the T_d samples, and use this classifier to regularize a classifier trained on the T_s samples. This idea is presented in the Adaptive SVM [42] (A-SVM). First a regular SVM on the T_d samples is trained, resulting in a different-distribution classifier $f^d(\boldsymbol{x})$. This classifier is then adapted to the target data by training a "delta function", $\Delta f(\boldsymbol{x})$, which adapts $f^d(\boldsymbol{x})$ to obtain the final classifier $f(\boldsymbol{x})$:

$$\begin{aligned} f(\boldsymbol{x}) &= f^d(\boldsymbol{x}) + \Delta f(\boldsymbol{x}) && (11)\\ &= f^d(\boldsymbol{x}) + \boldsymbol{v}^T\boldsymbol{x} + v_0. && (12) \end{aligned}$$

The parameters $\boldsymbol{v}$ and v_0 of $\Delta f(\boldsymbol{x})$ are determined from T_s by optimizing

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \frac{1}{2}\|\boldsymbol{v}\|^2 + C^s \sum_{i=1}^{N} \xi_i, && (13)\\ \text{s.t.} \quad & y_i f^d(\boldsymbol{x_i}) + y_i(\boldsymbol{v}^T\boldsymbol{x}_i + v_0) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall\, (\boldsymbol{x}_i, y_i) \in T_s. \end{aligned}$$

Note that the first constraint differs from the definition of the original SVM in (1). This constraint favors an answer where the total classifier $f(\boldsymbol{x})$ correctly classifies the T_s samples. The regularization term $\|\boldsymbol{v}\|^2$ in the objective function on the other hand, favors an answer close to $\Delta f(\boldsymbol{x}) = 0$, resulting in a total classifier $f(\boldsymbol{x})$ that is close to the different-distribution classifier $f^d(\boldsymbol{x})$. The above optimization criterion therefore results in a classifier $f(\boldsymbol{x})$ that is close to $f^d(\boldsymbol{x})$, but is also adapted to improve classification on the T_s samples.

Contrary to the parameter C in (1) the cost factor C^s in (13) does not balance between optimization of the margin and classification of the training samples. The role of C^s is to balance between a classifier $f(\boldsymbol{x})$ that is close to $f^d(\boldsymbol{x})$, and correctly classifying the T_s samples, where a higher value for C^s gives a larger weight to the T_s samples. As with the parameters in the other transfer classifiers, in our experiments C^s was set with cross validation.

Similar to the original SVM, A-SVM can also be used with kernels, by changing $\boldsymbol{x}_i$ in (12) and (13) to $\phi(\boldsymbol{x}_i)$.

An advantage of the A-SVM is that the classifier on the T_d samples only has to be calculated once, which reduces the computational load of the classifier. The memory load of the A-SVM is also lower than for the other classifiers, since all samples T need not be loaded in the memory at the same time.

IV. Experiments

We performed experiments on segmentation through voxelwise classification on data from multiple sources acquired with different MRI scanners. We evaluated two different applications of voxelwise classification: segmentation of white matter (WM) / gray matter (GM) / cerebrospinal fluid (CSF), and white-matter-lesion (WML) and multiple-sclerosis lesion (MSL) segmentation. In both cases we compared the performance of the four transfer classifiers to that of two regular supervised-learning classifiers: a regular SVM trained on all training samples, T, and an SVM trained on the same-distribution training samples, T_s only. Figure 1 schematically shows the usage of the different training sources in the different classifiers.

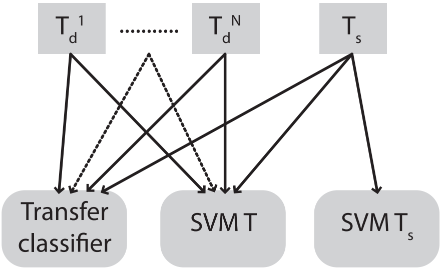


Figure 1. Schematic figure of the T_d data from sources 1 to N, the T_s data, and what training data is used in the different classifiers. The Transfer classifier denotes any of the four transfer classifiers presented.

A. Experimental Setup

Both in the WM/GM/CSF segmentations and the WML and MSL segmentations we used data from multiple sources: four for WM/GM/CSF segmentation, and three for lesion segmentation. We performed cross-validation experiments where in turn one source was selected as same-distribution source, where same-distribution training data and test data was obtained, while the data from the other sources was used as different-distribution training data.

In each experiment the performance of the four transfer classifiers was compared to the performance of the two supervised-learning classifiers. A fixed number of T_d samples was selected from the images of the different-distribution sources, while the number of T_s samples was varied, to study the influence of the amount of same distribution training data. All classifiers used exactly the same test samples and where possible the same T_s and T_d training samples.

All six classifiers were based on SVM classification with a Gaussian kernel. For the regular SVM and the weighted SVMs in WSVM, RSVM and TrAdaBoost an implementation in LIBSVM [4] was used. For A-SVM we used an adaptation to the LIBSVM algorithm by the authors of the A-SVM paper[1].

For the RSVM we chose $N_{it} = 20$, which is enough to achieve convergence. For TrAdaBoost we set $N_{it} = 100$, which should be sufficient according to [7].

For each source, suitable values for the SVM parameter C and the kernel parameter γ were determined with grid search

[1] http://www.cs.cmu.edu/ juny/AdaptSVM/

on T_d, where the best C and γ were selected according to the accuracy of a regular SVM. The same C and γ were used in all classifiers.

All four transfer classifiers have a transfer parameter that has to be tuned according to the data: for WSVM the ratio R_W, for RSVM the ratio R_R, for TrAdaBoost the initial weights $\boldsymbol{w}^1$ of the T_s samples, and for A-SVM the parameter C^s. For each of the sources this was done on the available T_d samples. Note that in all experiments T_d consisted of data from multiple sources. Each of the different-distribution sources was in turn selected as same-distribution source, where T_s training data and test data was selected, while the other different-distribution source/sources were used to extract T_d samples. In each experiment the transfer parameter optimizing the accuracy was recorded. The final parameters were obtained by averaging over the optimal parameters obtained for each of the different-distribution sources.

All images were corrected for non-uniformity using the N4 method [30], and basic image normalization was performed by a range-matching procedure that scaled the intensities such that the voxels between the 4th and the 96th percentage in intensity within the brain mask were mapped between zero and one. In each of the sources the features were normalized to zero mean and unit standard deviation.

For both applications the performance is reported in learning curves, showing the accuracy of the six classifiers as a function of the used number of T_s samples.

B. Brain-Tissue Segmentation Experiments

The segmentation of MRI brain images into the different tissues present (GM, WM, CSF) can give insight in the presence, severity, and location of brain atrophy. This can provide useful information about neuro-degenerative diseases such as dementia, as well as other brain disorders such as multiple sclerosis (MS) and schizophrenia. Many automated brain-tissue segmentation methods have been developed over the past 20 years, which are used in medical research as well as in the clinic.

In our experiments we performed brain-tissue segmentation by three-class voxelwise classification within a manually selected brain mask. Within this brain mask every voxel was classified as either WM, GM, or CSF.

