HHS Public Access

Author manuscript

IEEE Trans Artif Intell. Author manuscript; available in PMC 2024 April 01.

Published in final edited form as:

IEEE Trans Artif Intell. 2023 April ; 4(2): 383–397. doi:10.1109/tai.2022.3159510.

Improving Calibration and Out-of-Distribution Detection in Deep Models for Medical Image Segmentation

Davood Karimi,

Ali Gholipour [Senior Member, IEEE]

Department of Radiology, Boston Children's Hospital, and Harvard Medical School, Boston, Massachusetts, USA

Abstract

Convolutional Neural Networks (CNNs) have proved to be powerful medical image segmentation models. In this study, we address some of the main unresolved issues regarding these models. Specifically, training of these models on small medical image datasets is still challenging, with many studies promoting techniques such as transfer learning. Moreover, these models are infamous for producing over-confident predictions and for failing silently when presented with out-of-distribution (OOD) test data. In this paper, for improving prediction calibration we advocate for multi-task learning, i.e., training a single model on several different datasets, spanning different organs of interest and different imaging modalities. We show that multi-task learning can significantly improve model confidence calibration. For OOD detection, we propose a novel method based on spectral analysis of CNN feature maps. We show that different datasets, representing different imaging modalities and/or different organs of interest, have distinct spectral signatures, which can be used to identify whether or not a test image is similar to the images used for training. We show that our proposed method is more accurate than several competing methods, including methods based on prediction uncertainty and image classification.

Keywords

segmentation; convolutional neural networks; multi-task learning; uncertainty; out-of-distribution detection

I. Introduction

RECENT studies have shown that convolutional neural network (CNN)-based models outperform classical methods on many medical image segmentation tasks. Various aspects of the design and training of these models have been surveyed [1]. Most studies have focused on network architecture and loss function. However, it has been argued that more elaborate network architectures only marginally improve the performance of these models [2]. On the other hand, there are important unresolved issues regarding application of CNNs for medical image segmentation. One of these issues has to do with the training procedures and training data. The number of labeled images that are available for training is typically very small.

davood.karimi@childrens.harvard.edu .

Techniques such as transfer learning, error prediction and correction, unsupervised learning, and learning from noisy annotations have been proposed [3]-[5]. Another outstanding issue is a lack of understanding of the reliability and failure modes of these models. Deep learning models, in general, are known to produce over-confident predictions, even when the predictions are wrong [6]. Deep learning models also produce confident predictions on out-of-distribution (OOD) data, i.e., when the test data is from an entirely different distribution than the training data distribution [7]. Needless to say, there is no performance guarantee on OOD data and, in theory, model predictions on OOD data cannot be expected to be any more accurate than random output.

In order to improve the reliability of CNN-based medical image segmentation models for clinical use, effective solutions are needed for the above-mentioned challenges. In particular, we need methods to train well-calibrated models from limited data. We also need methods to inform us when these models fail.

II. Related works

A. Training procedures for CNN-based medical image segmentation models

Large labeled datasets are an essential requirement for deep learning [8]. Since such datasets are difficult to come by in medical imaging, many strategies have been proposed to tackle this limitation. We briefly review some of these methods.

In transfer learning [9], the model is first trained on another domain/task and then fine-tuned for the target task. Transfer learning has been reported to improve the performance of CNN-based models for medical image segmentation [10], [11]. A limitation of transfer learning is that most public image datasets include 2D images, whereas medical images are 3D. Semi-supervised constitute a large and diverse body of techniques [4]. These methods utilize a mix of labeled and unlabeled data or data that have not been labeled in detail. These methods have been used in deep learning-based medical image segmentation [12]. One can use less accurate automatic methods to generate approximate segmentations on large corpora of images and use those to train a more accurate CNN-based model [13]. The potential benefits of semi-supervised methods strongly depend on the nature of the task and training data [14]. It may be easier to obtain rough segmentations on large datasets. Rather than treating such approximate segmentation labels as ground truth, one can use more intelligent methods [5]. Studies have reported successful applications of such methods [15].

An alternative to transfer learning is multi-task learning [9]. Multi-task learning aims at learning multiple tasks simultaneously. Studies have reported successful application of multi-task learning for medical image segmentation [16]. Training on mixed and heterogeneous data for medical image segmentation has been addressed in many prior works. These works appear under different titles such as domain adaptation, domain generalization, multi-site training, and joint training [17]. Many of these studies address domain shift, whereby a model trained on a source domain dataset fails to perform accurately and reliably on a target domain dataset. This is an important consideration in clinical settings [18], [19]. Another aim is to remove or reduce the need for labeled data in the target domain [20]-[22]. To deal with the changing image intensity and

appearance between source and target domains, a wide range of solutions have been proposed [23]-[25]. Some studies incorporate domain-invariant prior information as well as unlabeled and weakly-labeled images [26]. Another example is the Synergistic Image and Feature Adaptation framework, which encourages image similarity and invariance of the learned features across domains [27]. Some studies resort to data simulation methods to improve model generalizability [28]. Federated learning methods are another class of related techniques, where the focus is on training using multi-site data while protecting institutional and patient privacy [29]. The above studies on multi-task learning have focused on segmentation *accuracy*. The effect of multi-task training on confidence calibration has not been explored in previous works.

B. Model calibration and uncertainty estimation

All machine learning models are bound to make wrong predictions. One would like the confidence of a model's predictions to be proportional to the probability of making a correct prediction. Suppose that for test sample x_i the model predicts the class $\hat{y}_i$ with a probability $\hat{p}_i$. In the ideal scenario with perfect confidence calibration, $\mathrm{P}(\hat{y} = y \mid \hat{p} = p) = p$ [30]. Standard deep learning models are poorly calibrated [6]. This is concerning for safety-critical applications such as medicine. Many methods have been proposed for improving the calibration of deep learning models. It has been shown that calibration can be improved by using a proper scoring rule as the loss function [6], [31], weight decay, avoiding batch normalization [6], training on adversarial examples [31], and Platt scaling [6]. For image classification, a more more accurate method was proposed by combining Platt scaling with histogram binning [32]. Another study proposed to train a separate model to map the un-calibrated output of a CNN to calibrated probabilities [33].

Another study showed that Mixup training, [34], can improve the model calibration in image classification [35]. Mixup was originally proposed as a data augmentation and regularization strategy [34]. In brief, it synthesizes additional training data using convex combinations of pairs of training data points and their labels. Thulasidasan et al. show that Mixup also improves the model's confidence calibration in image classification [35].

For medical image segmentation, studies have proposed methods to estimate the prediction uncertainty [36] or to use the uncertainty for improving the segmentation accuracy [37]. However, little attention has been paid to methods for *improving* the calibration of CNN-based segmentation models. An example of the latter is [38], where authors show that the average prediction of a model ensemble is better calibrated than that of a single model. However, their method requires training as many as 50 models, which is inefficient.

C. Detecting out-of-distribution data and model failure

Another important problem is detection of OOD test data. A central assumption of every machine learning method is that the training and test data come from the same distribution. When a test sample comes from an entirely different distribution, the model should include a mechanism to detect the OOD sample. This is challenging with deep learning models because of their black-box nature and the complex mapping between their input and output. Advancements in network design have not made deep learning models more robust to OOD

data [39]. Different methods have been proposed to increase the *robustness* of deep learning models to OOD data. For example, deep learning models trained with Mixup are less prone to making over-confident predictions on OOD test data in image classification [35]. One study suggests that histogram equalization and Adversarial Logit Pairing improve robustness to corrupted data [40]. However, the latter study noted that methods that work well on some datasets may fail on other datasets. More importantly, these studies focus on in-distribution data that have been perturbed, and do not address the true OOD data, on which the model is expected to fail.

