HHS Public Access

Author manuscript

IEEE Trans Med Imaging. Author manuscript; available in PMC 2018 November 01.

Published in final edited form as:

IEEE Trans Med Imaging. 2017 November ; 36(11): 2319–2330. doi:10.1109/TMI.2017.2721362.

Auto-context Convolutional Neural Network (Auto-Net) for Brain Extraction in Magnetic Resonance Imaging

Seyed Sadegh Mohseni Salehi[*] **[Student Member, IEEE]**, **Deniz Erdogmus**[*] **[Senior Member, IEEE]**, and **Ali Gholipour**[†] **[Senior Member, IEEE]**

[*]Electrical and Computer Engineering Department, Northeastern University, Boston, MA, 02115

[†]Radiology Department, Boston Children's Hospital; and Harvard Medical School, Boston MA 02115

Abstract

Brain extraction or whole brain segmentation is an important first step in many of the neuroimage analysis pipelines. The accuracy and robustness of brain extraction, therefore, is crucial for the accuracy of the entire brain analysis process. State-of-the-art brain extraction techniques rely heavily on the accuracy of alignment or registration between brain atlases and query brain anatomy, and/or make assumptions about the image geometry; therefore have limited success when these assumptions do not hold or image registration fails. With the aim of designing an accurate, learning-based, geometry-independent and registration-free brain extraction tool in this study, we present a technique based on an auto-context convolutional neural network (CNN), in which intrinsic local and global image features are learned through 2D patches of different window sizes. We consider two different architectures: 1) a voxelwise approach based on three parallel 2D convolutional pathways for three different directions (axial, coronal, and sagittal) that implicitly learn 3D image information without the need for computationally expensive 3D convolutions, and 2) a fully convolutional network based on the U-net architecture. Posterior probability maps generated by the networks are used iteratively as context information along with the original image patches to learn the local shape and connectedness of the brain to extract it from non-brain tissue.

The brain extraction results we have obtained from our CNNs are superior to the recently reported results in the literature on two publicly available benchmark datasets, namely LPBA40 and OASIS, in which we obtained Dice overlap coefficients of 97.73% and 97.62%, respectively. Significant improvement was achieved via our auto-context algorithm. Furthermore, we evaluated the performance of our algorithm in the challenging problem of extracting arbitrarily-oriented fetal brains in reconstructed fetal brain magnetic resonance imaging (MRI) datasets. In this application our voxelwise auto-context CNN performed much better than the other methods (Dice coefficient: 95.97%), where the other methods performed poorly due to the non-standard orientation and geometry of the fetal brain in MRI. Through training, our method can provide accurate brain

Corresponding author: S.S.M.Salehi (ssalehi@ece.neu.edu).

Relevant code can be found at: https://github.com/SadeghMSalehi/AutoContextCNN

extraction in challenging applications. This in-turn may reduce the problems associated with image registration in segmentation tasks.

Index Terms

Brain extraction; Whole brain segmentation; MRI; Convolutional neural network; CNN; U-net; Auto-Context

I. Introduction

Whole brain segmentation, or brain extraction, is one of the first fundamental steps in the analysis of magnetic resonance images (MRI) in advanced neuroimaging applications such as brain tissue segmentation and volumetric analysis [1], longitudinal and group analysis [2], cortical and sub-cortical surface analysis and thickness measurement [3], [4], and surgical planning. Manual brain extraction is time consuming especially in large-scale studies. Automated brain extraction is necessary but its performance and accuracy are critical as the output of this step can directly affect the performance of all following steps.

Recently neural networks and deep learning have attracted enormous attention in medical image processing. Brebisson et.al. [5] proposed the SegNet, a convolutional neural network system to segment different parts of the brain. Recently, CNN-based methods have also been used successfully in tumor segmentation [6], [7], [8], brain lesion segmentation [9], [10], and infant brain image segmentation [11]. In what follows we review the state-of-the-art in whole brain segmentation and the related work that motivated this study. We then introduce a CNN-based method that generates accurate brain extraction.

II. Related Work

Many algorithms have been developed and continuously improved over the past decade for whole brain segmentation, which has been a necessary component of large-scale neuroscience and neuroimage analysis studies. As the usage of these algorithms dramatically grew, the demand for higher accuracy and reliability also increased. Consequently, while fully-automated, accurate brain extraction has already been investigated extensively, it is still an active area of research. Of particular interest is a recent deep learning based algorithm [12] that has shown to outperform most of the popular routinely-used brain extraction tools.

The state-of-the-art brain extraction methods and tools use evolved combinations of image registration, atlases, intensity and edge feature information, and level sets/graph cuts to generate brain masks in MRI images. The majority of these algorithms rely heavily on the alignment of the query images to atlases or make strong assumptions about the geometry, orientation, and image features. Yet the outcome of most of these tools is often inaccurate and involves non-brain structures or cuts parts of the brain. Therefore most of these tools offer options and multiple parameters to set and try, that ultimately make brain extraction a semi-automatic or supervised task rather than fully automatic.

Among brain extraction methods four algorithms that are distributed with the widely-used neuroimage analysis software packages, have been evolved and are routinely used. These are the Brain Extraction Tool (BET) from FSL [13], [14], 3dSkullStrip from the AFNI toolkit [15], the Hybrid Watershed Algorithm (HWA) from FreeSurfer [16], and Robust Learning-Based Brain Extraction (ROBEX) [17]. BET expands a deformable spherical surface mesh model initialized at the center-of-gravity of the image based on local intensity values and surface smoothness. 3dSkullStrip, which is a modified version of BET, uses points outside of the expanding mesh to guide the borders of the mesh. HWA uses edge detection for watershed segmentation along with an atlas-based deformable surface model. ROBEX fits a triangular mesh, constrained by a shape model, to the probabilistic output of a brain boundary classifier based on random forests. Because the shape model alone cannot perfectly accommodate unseen cases, Robex also uses a small free-form deformation which is optimized via graph cuts.

The current methods are prone to significant errors when certain geometric assumptions do not hold, features are not precisely identified, or image registration, which is often not guaranteed to converge to an exact solution, fails. The problems associated with registration-based segmentation, and the recent promising results in neural network based image segmentation motivate further development and use of learning-based, geometry-independent, and registration-free brain image segmentation.

Recently, Kleesiek et. al. [12] proposed a deep learning based algorithm for brain extraction, which will be referred to as PCNN in this paper. PCNN uses seven 3D convolutional layers for voxelwise image segmentation. Cubes of size 53×53×53 around the grayscale target voxel are used as inputs to the network. In the extensive evaluation and comparison reported in [12], PCNN outperformed state-of-the-art brain extraction algorithms in publicly available benchmark datasets.

In this study we introduce auto-context CNNs with two network architectures to significantly improve brain extraction accuracy. In our first network, which is a voxelwise architecture, instead of using 3D convolutional layers with one window size (used in PCNN), we use 2D patches of three different sizes as proposed by Moeskops et al. [18]. In addition, to account for 3D structure, and efficiently learn from 3D information to identify brain voxels from non-brain voxels, we use three parallel pathways of 2D convolutional layers in three planes (i.e. axial, coronal and sagittal planes). Our second architecture is a U-net [19] style network, in which we use a weighted cost function to balance the number of samples of each class in training. We discuss the details of our proposed auto-context networks, generally referred to as Auto-Net, in this paper.

Context information has shown to be useful in computer vision and image segmentation tasks. Widely-used models, such as conditional random fields [20], rely on fixed topologies thus offer limited flexibility; but when integrated into deep CNNs, they have shown significant gain in segmentation accuracy [21], [10]. To increase flexibility and speed of computations, several cascaded CNN architectures have been proposed in medical image segmentation [6], [8], [22]. In such networks, the output layer of a first network is concatenated with input to a second network to incorporate spatial correspondence of labels.

To learn and incorporate context information in our CNN architectures, we adopt the auto-context algorithm [23], which fuses low-level appearance features with high-level shape information. As compared to a cascaded network, an auto-context CNN involves a generic and flexible procedure that uses posterior distribution of labels along with image features in an iterative supervised manner until convergence. To this end, the model is flexible and the balance between context information and image features is naturally handled.

Experimental results in this study show that our Auto-Net methods outperformed PCNN and the four widely-used, publicly-available brain extraction techniques reviewed above on two benchmark datasets (i.e. LPBA40 and OASIS, described in Section IV.A). On these datasets we achieved significantly higher Dice coefficients by the proposed Auto-Nets compared to the routinely-used techniques, as autocontext significantly boosted sensitivity while improving or maintaining specificity. We also examined the performance of the Auto-Net in the challenging problem of extracting fetal brain from reconstructed fetal brain MRI. In this case we only compared our results to BET and 3dSkullStrip as the other methods were not designed to work with the non-standard orientation and geometry of the fetal brain in MRI. We present the methods, including the network architectures and the autocontext CNN, in the next section and follow with experimental results in Section IV and a discussion in Section V.

