Intelligent Labeling Based on Fisher Information for Medical Image Segmentation Using Deep Learning

Jamshid Sourati, Ali Gholipour, *Senior Member, IEEE*, Jennifer G. Dy, *Member, IEEE*, Xavier Tomas-Fernandez, Sila Kurugol, *Member, IEEE*, and Simon K. Warfield, *Fellow, IEEE*

***Abstract*— Deep convolutional neural networks (CNN) have recently achieved superior performance at the task of medical image segmentation compared to classic models. However, training a generalizable CNN requires a large amount of training data, which is difficult, expensive, and time-consuming to obtain in medical settings. Active Learning (AL) algorithms can facilitate training CNN models by proposing a small number of the most informative data samples to be annotated to achieve a rapid increase in performance. We proposed a new active learning method based on Fisher information (FI) for CNNs for the first time. Using efficient backpropagation methods for computing gradients together with a novel low-dimensional approximation of FI enabled us to compute FI for CNNs with a large number of parameters. We evaluated the proposed method for brain extraction with a patch-wise segmentation CNN model in two different learning scenarios: universal active learning and active semi-automatic segmentation. In both scenarios, an initial model was obtained using labeled training subjects of a source data set and the goal was to annotate a small subset of new samples to build a model that performs well on the target subject(s). The target data sets included images that differed from the source data by either age group (e.g. newborns with different image contrast) or underlying pathology that was not available in the source data. In comparison to several recently proposed AL methods and brain extraction baselines, the results showed that FI-based AL outperformed the competing methods in improving the performance of the model after labeling a very small portion of target data set (<0.25%).**

***Index Terms*— Convolutional neural network, active learning, Fisher Information, brain extraction, patch-wise segmentation.**

I. Introduction

IMAGE segmentation is an important component of automatic medical image analysis. Given the segmentation, important quantitative imaging markers of disease can be extracted for improved diagnosis, personalized treatment planning and monitoring response to therapy. Supervised deep learning models, including convolutional neural networks (CNNs) have gained tremendous success in performing segmentation tasks in the last few years [1]–[4]. These models have also been among the highest ranked competitors in medical image segmentation challenges.

Deep CNNs require a large amount of labeled data in order to generalize well to unseen test data. However, labeled samples are not available for many populations in medical settings, such as patients that come from different age groups (e.g., newborns) or have different pathological conditions. This mismatch between labeled data sets and target populations arises frequently in clinical applications and degrades the performance and generalization capacity of deep models. To address this problem, we must label additional samples from the targeted, unlabeled data set, which is very costly, especially in medical applications where both time and expertise are needed for labeling. In contrast, obtaining unlabeled data is becoming less expensive. Simply labeling all of these data is not affordable. The first and simplest solution, i.e. random selection of samples to label, is sub-optimal leading to an unnecessarily large labeled data set. To address these issues, *active learning (AL)* methods aim to intelligently select a limited number of most informative samples to be annotated by an expert, maximizing the prediction performance for large, unlabeled data sets.

A. Related Work

Among AL methods developed for CNNs, uncertainty sampling (US) has been one of the most popular methods [5]–[12],

Manuscript received January 17, 2019; revised March 12, 2019; accepted March 19, 2019. Date of publication March 27, 2019; date of current version October 25, 2019. This work was supported in part by NIH under Grant R01 NS079788, Grant R01 EB019483, Grant R01 DK100404, Grant IDDRC U54 HD090255, and in part by the Research Grant from the Boston Children's Hospital Translational Research Program. The work of A. Gholipour was supported in part by NIH under grants R01EB018988 and R01NS106030, and a Technological Innovations in Neuroscience Award from the McKnight Foundation. The work of J. G. Dy was supported by the NSF under Grant IIS-1546428. The work of S. Kurugol was supported in part by the Crohn's and Colitis Foundation of America (CCFA), and in part by the American Gastroenterological Association (AGA). *(Corresponding author: Jamshid Sourati.)*

J. Sourati, A. Gholipour, X. Tomas-Fernandez, S. Kurugol, and S. K. Warfield are with the Computational Radiology Laboratory, Radiology Department, Boston Children's Hospital, Boston, MA 02115 USA (e-mail: jamshid.sourati@childrens.harvard.edu).

J. G. Dy is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2019.2907805

which queries the most uncertain samples to be labeled. Various uncertainty measures have been used. Gal et. al [7] measured uncertainty via Monte-Carlo dropout and used it in a Bayesian setting in order to consider both aleatoric uncertainty —the noise in the data and epistemic uncertainty —the uncertainty over the parameters of the CNN. Heteroscedastic aleatoric uncertainties were also considered for selecting uncertain samples by [8] through building Bayesian neural networks that predict sample-dependent variance [13]. Zhou *et al.* [9] utilized uncertainty together with Kullback-Leibler metric to query slices for labeling. Recently, Beluch *et al.* [10] showed that estimating uncertainty based on network ensembles resulted in a better performing AL algorithm. Similarly, Kuo *et al.* [12] computed disagreement between an ensemble of networks through Jensen-Shannon entropy (as opposed to classic Shannon entropy) as an estimate for uncertainty in their AL technique. Uncertainty-based AL techniques were successful in training models with fewer labeled samples in comparison to passive learning (i.e. random query selection). Nevertheless, these methods are prone to 1) redundancy among queries and 2) outliers. Although some of the work mentioned above attempted to address these issues with ad-hoc solutions, the question that remains to be answered is whether one can solve these issues more principally by using an information theory based objective.

Other information gain measures have been proposed to address shortcomings of US, such as mutual information between unlabeled samples and the model parameters [7], [14], [15], which, however, did not improve uncertainty results for CNNs, and mutual information between query candidates and unlabeled samples [16], which is computationally intractable for CNN models. Geometrical AL algorithms that use geometrical measures to select the most informative samples are more suitable for large learning models. Group of methods combined uncertainty with pair-wise similarities between candidates and the rest of unlabeled population to address the issue of outlier selection [17], [18]. Iterative optimization of such similarity-based objectives could be used instead of simply ranking the candidates to address the redundancy as well [19], [20]. Moreover, recently Sener and Savarese [21] chose queries that best covered various parts of the feature space occupied by the rest of the unlabeled samples. Although these algorithms are scalable to complex CNN models, they become very slow for large pool of unlabeled samples, and as we will show in our experiments, their performance do not differ significantly from the US methods.

B. Contributions

Here, we proposed to use a more sophisticated objective based on Fisher information (FI) for active learning in deep CNNs. FI has been shown to be theoretically beneficial for active learning in classical shallow models such as the logistic regression classifier [22]–[24]. In statistics, FI is known to measure the amount of information that a single sample carries about the parameters of the underlying distribution from which it is drawn. Intuitively, FI-based AL selects those queries that are expected to carry more information about the optimal model parameters. Recent studies showed FI-based objectives outperformed US when applied in AL on logistic regression [25], [26]. However, FI-based AL has not previously been applied to CNN models. The main reason is the significantly large parameter space of the CNN models which leads to very large FI matrices that are difficult to form and manipulate. Here, we developed the first FI-based AL method for deep CNNs by means of a method that exploits dimensionality reduction in parameter space. This paper is an extension of our short preliminary work that was recently presented [27].

FI has been previously used with deep models for purposes different than active learning including natural gradient descent [28], [29] and parameter weighting [30]. Computing FI, or Hessian matrix of the loss function as an approximation to FI, is not practical for most neural network architectures with millions of parameters. In some specific applications, formation of the entire FI matrix can be avoided [31], but in others, including our FI-based AL problem, FI matrix needs to be explicitly computed. Few approximate formations have been proposed which usually impose certain structural assumptions over the model parameters. For example, (block) diagonal approximations of FI [30], [32] assume that (blocks of) the parameters are uncorrelated, and Kronecker factorization of the matrix blocks [29], [33] makes an extra assumption of independence between pre- and post-activations of the layers' outputs.

Our contribution in this paper is threefold:

- We developed a FI-based AL for CNNs by applying a tractable approximation of FI that is based on an implicit re-parameterization of the model. Our approximation technique makes no assumptions about the structure of the model parameters.
- We used and evaluated AL for CNNs in two transfer learning scenarios, where we fine-tuned a pre-trained model obtained from a source data set to 1) build a model that is generalizable to a target data set of multiple subjects with different properties than the source (*Universal AL*), and 2) specialize a model for a specific patient with different anatomy or pathology by personalized refinement of an insufficient initial segmentation (*active semi-automatic segmentation*). We showed that our proposed FI-based AL achieve at least 99.7% accuracy of a fully trained model by only labeling a small portion (less than 0.25%) of the target data in the universal AL.
- We evaluated a comprehensive list of the most recently proposed AL approaches and state-of-the-art segmentation methods for patch-wise brain extraction to 1) compare the relative effectiveness of different AL methods and objectives in a widely-studied, benchmark medical image segmentation task, and 2) assess whether our proposed FI-based AL method can outperform the other methods in both segmentation scenarios considered here.