Several studies have proposed methods for OOD detection in deep learning. One work proposed a simple method that used the statistical distribution of the softmax values for detecting OOD data [41]. It showed that the maximum softmax value was higher for in-distribution and correctly-classified data samples than for OOD data samples. More recent studies have improved upon this basic method [7], [42]. One study showed that higher OOD detection accuracies could be achieved by 1) temperature scaling and 2) input perturbation [42]. The proposed image perturbation pushes the image in the direction that increases its predicted class. Authors show that this way of perturbing the input images increases the separation between in-distribution and OOD images. Although this operation pushes both in-distribution and OOD images closer to their predicted class, the shift is larger for in-distribution data samples than it is for OOD data samples. For image classification, one study trained Gaussian discriminant models on the penultimate network layer and used the Mahalanobis Distance to detect OOD data samples [7]. The authors extended their method by carrying out the same computation on all network layers. A different solution, again for image classification, has been proposed in [43]. In addition to the prediction head, a confidence head is added to the end of the CNN. The model directly estimates the prediction confidence as a scalar value. During training, the model is allowed to obtain a “hint" about the true class by using a weighted average of the true class and the model-predicted class in computing the prediction loss. However, the model is penalized for low-confidence predictions. At test time, OOD data samples are detected by thresholding the confidence score.

Deep k-Nearest Neighbors (DkNN) detects OOD test samples based on nonconformity of their representations with the representations of the training set [44]. Given a test sample, x_{test}, DkNN finds its k nearest neighbors in the training set in terms of representations in each of the network layers, $l \in 1 \ldots L$, and records the labels of those neighbors, Ω_l. Nonconformity of x_{test} with label j is defined as $\alpha(x_{\text{test}}, j) = \sum_l \mid i \in \Omega_l : i \neq j \mid$. DkNN also computes and stores nonconformity values for a *calibration set*, which is separate from the training set. Let us denote the set of nonconformity values of the calibration set with A. Then for a test sample x_{test} and label j, DkNN computes $p(x_{\text{test}}, j) = \mid \mid \{\alpha \in A : \alpha > \alpha(x\text{test}, j)\} \mid / \mid A \mid$. The largest $p(x_{\text{test}}, j)$ across all labels is referred to as prediction credibility. It quantifies the degree of support from the training set for the model prediction and is proposed as a measure of prediction confidence. We refer the reader to [44] for details.

Most methods proposed in prior works have been devised for 2D image classification settings and cannot be used for OOD detection in 3D medical image segmentation. This is

because, in volumetric image segmentation, feature maps are extremely large and typically only tens of training images are available. Some studies have proposed to detect OOD data based on prediction uncertainty [45]. However, such methods are not accurate in semantic segmentation applications [46]. In fact, compared with image classification, OOD detection in image segmentation has received much less attention. A recent study found that methods proposed for OOD detection in image classification do not translate well to segmentation tasks [47]. For segmentation of street view images, one study proposed a dedicated neural network to detect OOD data samples [46]. Their method classified an image as in-distribution or OOD using a very large “background dataset" to represent the variety of scenes outside of the training data distribution. However, it is difficult to obtain or even define the background set in medical imaging. The authors of [48] have proposed a method to automatically generate OOD data using a generative adversarial network (GAN) model that is trained in parallel with the main image classification network. One study evaluated the performance of several of the state of the art OOD detection methods for classification of 2D medical images and found that no single method achieved consistently-satisfying results, especially on samples that were closer to the in-distribution data [49]. Another study used prediction uncertainty measures to identify OOD data in medical image segmentation [38]. However, they evaluated their method on data that were hard to segment, not on true OOD data. As shown below by our results, methods based on prediction uncertainty cannot accurately detect OOD data in medical image segmentation.

It is worth mentioning that a related topic to OOD detection is the topic of adversarial examples [50]. These are examples that are crafted to fool a model into making wrong predictions. Adversarial examples may be important in some applications, but they are beyond the scope of this paper, which focuses on *natural* OOD data samples. By “natural OOD" we mean OOD data that exist due to such factors as a change in subject age, scanning protocol, or similar factors, as opposed to *artificially*-crafted OOD data created by an adversary.

D. Contributions of this work

In this paper, we address the problems outlined above and make the following contributions.

- *We show that multi-task learning can improve the confidence calibration of CNN models for medical image segmentation*. Through extensive experiments on a diverse collection of medical image segmentation datasets, we show that the confidence calibration of deep learning-based medical image segmentation models improves with multi-task learning. Importantly, in general, multi-task learning does not negatively impact the segmentation accuracy, and on some datasets it may slightly improve accuracy as well, even compared with other well-known training strategies such as transfer learning.
- *We propose a novel and accurate OOD detection method for CNN-based medical image segmentation*. Our method is based on spectral analysis of CNN feature maps. We show that whereas methods based on prediction uncertainty or image classification are inaccurate for OOD detection in 3D medical image

segmentation, our proposed method accurately detects OOD test data in different experimental settings.

III. Materials and Methods

A. Data

Table I summarizes the information about the datasets used in this work. The numbers of images in our datasets range from 15 to 400, which are typical of manually-labeled datasets in medical image segmentation, especially for in-house datasets because manual segmentation of complex 3D structures such as the brain cortical plate is time-consuming. Given the small size of the datasets, unless otherwise specified, we used a 3-fold cross-validation strategy. This way, all data are used for test, thereby increasing the power of the statistical significance tests. We always used a patient-wise data split. Computed Tomography (CT) images were normalized by mapping the Hounsfield Unit values in the range [−1000, 1000] to intensity range [0, 1]. Magnetic Resonance (MR) images were normalized by dividing each image by the standard deviation of voxel intensities. The first column in Table I shows the names that we use to refer to each dataset throughout this paper.

B. Network architecture and training details

The CNN architecture used in this work was based on the 3D U-Net [56], which we substantially modified by adding residual connections with short and long skip connections. The skip connections connect every fine feature map directly to all coarser feature maps in the encoder section of the network, similar to the DenseNet [57]. We set the number of features in the first stage of the encoder part of our network to 14, which was the largest allowed by our GPU memory. The model worked on 96^3-voxel image blocks. During training, we sampled blocks from random locations in the training images, which acts as a form of data augmentation. Other data augmentation methods that we used during training included random flips and rotations (by integer multiples of $\pi/2$) in all directions as well as addition of random Gaussian noise to voxel intensity values. We also experimented with elastic deformation for data augmentation, [58], but we did not pursue that augmentation method because it negatively impacted segmentation of fine structures such as brain cortical plate. On a test image, a sliding window approach with a 24-voxel overlap between adjacent blocks was used to process the image. We used the negative of the Dice Similarity Coefficient (DSC) as the loss function and Adam [59] as the optimization method. We used an initial learning rate of 10^{-4}, which was reduced by 0.90 after every 2000 training iterations if the loss did not decrease. If the loss did not decrease for two consecutive evaluations, we stopped the training. Since the focus of the study is on model calibration and OOD detection, we used the same settings mentioned above in all experiments.

C. Multi-task learning

In this work we advocate for training on heterogeneous data, which we refer to as “multi-task learning", although multi-task learning has also been used in other settings as we have explained in Section II-A. We train a single model on a mix of training datasets that can come from different imaging modalities with different organs of interest to be segmented. We do not use additional inputs to inform the model of the image modality or the organ

that needs to be segmented. Furthermore, the network will still have a single segmentation head (i.e., single output layer). Given the input image from any modality and organ of interest, the network will output the predicted segmentation map of the organ of interest on the segmentation head. Given the GPU memory limit, we use a training batch size of one. For multi-task learning, we sample a block from one of the images in the training set and use that block and its corresponding ground truth segmentation map to update the model parameters. In other words, no changes are made to the network architecture and overall training procedures compared with training on a single dataset. The only point worth mentioning is the frequency of sampling from different training datasets. We sample from each dataset with a probability proportional to the inverse of the square root of dataset size, $1 / \sqrt{n}$. This way, if for example we train on two datasets with 10 and 100 images each, the probability of sampling an image from these two datasets will be 0.24 and 0.76, respectively ($\frac{\sqrt{10}}{\sqrt{10}+\sqrt{100}} = 0.24$ and $\frac{\sqrt{100}}{\sqrt{10}+\sqrt{100}} = 0.76$). We found that, especially on datasets with fewer images, using $1 / \sqrt{n}$ resulted in higher accuracy than using $1 / n$, which is equivalent to uniform sampling.