III. Method

A. Network Architecture

We design and evaluate two Auto-Nets with two different network architectures: 1) a voxelwise CNN architecture [24], and 2) a fully convolutional network [25], [26] based on the U-net architecture [19]. We describe the details of the network architectures here and follow with our proposed auto-context CNN algorithm in the next subsection.

1) A voxelwise network—The proposed network has nine types of input features and nine corresponding pathways which are merged in two levels. Each pathway contains three convolutional layers. This architecture segments a 3D image voxel-by-voxel. For all voxels in the 3D image three sets of in-plane patches in axial, coronal, and sagittal planes are used. Each set contains three patches with window sizes of 15 × 15, 25 × 25 and 51 × 51. By using these sets of patches with different window size, both local and global features of each voxel are considered during training. Network parameters are learned simultaneously based on orthogonal-plane inputs, so 3D features are learned without using 3D convolution which is computationally expensive.

Figure 1(a) shows the schematic architecture of the parallel 2D pathways for one of the 2D views. In the first layer, 24 5 × 5 kernels for the patches of size 15 × 15 and 25 × 25, and 7 × 7 kernels for the patches of size 51 × 51 are used. After the first convolutional layer, ReLU nonlinear function and batch normalization is applied. For the second convolutional layer, ReLU nonlinear function is used after applying convolutional layer with 32 convolutional kernels of sizes 3 × 3, 3 × 3 and 5 × 5, for each patch, respectively. In the last convolutional layer 48 kernels of size 3 × 3 are used. In the proposed architecture, fully convolutional layers are used instead of fully connected layers [27] to achieve much faster testing time, as

the whole image can be tested in a network with convolutional layers while voxels are tested in a network with fully connected layers. After applying ReLU function, the output of the third convolutional layer is connected to a convolution-type, fully-connected layer with 256 kernels. Then, the nodes for each patch are concatenated and a 1 × 1 convolution with 64 kernels is applied. Each of the 2D pathways collects the information of a 2D plane.

To combine the information of 2D planes, the outputs of each set of in-plane patches are concatenated. This results in 192 nodes in total. Two kernels of 1 × 1 convolutional layers (for brain and non-brain classes) are applied on concatenated nodes with a softmax output layer. Figure 1(b) illustrates this step. We refer to this combination of three 2D pathways network as our 2.5D-CNN. By adding the auto-context algorithm to this architecture (Auto-2.5D-CNN), we aim to combine low-level features from patches with context information learned by the network to improve classification accuracy.

2) A fully convolutional network—The voxelwise approach has two drawbacks: 1) although using fully convolutional layers instead of fully connected layers makes the algorithm faster, it is still relatively slow; and 2) there is a tradeoff between finding local features and global features that involves choosing the window size around voxels. In the previous section we described how we conquered the latter problem by choosing different window sizes. Nonetheless, these drawbacks can also be addressed by using a fully convolutional network (FCN) [25]. To this end, we use the U-net [19] which consists of a contracting path that captures global features and an expanding path that enables precise localization.

The U-net style architecture is shown in Figure 1(c). This architecture consists of a contracting path (to the right) and an expanding path (to the left). The contracting path contains padded 3 × 3 convolutions followed by ReLU non-linear layers. A 2 × 2 max pooling operation with stride 2 is applied after every two convolutional layers. After each downsampling by the max pooling layers, the number of features is doubled. In the expanding path, a 2 × 2 upsampling operation is applied after every two convolutional layers, and the resulting feature map is concatenated to the corresponding feature map from the contracting path. At the final layer a 1 × 1 convolution with linear output is used to reach the feature map with a depth equal to the number of classes (brain or non-brain tissue). We refer to this network as the U-net, as we aim to augment it with the auto-context algorithm (Auto-U-net).

B. Auto-Context CNN

We propose auto-context convolutional neural networks by adopting the auto-context algorithm developed in [23]. Assuming m training image pairs $\{(X^{(j)}, Y^{(j)}), j = 1 \ldots m\}$, each 3D image is flattened into a 1D vector $X^{(j)}=(x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)})$ and its corresponding label image is flattened into the vector $Y^{(j)}=(y_1^{(j)}, y_2^{(j)}, \ldots, y_n^{(j)})$ where $y_i^{(j)}$ is the label of voxel i in image j. In each image the posterior probability of voxel i having label l, computed through a CNN $f_{y_i}(.)$, by the softmax classifier can be written as:

$$p(y_i=l|X(N_i))=\frac{e^{f_{y_l}(N_i)}}{\sum_c e^{f_{y_c}(N_i)}} \quad (1)$$

where N_i is the set of patches around voxel i, $i = 1, \ldots, n$, and c is the number of classes ($l = 0, \ldots, c-1$). During the optimization, the cross-entropy between the true distribution q and the estimated distribution p, i.e. $H(q, p) = -\Sigma_i\, q(y_i)\log p(y_i|X(N_i))$, is minimized. The true distribution follows the Dirac function, i.e. $q(y_i)$ is 1 for the true label and 0 otherwise. The cost function, therefore, would be:

$$H=-\sum_i \log\, p(y_i=\text{trueLabel}|X(N_i)) \quad (2)$$

In auto-context CNN, a sequence of classifiers is designed in a way that, to train each classifier, the posterior probabilities computed by the previous classifier are used as features. More specifically, for each image at step t the pair of $X(N_i)$, $p_{(t-1)}(N_i)$ is considered as a feature for classification of voxel i, where $p_{(t-1)}(N_i)$ is the posterior probability of voxels around voxel i. Algorithm 1 shows how the sequence of weights in the network are computed for the sequence of classifiers. The learned weights are used at test time for classification. The proof of convergence of Algorithm 1 is shown in Appendix A.

Algorithm 1

The auto-context CNN algorithm

The training MRI image pairs $\{(X^{(j)}, Y^{(j)}), j = 1...m\}$
construct uniform distribution of $p_0^{(j)}(N_i)$ on the labels;
repeat
  Make a training set $S_{(t)} = \{(y_i^{(j)}, (X^{(j)}(N_i), p_{(t-1)}^{(j)}(N_i)), j = 1...m, i = 1...n\}$;
  Train CNN network using architecture described in figure 1 (a,b: for voxel-wised, c: for FCN);
  Calculate $p_{(t)}^{(j)}(N_i)$ for $\{j = 1...m, i = 1...n\}$ using (1);
  Calculate H_t using (2);
  $I = |H_{(t)} - H_{(t-1)}|$;
until $I < \epsilon$;

To illustrate more on the effect of the auto-context algorithm, consider the first convolutional layer of each 2D pathway in the 2.5D-CNN. Suppose y is an input 3D patch result of concatenating the predicted label and data patches, and x is the output of the first layer for one of the kernels. For the convolution operation with kernel size k we have

$$x=\sum_{i=1}^{d} W_i * y_i + b \quad (3)$$

where W is a $k \times k \times d$ weight matrix, * is the 2D convolution operation, d is the depth of the input feature which is 2, and b is the bias. Expanding the summation in equation (3) we have

$$x = W_1 * y_1 + W_2 * y_2 + b \quad (4)$$

where W_1 and W_2 are $k \times k$ weight matrices corresponding to the intensity input (y_1) and label input (y_2), respectively. W_2 values are optimized such that they encode information regarding the shape of the brain labels, their respective location, and the connectedness of the labels. During the training of the network at step 0, the weights corresponding to the label input, W_2, are assigned much lower values than the weights corresponding to the intensity input (i.e. $W_2 << W_1$) since the label input carries no information about the image at the beginning. Note that $p_j^0(N_i)$ is constructed with uniform distribution over classes. On the other hand, in the following steps, the weights corresponding to the label input, W_2, are assigned higher values than the weights corresponding to the intensity input (i.e. $W_2 > W_1$). Consequently, in testing, the filters corresponding to the predicted labels are more effective than the filters corresponding to intensities.

C. Training

1) Voxelwise network—MRI image labels are often unbalanced. For brain extraction the number of non-brain voxels is on average roughly 10 times more than the number of brain voxels. The following process was used to balance the training samples: for each training image, 15000 voxels were randomly selected such that 50% of the training voxels were among border voxels. The voxels which had two different class labels in a cube of five voxels around them were considered border voxels. Of the remaining 50% of samples, 25% were chosen randomly from the brain class and 25% were chosen from the non-brain class.