1) Data Description: MR images with corresponding manual segmentations from the following four sources were used:

1) 6 T1-weighted images from the Rotterdam Scan Study [16], acquired with a 1.5T GE scanner with $0.49 \times 0.49 \times 0.80$ mm^3 voxel size
2) 12 half-Fourier acquisition single-shot turbo spin echo (HASTE) images scanned with a HASTE-Odd protocol (inversion time = 4400 ms, TR = 2800 ms, TE = 29 ms) from the Rotterdam Scan Study [16], acquired with a 1.5T Siemens scanner with $1.25 \times 1 \times 1$ mm^3 voxel size. These HASTE-Odd images have image contrast comparable to inverted T1 intensity.
3) 18 T1-weighted images from the Internet Brain Segmentation Repository (IBSR) [40], acquired with an unknown scanner, with voxel sizes between $0.84\times0.84\times 1.5$ mm^3 and $1 \times 1 \times 1.5$ mm^3
4) 20 T1-weighted images from the IBSR [40], 10 acquired with a 1.5T Siemens scanner, 10 acquired with a 1.5T GE scanner, all with $1 \times 3.1 \times 1$ mm^3 voxel size

All four sources used different scanners and different scanning parameters. Figure 3 shows a slice of an image from each of the four sources. The HASTE-Odd images were inverted prior to classification, because of their inverted tissue intensities compared to the T1-weighted images.

2) Features: To study the influence of the number of features, we performed classification on two different feature sets. The first feature set consisted of four features:

- The intensity
- The x, y, and z coordinate of the voxel, divided by the maximum width, length and height of the brain.

The second feature set consisted of 13 features – the four features mentioned above, together with nine scale-space features:

- The intensity after convolution with a Gaussian kernel with $\sigma = 1, 2,$ and 3 mm^3
- The gradient magnitude of the intensity after convolution with a Gaussian kernel with $\sigma = 1, 2,$ and 3 mm^3
- The absolute value of the Laplacian of the intensity after convolution with a Gaussian kernel with $\sigma = 1, 2,$ and 3 mm^3.

3) Train and Test Sets: From the same-distribution source in turn one image was selected, where between 3 (1 for every class) and 200 T_s samples were selected randomly, while the other images in the source were used as test images. For training a total of 1 500 T_d training samples per source were selected randomly from all images of the three different-distribution sources. From each of the test images 4 000 random samples were selected, on which the accuracy was evaluated. Mean classification errors were obtained by performing multiple experiments where every image in the source was once selected as training image.

4) Comparison with Existing Methods: To compare the performance of our SVM classification framework with that of existing methods, complete image segmentations were obtained and compared against manual segmentations and segmentations obtained with SPM8 [3]. SPM8 is a state-of-the-art brain-tissue-segmentation tool. It performs automatic segmentation based on mixture of Gaussians with incorporation of tissue probability maps of the three tissues, that are non-linearly registered to the target image, and intensity non-uniformity estimation. The segmentation is determined with the expectation-maximization algorithm.

Evaluations were performed with the Dice coefficient [10] on the WM, GM, and CSF. The Dice coefficient is defined as

$$\text{Dice} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}}, \tag{14}$$

where TP denotes the true positives, FP the false positives, and FN the false negatives.

The performance on the data from Source 4 was compared to that of several other automatic techniques as reported in literature. For this, the Tanimoto coefficient (which is also known as the Jaccard index) was used:

$$\mathrm{TC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \quad (15)$$

Note that TC ≤ Dice.

5) Influence of Normalization: We also performed classification with two different types of image normalization in order to study the added value of the transfer classifiers over various normalization techniques. In the experiments mentioned above all images were normalized by a range-matching procedure which maps the 4th and the 96th percentage of intensity within the brain mask to zero and one. We studied the influence of two other normalization techniques. For the first method the minimum intensity within the brain mask is mapped to zero, and the maximum to one. This method should be less robust to outliers in intensity than mapping the 4th and 96th percentile. For the second method we performed the tenth-percentile normalization procedure of Nyúl et al. [21] within the 4th and 96th percentage of intensity. This procedure first applies a range matching which maps the 4th and 96th percentile to zero and one, and next maps every tenth percentile within zero and one to the mean intensity over all (training and target) images.

Normalization experiments were performed on 13 features with the SVM T, SVM T_s, WSVM, RSVM, and A-SVM classifier. TrAdaBoost was omitted in these experiments because of its high computational load.

C. MSL and WML Segmentation Experiments

MS is a chronic inflammatory disease that affects the white matter in the brain, resulting in the formation of WMLs. Automatic methods to segment these lesions in MRI images enable the diagnosis and monitoring of the disease without the tedious task of performing manual segmentations. WMLs also occur in individuals who do not have MS. Typically, WML load increases with age, and a higher WML load is associated with cognitive decline [9], increased risk of stroke [36], and increased risk of dementia [24]. Automatic segmentation of WMLs therefore provides useful information in these research areas, as well as for the monitoring of patients.

In our experiments we performed WML and MSL segmentation by voxelwise classification. First a brain mask was determined with the brain-extraction tool [31], after which every voxel within the brain mask was classified as either lesion (WML or MSL were treated the same) or non-lesion tissue.

1) Data Description: We used data with manual segmentations from three different sources:

1) 20 healthy elderly subjects from the Rotterdam Scan Study [16], scanned with three sequences: T1, PD, and FLAIR, with $0.49 \times 0.49 \times 0.80$ mm^3 voxel size
2) 10 MS patients from the MS Lesion Challenge [32], scanned at the Children's Hospital of Boston with three sequences: T1, T2, and FLAIR, with $0.5 \times 0.5 \times 0.5$ mm^3 voxel size
3) 10 MS patients from the MS Lesion Challenge [32], scanned at the University of North Carolina with three sequences: T1, T2, and FLAIR, with $0.5 \times 0.5 \times 0.5$ mm^3 voxel size

Figure 6 shows slices of the three sequences for the three sources. As the PD images of Source 1 appear similar to the T2 images of Sources 2 and 3, we decided to treat these modalities to be the same.

2) Features: We performed experiments with a small and a large feature set, which were composed similarly to the feature sets for WM/GM/CSF segmentation discussed in Section IV-B2, with the difference that three MRI sequences were used instead of one, and the Gaussian kernels used for the convolution had sizes $\sigma = 0.5, 1,$ and 2 mm^3. The smaller kernel sizes account for the higher resolution of the images compared to the images used in the WM/GM/CSF experiments. This resulted in a feature set of 6 features and a set of 33 features.

3) Train and Test Sets: Since lesion voxels appear bright on FLAIR scans, we first discarded all voxels with a low FLAIR intensity. The threshold was set to 0.75 on the normalized FLAIR image, discarding most of the CSF and some GM voxels. For the reported learning curves only voxels with a FLAIR intensity above this threshold were selected for training and testing.

For Sources 1 and 2 train and test data was obtained by randomly selecting 1% of the lesion voxels in the image and then randomly selecting non-lesion voxels above the FLAIR threshold, so that a total of 5 000 samples per image were selected. The images of Source 3 contain only few lesion voxels, since these subjects were less affected and the images were also more conservatively segmented. To still have a reasonable number of lesion samples in Source 3 4% of all lesion voxels was selected. This resulted in training and test sets with a lesion percentage of 13% for Source 1, 15% for Source 2, and 10% for Source 3.

One to eight same-distribution training images different from the test images were selected from the same-distribution source, where from each image 200 same-distribution training samples were randomly selected in the way mentioned above. From the different-distribution sources 2 000 T_d samples were selected per source.