D. OOD detection using spectral analysis of feature maps

We propose a novel method for detecting OOD test data for CNN-based medical image segmentation models. As mentioned above, these models produce over-confident predictions even when a test sample is entirely outside the distribution of the training data. We show in Section IV that methods based on prediction uncertainty are unable to accurately detect OOD data. Moreover, because of the very large size of 3D medical images and their computed feature maps and small number of training images, methods based on analyzing the feature maps in their native space are ineffective. This is because these methods have been developed for scenarios where the size of the feature vector is on the order of hundreds or a few thousands and millions of training samples are available. We described several of these methods in Section II-C above. In Section IV, we show that such methods are not accurate for OOD detection in medical image segmentation applications. Instead, we propose computing the spectrum of the feature maps, which we define as the vector of singular values computed using singular value decomposition (SVD). Consider a test image x_i and denote the feature map computed for this image at a certain stage (i.e., layer) of the network with $F_i \in \mathbb{R}^{w,h,d,n}$, where w, h, d denote the dimensions of the feature map and n is the number of features. We reshape F_i as $\mathbb{R}^{whd,n}$ and compute the SVD of F_i as $F_i = USV$, where U and V are orthonormal matrices and the diagonal matrix S contains the singular values of F_i, which is referred to as its spectrum [60]. The vector of singular values, $s = \mathrm{diag}(S)$, depends on the magnitude of the feature values, which in turn depend on the image voxel intensities. Moreover, the spectrum has a very large dynamic range. To eliminate these effects, we take the logarithm of the spectrum s and then normalize it so that it has an ℓ_2 norm of unity. We refer to the normalized logarithmic spectrum of the feature maps computed as explained above as “the spectral signature". We still denote this spectral signature with s in the following.

Figure 2 shows example signatures, where a model trained on several datasets from Table I including CP-younger fetus and Liver-MRI-SPIR datasets but not including Pancreas and

Hippocampus. The figure shows example spectral signatures of training images from these four datasets. Each dataset has a distinct spectral signature. Note that this model segments CP-younger fetus and Liver-MRI-SPIR accurately, but fails on Pancreas and Hippocampus, which have not been seen during training. Nonetheless, as we show in Section IV, methods based on prediction uncertainty cannot detect these as OOD.

We propose detecting ooD data based on the similarity of spectral signatures. We compute the spectral signatures of all training images and store them as S_{train}. Given a test image, x^i_{test}, we compute its spectral signature s^i_{test}. We define Out-Of-Distribution Measure (OODM) of x^i_{test} as the Euclidean distance of its spectral signature, s^i_{test}, to its nearest neighbor in the training set:

$$\text{OODM}(x^i_{\text{test}}) = \min_j \left(\| s^i_{\text{test}} - s^i_{\text{train}} \|_2 \,,\ s^j_{\text{train}} \in S_{\text{train}} \right) \tag{1}$$

We expect OODM to be smaller for test images coming from the distribution of the training set than for images from other distributions. We declare a test image x^i_{test} to be OOD if $\text{OODM}(x^i_{\text{test}}) > \tau$. We determine the threshold τ using the training set. On the training set we compute the vector of $\text{OODM}_{\text{train}}$ using Eq. (1) using a leave-one-out strategy. We then compute τ as:

$$\tau = \text{mean}(\text{OODM}_{\text{train}}) + C \times \text{std}(\text{OODM}_{\text{train}}), \tag{2}$$

where we set $C = 2.5$ for computing the detection accuracy. The value of τ, and hence C, determine the trade-off between sensitivity and specificity. A larger τ reduces the false positive rate while also reducing the true positive rate. The area under the receiver-operating characteristic curve (AUC) is the standard measure that is used to characterize this trade-off [61], [62]. The receiver operating characteristic curve is the plot of the true positive rate versus the false positive rate at various settings of the detector threshold [61]. In our proposed model, we compute the AUC by changing our threshold, τ.

One can apply the above method on any of the network's feature maps. We found that using the deepest feature maps (i.e., feature maps closest to the output) led to more accurate OOD detection. This is in agreement with the fact that deeper layers provide more disentangled manifolds [63]. We applied the method on the last feature maps, which had 14 channels. Hence, the length of the spectral signatures in this work is 14. Figure 2 displays example histograms of OODM values for training data, in-distribution test data, and OOD test data, showing that OODM easily separates in-distribution from OOD data in this experiment.

We compare our proposed method with:

(a) A common method based on prediction uncertainty [45]. We train our model using dropout (with a rate of 10%) in each layer. At test time, we draw $N = 10$ random dropout masks and compute the entropy of the mean of the segmentation probability maps, $H(\bar{p}) = -\bar{p}\log(\bar{p})$ as the estimated voxel-wise prediction uncertainty map. We use the average of voxel-wise uncertainty on the predicted foreground as image-wise uncertainty.

Similar to our proposed method, we compute a threshold similar to Eq. (2) on the training set. We refer to this method as UNC-Dropout.

(b) A method based on model ensembles [31]. Recently, this method was used in medical image segmentation in [38]. We refer to this method as UNC-Ensemble. **(c)** DkNN [44], explained in Section II-C We use 20% of images as the calibration set. From each image, we generate 100 data samples via augmentation. For a test image, we sample 10 blocks overlapping the predicted foreground and estimate credibility as the mean of $p(x_{\text{test}}, j')$, where j' is the most frequent label on the 10 blocks. We use credibility as a measure of confidence as suggested in [44]. This method can only be applied when the training set has more than one classes.

(d) Method of Outlier Exposure [64]. This method is based on training the model against a dataset of outliers. The proposed loss function includes an outlier exposure term that has the form $\mathbb{E}_{x' \sim \mathcal{D}_{\text{out}}^{OE}}(\mathcal{L}_{OE}(f(x'), f(x), y)$, where $\mathcal{D}_{\text{out}}^{OE}$ is the outlier exposure dataset used during training. As suggested by [64], we use the cross-entropy, which is also the loss proposed by [48]. This method assumes the exact nature of the outlier data is unknown. Therefore, following the recommendations of [64], we used all other datasets listed in Table I. For example, if in an experiment we use “CP- younger fetus" as in-distribution and “CP- older fetus" and “CP- newborn" as OOD, we use all other 11 datasets in Table I as $\mathcal{D}_{\text{out}}^{OE}$. We refer to this method as Outlier Exposure.

(e) We compare with the method of [48], which is based on the inspection of softmax values as proposed by [41]. Specifically, OOD samples are detected as those for which the maximum softmax value is smaller than a threshold. However, authors of [48] propose two training strategies. First, they use an extra loss term to encourage the distribution of softmax values for OOD data samples to be close to a uniform distribution. They also use a GAN, whose task is to generate informative OOD training samples. The GAN model is trained in an alternative optimization framework in parallel with the main model. We refer to this method as Lee-2017.

(f) We compare with ODIN (Out-of-DIstribution detector for Neural networks) [42]. This method is based on the softmax values originally proposed by [41] that we have explained above. ODIN introduced two additional tricks: 1) temperature scaling, and 2) image perturbation. We have provided more detail on this method above in Section II-C.

(g) We also compare with the method proposed in [7], which followed the work of [42] and showed to be more accurate in image classification. As we briefly explained above, this method is based on the idea of a generative classifier. It assumes that the features can be modeled with class-conditional Gaussian distributions. Based on this assumption, the authors propose a confidence score based on the Mahalanobis Distance. Given the very large size of the feature maps, we use average pooling, as suggested by [7]. Since this method is based on the Mahalanobis Distance, we refer to it as Mah-Dist.

E. Evaluation metrics

We quantify segmentation accuracy using DSC, 95 percentile of the Hausdorff Distance (HD95) and Average Symmetric Surface Distance (ASSD). We assess model calibration by computing the Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) [65]. ECE and MCE are computed by dividing the probability range, [0,1], into K bins. Denote by S_k the set of points (i.e., voxels in a semantic segmentation problem) whose predicted probabilities fall in the interval $[\frac{k-1}{K}, \frac{k}{K}]$, for $k \in [1, K]$. Then ECE is defined as [6], [65]:

$$ECE = \sum_{k=1}^{K} \frac{\mid S_k \mid}{N} \mid \text{acc}(S_k) - \text{conf}(S_k) \mid , \tag{3}$$

where $\text{acc}(S_k)$ is the average prediction accuracy for the points in S_k and $\text{conf}(S_k)$ is the mean prediction probability for the points in S_k. MCE is defined as [6], [65]:

$$MCE = \max_k \mid \text{acc}(S_k) - \text{conf}(S_k) \mid . \tag{4}$$

Values of calibration measures such as ECE can be dramatically influenced by the large percentage of background voxels. These voxels are usually correctly and confidently segmented by a relatively good model. Including these voxels in the computation of calibration measures artificially improves the values of these measures. Hence, we dilate the surface bound-ary of the ground-truth foreground (in 3D, in both directions in/out) in each image by 10 voxels and use the obtained mask for computing the calibration measures. A similar strategy was used in [38]. For OOD detection, we report accuracy, sensitivity, and specificity. Furthermore, we compute the area under the receiver-operating characteristic curve (AUC) by changing the value of τ.