For training, the cross-entropy loss function was minimized using ADAM optimizer [28]. Three different learning rates were employed during the training: In the first step, a learning rate of 0.001 was used with 5000 samples for each MRI data pair and 15 epochs. In the second step, learning rate of 0.0001 was used to update the network parameters with another 5000 samples for each MRI data and 15 epochs. Finally, the last 5000 samples for each MRI data were used with a learning rate of 0.00005 to update the network parameters. The total training time for this architecture was less than two hours.

2) Fully convolutional network—The output layer in the FCN consists of c planes, one per class ($c = 2$ in brain extraction). We applied softmax along each pixel to form the loss. We did this by reshaping the output into a $width \times height \times c$ matrix and then applying cross entropy. To balance the training samples between classes we calculated the total cost by computing the weighted mean of each class. The weights are inversely proportional to the probability of each class appearance, i.e. higher appearance probabilities led to lower weights. Cost minimization on 15 epochs was performed using ADAM optimizer [28] with an initial learning rate of 0.001 multiplied by 0.9 every 2000 steps. The training time for this

network was approximately three hours on a workstation with an Nvidia Geforce GTX1080 GPU.

Figure 1d illustrates the procedure of using Algorithm 1. To create patches for each voxel in the network, two sets of features are used; first, patches of different sizes around each voxel are considered as inputs, i.e. $X(N_i)$. Second, exact same patch windows are considered around the posterior probability maps calculated in the previous step, $p_j^{t-1}(N_i)$, as additional sets of inputs. The posterior probabilities are multiplied to the mean of the data intensity to be comparable with data intensities. Concatenating these two 2D features provides 3D inputs to the network in two different domains.

Training was stopped when it reached convergence, i.e. when the change in the cross-entropy cost function became asymptotically smaller than a predefined threshold ε:

$$I_t = |H_{(t)} - H_{(t-1)}| < \varepsilon \quad (5)$$

For testing, the auto-context algorithm was used with two steps.

IV. Experiments

A. Datasets

We evaluated our algorithm first on two publicly available benchmark datasets and then on fetal MRI data which exhibits specific challenges such as non-standard, arbitrary geometry and orientation of the fetal brain, and the variability of structures and features that surround the brain. We used two-fold cross-validation in all experiments. The output of all algorithms was evaluated against the ground truth which was available for the benchmark datasets and was manually obtained prior to this study for the fetal MRIs.

The first dataset came from the LONI Probabilistic Brain Atlas Project (LPBA40) [29]. This dataset consists of 40 T1-weighted MRI scans of healthy subjects with spatial resolution of 0.86 × 1.5 × 0.86 mm. The second dataset involved the first two disks of the Open Access Series of Imaging Studies (OASIS) [30]. This consisted of 77 1 × 1 × 1 mm T1-weighted MRI scans of healthy subjects and subjects with Alzheimer's disease.

The third dataset contained 75 reconstructed T2-weighted fetal MRI scans. Fetal MRI data was obtained from fetuses scanned at a gestational age between 19 and 39 weeks (mean=30.1, stdev=4.6) on 3-Tesla Siemens Skyra scanners with 18-channel body matrix and spine coils. Repeated multiplanar T2-weighted single shot fast spin echo scans were acquired of the moving fetuses, Ellipsoidal brain masks defining approximate brain regions and bounding boxes in the brain region were defined in ITKSNAP [31], and the scans were then combined through robust super-resolution volume reconstruction by either of the algorithms developed in [32] or [33] for motion correction and volume reconstruction at isotropic resolution of either 0.75 or 1 mm. Brain masks were manually drawn on the reconstructed images by two experienced segmenters. Manual brain extraction took between

1 to 4 hours per case depending on the age and size of the fetal brain and the quality of the images.

B. Results

To evaluate the performance of the algorithms, Dice overlap coefficient was used to compare the predicted brain mask P with ground truth mask (extracted manually) R. The Dice coefficient was calculated as follow:

$$D=\frac{2|P\cap R|}{|P|+|R|}=\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \quad (6)$$

where TP, FP, and FN are the true positive, false positive, and false negative rates, respectively. We also report specificity, $\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$, and sensitivity, $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$, to compare algorithms.

Figure 2 shows the Dice coefficient for the different steps of the training session for all datasets in the auto-context CNN algorithm. Improvement in the Dice coefficient is observed in both network architectures (U-net and 2.5D-CNN) through the steps of the auto-context algorithm.

Table I shows the results of our proposed method compared to the other methods on the two benchmark datasets. The results for PCNN were taken from [12]. Auto-context CNNs (Auto-Nets) showed the highest Dice coefficients among all methods, with an increase of about 0.8% over the best performing methods in the LPBA40 dataset. This significant boost in performance was achieved in Auto-Nets through the autocontext algorithm which, by incorporating local shape context information along with local patches, allowed a significant increase in sensitivity and an increase in specificity.

The main advantage of our CNN-based method was revealed in the fetal MRI application where the fetal brains were in different orientations and surrounded by a variety of non-brain structures. Figure 3 shows an example, and Table II shows the results of whole brain segmentation on reconstructed fetal MRI. Only Auto-Net styles, BET and 3dSkullStrip were included in this comparison as the other methods were not designed to work with arbitrary brain orientation in fetal MRI and thus performed poorly. As expected, the auto-context algorithm improved the results significantly, and the Auto-Nets performed much better than the other algorithms in this application, with average Dice coefficients that were more than 12% higher than the other techniques, and sensitivities that were higher by a margin of more than 20%. In fact, as seen in Figure 3, the other two algorithms generated conservative brain masks which resulted in high specificity (close to 1) but very low sensitivity. The Dice coefficient, sensitivity, and specificity, calculated based on the ground truth for this case, are shown underneath each image in this figure.

The effect of using the auto-context algorithm can also be seen in Figure 3, where the voxelwise and fully convolutional networks on the right (i.e. 2.5D and U-Net, respectively)

are the networks without auto-context. Three different improvements are observed after using auto-context steps. First, the label of the brain voxels considered as non-brain by the first networks in the middle of the brain voxels (i.e. false negatives) were changed to brain voxels (yellow arrows). Second, the very small number of the non-brain voxels considered as brain voxels in the first networks (white arrows) were changed to non-brain voxels. Third, the auto-context algorithm slightly pushed the edges of the brain to the outside (cyan arrows). These three improvements resulted in remarkable improvement in sensitivity at the cost of only a slight decrease in specificity in this case. The result is a significant boost in segmentation accuracy also shown by a significant increase in the Dice overlap coefficient.

It is worth noting that based on the data in Tables I and II the FCN (Auto-U-net) performed slightly better than the voxelwise CNN (Auto-2.5D-CNN) for the LPBA40 and OASIS datasets, but the voxelwise CNN outperformed FCN for the fetal MRI data. Our explanation is that there was higher level of commonality in shape and features of the samples in the LPBA40 and OASIS benchmark datasets compared to the fetal MRI dataset. This information was learned by the FCN, resulting in better performance compared to the voxelwise approach. For the fetal brain images that were arbitrarily located and oriented in the image space and surrounded by various structures, global geometric features were less important, and the voxelwise network performed better than the FCN as it learned and relied on 3D local image features.

Figure 4 shows an example of a challenging fetal MRI case, where the voxelwise approach (Auto 2.5D) performed much better than the FCN approach (Auto U-net) as well as the other methods (BET and 3dSkullStrip). As can be seen from both Figures 3 and 4, fetal brains can be in non-standard arbitrary orientations, and the fetal head may be surrounded by different tissue or organs such as the amniotic fluid, uterus wall or placenta, or other fetal body parts such as hands or feet, or the umbilical cord. Despite the challenges raised, our Auto-Net methods, in particular the voxelwise CNN performed significantly better than the other methods in this application.

Figure 5 shows the box plots of the Dice coefficient, sensitivity, and specificity of the different algorithms on all three datasets. Among the non-CNN methods Robex performed well and was comparable to the 2.5D-CNN on the benchmark datasets, but could not be used reliably in the fetal dataset because of the geometric assumptions and the use of an atlas. On the other hand, BET and 3dSkullStrip had more relaxed assumptions thus could be used, albeit with limited accuracy. It should be noted that none of these methods were designed and tested for fetal brain MRI, so it was not expected that they worked well under the conditions of this dataset. In all datasets, Auto-Nets performed significantly better than all other methods as the auto-context significantly improved the results of both CNN architectures (2.5D and U-net).

Paired t-test was used to compare the results of different algorithms. The Dice coefficient of the proposed algorithm, Auto-Net (both Auto-2.5D and Auto-U-net), was significantly higher than BET, 3dSkullStrip, Robex, and HWA for LPBA40 and OASIS datasets at α threshold of 0.001 ($p < 0.001$). Moreover, it revealed significant differences ($p < 0.001$) between the Dice coefficient of the proposed algorithm (Auto-2.5D and Auto-U-net) with

BET and 3dSkullStrip in fetal MRI. Paired t-test also showed significant improvement in the Dice coefficients obtained from the voxelwise network and the FCN through the use of the auto-context algorithm (i.e. Auto-2.5D vs. 2.5D and Auto-U-net vs. U-net).