Mean classification errors were obtained by performing multiple experiments for differing numbers of T_s images, where every image in the same-distribution source was once used as first training image, once as second training image, etcetera. The images from the same-distribution source that were not used for training were used for testing, where the accuracy was determined on test sets of 5 000 samples per image.

4) Experiments for MS Lesion Challenge: We also calculated complete segmentations on 30 test images of the MS Lesion challenge, and submitted these to the challenge. Of the 30 test images 17 were acquired at the Children's Hospital of Boston (CHB, Source 2), and 13 at the University of North Carolina (UNC, Source 3). Segmentations were performed with RSVM on 33 features, which was the classifier that overall performed best in the experiments with eight same-distribution images.

In order to obtain a competing segmentation framework, the classifier was trained on more T_s samples than used in the learning curves. To speed up the calculation, only few T_d samples were used. A total of $50\,000$ T_s samples were selected from the ten same-distribution training images and $4\,000$ T_d samples were selected from the two different-distribution sources.

The classification parameters were set in a slightly different way than for the previous experiments. The SVM parameters C and γ were obtained with a grid-search experiment on the ten same-distribution images with a regular SVM. The parameter R_R was determined with a cross-validation experiment on the ten same-distribution images. In turn one same-distribution image was selected as test image, while the other nine same-distribution images were used as training data, together with the T_d samples. The value for R_R with the highest accuracy was selected.

The RSVM classifier was used to calculate a posterior probability $P(y = 1|\boldsymbol{x})$ per test voxel. The final segmentations were obtained by thresholding the posterior probability. The threshold was set differently for the two sources in the challenge data, in such a way that for the ten same-distribution training images the lesion volume in the manual and the automatic segmentation was equal.

We noticed that lesions voxels in the middle of large lesions often had lower intensities than the surrounding lesion voxels, which sometimes caused these voxels to be misclassified as non-lesion voxels. This was solved by a post-processing step, where groups of non-lesion-voxels that in the x and y direction were surrounded by lesion voxels, were classified to be lesion voxels.

The performance of our classifier on the test images of the MS Lesion Challenge was evaluated against two expert manual segmentations: segmentations from the expert who segmented the training data in Source 2, and segmentations from the expert who segmented the training data in Source 3. The segmentations were evaluated on four error metrics: relative absolute volume difference (RAVD), average symmetric surface distance (ASSD), true positive rate (TPR), and false positive rate (FPR) [32].

5) Influence of Normalization: We performed experiments with two different types of normalization, similar to the experiments on GM/WM/CSF segmentation. In these experiments images from the three modalities (T1, T2/PD, and FLAIR) were all normalized with 4-96th percentile range matching, min-max range matching, and the tenth-percentile matching of Nyúl et al. [21]. These experiments were performed on the dataset with 33 features for the SVM T, SVM T_s, WSVM, RSVM, and A-SVM classifier.

V. Results

A. Brain-Tissue Segmentation

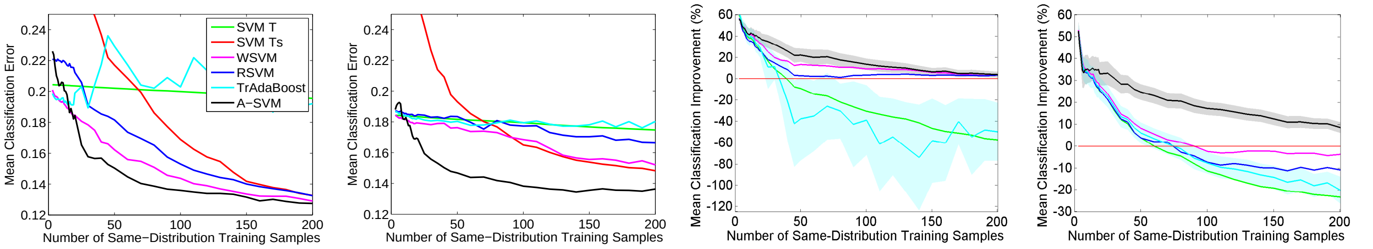
1) Comparison of Classifiers: Figures 2(a) and (b) give the learning curves for all classifiers on 4 and 13 features respectively. These learning curves show the mean classification errors on all 56 target images as a function of the number of same-distribution samples T_s, which were obtained from a single image. For both feature sets the SVM on all training samples T outperformed the SVM on only T_s when the number of T_s samples was small. When more T_s samples were added SVM T_s outperformed the SVM T classifier. Transfer learning improved classification compared to these two supervised-learning techniques. For SVM T classification errors were slightly lower for 13 features than for 4 features, which shows that the nine extra features hold additional information over the first four features. For the SVM T_s classifier errors were lower for four features, because of the curse of dimensionality.

Overall, the use of transfer learning improved classification compared to the two supervised-learning techniques. Figures 2(c) and 2(d) show the percentage reduction in classification error of the different classifiers compared to the SVM T_s classifier. These two figures include 95%-confidence intervals (CIs) of the mean improvement of TrAdaBoost and A-SVM. To avoid cluttering the figure not all CIs are shown, but those of SVM T, WSVM, and RSVM were similar to the CI of A-SVM. Overall A-SVM performed best, except for fewer than 15 T_s samples, where WSVM performed best. A-SVM significantly outperformed SVM T_s for the whole range of T_s samples, WSVM significantly outperformed SVM T_s for up to 150 T_s samples for four features, and 70 T_s samples for 13 features. RSVM performed slightly worse than WSVM for both configurations, and only outperformed SVM T_s for less than 50 T_s samples on four features. TrAdaBoost performed poorly for both feature sets, and showed much higher variance than the other classifiers.

2) Comparison with Existing Methods: We compared full image segmentations with existing brain-tissue-segmentation methods for the rightmost point in the learning curves (200 T_s samples). Except for A-SVM on 13 features, the transfer classifiers did not give an improvement over SVM T_s at this point of the feature curves, as can be seen from Fig. 2. The goal of these experiments was therefore not to investigate whether the transfer classifiers improve over other techniques, but to investigate whether our SVM T_s and transfer classifiers compare to available segmentation techniques.

Table I compares the performance of the SVM T_s classifier and three transfer classifiers: WSVM, RSVM, and A-SVM, with segmentations obtained with SPM8 [3]. For SVM T_s and the three transfer classifiers four features were used. TrAdaBoost was not included because of its poor performance and high computational load, which was caused by the large number of iterations. SVM T_s, WSVM, RSVM, and A-SVM were all significantly better than SPM8, but not significantly different from each other, based on a Friedman test with the significance threshold at $P = 0.05$. The table also includes the Dice scores of the best classifier in the brain-tissue-segmentation accuracy study of De Boer et al. [8], who used the Source 1 data to evaluate several brain-tissue-segmentation methods. In this study the best results were obtained with a kNN classifier [37]. Our classifiers obtained similar scores on WM and GM as the kNN classifier. Our classifiers also greatly outperformed the kNN classifier on CSF, but the main reason for this is that we tested within the manually segmented brain mask, whereas for the kNN classifier the brain was segmented by atlas registration. This causes additional errors, especially in the sulcal CSF.