IV. Results and Discussion

A. Feasibility and benefits of multi-task learning

As we mentioned above, we advocate for training a single model to segment different organs in different imaging modalities. To show that this is a viable approach, we trained a model on twelve datasets spanning various organs in MRI and cT images. We then trained twelve separate models, one on each of the datasets. We show a comparison of the test performance of these two training strategies in Table II, where the statistically significant differences have been marked using bold type. All statistical significance tests were performed using paired t-tests with a significance threshold of $p = 0.01$.

The results show that in general multi-task learning improves model prediction calibration. The joint model was significantly better-calibrated than the dedicated models on 9 out of 12 datasets in terms of ECE and on 10 out of 12 datasets in terms of McE. only on the Hippocampus dataset the dedicated model was significantly better calibrated than the joint model. Hippocampus dataset included 260 images, compared with 15-32 images in eight of the other datasets used in this experiment. Therefore, this could be the influence of dataset

size on the potential benefits of multi-task learning. However, there are three other large datasets in this experiment: CP-newborn ($n = 400$), Pancreas ($n = 281$), and KiTs ($n = 300$). For Pancreas and KiTs, multi-task learning resulted in better-calibrated models in terms of all four calibration measures (ECE and MCE). For CP-newborn, too, multi-task learning resulted in a better-calibrated model in terms of ECE. We repeated the above experiment with a randomly-selected subset of $n = 30$ images from this dataset. We observed that with this reduced dataset size, multi-task learning indeed resulted in statistically significantly better calibrated models. With $n = 30$, a single model trained on the Hippocampus dataset achieved ECE and MCE of 0.16±0.04 and 0.32±0.08, respectively, whereas multi-task learning achieved ECE and MCE of 0.14±0.03 and 0.27±0.05, respectively, which were both significantly lower. Hence, multi-task learning in general improves the model prediction calibration, even for relatively large datasets. However, on some segmentation tasks (e.g., Hippocampus in this experiment) the positive effects of multi-task learning may disappear if large datasets are available. Figure 3 shows examples of estimated uncertainty maps. For this figure, we have intentionally selected test images on which the segmentation accuracy was relatively low. The figure shows that the model displays high segmentation uncertainty at the locations where segmentation error occurs, visually confirming that the model is well-calibrated.

In terms of segmentation accuracy, the results are somewhat mixed. Nonetheless, on average Table II shows that training a single model on a pool of heterogeneous datasets can achieve results that are as good as or even better than when dedicated models. Multi-task learning achieved significantly higher DSC on five datasets, significantly lower HD95 on seven datasets, and significantly lower AssD on four datasets. only on one of the datasets (Hippocampus) multitask learning was significantly worse. Given the small size of most of our datasets, the fact that a single model can learn to automatically recognize the context and accurately segment the organ of interest is interesting. Figure 4 shows example slices of several test images from different datasets and the computed segmentations. They show that the joint model accurately segments different organs in different imaging modalities. Please note that here our aim was to improve the model confidence calibration and not to improve the segmentation accuracy. As we showed above, the proposed strategy of training on heterogeneous data did improve the model confidence calibration on the overwhelming majority of the datasets. The fact that it also “on average" improved the segmentation accuracy is an incidental advantage.

We investigated the effect of mixup, [34], on the calibration and segmentation accuracy with the same twelve datasets used in Table II. As we mentioned above, recent works have shown that mixup can improve calibration of image classification models [35] and segmentation accuracy [66]. However, those studies used images from the same modality and organ. In our setting with a mix of twelve different datasets, mixup resulted in a consistent *deterioration* of model performance and calibration. on all twelve datasets, mixup resulted in lower DSC, often by large margins. For example on CP-younger fetus, Heart, and Hippocampus datasets, DSC dropped by approximately 0.09-0.13 compared with the results shown in Table II, which were statistically significant reductions. similarly, model calibration was worse when mixup was used. Therefore, mixup is very ineffective on

datasets with heterogeneous modalities and organs of interest. Below, we present the results of using mixup on a more homogeneous set of data, i.e. CP segmentation in MRI.

Here we present an experiment to compare the multi-task learning approach with transfer learning using the three cortical plate datasets (see Table I). As shown in Figure 5, the shape and complexity of cortical plate evolves dramatically before and after birth. In addition, the sizes of the three datasets are highly unequal. Given the much smaller sizes of two of the datasets, transfer learning is the method that is recommended by some studies [4], [10]. Table III compares the results obtained with different transfer learning trials and the results obtained with multi-task learning. In each of the transfer learning trials, we first trained the model to convergence on one of the datasets and then fine-tuned it on another dataset. We then further fine-tuned the model on the third dataset. our fine-tuning strategy was “deep fine-tuning" [10]; we reduced the initial learning rate by half and fine-tuned all model layers. Shallow fine-tuning and keeping the initial learning rate produced inferior results. We also tried other fine-tuning curricula, i.e., orders of datasets used in fine-tuning, but did not achieve better results than those in Table III. For each dataset, we performed paired t-tests between the four results (i.e., the three transfer learning trials and the multi-task learning trial). statistically better results, at $p = 0.01$, were marked with bold type.

Table III shows that the joint model had better-calibrated predictions than models trained on individual datasets as well as all three transfer learning trials on all three datasets in terms of ECE and MCE. In terms of accuracy, the multi-task learning approach was similar to or better than transfer learning. Although transfer learning improved the segmentation accuracy in some cases, the improvements were marginal. Multi-task learning, on the other hand, resulted in statistically significant improvements in segmentation accuracy. For the smallest dataset, i.e., CP-older fetus, multi-task learning significantly improved DSC, HD95, and ASSD. For the other two datasets, multi-task learning significantly improved ASSD. Figure 6 shows segmentation results on example test images from the three cortical plate segmentations for models trained on each of the three datasets, separately, and also for the joint model. The results show that a model trained on all three datasets can segment the test data from all three datasets with high accuracy. However, a model trained on each one of the datasets may perform very poorly on the test images from the other two datasets.

Table IV presents a comparison of multi-task learning with mixup on the same three CP datasets. Unlike with the twelve heterogeneous datasets in Table II, in this experiment mixup works and it does improve the segmentation accuracy of the model compared with standard training results presented in Table III. However, improvements in model calibration due to mixup are marginal, and ECE and MCE values achieved with multi-task learning are significantly better than those with mixup on all three datasets.

An additional appeal of a joint model that accurately segments all three datasets is that one would need to maintain only one set of model weights. When trained on one of the datasets, the model will perform poorly on the other datasets, one would need to maintain three separate models, and for a test image one would need to know which of the three datasets the image belongs to. Furthermore, the time required to train a single model on several datasets is generally less than the time needed to train separate models. For example,

for CP segmentation, the training time for a model to segment cp-younger fetus, CP-older fetus, and CP-newborn is approximately 9 hours, whereas the training time for a model for each of these three datasets is approximately 5 hours, for a total of approximately 15 hours.

B. Detecting OOD test data

We present the results of OOD detection with our method and competing methods in four different experiments.

In the first experiment, we used a mixture of eight different datasets for training. These included CP-younger fetus, CP-older fetus, Prostate, Heart, Liver-CT, Liver-MRI-SPIR, Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out datasets. We used test images from the same eight dataset as in-distribution data. As OOD data, we used Pancreas, Hippocampus, and Spleen datasets. Histograms of OODM have been shown in the lower part of Figure 2. Table V compares different methods for OOD detection. Our method perfectly detected the OOD images. UNC-Dropout failed, achieving an accuracy of 0.55. The other methods performed better than UNC-Dropout, but none of them achieves the level of accuracy of our method. Among the competing methods, UNC-Ensemble achieved better results, but it requires training tens of models. Following the recommendations of [38], we trained 50 models for this method.