The semi-automatic segmentation scenario is tightly related to interactive image segmentation. Despite the advances in segmentation methods, interactive tools are still needed in practice because the automated segmentation results on clinical

medical images are rarely flawless. This is due to several sources of variation in data such as inter-subject variations and/or variations due to use of different scanners or protocols [34]–[36]. Hence, user interaction is often needed to refine the segmentation results before they are used in clinic. Traditionally, the user's interactions for refinement of the results are fully manual, which include scrolling through all slices and exhaustively searching for mis-segmented regions to correct. Unfortunately, this process can be very labor intensive and time-taking. Instead, we proposed an intelligent system that can learn from the users limited feedback at any given time and automatically refines the algorithm so that the algorithm automatically correct similar mistakes in other parts of the image. In this scenario, AL was used to accelerate the process of interactive image segmentation by converting its fully manual process into a semi-automatic workflow. There are existing semi-automatic interactive segmentation methods, however, they either lack any AL component [37], [38], or simply use uncertainty sampling [39], [40]. In this paper, an interactive image segmentation framework is proposed based on our FI-based AL algorithm.

II. Methods

This section is organized as follows. Section II-A gives a short introduction to active learning, Section II-B explains our FI approximation methods. Lastly, Section II-C describes the recently introduced AL methods that we compare our approach with.

A. Preliminaries

Active learning (AL) is usually done in an iterative fashion. The pseudo-code of this algorithmic process is shown in Fig. 1. Each AL iteration consists of two phases: (1) query selection from a pool of unlabeled samples, and (2) model update using the newly labeled queries possibly mixed with the previously labeled samples. In line 3, $\mathcal{A}$ denotes a querying module that takes the current model and the unlabeled pool of samples and select a subset of unlabeled samples of size k (query batch size). The selected queries are then labeled by an expert (line 4) and added to the labeled data set (line 5). The expanded labeled data are then used in the second phase to update $\mathcal{M}_{t-1}$ to $\mathcal{M}_t$ (line 6). Previous research has shown that fine-tuning $\mathcal{M}_{t-1}$ with the labeled queries converges faster and to a more robust model than training all layers of the model from scratch [9], [41].

While the initial labeled samples $\mathcal{L}_0$ can be considered a non-empty set, e.g. those samples that are used to train $\mathcal{M}_0$, in this paper we start from an empty $\mathcal{L}_0$. This is because our goal is not necessarily to preserve the performance of the model with respect to the source data set. Instead, we aim to adapt the network to achieve highest accuracy possible in the target data set with minimum number of additional samples to be queried and labeled. Moreover, if size of the initial training data is much larger than k ($|\mathcal{L}_0| \gg k$), the new labeled queries from the target data set will be overwhelmed by $\mathcal{L}_0$ and consequently $\mathcal{M}_t$ will be too close to $\mathcal{M}_{t-1}$.

Inputs: Initial model $\mathcal{M}_0$, unlabeled sample pool $\mathcal{U}_0$, size of query batch k
Outputs: Expanded training data $\mathcal{L}$, updated model $\mathcal{M}$

1: $\mathcal{L}_0 \leftarrow \varnothing$
2: **for** t: $1, 2, ..$ **do**
  /* Phase 1: Query selection */
3: $Q_t \leftarrow \mathcal{A}\left(\mathcal{U}_{t-1}, \mathcal{M}_{t-1}, k\right)$
4: $Y_t \leftarrow$ labels of samples in Q_t
  /* Phase 2: Updating */
5: $\mathcal{L}_t \leftarrow \mathcal{L}_{t-1} \cup \{(\mathbf{x}, y) | \mathbf{x} \in Q_t, y \in Y_t\}$
6: $\mathcal{M}_t \leftarrow$ fine-tuning $\mathcal{M}_{t-1}$ using $\mathcal{L}_t$
7: $\mathcal{U}_t \leftarrow \mathcal{U}_{t-1} \backslash Q_t$
8: **end for**
9: **return** $\mathcal{L}_t, \mathcal{M}_t$

Fig. 1. General pool-based AL iterations.

The family of models that we focus here include CNNs. Any model $\mathcal{M}$ in this family is capable of outputting class posterior probabilities for a given input $\mathbf{x}$, i.e. $\mathbb{P}(y|\,\mathbf{x}, \boldsymbol{\theta})$. Let us denote the ordered set of model parameters by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$, where $\boldsymbol{\theta}_i$ is the parameter set of the i-th layer and L is the number of hidden layers. We use d_i to represent number of the parameters involved in the i-th layer, that is $\boldsymbol{\theta}_i \in \mathbb{R}^{d_i}$, and $d = \sum_i d_i$ indicates total number of model parameters. We also define $\ell(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}; \mathbf{x}, y) := \log \mathbb{P}(y|\,\mathbf{x}, \boldsymbol{\theta})$, where we skip writing the dependence to $\mathbf{x}$ and y for simplifying the notations. The gradient of this function with respect to $\boldsymbol{\theta}$, i.e. $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$, is called the *score function* and plays key role in defining the FI.

Fisher information (FI) $\mathbf{I}(\boldsymbol{\theta})$ is defined as $\mathbb{E}_{\mathbf{x},y}\big[\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}^{\top} \ell(\boldsymbol{\theta})\big]$. Assuming that the underlying distribution of the data has the form $p(\mathbf{x}, y) = p(\mathbf{x})\mathbb{P}(y|\,\mathbf{x}, \boldsymbol{\theta}_0)$ with a conditional in the same parametric family as the model posterior $\mathbb{P}(y|\,\mathbf{x}, \boldsymbol{\theta})$, the FI $\mathbf{I}(\boldsymbol{\theta}_0)$ measures the amount of information that an observation carries about the true model parameter $\boldsymbol{\theta}_0$. Trace of (inverse) FI serves as a useful active learning objective [23], [24]. We optimize this objective with respect to a query distribution $\mathbf{q}$ defined over the pool $\mathcal{U}$ (hence q_i is the probability of querying $\mathbf{x}_i \in \mathcal{U}$):

$$\underset{\mathbf{q} \in [0,1]^n}{\arg\min}\ \operatorname{tr}\left[\mathbf{I}_{\mathbf{q}}(\boldsymbol{\theta}_0)^{-1}\right] \quad \text{s.t.} \ \sum_i q_i = 1. \tag{1}$$

where $\mathbf{I}_q$ denotes the FI when the underlying data distribution is $q(\mathbf{x})\mathbb{P}(y|\,\mathbf{x}, \boldsymbol{\theta}_0)$ for some *query distribution* q. In pool-based AL, there is a finite number of unlabeled samples from which the queries are chosen, and therefore q is a probability mass function (PMF). Furthermore, since $\boldsymbol{\theta}_0$ is not known, it is replaced by the available estimate $\hat{\boldsymbol{\theta}}$ in $\mathcal{M}_{t-1}$. The matrix inversion in optimization (1) makes the objective highly non-linear and hard to solve. Simply removing this non-linearity by discarding the inversion will lead to an objective that scores samples independently based on the expected ℓ_2-norm of their gradients (hence, expected gradient length AL [42]). However, such simplified objective ignores the interaction between the queries resulting in poor AL performance. Fortunately, the

optimization (1) can be reformulated in the form of a semi-definite programming (SDP) problem [26], [43]. Therefore, in FI-based AL, the querying module $\mathcal{A}(\mathcal{U}, \mathcal{M}, k)$ (line 3 in Fig. 1) consists of two steps: (1) solving this SDP for a query distribution $q(\mathbf{x})$, and (2) choosing distinct samples of k draws from this distribution.

B. Approximate FI-Based AL

For large models with millions of parameters, including deep CNNs, forming FI matrix explicitly is prohibitively expensive. Furthermore, computational complexity of solving the resulting SDP increases quadratically with sample size n and super-quadratically with respect to the number of parameters [26]. Therefore, even if we can find a way of forming exact FI matrix, tractability of FI-based AL is very sensitive to size of the model's parameter space and can easily become intractable for CNNs with even intermediate number of layers. To address this problem, we propose our method of approximating FI matrix with reduced dimensionality. Our approximation decreases dimensionality of the FI matrix from the number of parameters to the number of hidden layers in the neural network.