Figure 6 shows logarithmic-scale average absolute error heat maps of the different algorithms on the LPBA40 dataset in the MNI atlas space [34]. These maps show where most errors occurred for each algorithm, and indicate that the Auto-Nets performed much better than the other methods in this dataset.

Table III shows the average testing time (in seconds) for each dataset and each algorithm. It should be mentioned that the testing time for all the CNN-based methods including the PCNN were measured on GPUs, whereas the testing time for all non-CNN based methods were measured on multicore CPUs, therefore this data does not directly compare the computational cost of different algorithms. It is also noteworthy that by using fully convolutional layers instead of fully connected layers in the 2.5D CNN architecture the testing time was decreased by a factor of almost 15 fold. Nonetheless, the FCN U-net is still significantly faster.

V. Discussion

Our proposed auto-context convolutional neural networks outperformed the recent deep learning method [12] and four widely-used brain extraction techniques that were continuously evolved and improved over the past decade due to the significant demand for accurate and reliable automated brain extraction in the neuroscience and neuroimaging communities.

We achieved the highest Dice coefficients as well as a good sensitivity-specificity trade-off among the techniques examined in this paper. This was achieved by using the autocontext algorithm and FCN approach together for standard datasets and auto-context with multiple patch sizes as well as context information in a voxelwise CNN architecture.

While the auto-context FCN based on U-net was much faster than the auto-context voxelwise network, it performed only slightly better for the benchmark datasets. On the other hand, the auto-context voxelwise network performed much better than the auto-context FCN in the very challenging fetal MRI brain extraction problem. The auto-context algorithm dramatically improved the performance of both networks.

We trained and examined efficient voxelwise and FCN Auto-Nets in this paper. Extensions to 3D networks is analytically straightforward; but the 3D counterparts are typically more demanding on computational resources, in particular memory. Generally, in voxelwise networks each voxel is considered as an independent sample to be classified. A window or different-sized windows around voxels are chosen as features and the network is trained using those features. Kleesiek et al. [12] used one cube with constant window size around each voxel. Moeskops et al. [18] used different window sizes around voxels but in 2D views. The main reason that previous studies did not use both approaches together, is that the number of parameters increases significantly with 3D convolutional kernels, especially when different, typically large window sizes are used. Such a network can easily consume more

memory than what is available on most workstation GPUs. Our 2.5D network made a good trade-off in this regards.

To compare our 2.5D network (which consists of three 2D pathway networks) with its 3D counterpart, we calculate the number of parameters: The 2.5D network contains 68.81 million parameters whereas its 3D counterpart contains 793.84 million parameters. With direct implementation with a small batch size of 1, the 3D counterpart of our CNN consumes more than 40GB of GPU memory. On the other hand, in our 2.5D network architecture we efficiently used a batch size of 64. The nearest 3D counterpart of our network with similar memory usage contained two cubes with window sizes of 15 and 41. We tested this network on the LPBA40 dataset and observed 1.5% decrease in average Dice coefficients while the average testing time increased by a factor of 1.5. This architecture contained 154 million parameters. We also systematically evaluated the effect of the three pathways and different window sizes. To this end, we trained and tested networks with only one pathway in each plane. While the testing times were decreased by a factor of 4, we observed significant decrease in average Dice coefficients, at 2.8 – 4.3%. We also observed significant decrease in average Dice coefficients by using single window sizes instead of using different window sizes (i.e. 5%, 2.1%, and 0.9% drop in the Dice coefficients for window sizes of 15, 25, and 51, respectively).

With Auto-Net we overcome one of the persisting challenges in fetal brain MRI processing. The extraction of fetal brain from reconstructed fetal MRI previously required a significant amount of work to correct the masks provided by BET or other level set whole brain segmentation techniques [35], [36]. Atlas-based segmentation methods heavily rely on image registration which involves time-consuming search and optimization to match the arbitrary orientation of images [37], followed by deformable registration to age-matched templates [38], or patch-based label propagation [39], which are also time consuming and difficult due to the presence of residual non-brain tissue after initial alignments. Most of the work in the literature focused on brain detection and localization in original fetal brain MRI scans, mainly to improve automated motion correction and reconstruction, e.g. [40], [41], [42]. While accurate bounding boxes are detected around the fetal brain by these methods, leading to improved motion correction [41], the estimated brain masks are not exact and consequently the reconstructed images involve significant non-brain tissue. Therefore accurate brain extraction is critically needed after reconstruction. Rather than being dependent on difficult and time-consuming image registration processes, the Auto-Net fetal brain extractions, proposed here, work at the voxel level to mask the fetal brains and prepare them for registration to an atlas space [43] for further analysis. Brain masks are also useful in other processing tasks, such as intensity non-uniformity correction [44], which poses significant challenges in fetal MRI as can be seen in Figure 4.

In comparison with other methods, the features in CNN-based methods are learnt through the training step and no hand-crafted features are needed. After training, these methods are fast in testing. We noted that these methods do not use image registration nor do they make assumptions about global image geometry. Rather, the networks learn to classify voxels based on local and shape image features. An inherent assumption in such learning-based methods is that a suitable training set is available. This is a strict assumption both in terms of

the existence of the training set and in that any new test image should have the same feature distribution as the training set. We used one modality in this study. It is expected that if multiple modalities, such as T1-weighted, T2-weighted, FLAIR, or CT images along with their training data are available and used, they result in increased accuracy. The only change in the architecture will be the additional third dimension of the kernel of the first convolutional layer.

VI. Conclusion

We developed and evaluated auto-context convolutional neural networks with two different architectures (a voxelwise network with three parallel 2D pathways, and a FCN style U-net) for whole-brain segmentation in 3D MRI. The auto-context CNNs outperformed a recent deep learning method and four widely-used brain extraction methods in two publicly available benchmark datasets and in the very challenging problem of extracting fetal brain from reconstructed fetal MRI. Unlike the current highly evolved brain extraction methods that use a combination of surface models, surface evolutions, and edge and intensity features, CNN-based methods do not use image registration or assume global geometric features such as certain orientations, but require suitable training data.

Acknowledgments

This study was supported in part by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health (NIH) grant R01 EB018988. The content of this work is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

Appendix A

Theorem 1

The cross-entropy cost function in Algorithm 1 monotonically decreases during the training.

Proof

To show that the cross-entropy cost function decreases monotonically, we show that the cost at each level will be smaller or at least equal to the cost at previous level. At the arbitrary step *t*,

$$H_t = -\sum_i \log\, p_{(t),i}(y_i) = -\sum_i \log\, p_{(t)}(y_i|(X^{(j)}(N_i), p_{(t-1)}(N_i))) \quad (7)$$

and

$$H_{t-1} = -\sum_i \log\, p_{(t-1),i}(y_i) \quad (8)$$

Also, note that the posterior probability is:

$$p_{(t)}(y_i=k|X(N(i),p_{(t-1)}(N_i))=\frac{e^{f_{y_k}(X(N_i),p_{(t-1)}(N_i))}}{\sum_c e^{f_{y_c}(X(N_i),p_{(t-1)}(N_i))}} \quad (9)$$