In the second experiment, we trained a model on CP-newborn and applied the model on test images from the same dataset and the other two CP datasets. Figure 7 shows OODM histograms for this experiment. The OODM values for both CP-younger fetus and CP-older fetus fall outside of the distribution of CP-newborn. This model achieved DSC of 0.689 ± 0.095 and 0.781 ± 0.028 on CP-younger fetus and CP-older fetus datasets, respectively. These are very low compared with the results shown in Table III. Therefore, images from CP-younger fetus and CP-older fetus datasets are OOD. Our proposed method easily distinguished OOD data from in-distribution data. It is interesting to note that OODM values for CP-younger fetus are distributed farther away than those of CP-older fetus. This makes sense because CP-younger fetus is less similar to CP-newborn than CP-older fetus is. As shown in Table VI, our method accurately separated OOD data samples from in-distribution data samples. UNC-Dropout achieved an accuracy of 0.57, while UNC-Ensemble and Mah-Dist achieved 0.80. Outlier Exposure was the best of the competing methods, but still achieved accuracy, sensitivity, and specificity of 20% lower than the proposed method. DkNN cannot be applied in this experiment because the training set has only one class.

In another experiment, we trained our model on CP-younger fetus dataset and tested on the other two CP datasets and on four completely different datasets: Heart, Liver-CT, Hippocampus, and Pancreas. This model achieved a DSC of 0.788 ± 0.045 and 0.765 ± 0.061 on CP-older fetus and CP-newborn datasets, respectively, which are lower than the results in Table III. Also, as expected, on the other four datasets the model failed, achieving a mean DSC of 0.20-0.45. We present the results of this experiment separately for two CP datasets and the four non-CP datasets. Table VII shows the OOD detection accuracy results for different methods. For both CP and non-CP datasets our method achieved high detection accuracy and performed better than the other techniques. Among the competing methods, UNC-Ensemble and Outlier exposure achieved better results, but the accuracy for

our method was much higher. Figure 8 displays the histograms of the OODM values for this experiment.

Finally, we report an experiment with the three liver MRI datasets (See Table I). The top section of Figure 9 shows a sample image from each of these datasets. This experiment demonstrates that OOD data are often not easy to distinguish visually. Our experiments show that a model trained on Liver-MRI-SPIR and Liver-MRI-DUAL-In accurately segments images from Liver-MRI-DUAL-Out (mean DSC= 0.89). Similarly, a model trained on Liver-MRI-SPIR and Liver-MRI-DUAL-Out achieved a mean DSC of 0.86 on Liver-MRI-DUAL-In. Even a model trained on Liver-MRI-DUAL-SPIR alone, accurately segmented Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out. On the other hand, a model trained on Liver-MRI-DUAL-In and/or Liver-MRI-DUAL-Out failed on images from Liver-MRI-SPIR (mean DSC ≈ 0.40). These observations are not intuitive and are not at all easy to foretell by visually inspecting these images. Specifically, there are asymmetries that cannot be predicted by visual inspection. As an example, as mentioned above, Liver-MRI-DUAL-SPIR *is* OOD for a model trained on Liver-MRI-DUAL-In but Liver-MRI-DUAL-In *is not* OOD for a model trained on Liver-MRI-DUAL-SPIR. This example further highlights the importance and challenging nature of OOD detection in CNN-based medical image segmentation.

Figure 9(a) shows OODM histograms for an experiment where Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out were used for training. The OODM values were computed on the test data from the same datasets and Liver-MRI-SPIR, which is OOD for this model. Our proposed method easily separates in-distribution from OOD data. Table VIII shows that UNC-Dropout and DkNN have low accuracies. UNC-Ensemble achieved better results, but still has an accuracy and AUC that are approximately 20% lower than our proposed method. Figure 9(b) shows the OODM for an experiment where Liver-MRI-SPIR and Liver-MRI-DUAL-In were used for training. The trained model works well on Liver-MRI-DUAL-Out dataset as well. As desired, the OODM values for most images from Liver-MRI-DUAL-Out fall below τ, and hence correctly classified as in-distribution. Figure 9(c) shows the same for an experiment in which Liver-MRI-SPIR and Liver-MRI-DUAL-Out were used for training.

In terms of computation time, our method processes an image in approximately 5 seconds on a Linux machine with 32 GB of memory and an NVIDIA GeForce GTX 1080 GPU. With our implementation of UNC-Dropout, UNC-Ensemble, Mah-Dist, ODIN, and DkNN, they take approximately 4, 15, 5, 12, and 35 seconds, respectively. Another advantage of our OOD detection method is relative simplicity. The only hyperparameter in the OOD detection method itself is C that determines the threshold, τ. This value influences the trade-off between sensitivity and specificity. In all of our experiments we reported the value of AUC, which accounts for this trade-off. The accuracy/sensitivity/specificity values reported above were all obtained with $C = 2.5$. The actual impact of the value of C depends on the experiment. For example, for the experiment shown in Figure 8(a) using $C = 2.0$ would result in sensitivity and specificity of 0.96 and 0.86, respectively, and using $C = 3.0$ would result in sensitivity and specificity of 0.88 and 1.00, respectively. In Figure 8(b), on the other hand, sensitivity and specificity remain at 1.00 for all value in the range $C \in [2.0, 3.0]$.

Applying the method on other feature maps resulted in lower accuracy than on the last layer. As an example, in the experiment with brain cortical plate datasets reported in Table V, applying the method on the 2nd deepest feature map resulted in accuracy, sensitivity, and specificity of 0.92, 0.89, and 0.92, respectively. Moreover, applying the method on the 3rd deepest feature map resulted in accuracy, sensitivity, and specificity of 0.90, 0.88, and 0.91, respectively. On the coarsest feature maps (i.e., the last encoder feature maps) the achieved accuracy, sensitivity, and specificity were 0.84, 0.83, and 0.86, respectively. Better accuracy achieved at deeper layers is likely due to the higher degree of disentanglement in deeper layers [63].

This study used a large number of dataset to evaluate the proposed methods. Nonetheless, the variability in medical image data is very high. It would be instructive to explore the potential of the proposed methods in multi-label and multi-class settings. In one experiment, we repeated the experiment reported in Table VII by considering a two-class segmentation for the Hippocampus data, where the Hippocampus dataset has two segmentation labels (anterior and posterior). Therefore, the output layer in this experiment had one extra channel for the Hippocampus dataset. For other datasets this channel is padded with zeros in the training data. With this setting, our OOD detection method achieved an accuracies of 0.94 and 1.00 on the CP and non-CP test data, which are similar to the results presented in Table VII. Another potentially important factor is image resolution. Increasing image resolution beyond the native (acquisition) resolution by upsampling did not significantly improve the segmentation accuracy or confidence calibration. even for fine structures such as the brain cortical plate. This may be due to the fact that after image upsampling the corresponding segmentations also need to be upsampled. For fine structures such as cortical plate, expert-provided labels are most accurate in the resolution used during manual annotation. Upsampling leads to inevitable errors in the training labels of fine structures that may contribute to model inaccuracy and poor calibration. Downsampling the images and labels substantially reduced the segmentation accuracy and calibration on fine structures such as cortical plate. significantly increasing/decreasing image resolution can also render an in-distribution test image OOD with respect to a model trained on images with very different resolutions. In terms of segmentation accuracy, another factor is the probability of sampling from different datasets. As we mentioned in section III-C, we sampled from each dataset with a probability proportional to the inverse of the square root of dataset size, $1 / \sqrt{n}$. This usually resulted in higher accuracy than uniform sampling (i.e., $1 / n$). As an example, using a uniform sampling for the experiment reported in Table II resulted in DSC, HD95, and ASSD of 0.87±0.04, 0.88±0.03, and 0.24 ± 0.08 for CP- younger fetus, and 0.87 ± 0.11, 8.9 ± 7.8, and 2.20 ± 2.28 for the Heart dataset. These are slightly worse than the results presented in Table II.