Suppose $\mathbf{I}_q \in \mathbb{R}^{d\times d}$ is the full FI matrix of the model, where d denotes the overall number of parameters in the network. This matrix can be partitioned into L^2 sub-matrices, where the (i, j)-th sub-matrix $\mathbf{I}_q^{(i,j)}$ has dimensionality of $n_i \times n_j$ (for some $1 \le i, j \le L$) and includes all interactions between elements of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ (Fig. 2, first and second rows).

We transform down the full FI to a smaller matrix with a non-uniform average-pooling of $\mathbf{I}_q$. More specifically, we form $L \times L$ matrix $\tilde{\mathbf{I}}_q$ such that its (i, j)-th element is the average of all elements in sub-matrix $\mathbf{I}_q^{(i,j)}$. It is easy to verify that this transformation can be written as the following matrix multiplication:

$$\tilde{\mathbf{I}}_q \;=\; \mathbf{E}\,\mathbf{I}_q\,\mathbf{E}^\top \tag{2}$$

where $\mathbf{E}$ is a sparse $L \times n$ matrix. Considering the same ordering in block-wise set of parameters $\boldsymbol{\theta}$ shown above, the i-th row of $\mathbf{E}$ is an all-zero vector except for d_i-th block of elements with indices $(d_1 + \ldots + d_{i-1} + 1), \ldots, (d_1 + \ldots + d_{i-1} + d_i)$, which have value $\frac{1}{d_i}$ (Fig. 2, third row). This matrix can be viewed as the Jacobian of the re-parameterization that transforms $\mathbf{I}_q$ to $\tilde{\mathbf{I}}_q$.

It is easy to show that $\tilde{\mathbf{I}}_q$ can be computed without explicitly forming $\mathbf{I}_q$. Indeed, it is equivalent to Fisher information of a model with a score function equal to $\tilde{\mathbf{s}}(\boldsymbol{\theta}) = \left[\frac{1}{d_1}\sum_{\theta\in\boldsymbol{\theta}_1}\nabla_\theta \ell(\boldsymbol{\theta}), \ldots, \frac{1}{d_L}\sum_{\theta\in\boldsymbol{\theta}_L}\nabla_\theta \ell(\boldsymbol{\theta})\right]^\top$, which contains layer-wise average of the original scores. Now, then the (i, j)-th element of the new model's FI is

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x},y}\left[\tilde{\mathbf{s}}(\boldsymbol{\theta})\tilde{\mathbf{s}}(\boldsymbol{\theta})^\top\right]_{i,j} \\
&\quad= \mathbb{E}_{\mathbf{x},y}\left[\left(\frac{1}{d_i}\sum_{\theta\in\boldsymbol{\theta}_i}\nabla_\theta \ell(\boldsymbol{\theta})\right)\left(\frac{1}{d_j}\sum_{\theta\in\boldsymbol{\theta}_j}\nabla_\theta \ell(\boldsymbol{\theta})\right)\right] \\
&\quad= \frac{1}{d_i d_j}\sum_{\theta\in\boldsymbol{\theta}_i}\sum_{\theta'\in\boldsymbol{\theta}_j}\mathbb{E}_{\mathbf{x},y}\left[\nabla_\theta \ell(\boldsymbol{\theta})\nabla_{\theta'} \ell(\boldsymbol{\theta}')\right],
\end{aligned} \tag{3}$$

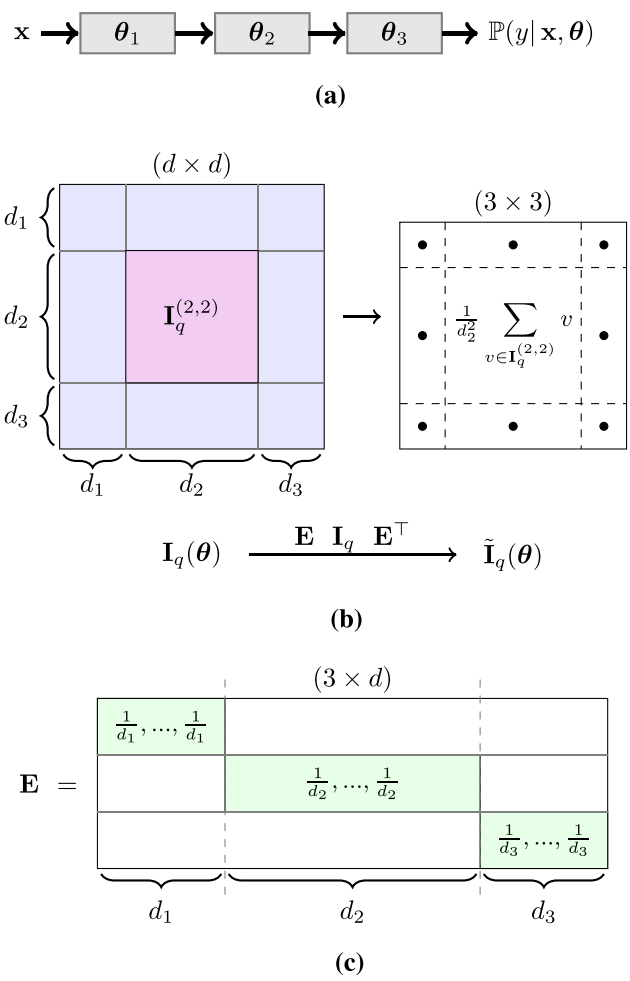

Fig. 2. Illustration of our FI approximation for a 3-layer network ($L = 3$). Parameters of the model are denoted by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ such that the i-th layer has d_i parameters, i.e. $\boldsymbol{\theta}_i \in \mathbb{R}^{d_i}$ (row **(a)**). The full FI matrix $\boldsymbol{I}_q(\boldsymbol{\theta})$ has dimensionality $d\times d$, where $d = d_1+d_2+d_3$, and it can be partitioned into $3\times3 = 9$ sub-matrices each of which corresponds interactions between a pair of layers. For example, the marked sub-matrix $\mathrm{I}_q^{(2,2)}$ contains all inter-layer terms of $\boldsymbol{\theta}_2$. In this example, the second layer is assumed to have more parameters than the other two ($d_2 > d_1, d_3$). The approximated FI matrix is 3×3, where the (i, j)-th element is the average of all elements in the (i, j)-th block in I_q (row **(b)**). This transformation can be done through a matrix multiplication of form $\mathrm{E}\,\mathrm{I}_q(\boldsymbol{\theta})\mathrm{E}^\top$ where $\mathbf{E}$ is a $3 \times d$ matrix structured according to the size and order of the layers, for example its second row has value $\frac{1}{d_2}$ in elements indexed by $d_1 + 1, \ldots, d_1 + d_2$ and zero elsewhere (row **(c)**).

which is, by definition, equal to the (i, j)-th element of $\tilde{\mathbf{I}}_q(\boldsymbol{\theta})$. Hence, in order to form $\tilde{\mathbf{I}}_q$ we only need to form $\tilde{s}(\boldsymbol{\theta})$ from the current model and use its outer-product from (3).

As mentioned before, computational complexity of solving (1) also depends on size of the unlabeled pool and increases quadratically with n. Hence, in practice it will be slow for large n values. In order to further accelerate our AL framework, we downsample the unlabeled pool to a subset that contains the most β uncertain samples ($\beta < n$). Such downsampling of the large pools via *uncertainty filtering* has already been used in accelerating pool-based AL methods [26], [44].