Using $f_{yk}(X(N_i), p_{(t-1)}(N_i)) = log\, p_{(t-1),i}(y_i)$ cross-entropy in level t will be equal to cross-entropy in level $t-1$. Since, during the training in step t we are minimizing the cross entropy cost function, $p_{(t),i}(y_i)$ should at least work better than $p_{(t-1),i}(y_i)$. Therefore:

$$H_{(t)} \leq H_{(t-1)} \quad (10)$$

References

1. Makropoulos A, Gousias IS, Ledig C, Aljabar P, Serag A, Hajnal JV, Edwards AD, Counsell SJ, Rueckert D. Automatic whole brain MRI segmentation of the developing neonatal brain. IEEE transactions on medical imaging. 2014; 33(9):1818–1831. [PubMed: 24816548]
2. Li G, Wang L, Shi F, Lyall AE, Lin W, Gilmore JH, Shen D. Mapping longitudinal development of local cortical gyrification in infants from birth to 2 years of age. The Journal of Neuroscience. 2014; 34(12):4228–4238. [PubMed: 24647943]
3. MacDonald D, Kabani N, Avis D, Evans AC. Automated 3D extraction of inner and outer surfaces of cerebral cortex from MRI. NeuroImage. 2000; 12(3):340–356. [PubMed: 10944416]
4. Clouchoux C, Kudelski D, Gholipour A, Warfield SK, Viseur S, Bouyssi-Kobar M, Mari J-L, Evans AC, Du Plessis AJ, Limperopoulos C. Quantitative in vivo MRI measurement of cortical development in the fetus. Brain Structure and Function. 2012; 217(1):127–139. [PubMed: 21562906]
5. de Brebisson, A., Montana, G. Deep neural networks for anatomical brain segmentation; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2015. p. 20-28.
6. Wachinger C, Reuter M, Klein T. DeepNAT: Deep convolutional neural network for segmenting neuroanatomy. NeuroImage. 2017
7. Pereira S, Pinto A, Alves V, Silva CA. Brain tumor segmentation using convolutional neural networks in MRI images. IEEE transactions on medical imaging. 2016; 35(5):1240–1251. [PubMed: 26960222]
8. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Pal C, Jodoin P-M, Larochelle H. Brain tumor segmentation with deep neural networks. Medical Image Analysis. 2016
9. Brosch T, Tang LY, Yoo Y, Li DK, Traboulsee A, Tam R. Deep 3D convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. IEEE transactions on medical imaging. 2016; 35(5):1229–1239. [PubMed: 26886978]
10. Kamnitsas K, Ledig C, Newcombe VF, Simpson JP, Kane AD, Menon DK, Rueckert D, Glocker B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. arXiv preprint arXiv:1603.05959. 2016
11. Zhang W, Li R, Deng H, Wang L, Lin W, Ji S, Shen D. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. NeuroImage. 2015; 108:214–224. [PubMed: 25562829]
12. Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, Bendszus M, Biller A. Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. NeuroImage. 2016; 129:460–469. [PubMed: 26808333]
13. Smith SM. Fast robust automated brain extraction. Human brain mapping. 2002; 17(3):143–155. [PubMed: 12391568]
14. Jenkinson M, Pechaud M, Smith S. BET2: MR-based estimation of brain, skull and scalp surfaces. 11th annual meeting of the organization for human brain mapping. 2005; 17:7.

15. Cox RW. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Computers and Biomedical research. 1996; 29(3):162–173. [PubMed: 8812068]
16. Lin G, Adiga U, Olson K, Guzowski JF, Barnes CA, Roysam B. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. Cytometry Part A. 2003; 56(1):23–36.
17. Iglesias JE, Liu C-Y, Thompson PM, Tu Z. Robust brain extraction across datasets and comparison with publicly available methods. IEEE transactions on medical imaging. 2011; 30(9):1617–1634. [PubMed: 21880566]
18. Moeskops P, Viergever MA, Mendrik AM, de Vries LS, Benders MJ, Išgum I. Automatic segmentation of MR brain images with a convolutional neural network. IEEE transactions on medical imaging. 2016; 35(5):1252–1261. [PubMed: 27046893]
19. Ronneberger, O., Fischer, P., Brox, T. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2015. U-net: Convolutional networks for biomedical image segmentation; p. 234-241.
20. Lafferty J, McCallum A, Pereira F, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the eighteenth international conference on machine learning, ICML. 2001; 1:282–289.
21. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062. 2014
22. Valverde S, Cabezas M, Roura E, González-Villà S, Pareto D, Vilanova JC, Ramió-Torrentà L, Rovira À, Oliver A, Lladó X. Improving automated multiple sclerosis lesion segmentation with a cascaded 3D convolutional neural network approach. NeuroImage. 2017
23. Tu Z, Bai X. Auto-context and its application to high-level vision tasks and 3D brain image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010; 32(10): 1744–1757. [PubMed: 20724753]
24. Ciresan D, Giusti A, Gambardella LM, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. Advances in neural information processing systems. 2012:2843–2851.
25. Long, J., Shelhamer, E., Darrell, T. Fully convolutional networks for semantic segmentation; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3431-3440.
26. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. IEEE transactions on pattern analysis and machine intelligence. 2017; 39(4):640–651. [PubMed: 27244717]
27. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, Le-Cun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229. 2013
28. Kingma D, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014
29. Shattuck DW, Prasad G, Mirza M, Narr KL, Toga AW. Online resource for validation of brain segmentation methods. NeuroImage. 2009; 45(2):431–439. [PubMed: 19073267]
30. Marcus DS, Wang TH, Parker J, Csernansky JG, Morris JC, Buckner RL. Open access series of imaging studies (OASIS): cross-sectional mri data in young, middle aged, nondemented, and demented older adults. Journal of cognitive neuroscience. 2007; 19(9):1498–1507. [PubMed: 17714011]
31. Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. Neuroimage. 2006; 31(3):1116–1128. [PubMed: 16545965]
32. Gholipour A, Estroff JA, Warfield SK. Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain MRI. IEEE transactions on medical imaging. 2010; 29(10): 1739–1758. [PubMed: 20529730]
33. Kainz B, Steinberger M, Wein W, Kuklisova-Murgasova M, Malamateniou C, Keraudren K, Torsney-Weir T, Rutherford M, Aljabar P, Hajnal JV, et al. Fast volume reconstruction from motion corrupted stacks of 2D slices. IEEE transactions on medical imaging. 2015; 34(9):1901–1913. [PubMed: 25807565]

34. Fonov V, Evans AC, Botteron K, Almli CR, McKinstry RC, Collins DL, et al. BDC. Group. Unbiased average age-appropriate atlases for pediatric studies. NeuroImage. 2011; 54(1):313–327. [PubMed: 20656036]
35. Gholipour A, Akhondi-Asl A, Estroff JA, Warfield SK. Multiatlas multi-shape segmentation of fetal brain MRI for volumetric and morphometric analysis of ventriculomegaly. Neuroimage. 2012; 60(3):1819–1831. [PubMed: 22500924]
36. Gholipour A, Estroff JA, Barnewolt CE, Connolly SA, Warfield SK. Fetal brain volumetry through MRI volumetric reconstruction and segmentation. International journal of computer assisted radiology and surgery. 2011; 6(3):329–339. [PubMed: 20625848]
37. Taimouri, V., Gholipour, A., Velasco-Annis, C., Estroff, JA., Warfield, SK. Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on. IEEE; 2015. A template-to-slice block matching approach for automatic localization of brain in fetal MRI; p. 144-147.
38. Tourbier S, Velasco-Annis C, Taimouri V, Hagmann P, Meuli R, Warfield SK, Cuadra MB, Gholipour A. Automated template-based brain localization and extraction for fetal brain MRI reconstruction. NeuroImage. 2017
39. Wright R, Kyriakopoulou V, Ledig C, Rutherford MA, Hajnal JV, Rueckert D, Aljabar P. Automatic quantification of normal cortical folding patterns from fetal brain MRI. NeuroImage. 2014; 91:21–32. [PubMed: 24473102]
40. Kainz, B., Keraudren, K., Kyriakopoulou, V., Rutherford, M., Hajnal, JV., Rueckert, D. Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on. IEEE; 2014. Fast fully automatic brain detection in fetal MRI using dense rotation invariant image descriptors; p. 1230-1233.
41. Keraudren K, Kuklisova-Murgasova M, Kyriakopoulou V, Malamateniou C, Rutherford MA, Kainz B, Hajnal JV, Rueckert D. Automated fetal brain segmentation from 2D MRI slices for motion correction. NeuroImage. 2014; 101:633–643. [PubMed: 25058899]
42. Keraudren, K., Kainz, B., Oktay, O., Kyriakopoulou, V., Rutherford, M., Hajnal, JV., Rueckert, D. International Conference on Medical Image Computing and Computer-Assisted Inter-vention. Springer; 2015. Automated localization of fetal organs in MRI using random forests with steerable features; p. 620-627.
43. Gholipour A, Rollins CK, Velasco-Annis C, Ouaalam A, Akhondi-Asl A, Afacan O, Ortinau CM, Clancy S, Limperopoulos C, Yang E, Estroff J, Warfield SK. A normative spatiotemporal MRI atlas of the fetal brain for automatic segmentation and analysis of early brain growth. Scientific Reports. 2017; 7(1):476. [PubMed: 28352082]
44. Tustison NJ, Avants BB, Cook PA, Zheng Y, Egan A, Yushkevich PA, Gee JC. N4ITK: improved N3 bias correction. IEEE transactions on medical imaging. 2010; 29(6):1310–1320. [PubMed: 20378467]