V. Conclusion

We showed that, compared with the standard approach of training on one dataset, multi-task learning can improve the confidence calibration of CNN-based medical image segmentation models. our results showed that multi-task learning leads to lower calibration errors in terms of ECE and MCE and strong spatial correlation between prediction confidence and segmentation accuracy. Additional benefits of multi-task learning include overall higher

segmentation accuracy and generalizability of the trained model across datasets, imaging modalities, and age groups. our proposed OOD detection method proved to be very accurate in several experiments with different datasets. As we showed in our experiments on liver segmentation in MRi, visually identifying OOD data could be non-trivial. Therefore, reliable deployment of CNN-based segmentation models for medical applications requires accurate OOD detection methods. This has been a challenging problem because of the massive size and complexity of deep learning models. Previous studies have used measures of prediction uncertainty for this purpose. But our experiments show that such methods are inaccurate. To the best of our knowledge, this is the first study to propose a method for OOD detection in medical image segmentation by analyzing CNN features. Some prior works, such as Mah-Dist [7], are based on modeling the distribution of features, which may be accurate for image classification, but, as our experiments show, are bound to fail in 3D medical image segmentation. Similarly, many previous methods have been tailored for natural image classification and perform poorly in medical image segmentation, as our experiments with DkNN have shown. Therefore, our proposed OOD detection method offers a solution to a hitherto-unsolved problem.

Acknowledgments

Research reported in this publication was supported in part by the National Institutes of Health (NIH) grants R01 EB018988, R01 NS106030, and R01 EB031849; and in part by the Office of the Director of the NIH under award number S10OD0250111. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1]. Taghanaki SA, Abhishek K, Cohen JP, Cohen-Adad J, and Hamarneh G, “Deep semantic segmentation of natural and medical images: A review,” Artificial Intelligence Review, pp. 1–42, 2020. [PubMed: 32836651]

[2]. Isensee F, Kickingereder P, Wick W, Bendszus M, and Maier-Hein KH, “No new-net,” in International MICCAI Brainlesion Workshop. Springer, 2018, pp. 234–244.

[3]. Xie Y, Zhang J, Lu H, Shen C, and Xia Y, “Sesv: Accurate medical image segmentation by predicting and correcting errors,” IEEE Transactions on Medical Imaging, vol. 40, no. 1, pp. 286–296, 2020. [PubMed: 32956049]

[4]. Cheplygina V, de Bruijne M, and Pluim JP, “Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis,” Medical image analysis, vol. 54, pp. 280–296, 2019. [PubMed: 30959445]

[5]. Karimi D, Dou H, Warfield SK, and Gholipour A, “Deep learning with noisy labels: exploring techniques and remedies in medical image analysis,” arXiv preprint arXiv:1912.02911, 2019.

[6]. Guo C, Pleiss G, Sun Y, and Weinberger KQ, “On calibration of modern neural networks,” arXiv preprint arXiv:1706.04599, 2017.

[7]. Lee K, Lee K, Lee H, and Shin J, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in Advances in Neural Information Processing Systems, 2018, pp. 7167–7177.

[8]. LeCun Y, Bengio Y, and Hinton G, “Deep learning,” nature, vol. 521, no. 7553, p. 436, 2015. [PubMed: 26017442]

[9]. Pan SJ and Yang Q, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[10]. Tajbakhsh N et al. , “Convolutional neural networks for medical image analysis: full training or fine tuning?” IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1299–1312, 2016. [PubMed: 26978662]

[11]. Karimi D, Warfield SK, and Gholipour A, “Critical assessment of transfer learning for medical image segmentation with fully convolutional neural networks,” arXiv preprint arXiv:2006.00356, 2020.

[12]. Enguehard J, OâĂŹHalloran P, and Gholipour A, “Semi-supervised learning with deep embedded clustering for image classification and segmentation,” IEEE Access, vol. 7, pp. 11093–11104, 2019. [PubMed: 31588387]

[13]. Zhang L, Gopalakrishnan V, Lu L, Summers RM, Moss J, and Yao J, “Self-learning to detect and segment cysts in lung ct images without manual annotation,” in 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). IEEE, 2018, pp. 1100–1103.

[14]. Oliver A, Odena A, Raffel CA, Cubuk ED, and Goodfellow I, “Realistic evaluation of deep semi-supervised learning algorithms,” in Advances in Neural Information Processing Systems, 2018, pp. 3235–3246.

[15]. Wang G, Liu X, Li C, Xu Z, Ruan J, Zhu H, Meng T, Li K, Huang N, and Zhang S, “A noise-robust framework for automatic segmentation of covid-19 pneumonia lesions from ct images,” IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2653–2663, 2020. [PubMed: 32730215]

[16]. Harouni A, Karargyris A, Negahdar M, Beymer D, and Syeda-Mahmood T, “Universal multi-modal deep network for classification and segmentation of medical images,” in 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). IEEE, 2018, pp. 872–876.

[17]. Huang C, Han H, Yao Q, Zhu S, and Zhou SK, “3d u2-net: a 3d universal u-net for multi-domain medical image segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2019, pp. 291–299.

[18]. Liu Q, Dou Q, Yu L, and Heng PA, “Ms-net: multi-site network for improving prostate segmentation with heterogeneous mri data,” IEEE transactions on medical imaging, vol. 39, no. 9, pp. 2713–2724, 2020. [PubMed: 32078543]

[19]. Li L, Zimmer VA, Ding W, Wu F, Huang L, Schnabel JA, and Zhuang X, “Random style transfer based domain generalization networks integrating shape and spatial information,” in International Workshop on Statistical Atlases and Computational Models of the Heart. Springer, 2020, pp. 208–218.

[20]. Li H, Loehr T, Sekuboyina A, Zhang J, Wiestler B, and Menze B, “Domain adaptive medical image segmentation via adversarial learning of disease-specific spatial patterns,” arXiv preprint arXiv:2001.09313, 2020.

[21]. Zhang Y, Miao S, Mansi T, and Liao R, “Task driven generative modeling for unsupervised domain adaptation: Application to x-ray image segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2018, pp. 599–607.

[22]. Dong J, Cong Y, Sun G, Zhong B, and Xu X, “What can be transferred: Unsupervised domain adaptation for endoscopic lesions segmentation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4023–4032.

[23]. Ouyang C, Kamnitsas K, Biffi C, Duan J, and Rueckert D, “Data efficient unsupervised domain adaptation for cross-modality image segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2019, pp. 669–677.

[24]. Xia Y, Yang D, Yu Z, Liu F, Cai J, Yu L, Zhu Z, Xu D, Yuille A, and Roth H, “Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation,” Medical Image Analysis, vol. 65, p. 101766, 2020. [PubMed: 32623276]

[25]. Chang W-G, You T, Seo S, Kwak S, and Han B, “Domain-specific batch normalization for unsupervised domain adaptation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7354–7362.

[26]. Bateson M, Dolz J, Kervadec H, Lombaert H, and Ayed IB, “Constrained domain adaptation for image segmentation,” IEEE Transactions on Medical Imaging, 2021.

[27]. Chen C, Dou Q, Chen H, Qin J, and Heng P-A, “Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 865–872.

[28]. Gholami A, Subramanian S, Shenoy V, Himthani N, Yue X, Zhao S, Jin P, Biros G, and Keutzer K, “A novel domain adaptation framework for medical image segmentation,” in International MICCAI Brainlesion Workshop. Springer, 2018, pp. 289–298.

[29]. Liu Q, Chen C, Qin J, Dou Q, and Heng P-A, “Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1013–1023.

[30]. Zadrozny B and Elkan C, “Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers,” in Icml, vol. 1. Citeseer, 2001, pp. 609–616.

[31]. Lakshminarayanan B, Pritzel A, and Blundell C, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Advances in Neural Information Processing Systems, 2017, pp. 6402–6413.

[32]. Kumar A, Liang PS, and Ma T, “Verified uncertainty calibration,” in Advances in Neural Information Processing Systems, 2019, pp. 3787–3798.

[33]. Maroñas J, Paredes R, and Ramos D, “Calibration of deep probabilistic models with decoupled bayesian neural networks,” arXiv preprint arXiv:1908.08972, 2019.

[34]. Zhang H, Cisse M, Dauphin YN, and Lopez-Paz D, “mixup: Beyond empirical risk minimization,” arXiv preprint arXiv:1710.09412, 2017.