Gathering everything in one place and introducing L auxiliary variables, $t_1, \ldots, t_L \in \mathbb{R}$, similar to the shallow version of FI-based AL [26], we get the following SDP for our

approximate FI-based AL:

$$\underset{q_1,\ldots,q_\beta,t_1,\ldots,t_L}{\arg\min} \quad t_1+\ldots+t_L$$
$$\text{s.t.} \sum_{i=1}^{\beta} q_i = 1, \quad \bigoplus_{j=1}^{L} \begin{bmatrix} \sum_i q_i \tilde{\mathbf{I}}_q(\hat{\boldsymbol{\theta}};\mathbf{x}_i) & \mathbf{e}_j \\ \mathbf{e}_j^\top & t_j \end{bmatrix} \succeq 0, \quad (4)$$

where $\bigoplus$ denotes matrix direct sum, q_i denotes the probability of querying the i-th unlabeled sample $\mathbf{x}_i$ after uncertainty filtering, and $\tilde{\mathbf{I}}_q(\mathbf{x}_i;\hat{\boldsymbol{\theta}})$ is the conditional FI at the single sample $\mathbf{x}_i$, defined as

$$\begin{aligned}\tilde{\mathbf{I}}_q(\mathbf{x}_i;\hat{\boldsymbol{\theta}}) &= \mathbb{E}_{y|\mathbf{x}_i,\hat{\boldsymbol{\theta}}}\left[\tilde{\mathbf{s}}(\boldsymbol{\theta})\tilde{\mathbf{s}}(\boldsymbol{\theta})^\top\right] \\ &= \sum_{y_i} \mathbb{P}(y_i|\mathbf{x}_i,\hat{\boldsymbol{\theta}})\tilde{\mathbf{s}}(\boldsymbol{\theta})\tilde{\mathbf{s}}(\boldsymbol{\theta})^\top. \quad (5)\end{aligned}$$

C. Non-FI AL Methods

Here, we list and briefly describe the non-FI AL methods that we will compare against the FI-based AL in the next section. The name with which each paragraph starts represent the label that we will use to represent the corresponding method in the results section.

random: randomly querying k samples without replacement from the unlabeled pool $\mathcal{U}$;

entropy [6]: uncertainty sampling method with uncertainty measured by means of Shannon entropy function computed over the network's class posterior probabilities; suppose the posterior of a sample $\mathbf{x}_i$ is denoted by $\mathbf{p}_i \in [0,1]^c$ where $p_{ij} = \mathbb{P}(y=j|\mathbf{x}_i,\hat{\boldsymbol{\theta}})$, then its uncertainty is measured by $H_S(\mathbf{p}_i|\hat{\boldsymbol{\theta}}) := -\sum_{j=1}^{c} p_{ij}\log p_{ij}$.

MC drop-out [7]: uncertainty sampling method with uncertainty measured as the Shannon entropy of the average of class posterior probabilities in $T=20$ Monte-Carlo (MC) parameter sets drawn from dropout distribution [7]; suppose the resulting posterior probability of the i-th sample in the τ-th MC run is denoted by $\mathbf{p}_i^{(\tau)}$, then the model's uncertainty for the i-th sample would be estimated as $H_S\left(\frac{1}{T}\sum_{\tau=1}^{T}\mathbf{p}_i^{(\tau)}\right)$.

ensemble-S [10]: an uncertainty sampling method where the uncertainty is measured as the average of Shannon entropies based on an ensemble of networks; in each AL iteration, an ensemble of $T=7$ networks was created as the following: in the first iteration where no labeled target sample was observed yet, multiple pre-trained models were obtained by repeatedly training models over the source data set with different random initializations and drop-out; in the intermediate t-th AL iteration ($t \geq 2$), an ensemble was obtained by fine-tuning $\mathcal{M}_{t-2}$ with $\mathcal{L}_{t-1}$ for multiple times, each time using drop-out for the FC layers in order to get a slightly different model. After creating the ensemble $\left\{\hat{\boldsymbol{\theta}}^{(1)},\ldots,\hat{\boldsymbol{\theta}}^{(T)}\right\}$, the uncertainty is measured by computing Shannon entropy over the average of posterior probabilities of the ensemble, i.e. $H_S\left(\frac{1}{T}\sum_{\tau=1}^{T}\mathbf{p}_i^{(\tau)}\right)$, where $p_{ij}^{(\tau)} := \mathbb{P}\left(y=j\left|\mathbf{x}_i,\hat{\boldsymbol{\theta}}^{(\tau)}\right.\right)$.

ensemble-JS [12]: an uncertainty model similar to ensemble-S, except it uses Jensen-Shannon entropy over the ensemble posteriors rather than Shannon entropy; for the i-th sample, Shannon-Jensen entropy is defined as $H_{JS}\left(\mathbf{p}_i^{(1)},\ldots,\mathbf{p}_i^{(T)}\right) := H_S\left(\frac{1}{T}\sum_{\tau=1}^{T}\mathbf{p}_i^{(\tau)}\right) - \frac{1}{T}\sum_{\tau=1}^{T}H_S\left(\mathbf{p}_i^{(\tau)}\right)$. This entropy measures the amount of disagreement among a given ensemble of posteriors, hence this algorithm can be viewed as a query by committee technique [45].

RepU [19], [20]: an AL algorithm that combines uncertainty with representativeness to query the most uncertain samples that also represent the unlabeled pool best [19]; although this algorithm was originally proposed for fully convolutional networks but it can be applied to patch-wise segmentation as well. It has the following steps: (1) selecting a certain number of most uncertain candidates B (size of B is set to 200 in our experiments in order to be consistent with our uncertainty filtering), (2) choosing a subset $Q \subset B$ of size k which is most similar to the rest of the unlabeled pool $\mathcal{U}\setminus B$, with set-wise similarity defined as $\text{sim}(Q,\mathcal{U}\setminus B) := \sum_{\mathbf{z}\in Q}\max_{\mathbf{z}'\in\mathcal{U}\setminus B}\cos(\mathbf{z},\mathbf{z}')$, where $\cos(\cdot,\cdot)$ is the cosine similarity, and $\mathbf{z},\mathbf{z}'$ are features extracted from the second FC layer of our network. Since choosing Q that maximizes this similarity is an NP-hard problem, a forward greedy selection algorithm is used.

core-set [21]: this AL approach selects queries that best covers feature space of the data, which is shown to be equivalent to the k-center problem (facility location problem) [46]; this algorithm requires computing similarities between the pool samples and the labeled data set $\mathcal{L}_{t-1}$ that is obtained until the t-th iteration, and greedily adding unlabeled samples that have the largest distance to their nearest neighbor in $\mathcal{L}_{t-1}$. This similarity can be computed using the same set-wise similarity that was used for performing RepU. In the first iteration, we do not have any labeled sample from the target subjects ($\mathcal{L}_0 = \varnothing$), hence we consider labeled voxels from the source subjects. Note that here it is assumed that we still have access to the source data set, which is not always the case in practice.

Among the competing methods explained above, entropy, MC dropout, ensemble-S and ensemble-JS are examples of different US techniques, and RepU and core-set are geometrical AL algorithms.

III. Experimental Settings

We evaluated performance of the proposed AL framework in the application of patch-wise brain extraction using T1- and T2-weighted MR images. Our experiments had flavor of transfer learning: given a CNN model pre-trained over a source data set, the goal was to fine-tune this model to a target data set with different properties than the source, using the smallest number of additional target data to be labeled. Depending on the number of target subject(s), we had the following two scenarios:

- *Universal AL*, where subjects of the target data set were divided into pool and test partitions. We used voxels of the pool subjects as the unlabeled pool from which the voxel queries were selected, whereas the test subjects were used only to evaluate the performance of the resulting model on the target data set.
- *Active semi-automatic segmentation*, where individual subjects of each data set were considered separately.

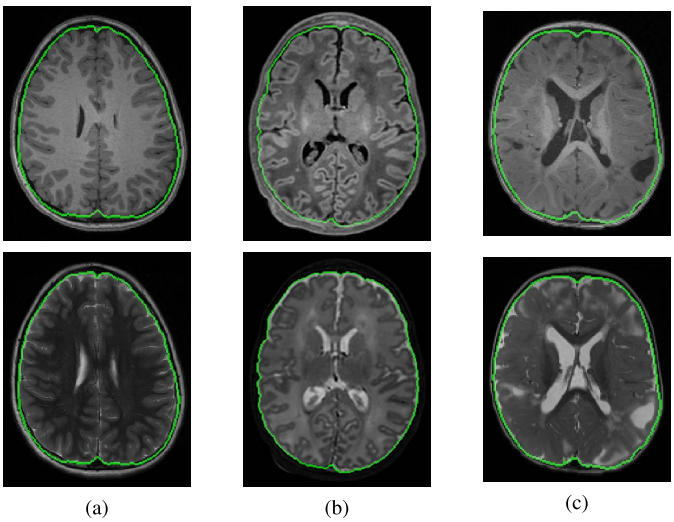

Fig. 3. Sample axial slices of T1-weighted (first row) and T2-weighted (second row) MRI images from the data sets used in the experiments. The interior region of the green boundary is the ground-truth brain mask. These samples show contrast differences between source and target populations, especially between newborns and adolescents. (a) Adolescent subject. (b) Newborn subject. (c) Subject with TSC.

We selected the queries from the same subject to refine the model for that specific subject by labeling only a small number of new samples from that subject. We evaluated the final model by comparing the resultant segmentation for each subject with the corresponding ground truth segmentation.