Figure 3 shows examples of the resulting segmentations



(a) Mean, 4 features (b) Mean, 13 features (c) Difference, 4 features (d) Difference, 13 features

Figure 2. Learning curves showing the mean classification error and classification improvement on the test sets as a function of the number of T_s samples, for the six classifiers: an SVM on T, an SVM on T_s, WSVM, RSVM, TrAdaBoost, and A-SVM. (a) and (b) show the average learning curves over all 56 images from the four sources, for 4 and 13 features respectively. (c) and (d) show the percentage reduction in mean classification error compared to the SVM T_s classifier, averaged over all 56 images, for 4 and 13 features respectively. For TrAdaBoost and A-SVM 95%-CIs for the mean improvement were included, the CIs of SVM T, WSVM, and RSVM were similar to the CI of A-SVM.

Table I
DICE COEFFICIENTS FOR CSF, GM, AND WM FOR COMPLETE IMAGE SEGMENTATIONS WITH THE BEST SUPERVISED-LEARNING CLASSIFIER(SVM T_s), WSVM, RSVM, AND A-SVM, AND SPM8. ALL DICES SCORES ARE GIVEN ON THE FOUR SOURCES WITH FOUR FEATURES. FOR EACH EXPERIMENT WE USED 200 T_s SAMPLES FROM ONE SAME-DISTRIBUTION TRAINING IMAGE AND 4 500 T_d SAMPLES. THE KNN CLASSIFIER IS THE BEST CLASSIFIER IN [8] ON THE DATA FROM SOURCE 1.

	Source 1			Source 2			Source 3			Source 4		
Classifier	CSF	GM	WM	CSF	GM	WM	CSF	GM	WM	CSF	GM	WM
SVM T_s	0.90	0.90	0.92	0.87	0.90	0.91	0.08	0.92	0.87	0.45	0.86	0.78
WSVM	0.92	0.92	0.93	0.83	0.91	0.94	0.47	0.92	0.88	0.40	0.86	0.78
RSVM	0.91	0.92	0.93	0.91	0.93	0.94	0.43	0.92	0.87	0.37	0.84	0.77
A-SVM	0.89	0.90	0.92	0.91	0.93	0.94	0.34	0.92	0.87	0.24	0.86	0.77
SPM8	0.81	0.86	0.91	0.85	0.89	0.95	0.19	0.82	0.86	0.19	0.79	0.81
kNN [8]	0.81	0.90	0.94	-	-	-	-	-	-	-	-	-

with the WSVM on the four sources.

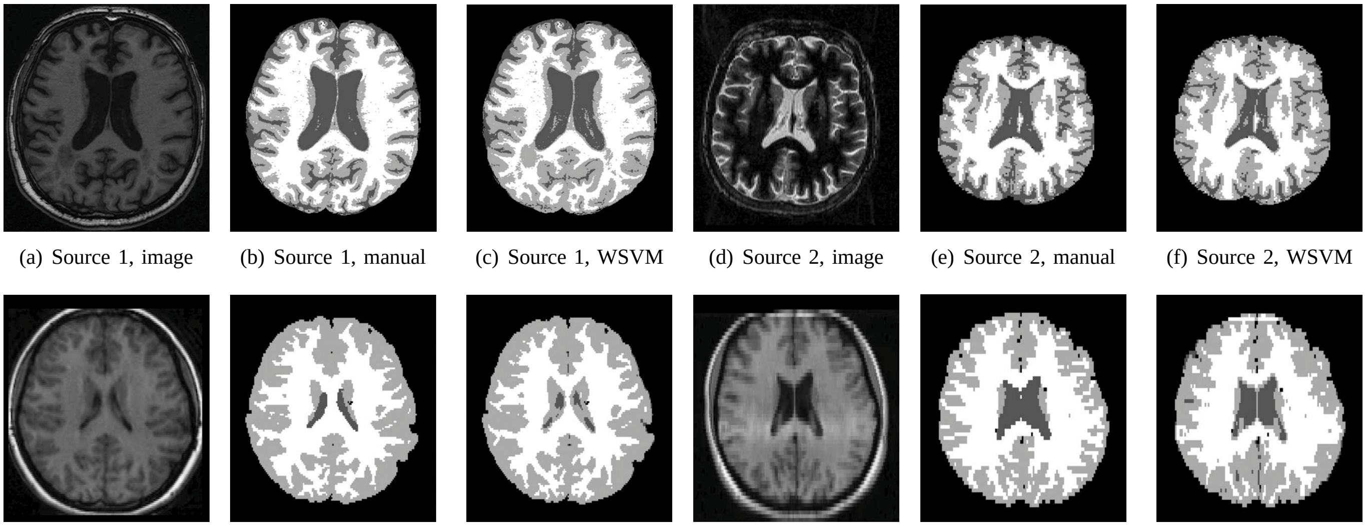
We also compared our complete segmentations on Source 4, the IBSR data with 20 subjects, to various methods that reported their performance on the same data. Table II shows mean Tanimoto coefficients for CSF, GM, WM, and the sum of GM and WM, and CSF, GM, and WM. The first six entries are clustering methods, reported on the IBSR website[2], the other nine methods were collected from literature. Not all methods reported overlap values for the CSF. The best results were obtained with a decision forest classifier [43], which was trained and tested in cross validation on all remaining images. Our SVM T_s classifier and the three tested transfer classifiers WSVM, RSVM, and A-SVM, outperformed most of the other methods as well as SPM8.

3) Influence of Normalization: Figure 4 shows the learning curves for the three types of normalization. Min-Max range matching, for which the results are shown in Figure 4(b) led to higher mean classification errors than 4-96th percentile range matching. Also, for min-max range matching the SVM T_s classifier performed worse than the SVM T classifier regardless of the number of T_s samples, indicating that the min-max normalization is not sufficient, even within the same source. Applying the more extensive normalization of Nyúl et al. [21], for which the result is shown in Figure 4(c), did not give better overall results than when 4-96th percentile range matching was applied. The performance of the SVM T_s and the transfer classifiers was similar for the two normalization techniques, but the SVM T classifier performed slightly better for the 4-96th percentile range matching. This indicates that more extensive normalization is not needed to normalize within sources, and slightly hurts the performance of classification between sources. The use of a transfer classifier improved classification of the two supervised-learning classifiers regardless of the used normalization technique.

B. MSL and WML Segmentation

1) Comparison of Classifiers: We performed a similar set of cross-validation experiments for MSL and WML segmentation. Figures 5(a) and (b) show the mean learning curves of the six classifiers on 6 and 33 features respectively. A very similar pattern can be seen as in the learning curves for GM/WM/CSF segmentation: for a small amount of T_s data SVM T was the best supervised-learning classifier, whereas for more T_s data SVM T_s performed better. The transfer classifiers WSVM and RSVM improved over these two supervised-learning classifiers up to a point where a relatively large number of same-distribution training images was available. At this point SVM T_s, WSVM and RSVM converged to the same error rate. All classifiers performed better on 33 features than on 6. Figures 5(c) and (d) show the improvement over SVM T_s for SVM T, WSVM, RSVM, TrAdaBoost and A-SVM, for 6 and 33 features. The figures include CIs for some of the classifiers. The CIs of the other classifiers were similar to that of WSVM. WSVM and RSVM overall performed best, significantly outperforming SVM T_s for up to five T_s images (three for RSVM on 33 features). WSVM seems to perform slightly worse than RSVM on 33 features, but this difference is not significant. Similar to the GM/WM/CSF experiments

[2] http://www.cma.mgh.harvard.edu/ibsr/



(a) Source 1, image (b) Source 1, manual (c) Source 1, WSVM (d) Source 2, image (e) Source 2, manual (f) Source 2, WSVM

(g) Source 3, image (h) Source 3, manual (i) Source 3, WSVM (j) Source 4, image (k) Source 4, manual (l) Source 4, WSVM

Figure 3. Segmentations with the WSVM classifier with four features. The classifier was trained on a total of 4 500 T_d samples and 200 T_s samples from one image from the target source, which corresponds to the right-most point of the learning curves in Figure 2(a). The classification errors for the shown slices were (c) 8.1%, (f) 9.2%, (i) 6.9%, (l) 15.2%.