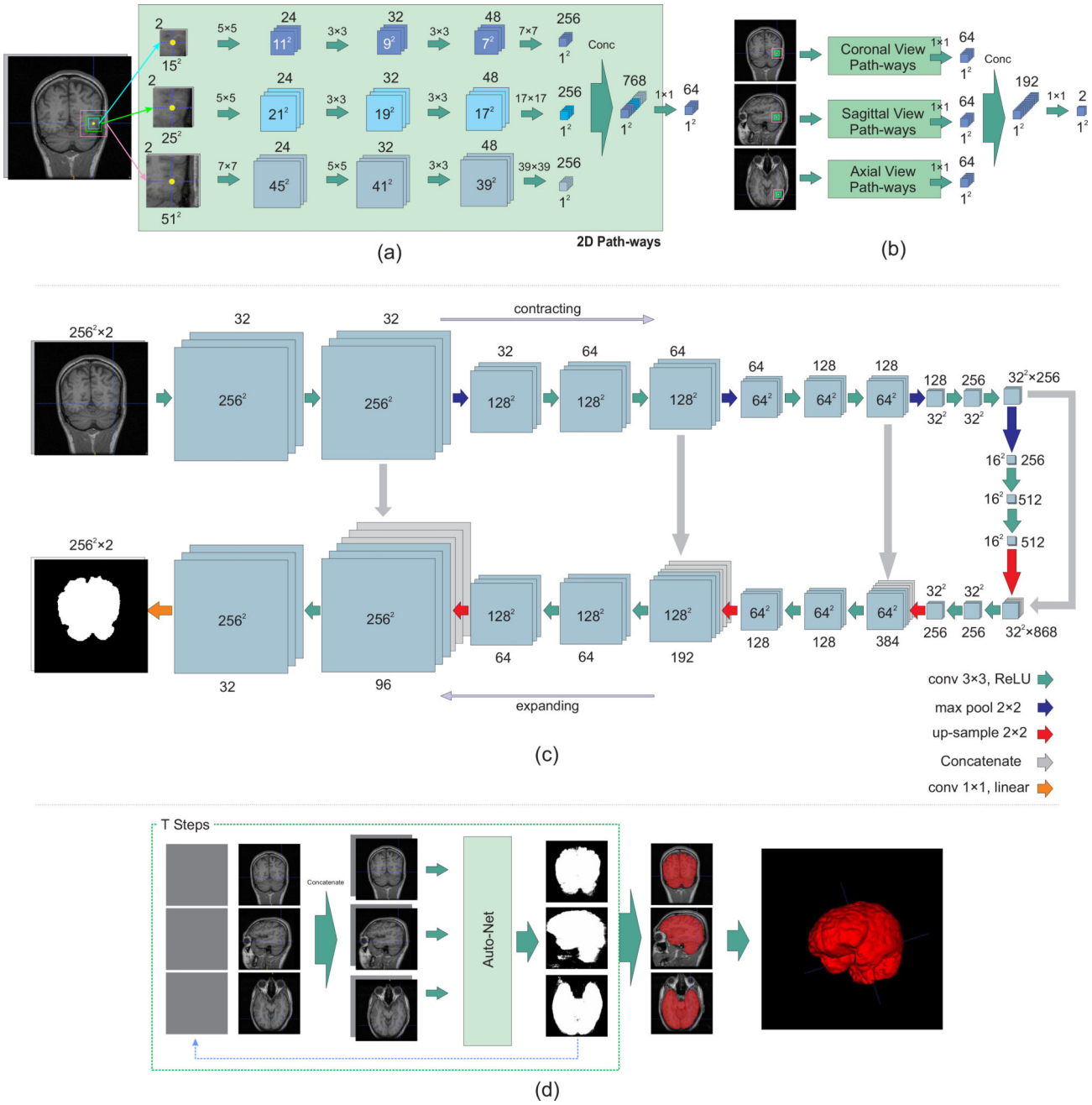

Fig. 1.

Schematic diagram of the proposed networks: a) The proposed voxelwise architecture for 2D image inputs; b) the network architecture to combine the information of 2D pathways for 3D segmentation; c) the U-net style architecture. The 2D input size for the LPBA40 and fetal MRI datasets was 256 × 256 and for the OASIS dataset was 176 × 176; and d) the auto-context formation of the network to reach the final results using network (a) as example. The context information along with multiple local patches are used to learn local shape information from training data and predict labels for the test data.

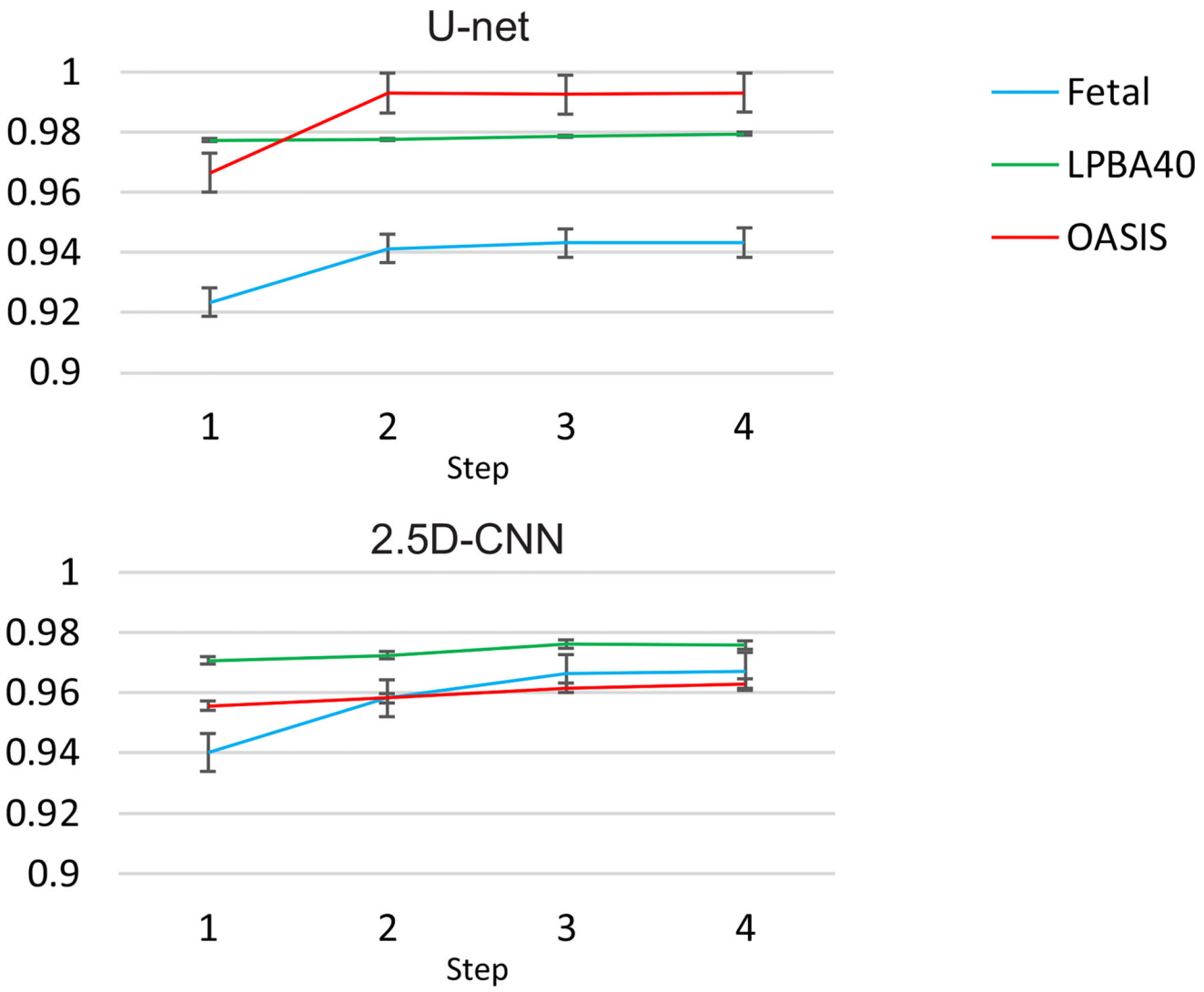

Fig. 2.

The Dice coefficient of training at four steps of the auto-context algorithm on all datasets based on the U-net (up) and the voxelwise 2.5D CNN approach (bottom). These plots show that the networks learned the context information through iterations and they converged.