A. Data

We used brain images of 10 healthy adolescent subjects as our source data set to train $\mathcal{M}_0$. We considered four pediatric data sets as target: newborns and subjects with tuberous sclerosis complex (TSC) divided into three age groups of (Gr1) younger than six months, (Gr2) between six months and one year old, and (Gr3) older than one year. The latter data set contained 25 normal newborn subjects provided by the Developing Human Connectome Project [47], and the other three groups contained 26 subjects from 2 months old to 2.5 years old, whose MRI images were acquired in five TSC centers throughout the United States (data set with different health conditions). The three age groups of this data set had nine, nine and eight subjects, respectively. Sample 2D axial slices from each data set are shown in Figs. 3a to 3c, indicating different visual characteristics. Pediatric brain presents specific challenges for segmentation because of marked intra- and inter-variation in head size and shape in early life, rapid changes in tissue contrast associated with myelination, and low contrast to noise ratio compared to adult brain MRIs. Newborn MR images in particular have different intensity contrast compared to older children and adult brain images.

B. Technical Details

In patch-wise segmentation, any voxel is a data sample represented by a patch around it containing the voxel itself and some of its neighbors. In our initial experiments of brain extraction, we examined different patch sizes from axial slices and chose 2D patches of size 25×25 as the best choice. Having two modalities of T1- and T2-weighted images, each voxel is represented by a $25 \times 25 \times 2$ patch from each modality. Architecture of the CNN model that we used is shown in Fig. 4. This model has four convolutional layers and three fully connected (FC) layers ($L = 7$). Furthermore, there exists a ReLU activation function after each layer, except the last layer that is followed by a soft-max activation to give class posteriors.

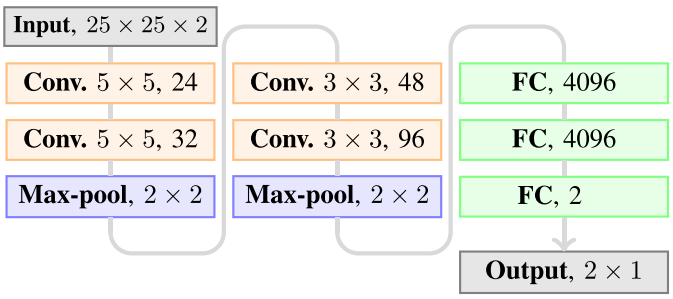

Fig. 4. Architecture of the CNN model used in our experiments. The input is concatenation of two 25×25 patches from T1w and T2w images, and the output is a binary posterior probability. The numbers shown in "**Conv.**" blocks represent the size and number of the kernels, and those in "**FC**" blocks show the number of output nodes.

An initial model $\mathcal{M}_0$ was trained with the network's parameters initialized according to [48]. Then, we ran various AL algorithms to make improvement on $\mathcal{M}_0$ with $k = 50$ until the total number of queries reach 600. Note that in practice, our algorithm requires a single click for labeling the central voxel of a query patch. In other words, each AL iteration consists of k user clicks. In the t-th AL iteration, the model got updated by fine-tuning $\mathcal{M}_{t-1}$ for 60 epochs using all labeled samples available from the previous iterations. In both initial training of $\mathcal{M}_0$ and partial fine-tuning of $\mathcal{M}_t$, we used Adam optimizer with learning rate of 10^{-5} and drop-out rate of 0.5 only for FC layers. The CNN models were built using the TensorFlow package and the SDP of FI-based AL in (4) was solved using MOSEK solver [49] of CVXPY package [50], [51].

We compared our FI-based AL algorithm with a list of other AL algorithms explained in Section II-C. In addition to these techniques, we also compared the results of the networks trained using the competing AL algorithms with several brain extraction baseline methods from the literature, including BET [52], ROBEX [53] and volBrain [54]. When running BET, we set the fractional intensity threshold to 0.1 and the vertical gradient of fractional intensity to -0.1. The results of volBrain algorithm were obtained through their online automated system.[1]

For evaluating any given model resulted from AL iterations or baseline methods, we compared the corresponding predicted segmentation map after post-processing with the ground-truth mask. The post-processing steps included 3D connected component analysis, i.e., keeping only the largest

[1] http://volbrain.upv.es/.

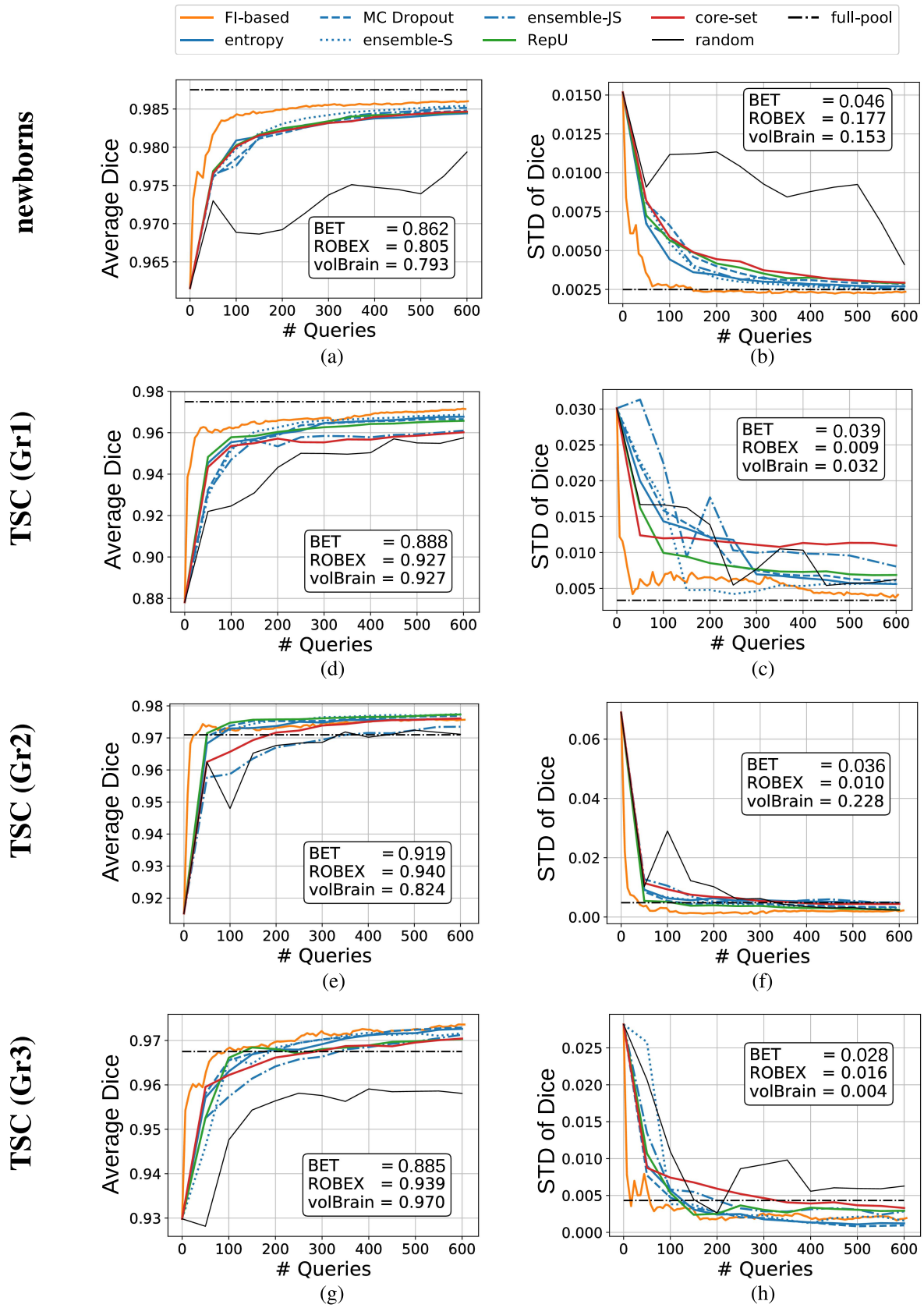

Fig. 5. Evaluating Dice coefficients of the models obtained from universal AL algorithms that were executed to select up to 600 queries (colored curves), models that were trained using all unlabeled samples in the pool (the horizontal dash-dotted line labeled as "full-pool"), and the baseline methods (reported in text-boxes within the figures).