Table II
MEAN TANIMOTO COEFFICIENTS ON CSF, GM, AND WM FOR A VARIETY OF METHODS ON THE IBSR DATA WITH 20 SUBJECTS. G+W DENOTES THE AVERAGE SCORE ON GM AND WM, AND C+G+W DENOTES AVERAGE SCORE ON CSF, GM, AND WM. FOR THE SVM T_s, WSVM, RSVM, AND A-SVM CLASSIFIER 200 T_s SAMPLES WERE RANDOMLY SELECTED FROM ONE IMAGE, FOR THE TRANSFER CLASSIFIERS 4 500 T_d SAMPLES WERE ADDED. CLASSIFICATION WAS PERFORMED ON FOUR FEATURES.

Classifier	CSF	GM	WM	G+W	C+G+W	Classifier	CSF	GM	WM	G+W	C+G+W
Adaptive MAP [25]	0.069	0.564	0.567	0.566	0.400	FC-PABIC [44]	-	0.770	0.658	0.714	-
Biased MAP [25]	0.071	0.558	0.562	0.560	0.379	Modified FCM [29]	-	0.750	0.724	0.737	-
Fuzzy c-means [25]	0.048	0.473	0.567	0.519	0.362	MPM-MAP [19]	0.227	0.662	0.683	0.673	0.524
MAP [25]	0.071	0.550	0.554	0.552	0.392	SV-GMM [23]	-	0.768	0.734	0.751	-
ML [25]	0.062	0.535	0.551	0.543	0.383	TMCD [33]	-	0.676	0.669	0.673	-
TS k-means [25]	0.049	0.477	0.571	0.524	0.366	SPM8	0.107	0.650	0.684	0.667	0.480
AMS [20]	-	0.683	0.691	0.687	-	SVM T_s	0.309	0.757	0.645	0.701	0.570
BSE-BFC-PVC [28]	-	0.595	0.664	0.630	-	WSVM	0.266	0.754	0.648	0.701	0.556
C-GMM [15]	-	0.680	0.660	0.670	-	RSVM	0.240	0.730	0.633	0.682	0.534
Decision Forest [43]	0.614	0.838	0.731	0.785	0.728	A-SVM	0.162	0.759	0.633	0.696	0.518

TrAdaBoost overall performed poorly, with a larger variance than the other classifiers. A-SVM, which overall performed best on the WM/GM/CSF experiments, did not perform well in the lesion-segmentation experiments.

Figure 6 shows resulting segmentations of the RSVM classifier on 33 features with eight same-distribution images, where the threshold on the posterior probabilities was selected so that the total lesion volume equaled that in the manual segmentation.

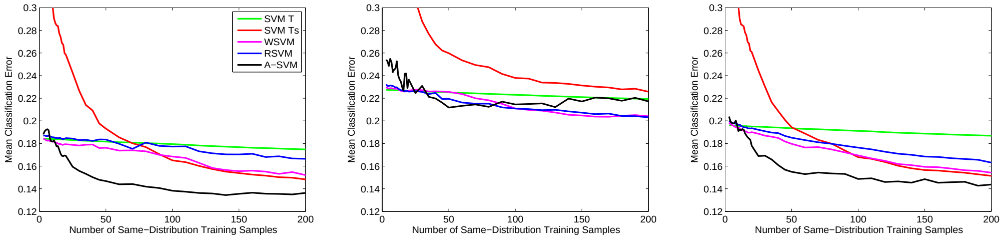
2) MS Lesion Challenge: Table III shows the mean scores obtained on the 30 test images of the MS Lesion Challenge data. The scores were designed such that a score of 90 is comparable to expert segmentations. Our method performed slightly better on the CHB data than on the UNC data, with scores of 80 and 75 respectively. At the moment of writing the website of the MS Lesion Challenge[3] listed the performance of 35 segmentation algorithms. With a total score of 77.9083 our method ranked second on a total of eight methods that segmented all 30 test images. The other 27 methods segmented only 23 test images (14 CHB, 9 UNC), on which our algorithm obtained a total score of 81.2174. Nine of the 27 methods had a higher score on the 23 test images than our algorithm.

3) Influence of Normalization: Figure 7 shows the learning curves for the three types of normalization. Like for WM/GM/CSF segmentation, the Min-Max range matching, shown in Figure 4(b), led to higher mean classification errors than 4-96th percentile range matching. The more extensive normalization of Nyúl et al. [21], for which the result is shown in Figure 7(c), gave similar results as 4-96th percentile range matching for SVM T, WSVM, and RSVM, but not for SVM T_s and A-SVM. For all three normalization techniques a WSVM or RSVM classifier improved performance. Remarkably, the performance of the SVM T_s classifier deteriorates when the normalization of Nyúl et al. [21] is used, compared to 4-96th percentile range matching.

VI. CONCLUSION AND DISCUSSION

We presented a transfer-learning approach to image segmentation, which enables supervised segmentation of images acquired with different MRI scanners and/or imaging protocols. The presented transfer classifiers benefit from training

[3] http://www.ia.unc.edu/MSseg/results_table.php

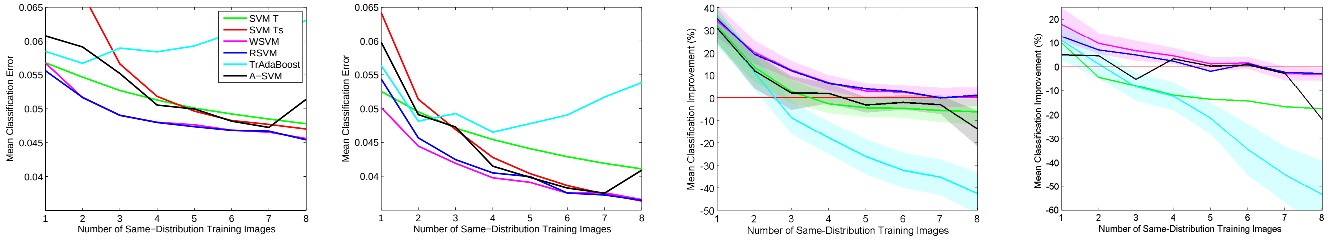


(a) range matching 4-96%

(b) range matching min-max

(c) Nyúl et al. [21] normalization

Figure 4. Learning curves for WM/GM/CSF segmentation, showing mean classification errors as a function of the number of T_s samples on 13 features for three different normalization techniques. (a) equals the image in Fig. 2(b) and includes range matching between the 4th and 96th percentile, (b) includes range matching between the minimum and maximum value, and (c) includes the normalization of Nyúl et al. [21].