[35]. Thulasidasan S, Chennupati G, Bilmes JA, Bhattacharya T, and Michalak S, “On mixup training: Improved calibration and predictive uncertainty for deep neural networks,” in Advances in Neural Information Processing Systems, 2019, pp. 13888–13899.

[36]. Wang G, Li W, Aertsen M, Deprest J, Ourselin S, and Vercauteren T, “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks,” Neurocomputing, vol. 338, pp. 34 – 45, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219301961

[37]. Karimi D et al. , “Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound images,” Medical Image Analysis, vol. 57, pp. 186 – 196, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841519300623 [PubMed: 31325722]

[38]. Mehrtash A, Wells III WM, Tempany CM, Abolmaesumi P, and Kapur T, “Confidence calibration and predictive uncertainty estimation for deep medical image segmentation,” arXiv preprint arXiv:1911.13273, 2019.

[39]. Hendrycks D and Dietterich T, “Benchmarking neural network robustness to common corruptions and perturbations,” arXiv preprint arXiv:1903.12261, 2019.

[40]. Kannan H, Kurakin A, and Goodfellow I, “Adversarial logit pairing,” arXiv preprint arXiv:1803.06373, 2018.

[41]. Hendrycks D and Gimpel K, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” arXiv preprint arXiv:1610.02136, 2016.

[42]. Liang S, Li Y, and Srikant R, “Enhancing the reliability of out-of-distribution image detection in neural networks,” arXiv preprint arXiv:1706.02690, 2017.

[43]. DeVries T and Taylor GW, “Learning confidence for out-of-distribution detection in neural networks,” arXiv preprint arXiv:1802.04865, 2018.

[44]. Papernot N and McDaniel P, “Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning,” arXiv preprint arXiv.1803.04765, 2018.

[45]. Kendall A and Gal Y, “What uncertainties do we need in bayesian deep learning for computer vision?” in Advances in neural information processing systems, 2017, pp. 5574–5584.

[46]. Bevandić P, Krešo I, Oršić M, and Šegvić S, “Discriminative out-of-distribution detection for semantic segmentation,” arXiv preprint arXiv:1808.07703, 2018.

[47]. Angus M, Czarnecki K, and Salay R, “Efficacy of pixel-level ood detection for semantic segmentation,” arXiv preprint arXiv:1911.02897, 2019.

[48]. Lee K, Lee H, Lee K, and Shin J, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” arXiv preprint arXiv:1711.09325, 2017.

[49]. Cao T, Huang C-W, Hui DY-T, and Cohen JP, “A benchmark of medical out of distribution detection,” arXiv preprint arXiv:2007.04250, 2020.

[50]. Szegedy C et al. , “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.

[51]. Bastiani M et al. , “Automated processing pipeline for neonatal diffusion mri in the developing human connectome project,” NeuroImage, vol. 185, pp. 750–763, 2019. [PubMed: 29852283]

[52]. Heimann T et al. , “Comparison and evaluation of methods for liver segmentation from CT datasets,” IEEE transactions on medical imaging, vol. 28, no. 8, pp. 1251–1265, 2009. [PubMed: 19211338]

[53]. Kavur A, Selver M, Dicle O, Barıs M, and Gezer N, “Chaos-combined (ct-mr) healthy abdominal organ segmentation challenge data. accessed: 2019-04-11,” 2019.

[54]. Heller N et al. , “The kits19 challenge data: 300 kidney tumor cases with clinical context, ct semantic segmentations, and surgical outcomes,” arXiv preprint arXiv:1904.00445, 2019.

[55]. Bilic P et al. , “The liver tumor segmentation benchmark (lits),” arXiv preprint arXiv:1901.04056, 2019.

[56]. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, and Ronneberger O, “3d u-net: learning dense volumetric segmentation from sparse annotation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 424–432.

[57]. Huang G, Liu Z, Weinberger KQ, and van der Maaten L, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, vol. 1, 2017, p. 3.

[58]. Karimi D, Samei G, Kesch C, Nir G, and Salcudean SE, “Prostate segmentation in mri using a convolutional neural network architecture and training strategy based on statistical shape models,” International Journal of Computer Assisted Radiology and Surgery, vol. 13, no. 8, pp. 1211–1219, Aug 2018. [Online]. Available: 10.1007/s11548-018-1785-8 [PubMed: 29766373]

[59]. Kingma DP and Ba J, “Adam: A method for stochastic optimization,” in Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2014.

[60]. Golub G and Van-Loan CF, Matrix Computations. Baltimore, MD: Johns Hopkins University Press, 2013.

[61]. Fawcett T, “An introduction to roc analysis,” Pattern recognition letters, vol. 27, no. 8, pp. 861–874, 2006.

[62]. Murphy KP, “Machine learning: a probabilistic perspective,” 2012.

[63]. Bengio Y, Mesnil G, Dauphin Y, and Rifai S, “Better mixing via deep representations,” in International conference on machine learning, 2013, pp. 552–560.

[64]. Hendrycks D, Mazeika M, and Dietterich T, “Deep anomaly detection with outlier exposure,” arXiv preprint arXiv:1812.04606, 2018.

[65]. Naeini MP, Cooper G, and Hauskrecht M, “Obtaining well calibrated probabilities using bayesian binning,” in Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[66]. Panfilov E, Tiulpin A, Klein S, Nieminen MT, and Saarakkala S, “Improving robustness of deep learning based knee mri segmentation: Mixup and adversarial domain adaptation,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019, pp. 0–0.

Impact Statement—

Modern artificial intelligence methods have great potential for automatic medical image analysis, with applications in disease detection, assessment, and computer-aided intervention. However, their predictions are usually over-confident, even when the predictions are completely wrong. This presents a serious shortcoming of these methods for deployment in medical and clinical applications. In this paper, we address this problem for medical image segmentation, which is a central task in medical image analysis. We propose techniques that can reduce the over-confidence of these artificial intelligence methods on erroneous predictions. We also develop techniques that can detect when these methods fail. The techniques that we have developed in this paper can significantly improve the reliability of artificial intelligence methods for medical image analysis applications.

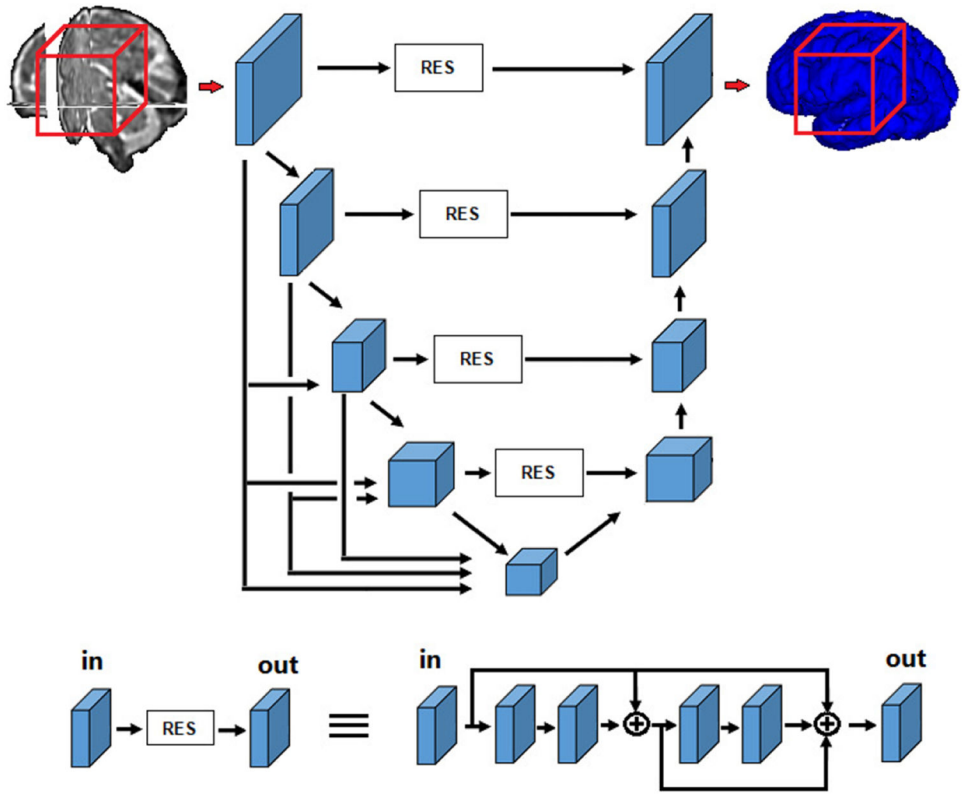

Fig. 1.
A schematic of our network architecture.