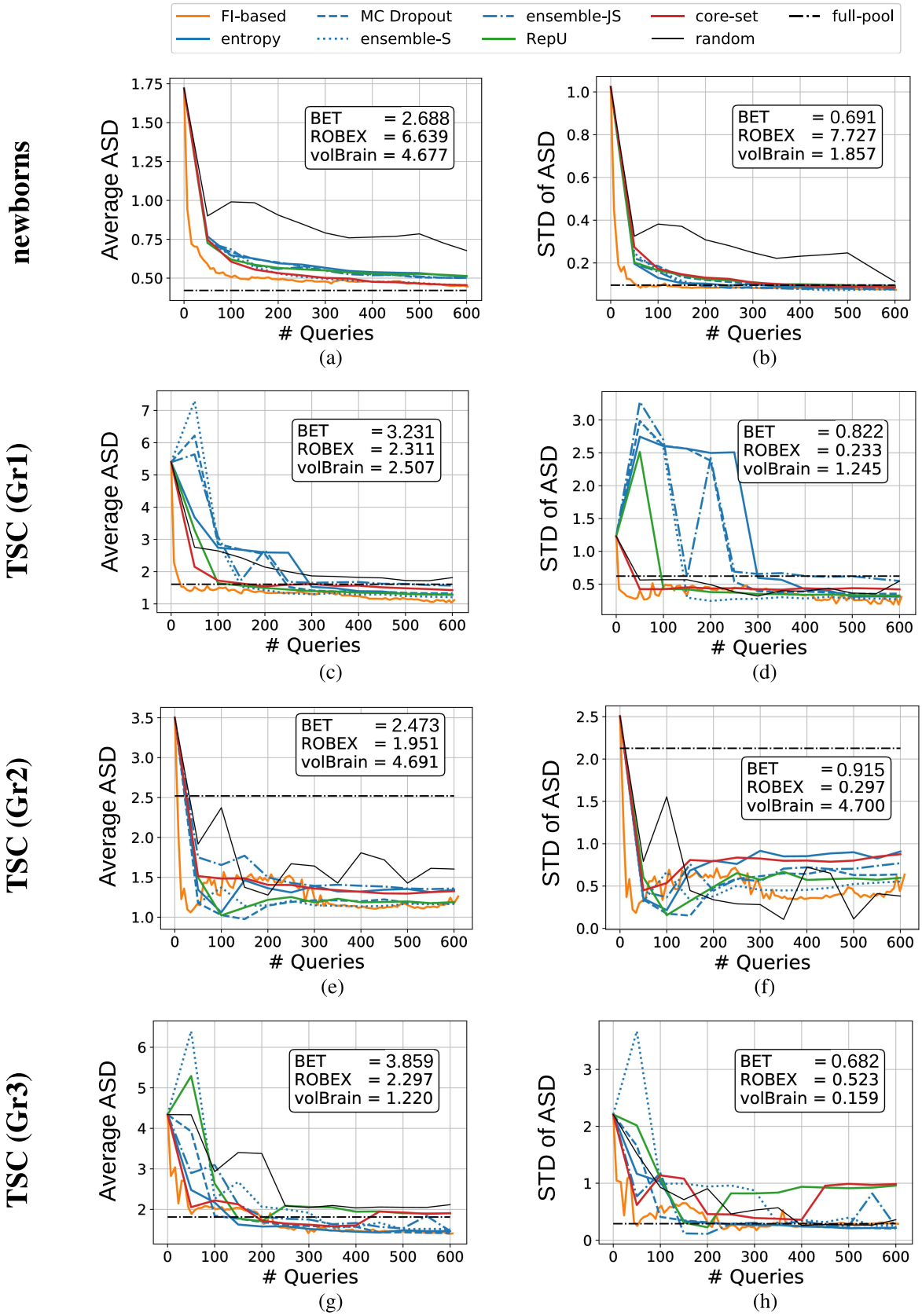

Fig. 6. Evaluating average surface distance (ASD) criterion for the models obtained from universal AL algorithms that were executed to select up to 600 queries (colored curves), models that were trained using all unlabeled samples in the pool (the horizontal dash-dotted line labeled as "full-pool"), and the baseline methods (reported in text-boxes within the figures).

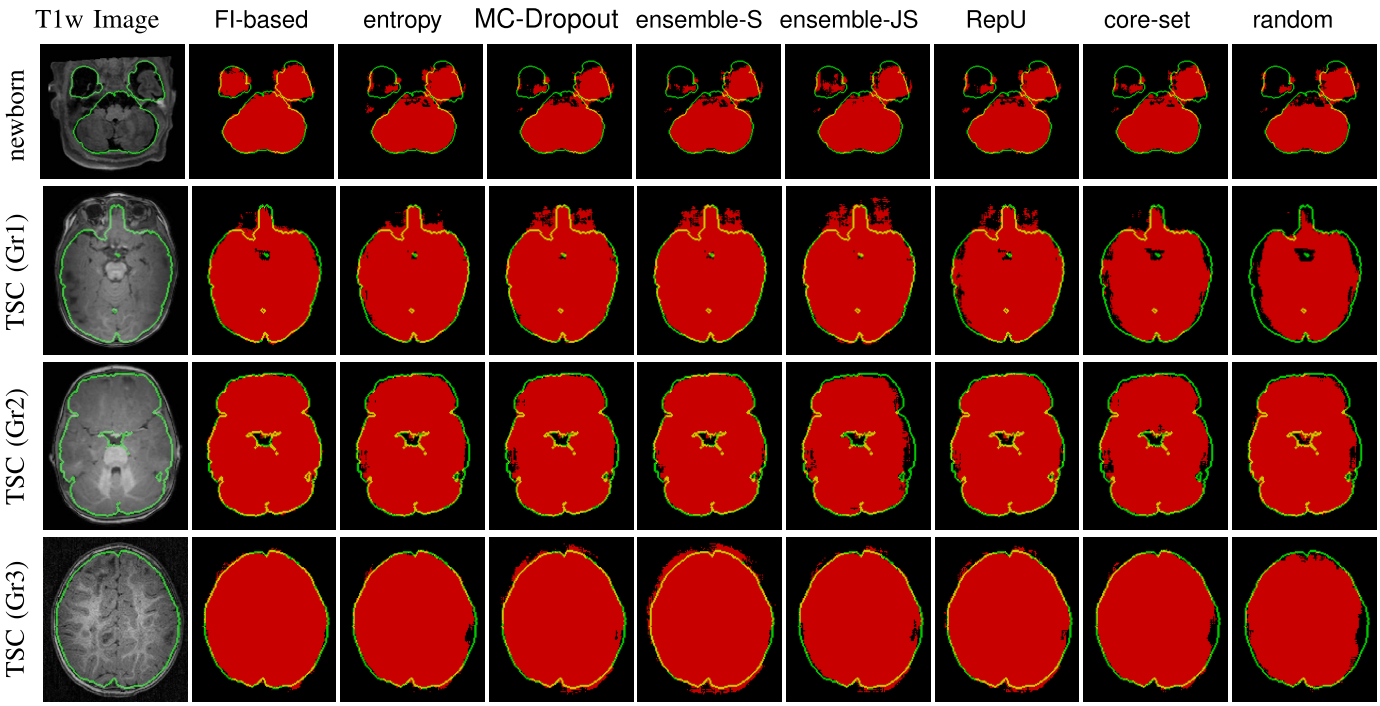

Fig. 7. Segmentation of a sample test slice from each target data set using models from early AL iterations (after labeling up to 50 queries by each method). Each row corresponds to a single target data set, where the first image is the original slice from the T1-weighted MRI image together with the ground-truth mask (interior region of the green boundary), and the rest show the segmentation results. The red voxels represent those that are marked as brain by the corresponding models. Ideally, all the voxels inside the green boundaries should be red.

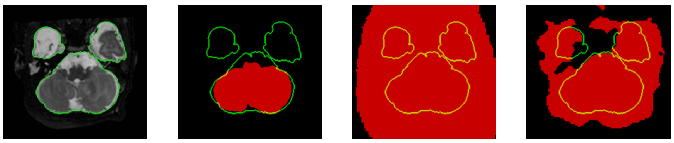

Fig. 8. T2-weighted image and the baseline results of the newborn slice shown in Fig. 7. (a) T2w image. (b) BET. (c) ROBEX. (d) volBrain.

3D component, and morphological operations for filling its holes. We used Dice score and average surface distance (ASD) as two evaluation metrics.

IV. Results

In this section, we demonstrated the results of our experiments using the proposed FI-based AL, other competing AL methods and the brain extraction baselines. See section III for more details on the experimental setting.