(a) Mean, 6 features

(b) Mean, 33 features

(c) Improvement, 6 features

(d) Improvement, 33 features

Figure 5. Learning curves showing the mean classification error and mean classification improvement on the test sets as a function of the number of T_s images, for WML segmentation with the six classifiers: SVM on T, SVM on T_s, WSVM, RSVM, TrAdaBoost, and A-SVM. (a) and (b) show the mean learning curves over all 40 images from the three sources for 6 and 33 features respectively. (c) and (d) show the percentage reduction in mean classification error compared to SVM T_s averaged over all 40 images, for 6 and 33 features respectively. The shaded areas show 95%-CIs for the mean improvement. For (a) and (b) the CI of SVM T and RSVM were similar to the one of WSVM, and for (b) the CI of A-SVM was similar to the one of WSVM.

Table III
AVERAGE SCORES OBTAINED ON THE TWO DATASETS (CHB = CHILDREN'S HOSPITAL OF BOSTON, UNC = UNIVERSITY OF NORTH CAROLINA) OF THE MS LESION CHALLENGE, FOR RSVM WITH 33 FEATURES. RAVD = RELATIVE ABSOLUTE VOLUME DIFFERENCE (%), ASSD = AVERAGE SYMMETRICAL SURFACE DISTANCE (MM), TPR = TRUE POSITIVE RATE (%), FPR = FALSE POSITIVE RATE (%), SC = SCORE.

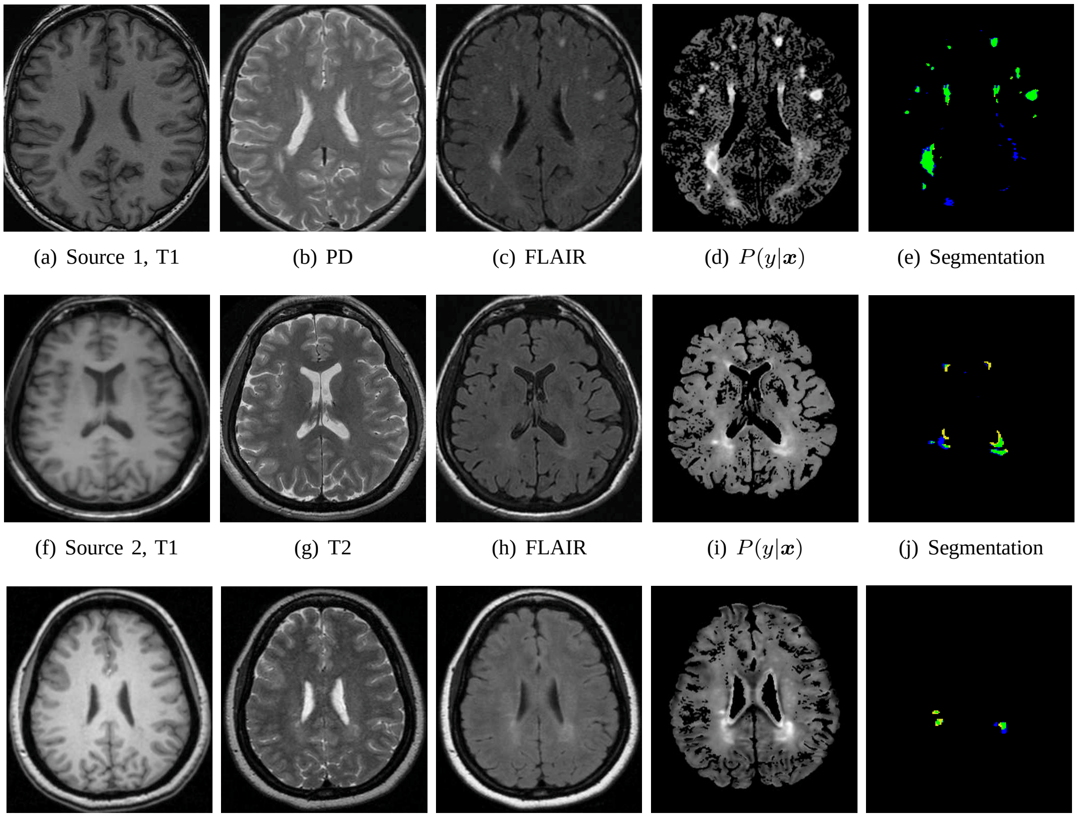
Ground Truth	CHB Rater								UNC Rater								
Dataset	RAVD	Sc	ASSD	Sc	TPR	Sc	FPR	Sc	RAVD	Sc	ASSD	Sc	TPR	Sc	FPR	Sc	Total
All CHB	112.0	84	7.2	85	53.1	81	78.8	62	114.6	89	4.5	91	58.7	84	70.3	67	80
All UNC	151.1	84	12.4	74	28.2	68	65.9	70	300.4	87	13.0	73	44.9	76	69.6	67	75
All Average	128.9	84	9.5	80	42.3	75	73.2	65	195.1	88	8.2	83	52.7	81	70.0	67	78

data acquired with different protocols, so-called different-distribution training data (T_d), and therefore compared to a regular supervised classifier, need fewer labeled samples that are exactly representative of the target data, the so-called same-distribution training data (T_s), to obtain the same result.

The benefits of transfer learning over standard supervised learning were evaluated with experiments on WM/GM/CSF segmentation and WML and MSL segmentation on MRI brain scans obtained with various scanners and scan protocols, with varying numbers of T_s samples. The experiments showed a clear improvement in performance when transfer learning was used. Specifically, for a small number of T_s samples transfer learning greatly outperformed the supervised-learning classifiers, minimizing mean classification errors by up to 60%. Also, when the required accuracy is set, the use of a transfer classifier typically reduces the required number of T_s training samples. Ultimately, when enough T_s samples were available to reliably train a supervised classification scheme, a regular SVM on T_s and the best-performing transfer classifiers reached similar performance.

For GM/WM/CSF segmentation, a relatively easy task, a regular SVM on T_s reached the same performance as the best transfer classifiers at an earlier point than was the case for the lesion segmentation. Also, intuitively transfer learning could bring more improvement when more features are used, since higher-dimensional feature spaces generally require more training samples. This could clearly be seen in the experiments on WML/MSL segmentation. On the experiments on brain-tissue classification however, only one of the transfer classifiers gave more improvement on the larger feature set.

We presented and evaluated four transfer classifiers: Weighted SVM (WSVM), Reweighted SVM (RSVM), TrAdaBoost, and Adaptive SVM (A-SVM). WSVM showed to be the most consistent classifier of the four; for a small number of T_s samples, it outperformed the regular SVMs on all training data T and on only T_s in all learning curves. RSVM generally performed similar to the WSVM on the lesion segmentation experiments, but worse than the WSVM on the WM/GM/CSF segmentation experiments. TrAdaBoost

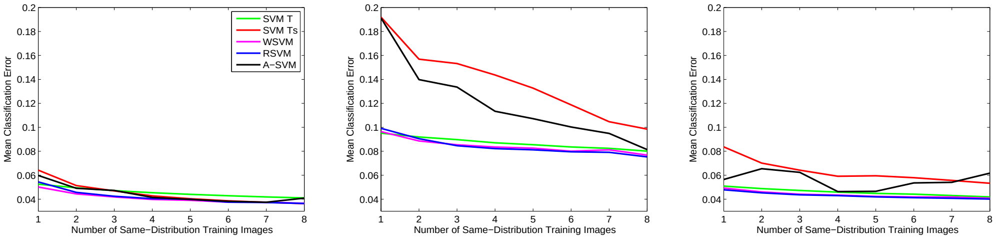


(a) Source 1, T1 (b) PD (c) FLAIR (d) $P(y|\boldsymbol{x})$ (e) Segmentation

(f) Source 2, T1 (g) T2 (h) FLAIR (i) $P(y|\boldsymbol{x})$ (j) Segmentation

(k) Source 3, T1 (l) T2 (m) FLAIR (n) $P(y|\boldsymbol{x})$ (o) Segmentation

Figure 6. Examples of resulting segmentations of the RSVM classifier on 33 features after training on T_s samples from eight images and 4 000 T_d samples. Figures (d),(i),(n) show the posterior outputs of the classifier, and Figures (e),(j),(o) show the final segmentation in blue, the manual segmentation in yellow, and the overlap between the two in green. The true positive rates (TPRs) and false positive rates (FPRs) for the showed slices were (e): TPR=92%, FPR=14%, (j): TPR=47%, FPR=49%, (o): TPR=48%, FPR=45%.