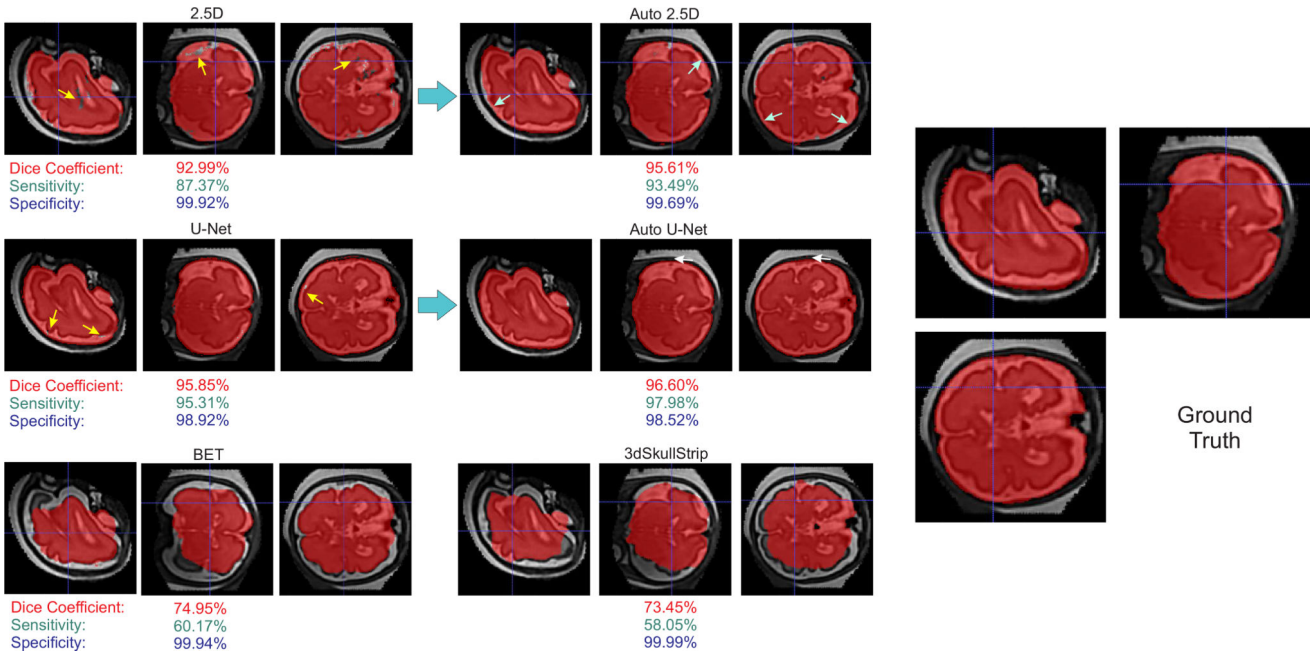

Fig. 3.

Predicted masks overlaid on the data for fetal brain MRI; the top images show the improvement of the predicted brain mask in different steps of the Auto-Net using 2.5D-CNN. The middle images show the improvement of the predicted brain mask in different steps of the Auto-Net using U-Net. The bottom left and right images show the predicted brain masks using BET and 3dSkullStrip, respectively. The right image shows the ground truth manual segmentation. Despite the challenges raised, our method (Auto-Net) performed very well and much better than the other methods in this application. The Dice coefficient, sensitivity, and specificity, calculated based on the ground truth for this case, are shown underneath each image in this figure.

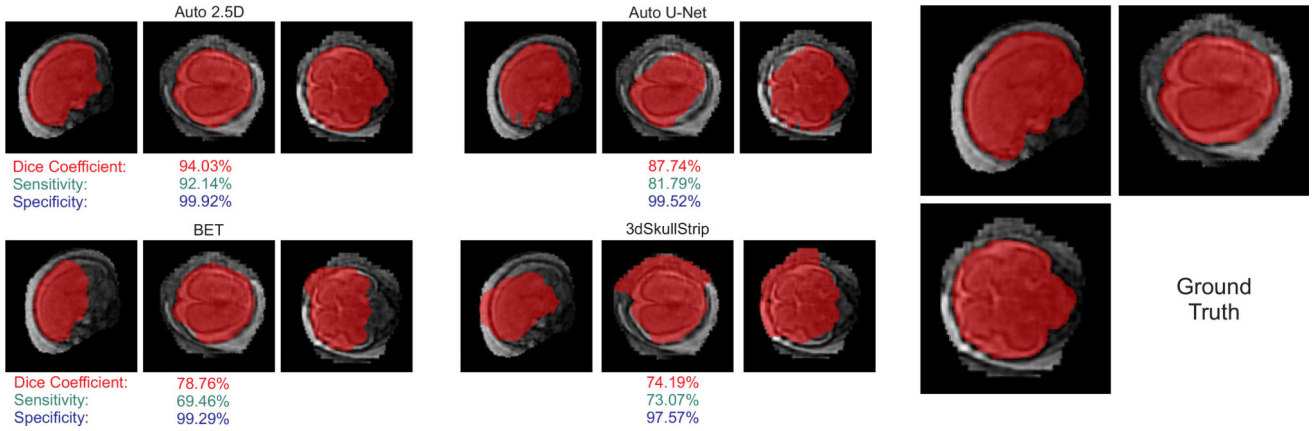

Fig. 4.

Predicted masks overlaid on the reconstructed fetal brain MRI for a challenging case with decent image reconstruction quality and intensity non-uniformity due to B1 field inhomogeneity; the top images show the predicted brain masks by Auto-Net using 2.5D-CNN (left) and U-net (right). The bottom left and right images show the predicted brain masks using BET and 3dSkullStrip, respectively. The right image shows the ground truth manual segmentation. As can be seen, fetal brains can be in non-standard arbitrary orientations. Moreover, the fetal head may be surrounded by different tissue or organs. Despite all these challenges, the Auto-2.5D CNN performed well and much better than the other methods in this case. The Dice coefficient, sensitivity, and specificity, calculated based on the ground truth, are shown underneath each image in this figure.

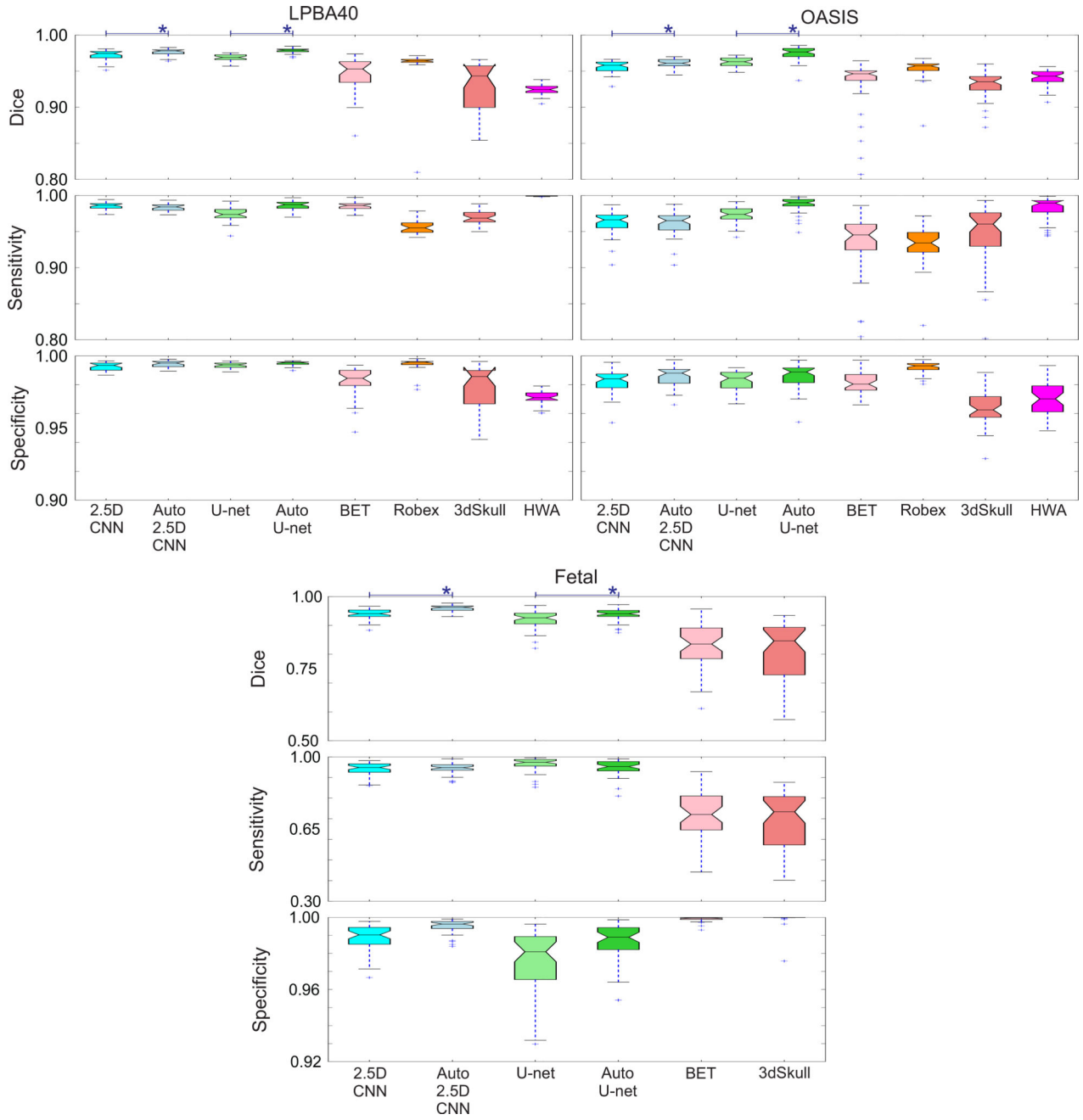

Fig. 5.