A. Universal Active Learning

Average values and standard deviation of the two evaluation metrics, i.e. Dice coefficient and ASD, computed for all universal AL iterations are shown in Figs. 5 and 6. For each AL iteration, the statistics are taken over the test subjects of the corresponding target data set. These curves show that based on both metrics, FI-based AL could converge to a high-accuracy model faster (i.e. using less samples to be annotated) than other AL methods. The converged accuracy of the proposed FI-based AL is comparable to that of the full-pool training, when the number of labeled samples at the end of the AL experiments is less than 0.5% of size of the pool. Indeed, for two data sets the AL models could achieve even higher accuracy than the model that used all the pool samples for training. Moreover, observe that all the baseline methods resulted in significantly weaker performance, in some cases even worse than the pre-trained model.[2]

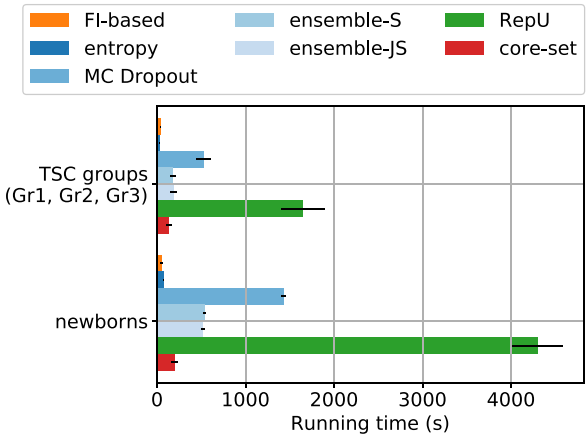

Fig. 9. Average running time of a single AL iteration for different methods. The average values are reported for the three TSC groups and newborns data sets separately as the image dimensions in the latter are larger. Notice that FI-based AL is significantly faster than most other competents.

[2]volBrain failed to generate segmentation for one of the newborn test subjects.

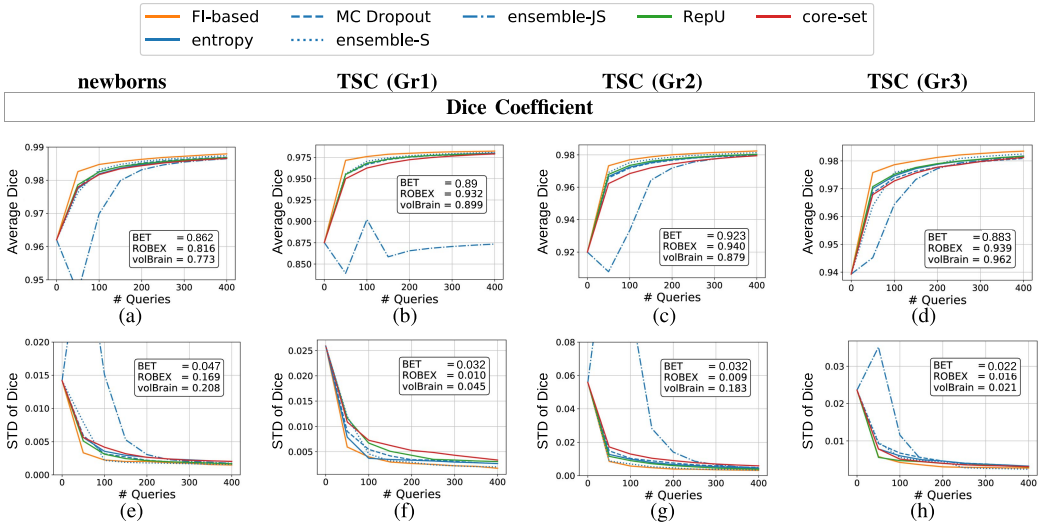

Fig. 10. Evaluating models obtained from semi-automatic segmentation AL algorithms that were executed to select up to 400 queries (colored curves), and the baseline methods (reported in text-boxes within the figures). For each metric, the first row shows the average values and second row shows the standard deviations (taken over all the subjects in target data set).

We also visualized segmentation of sample slices by different AL models in Fig. 7. These are the results of early AL iterations. More specifically, for a given target data set, we took models obtained after labeling up to 50 queries from the corresponding target data set and used them to segment a sample slice from a test subject. The results indicate the fine details where FI-based AL outperformed other methods. The difference is obvious for the newborn's sample slice where T1w image was not enough for extracting several parts of the brain. Visualizing the same slice from T2 modality (see Fig. 8a) reveals that T2w image can be used to remove this ambiguity. However, only FI-based AL was successful in creating a model which could properly grasp information from both modalities. Failure of baseline methods to segment this slice is also demonstrated in Figs. 8b to 8d. BET and ROBEX missed the main structure by undersegmenting and oversegmenting the brain in the selected slice, respectively. Moreover, volBrain could distinguish the overall structure but resulted in too many false positives.

We noticed that FI-based AL algorithm was faster than other methods in all the experiments. Fig. 9 shows average running time duration of all the AL techniques used in this paper. All the experiments were run on a 2.6GHz Intel Xenon CPU and an NVIDIA GeForce GTX 1070 Ti GPU. The slowest AL method was RepU which showed best performance among the other non-FI AL algorithms based on Fig. 5.

B. Active Semi-Automatic Segmentation

Average and standard deviation of Dice coefficients computed for all the AL iterations are reported in Fig. 10. Similar to universal active learning, we observed that FI-based AL in average achieved higher model accuracy with less number of queries. Among other non-FI AL methods, ensemble-JS was significantly worse than others, and RepU and ensemble-S were comparable to FI-based approach. Furthermore, note that the convergent Dice scores in semi-automatic segmentation experiments were generally better than those in universal AL.

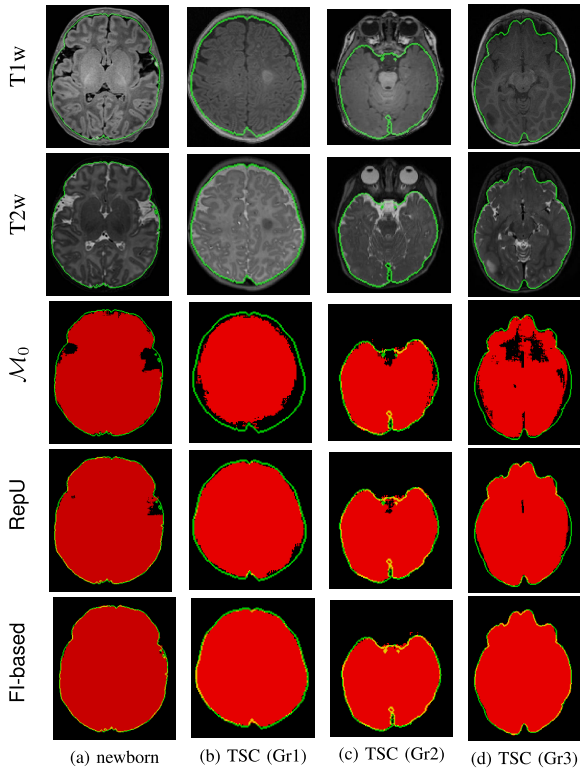

Fig. 11. Segmentation of sample slices from different subject groups in active semi-automatic segmentation experiments. Each column corresponds to a single subject. The first two rows show the slice from T1- and T2-weighted images together with the ground truth mask (green boundaries). The third row shows the results obtained by the initial pre-trained model, and the last two rows show the results of RepU method, as the best method among non-FI AL techniques in terms of average Dice scores, and FI-based AL. For the subjects considered in columns (a) to (d) and the AL iterations whose results were shown, the FI-based AL (considering all slices) improved Dice scores of $\mathcal{M}_0$ by 3.16%, 8.21%, 1.19% and 6.54%, respectively. (a) Newborn. (b) TSC (Gr1). (c) TSC (Gr2). (d) TSC (Gr3).

This is expected because specializing a learning model to smaller data set (e.g., a single subject) is generally easier than training a model that is generalizable to large data sets.

Segmentation of sample slices with different models in these experiments are shown in Fig. 11. One subject per group was chosen to visualize segmentation results obtained by the pre-trained model $\mathcal{M}_0$, and the models achieved by two AL methods in early iterations (after labeling up to 50 queries): FI-based and the best non-FI AL method in terms of the average Dice scores, which was RepU. It is clear that FI-based AL resulted in more accurate segmentation results with much fewer false negatives.