(a) range matching 4-96% (b) range matching min-max (c) Nyúl et al. [21] normalization

Figure 7. Learning curves for WML/MSL segmentation, showing mean classification error as a function of the number of T_s samples on 13 features for three different normalization techniques. (a) equals the image in Fig. 5(b) and includes range matching between the 4th and 96th percentile, (b) includes range matching between the minimum and maximum value, and (c) includes the normalization of Nyúl et al. [21].

showed the worst results, never outperforming the two baseline supervised-learning classifiers. It especially performed badly for lesion segmentation, where classification errors increased for an increasing number of T_s samples. We think TrAdaBoost is likely to experience difficulties when there is class overlap in the T_s samples. Since the weights of misclassified T_s samples are increased, this can make the classifier focus too much on a few initially misclassified T_s samples. The performance of A-SVM was dependent on the classification task. It performed very well on the WM/GM/CSF segmentation experiments when more than 15 T_s samples were available, in most cases greatly outperforming all other classifiers, whereas it did not give an overall good performance on the lesion-segmentation experiments. A-SVM is the only classifier of the four that does not explicitly take the T_d samples into account. It therefore incorporates less knowledge of the distribution of the T_d samples, such as the amount of class overlap and the class prior probabilities, which might be disadvantageous in some cases. Further investigating which of the different transfer classifiers are best suitable for which situation might be an interesting direction for future work.

We also investigated the influence of three image normalization techniques. In our experiments min-max normalization led to larger classification errors than the 4th-96th percentile range matching, both between and within sources. The more extensive normalization method of Nyúl et al. [21] overall slightly increased classification errors for the WM/GM/CSF segmentation experiments, and slightly decreased errors for WML and MSL segmentation. This is in contrast to results by Shah et al. [27], who showed that this normalization can greatly improve performance on images from different sources. For all normalization techniques a transfer-learning classifier could still bring improvement over the regular classifiers, indicating that although a more advanced normalization procedure could reduce differences between images from different scanners, transfer learning is still beneficial.

In the experiments the SVM parameters C and γ were determined with a regular SVM on T_s, which gave the regular SVMs an advantage over the transfer classifiers. The performance of the transfer classifiers may therefore still be improved by determining the optimal C and γ for each classifier separately, for instance by grid search on the different-distribution sources. To facilitate the large number of experiments required for the learning curves however, we chose to keep these parameters fixed. The classifier-specific parameters of the transfer classifiers were tuned using cross validation on the different-distribution sources, assuming that the differences and similarities between those sources were representative of the differences and similarities that can be expected in general between T_s and T_d data. Another option would be to include T_s data when determining the transfer parameters. As we wanted to study the behavior of the transfer classifiers also for very few T_s samples, this technique was only used in the experiments for the MS-lesion challenge. It could also be beneficial to apply different transfer parameters for each of the different-distribution sources, depending on the similarity with the target data. Exploring other ways of determining classifier parameters will be a topic of further research.

The three transfer classifiers WSVM, RSVM, and A-SVM provided good segmentations in comparison to results reported in literature. In Table I we compared our WM/GM/CSF segmentations to segmentations obtained with SPM8 [3], a state-of-the-art brain-tissue segmentation tool. On all four sources WSVM, RSVM, and A-SVM outperformed SPM8. In Table II we reported the performance of various methods on the IBSR data with 20 subjects. Our transfer classifiers outperformed 12 of the 16 methods. One of the methods that outperformed our classifiers was trained and tested in cross validation, using many more same-distribution training images than our methods, and the other three used a much more sophisticated bias-correction scheme. Using our methods as part of such a scheme could increase the performance on this dataset. Also, in the MRBrainS13[4] brain-tissue-segmentation challenge our SVM classification scheme ranked second, only to be beaten by a semi-automatic method. In the MS-lesion challenge our RSVM ranked second out of nine methods on all test data, and tenth out of 26 methods on a subset.

For MRI brain-tissue segmentation several other techniques have been developed to facilitate image segmentation across scanners. Cocosco et al. [5] used a registered probabilistic tissue atlas to automatically select "training" samples from target images, based on which a kNN classifier was trained to segment the whole image. Freesurfer [11] first automatically segments the voxels with the highest intensities (within a brain mask) as WM, after which the GM is identified by dilation of the WM tissue following the intensity gradients up until the point where a decrease in intensity indicates the boundary between GM and CSF. A different often used approach is unsupervised classification by the expectation-maximization (EM) algorithm [19], [34], [38], [39]. Here segmentation of the target data is performed by alternating between optimization of the source-specific model parameters given the segmentation of the previous step, and optimizing the segmentation given the determined model parameters. The state-of-the-art brain-tissue-segmentation method SPM is also based on such an EM-optimization [3]. All these methods do not use any labeled samples of the target data. This makes it easy to apply these techniques to new data. However, as our experiment prove superiority of transfer learning over SPM, we may conclude that a small amount of manually labeled T_s data used in a transfer-learning framework, can greatly improve the performance.

Many of the techniques mentioned above combine voxelwise classification with atlas-based prior tissue probabilities, partial-volume modeling, and/or Markov-Random-Field modeling. In this work we have restricted ourselves to voxelwise classification, to allow for a direct comparison of the different learning techniques. However, the established transfer-learning framework could also be used as the basis of a more advanced segmentation scheme, replacing the voxelwise classification step in any of the mentioned techniques.

In the experiments we have focused on MRI brain segmentation. However, the variability in imaging protocols forms a common problem across most applications. We expect that transfer learning can also improve supervised algorithms in many other segmentation and image analysis tasks.

We believe that transfer learning is a promising approach to biomedical image analysis. In applications for which data with ground truth labels is available from other studies, transfer learning can significantly decrease the amount of representative training data needed. This facilitates the application of supervised techniques in large multi-center studies and in clinical practice.

VII. Acknowledgments

This research was performed as part of the research project 'Transfer learning in biomedical image analysis' which is financed by The Netherlands Organization for Scientific Research (NWO).

The IBSR MR brain data sets and their manual segmentations were provided by the Center for Morphometric Analysis at Massachusetts General Hospital.

References

[1] P. Anbeek, K.L. Vincken, G.S. Van Bochove, M.J. Van Osch, J. van der Grond, et al. Probabilistic segmentation of brain tissue in MR imaging. *Neuroimage*, 27(4):795, 2005.

[4] http://mrbrains13.isi.uu.nl/

[2] P. Anbeek, K.L. Vincken, M.J.P. van Osch, R.H.C. Bisschops, and J. van der Grond. Automatic segmentation of different-sized white matter lesions by voxel probability estimation. *Medical image analysis*, 8(3):205–215, 2004.