Evaluation scores (Dice, sensitivity, and specificity) for three data sets (LPBA40, OASIS, and fetal MRI). Median is displayed in boxplots; blue crosses represent outliers outside 1.5 times the interquartile range of the upper and lower quartiles, respectively. For the fetal dataset the registration-based algorithms were removed due to their poor performance. Those algorithms were not meant to work for images of this kind with non-standard geometry. Overall, these results show that our methods (Auto-Nets: Auto 2.5D and Auto U-net) made a very good trade-off between sensitivity and specificity and generated the highest Dice coefficients among all methods including the PCNN [12]. The performance of Auto-

Nets was consistently superior in the fetal MRI application where the other methods performed poorly due to the non-standard image geometry and features. Using Auto-context algorithm showed significant increase in Dice coefficients in both voxelwise and FCN style networks.

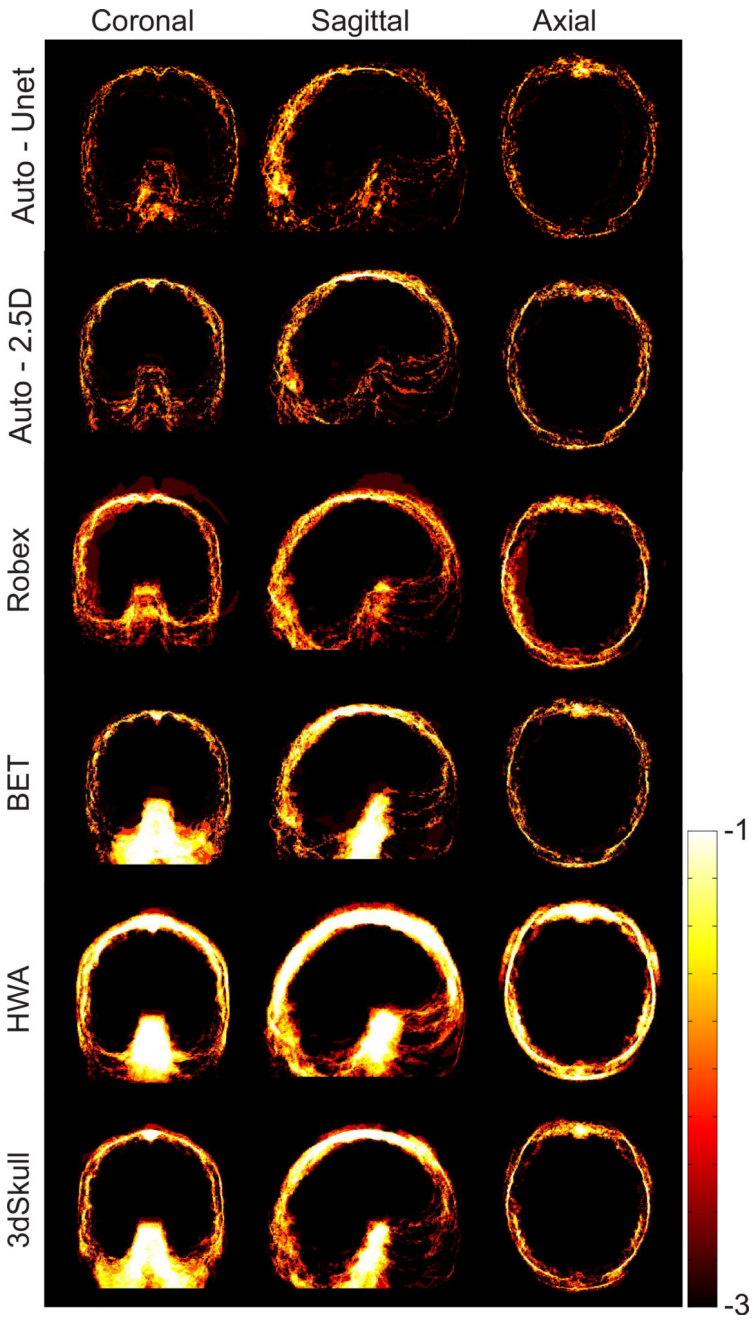

Fig. 6.

Logarithmic-scale absolute error maps of brain extraction obtained from six algorithms on the LPBA40 dataset. This analysis shows that Auto-Nets performed much better than the other methods in this dataset.

TABLE I

Mean and standard deviation of the scores for different algorithms on LPBA40 and OASIS datasets. The results show that our algorithm increased both sensitivity and specificity and resulted in highest Dice scores among all widely-used tools and the recent PCNN method [12].

	LPBA40			OASIS		
Method	**Dice**	**Sensitivity**	**Specificity**	**Dice**	**Sensitivity**	**Specificity**
Auto-U-net	**97.73** (±0.003)	98.31 (±0.006)	**99.48** (±0.001)	**97.62** (±0.01)	**98.66** (±0.01)	98.77 (±0.01)
U-net	96.79 (±0.004)	97.22 (±0.01)	99.34 (±0.002)	96.22 (±0.006)	97.29 (±0.01)	98.27 (±0.007)
Auto-2.5D-CNN	97.66 (±0.01)	98.25 (±0.01)	99.47 (±0.002)	96.06 (±0.007)	96.21 (±0.01)	98.56 (±0.006)
2.5D-CNN	97.17 (±0.005)	98.52 (±0.01)	99.24 (±0.002)	95.61 (±0.007)	96.3 (±0.01)	98.20 (±0.01)
PCNN	96.96 (±0.01)	97.46 (±0.01)	99.41 (±0.003)	95.02 (±0.01)	92.40 (±0.03)	**99.28** (±0.004)
BET	94.57 (±0.02)	98.52 (±0.005)	98.22 (±0.01)	93.44 (±0.03)	93.41 (±0.04)	97.70 (±0.02)
Robex	95.40 (±0.04)	94.25 (±0.05)	99.43 (±0.004)	95.33 (±0.01)	92.97 (±0.02)	99.21 (±0.004)
3dSkullStrip	92.99 (±0.03)	96.95 (±0.01)	97.87 (±0.01)	92.77 (±0.01)	94.44 (±0.04)	96.82 (±0.01)
HWA	92.41 (±0.007)	**99.99** (±0.0001)	97.07 (±0.004)	94.06 (±0.01)	98.06 (±0.01)	96.34 (±0.01)

TABLE II

Mean and standard deviation of the scores of different algorithms on the fetal dataset. The results show that highest Dice coefficients were obtained by Auto-Net compared to BET and 3dSkullStrip among the techniques that could be used in this application. Also, the voxelwise approach (Auto-2.5D-CNN) performed much better than the FCN (Auto-U-net) in this application.

Method	Dice	Sensitivity	Specificity
Auto-U-net	93.80(±0.02)	94.64(±0.04)	98.65(±0.01)
U-net	92.21(±0.03)	**96.46**(±0.03)	97.57(±0.01)
Auto-2.5D-CNN	**95.97**(±0.02)	94.63(±0.02)	99.53(±0.004)
2.5D-CNN	94.01(±0.01)	94.20(±0.03)	98.88(±0.008)
BET	83.68(±0.07)	73.00(±0.1)	99.91(±0.001)
3dSkullStrip	80.57(±0.12)	69.19(±0.16)	**99.97**(±0.001)

TABLE III

Average runtimes (seconds) of the methods compared in this study: the non-CNN methods were tested on an Intel(R) Core(TM) i7-5930K CPU with 3.50 GHz and 64 GB RAM for all data sets (LPBA40, OASIS and Fetal). The CNN-based methods were tested on an NVIDIA GeForce GTX 1080 (Pascal architecture). The PCNN timings are based on those reported in [12] using an NVIDIA Titan GPU with Kepler architecture.

Method	LPBA40	OASIS	Fetal
Auto-U-net	10.03	22.85	14.11
U-net	4.57	11.36	6.87
Auto-2.5D-CNN	794.42	641.26	501.73
2.5D-CNN	396.23	320.12	244.9
PCNN	36.51	40.99	-
BET	**2.04**	**1.96**	**1.62**
3dSkullStrip	130.4	119.12	82.72
Robex	52.10	63.25	-
HWA	18.73	13.42	-