V. Conclusion

In this paper, we proposed to use an active learning (AL) objective based on Fisher information (FI) for CNN models. In order to make FI-based computations tractable for deep models, we shrunk the parameter space by an implicit model reparametrization. Our experiments included various transfer learning scenarios, where we started from a pre-trained model obtained from a source data set and executed AL for intelligent labeling of small number of target samples in order to fine-tune the initial model. We used 10 adolescent subjects as our source, and considered four target data sets: newborn subjects and patients with TSC lesions that came from 3 different age groups. We evaluated the proposed AL framework in the context of patch-wise brain extraction. The results were presented in two scenarios of (1) universal AL to build a model for multiple target subjects, and (2) active semi-automatic segmentation to personalize a model for a single subject. The results showed that FI-based AL outperformed a comprehensive list of recently proposed AL techniques and brain extraction baseline approaches in reducing the number of samples that need to be labeled for training a high-quality patch-wise segmentation method. We saw that, in the worst case of universal AL, labeling only a 0.25% of the target subjects with FI-based AL achieved about 99.7% of the model that used all target subjects for training. Furthermore, in best the case of universal AL, the FI-based model even improved the fully-trained model by around 0.6% of Dice score.

There are still some difficulties in implementing FI-based AL. One of them is the computational complexity of the SDP that is involved. Here, we did uncertainty filtering in order to downsample the unlabeled pool and speed-up solving this optimization. The downside of this filtering is that the true performance of FI-based AL cannot be reached because such filtering could also throw away useful samples. Future work involves developing other approaches to reduce the complexity of optimization while realizing the full capacity of FI-based deep AL. Another important direction of future work is to extend FI-based AL to fully convolutional networks (FCNs). In contrast to CNNs with fully connected final layers that output only a single probability distribution, FCNs output a map of class probabilities associated with the full segmentation map of the input. Therefore, computing FI matrix for FCNs will require modeling the joint distribution of class labels of all voxels.

References

[1] G. Litjens *et al.*, "anchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.

[2] Y. Lamash *et al.*, "Curved planar reformatting and convolutional neural network-based segmentation of the small bowel for visualization and quantitative assessment of pediatric Crohn's disease from MRI," *J. Magn. Reson. Imag.*, 2018. to be published.

[3] Y. Lamash, S. Kurugol, and S. K. Warfield, "Semi-automated extraction of Crohns disease MR imaging markers using a 3D residual CNN with distance prior," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 218–226.

[4] M. Haghighi, S. K. Warfield, and S. Kurugol, "Automatic renal segmentation in DCE-MRI using convolutional neural networks," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 1534–1537.

[5] S. Zhou, Q. Chen, and X. Wang, "Active deep networks for semi-supervised sentiment classification," in *Proc. 23rd Int. Conf. Comput. Linguistics, Posters*. Association for Computational Linguistics, 2010, pp. 1515–1523.

[6] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2591–2600, Dec. 2017.

[7] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1183–1192.

[8] D. Mahapatra, B. Bozorgtabar, J.-P. Thiran, and M. Reyes, "Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 580–588.

[9] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: Actively and incrementally," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 7340–7351.

[10] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The power of ensembles for active learning in image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9368–9377.

[11] W. Shao, L. Sun, and D. Zhang, "Deep active learning for nucleus classification in pathology images," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 199–202.

[12] W. Kuo, C. Häne, E. Yuh, P. Mukherjee, and J. Malik, "Cost-sensitive active learning for intracranial hemorrhage detection," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 715–723.

[13] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5574–5584.

[14] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. (2011). "Bayesian active learning for classification and preference learning." [Online]. Available: https://arxiv.org/abs/1112.5745

[15] X. Li and Y. Guo, "Adaptive active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 859–866.

[16] J. Sourati, M. Akcakaya, J. G. Dy, T. K. Leen, and D. Erdogmus, "Classification active learning based on mutual information," *Entropy*, vol. 18, no. 2, p. 51, 2016.

[17] J. Zhu, H. Wang, B. K. Tsou, and M. Y. Ma, "Active learning with sampling by uncertainty and density for data annotations," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1323–1331, Aug. 2010.

[18] D. Mahapatra, P. J. Schüffler, J. A. W. Tielbeek, F. M. Vos, and J. M. Buhmann, "Semi-supervised and active learning for automatic segmentation of crohn's disease," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2013, pp. 214–221.

[19] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2017, pp. 399–407.

[20] W. Wang, Y. Lu, B. Wu, T. Chen, D. Z. Chen, and J. Wu, "Deep active self-paced learning for accurate pulmonary nodule segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2018, pp. 723–731.

[21] O. Sener and S. Savarese. (2018). "Active learning for convolutional neural networks: A core-set approach." [Online]. Available: https://arxiv.org/abs/1708.00489

[22] T. Zhang and F. Oles, "The value of unlabeled data for classification problems," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 1191–1198.

[23] K. Chaudhuri, S. M. Kakade, P. Netrapalli, and S. Sanghavi, "Convergence rates of active learning for maximum likelihood estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1090–1098.

[24] J. Sourati, M. Akcakaya, T. K. Leen, D. Erdogmus, and J. G. Dy, "Asymptotic analysis of objectives based on Fisher information in active learning," *J. Mach. Learn. Res.*, vol. 18, no. 34, pp. 1–41, 2017.

[25] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 417–424.

[26] J. Sourati, M. Akcakaya, D. Erdogmus, T. K. Leen, and J. G. Dy, "A probabilistic active learning algorithm based on Fisher information ratio," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 2023–2029, Aug. 2018.

[27] J. Sourati, A. Gholipour, J. G. Dy, S. Kurugol, and S. K. Warfield, "Active deep learning with Fisher information for patch-wise semantic segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2018, pp. 83–91.

[28] G. Desjardins, K. Simonyan, R. Pascanu, and k. kavukcuoglu, "Natural neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2071–2079.

[29] R. B. Grosse and J. Martens, "A kronecker-factored approximate Fisher matrix for convolution layers," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 573–582.

[30] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4652–4662.

[31] P. W. Koh and P. Liang. (2017). "Understanding black-box predictions via influence functions." [Online]. Available: https://arxiv.org/abs/1703.04730

[32] N. L. Roux, P.-A. Manzagol, and Y. Bengio, "Topmoumoute online natural gradient algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 849–856.

[33] J. Martens and R. Grosse, "Optimizing neural networks with kronecker-factored approximate curvature," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2408–2417.

[34] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Ann. BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[35] D. F. Pace, A. V. Dalca, T. Geva, A. J. Powell, M. H. Moghari, and P. Golland, "Interactive whole-heart segmentation in congenital heart disease," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 80–88.

[36] C. Rupprecht, I. Laina, N. Navab, G. D. Hager, and F. Tombari. (2018). "Guide me: Interacting with deep networks." [Online]. Available: https://arxiv.org/abs/1803.11544

[37] G. Wang *et al.*, "DeepIGeoS: A deep interactive geodesic framework for medical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[38] G. Wang *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Trans. Med. Imag.*, vol. 37, no. 7, pp. 1562–1573, Jul. 2018.

[39] H. Veeraraghavan and J. V. Miller, "Active learning guided interactions for consistent image segmentation with reduced user interactions," in *Proc. IEEE Int. Symp. Biomed. Imag., Nano Macro*, Mar./Apr. 2011, pp. 1645–1648.

[40] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3D image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2011, pp. 603–610.

[41] N. Tajbakhsh *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[42] S. Otálora, O. Perdomo, F. González, and H. Müller, "Training deep convolutional neural networks with active learning for exudate classification in eye fundus images," in *Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Cham, Switzerland: Springer, 2017, pp. 146–154.

[43] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[44] K. Wei, R. Iyer, and J. Bilmes, "Submodularity in data subset selection and active learning," in *Proc. 21st Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 1954–1963.

[45] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 287–294.

[46] R. S. Garfinkel, A. W. Neebe, and M. R. Rao, "The m-center problem: Minimax facility location," *Manage. Sci.*, vol. 23, no. 10, pp. 1133–1142, 1977.

[47] A. Makropoulos *et al.*, "The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction," *NeuroImage*, vol. 173, pp. 88–112, Jul. 2018.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[49] M. ApS. (2017). *The MOSEK Optimizer API for Python Manual. Version 8.1*. [Online]. Available: https://docs.mosek.com/8.1/pythonapi/index.html

[50] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 2909–2913, 2016.

[51] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[52] S. M. Smith, "Fast robust automated brain extraction," *Human Brain Mapping*, vol. 17, no. 3, pp. 143–155, 2002.

[53] J. E. Iglesias, C. Y. Liu, P. M. Thompson, and Z. Tu, "Robust brain extraction across datasets and comparison with publicly available methods," *IEEE Trans. Med. Imag.*, vol. 30, no. 9, pp. 1617–1634, Sep. 2011.

[54] J. V. Manjón and P. Coupé, "volbrain: An online MRI brain volumetry system," *Frontiers Neuroinform.*, vol. 10, p. 30, Jul. 2016.