[3] J. Ashburner and K.J. Friston. Unified segmentation. *Neuroimage*, 26(3):839–851, 2005.

[4] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[5] C.A. Cocosco, A.P. Zijdenbos, and A.C. Evans. A fully automatic and robust brain MRI tissue classification method. *Medical Image Analysis*, 7(4):513–527, 2003.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[7] W. Dai, Q. Yang, G.R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[8] R. De Boer, H.A. Vrooman, M.A. Ikram, M.W. Vernooij, M. Breteler, A. Van Der Lugt, and W.J. Niessen. Accuracy and reproducibility study of automatic MRI brain tissue segmentation methods. *Neuroimage*, 51(3):1047–1056, 2010.

[9] J.C. De Groot, F.E. De Leeuw, M. Oudkerk, J. Van Gijn, A. Hofman, J. Jolles, and M. Breteler. Periventricular cerebral white matter lesions predict rate of cognitive decline. *Annals of neurology*, 52(3):335–341, 2002.

[10] L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[11] B. Fischl. FreeSurfer. *Neuroimage*, 62(2):774–781, 2012.

[12] J. Folkesson, E.B. Dam, O.F. Olsen, P.C. Pettersen, and C. Christiansen. Segmenting articular cartilage automatically using a voxel classification approach. *Medical Imaging, IEEE Transactions on*, 26(1):106–115, 2007.

[13] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[14] E. Geremia, O. Clatz, B.H. Menze, E. Konukoglu, A. Criminisi, and N. Ayache. Spatial decision forests for ms lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011.

[15] H. Greenspan, A. Ruf, and J. Goldberger. Constrained Gaussian mixture model framework for automatic segmentation of MR brain images. *Medical Imaging, IEEE Transactions on*, 25(9):1233–1245, 2006.

[16] M.A. Ikram, A. van der Lugt, W.J. Niessen, G.P. Krestin, P.J. Koudstaal, A. Hofman, M.M.B. Breteler, and M.W. Vernooij. The Rotterdam Scan Study: design and update up to 2012. *European journal of epidemiology*, 26(10):811–824, 2011.

[17] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 264, 2007.

[18] F. Liu, D. Xu, M.S. Ferguson, B. Chu, T. Saam, N. Takaya, T.S. Hatsukami, C. Yuan, and W.S. Kerwin. Automated in vivo segmentation of carotid plaque mri with morphology-enhanced probability maps. *Magnetic Resonance in Medicine*, 55(3):659–668, 2006.

[19] J.L. Marroquin, B.C. Vemuri, S. Botello, E. Calderon, and A. Fernandez-Bouzas. An accurate and efficient Bayesian method for automatic segmentation of brain MRI. *Medical Imaging, IEEE Transactions on*, 21(8):934–945, 2002.

[20] A. Mayer and H. Greenspan. An adaptive mean-shift framework for MRI brain segmentation. *Medical Imaging, IEEE Transactions on*, 28(8):1238–1250, 2009.

[21] L.G. Nyul, J.K. Udupa, and X. Zhang. New variants of a method of MRI scale standardization. *Medical Imaging, IEEE Transactions on*, 19(2):143–150, 2000.

[22] S.J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[23] Z. Peng, W. Wee, and J.H. Lee. Automatic segmentation of MR brain images using spatial-varying Gaussian mixture and Markov random field approach. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 80–80. IEEE, 2006.

[24] N.D. Prins, E.J. van Dijk, T. den Heijer, S.E. Vermeer, P.J. Koudstaal, M. Oudkerk, A. Hofman, and M.M.B. Breteler. Cerebral white matter lesions and the risk of dementia. *Archives of neurology*, 61(10):1531–1534, 2004.

[25] J.C. Rajapakse and F. Kruggel. Segmentation of MR images with intensity inhomogeneities. *Image and Vision Computing*, 16(3):165–180, 1998.

[26] B. Scholkopf and A.J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[27] M. Shah, Y. Xiao, N. Subbanna, S. Francis, D.L. Arnold, D.L. Collins, and T. Arbel. Evaluating intensity normalization on mris of human brain with multiple sclerosis. *Medical image analysis*, 15(2):267–282, 2011.

[28] D.W. Shattuck, S.R. Sandor-Leahy, K.A. Schaper, D.A. Rottenberg, R.M. Leahy, et al. Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876, 2001.

[29] M.Y. Siyal and L. Yu. An intelligent modified fuzzy *c*-means based algorithm for bias estimation and segmentation of brain MRI. *Pattern recognition letters*, 26(13):2052–2062, 2005.

[30] J.G. Sled, A.P. Zijdenbos, and A.C. Evans. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *Medical Imaging, IEEE Transactions on*, 17(1):87–97, 1998.

[31] S.M. Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.

[32] M. Styner, J. Lee, B. Chin, M. Chin, O. Commowick, H. Tran, S. Markovic-Plese, V. Jewells, and S. Warfield. 3D segmentation in the clinic: A grand challenge II: MS lesion segmentation. *MIDAS Journal*, pages 1–5, 2008.

[33] J. Tohka, A. Zijdenbos, A. Evans, et al. Fast and robust parameter estimation for statistical partial volume models in brain MRI. *Neuroimage*, 23(1):84–97, 2004.

[34] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based tissue classification of MR images of the brain. *Medical Imaging, IEEE Transactions on*, 18(10):897–908, 1999.

[35] A. van Opbroek, M. Ikram, M. Vernooij, and M. de Bruijne. Supervised image segmentation across scanner protocols: A transfer learning approach. *Machine Learning in Medical Imaging*, pages 160–167, 2012.

[36] S.E. Vermeer, M. Hollander, E.J. van Dijk, A. Hofman, P.J. Koudstaal, and M.M.B. Breteler. Silent brain infarcts and white matter lesions increase stroke risk in the general population the rotterdam scan study. *Stroke*, 34(5):1126–1129, 2003.

[37] H.A. Vrooman, C.A. Cocosco, F. van der Lijn, R. Stokking, M A. Ikram, M.W. Vernooij, M.M. Breteler, and W.J. Niessen. Multi-spectral brain tissue segmentation using automatically trained k-nearest-neighbor classification. *Neuroimage*, 37(1):71, 2007.

[38] W.M. Wells III, W.E.L. Grimson, R. Kikinis, and F.A. Jolesz. Adaptive segmentation of MRI data. *Medical Imaging, IEEE Transactions on*, 15(4):429–442, 1996.

[39] M. Wels, Y. Zheng, M. Huber, J. Hornegger, and D. Comaniciu. A discriminative model-constrained EM approach to 3D MRI brain tissue classification and intensity non-uniformity correction. *Physics in Medicine and Biology*, 56(11):3269, 2011.

[40] A.J. Worth. The Internet Brain Segmentation Repository (IBSR).

[41] P. Wu and T.G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *Proceedings of the twenty-first international conference on Machine learning*, page 110. ACM, 2004.

[42] J. Yang, R. Yan, and A.G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.

[43] Z. Yi, A. Criminisi, J. Shotton, and A. Blake. Discriminative, semantic segmentation of brain tissue in MR images. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 558–565, 2009.

[44] Y. Zhou and J. Bai. Atlas-based fuzzy connectedness segmentation and intensity nonuniformity correction applied to brain MRI. *Biomedical Engineering, IEEE Transactions on*, 54(1):122–129, 2007.