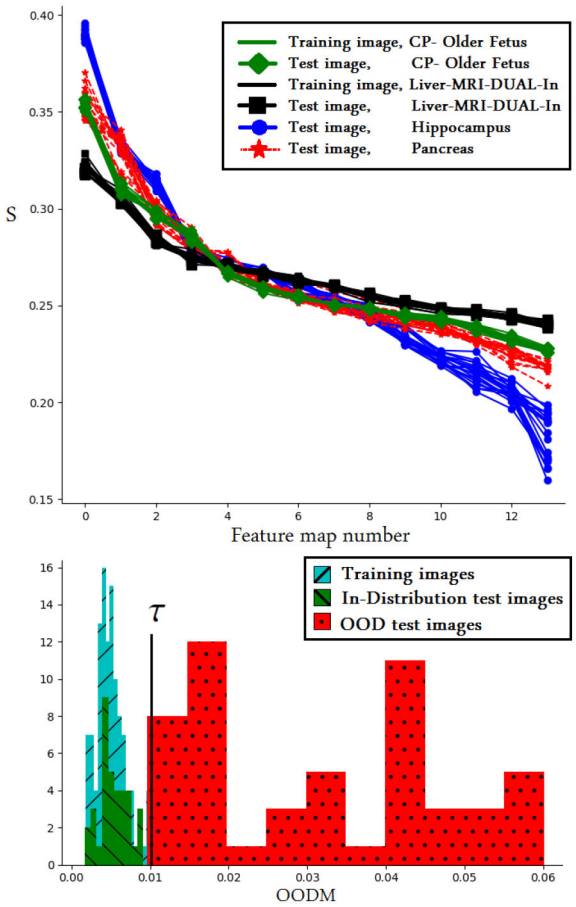

Fig. 2.

A demonstration of our proposed OOD detection method. Here, the model is trained on several datasets including CP-older fetus, Heart, and Liver-MRI-DUAL-In. TOP: Spectral signatures of four datasets, two of which (CP-older fetus and Liver-MRI-DUAL-In) are from the training data, while the other two (Hippocampus and Pancreas) are OOD. The spectra for the training and test samples for in-distribution data are very similar and not visually distinguishable. BOTTOM: Histograms of OODM, where OOD test images are from Pancreas, Hippocampus, and Spleen datasets. The value of the threshold $\tau = 0.011$ is marked with the vertical black line.

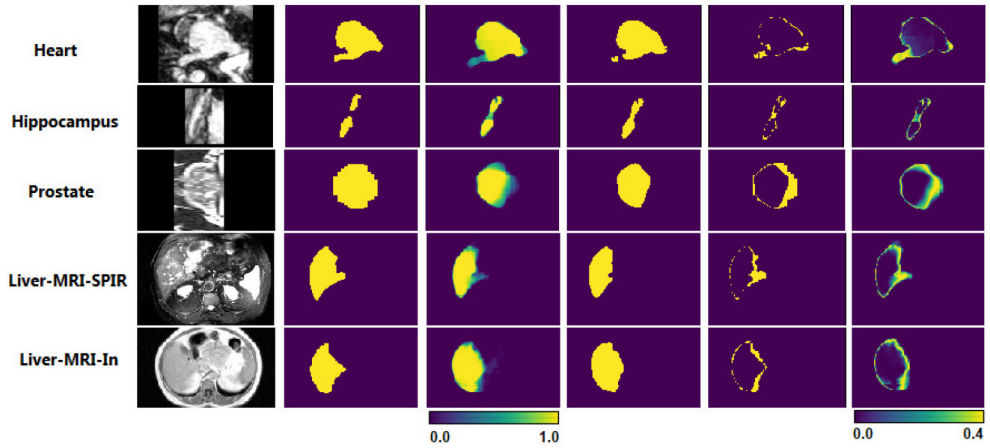

Fig. 3.

Examples prediction uncertainty maps produced by a model trained to segment a heterogeneous pool of datasets. From left, the first column shows a slice of the image. The second column is the ground-truth segmentation map. The third column is the model's predicted probability map (in the range [0,1]) that each voxel is a foreground voxel. The fourth column is the probability map thresholded at 0.5, showing the binary prediction of the model. The fifth column is the binary difference between the ground-truth (second column) and prediction (fourth column). In other words, the fifth column shows voxels where the model makes wrong predictions. The last column shows a voxel-wise prediction uncertainty map computed as $-p\log(p)$ where p is the predicted class probability for the voxel (in the range [0, −0.5 log(0.5)]). Note that all images in this figure are in-distribution data.

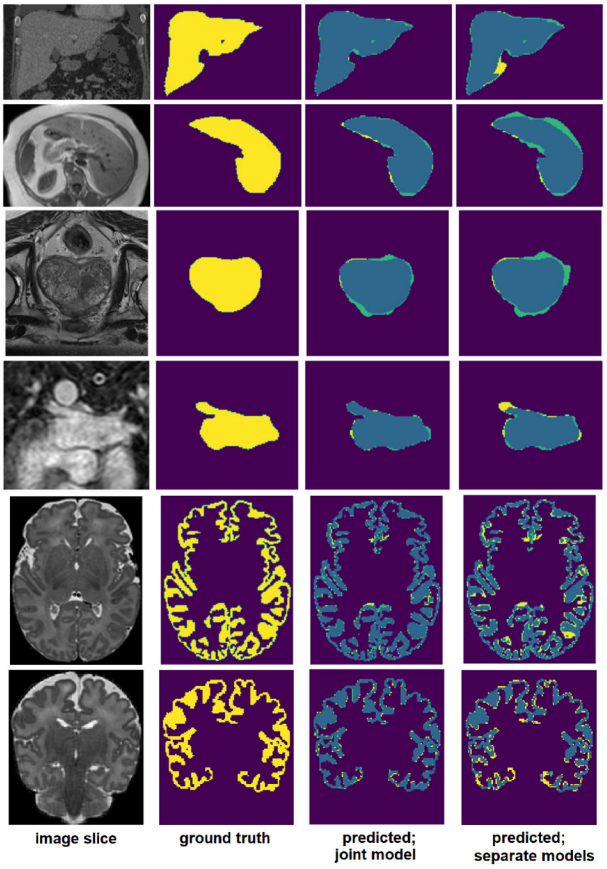

Fig. 4.

A slice from selected test images in the experiment reported in Table II and the output segmentation of the joint model that was trained on all datasets as well as individual models trained on each dataset. This is able to accurately segment different organs in different modalities. Moreover, it performs better than or comparable with dedicated models trained to segment each dataset separately. Note that all images used in this study are 3D; we have shown selected slices for visualization.

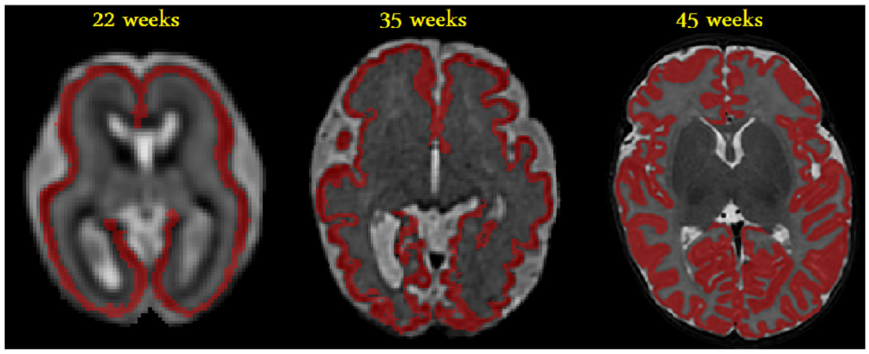

Fig. 5.

Example images and segmentations (in red) from the cortical plate datasets. From left, images come from CP-younger fetus, CP-older fetus, and CP-newborn. Postmenstrual age of each subject is displayed above the image.

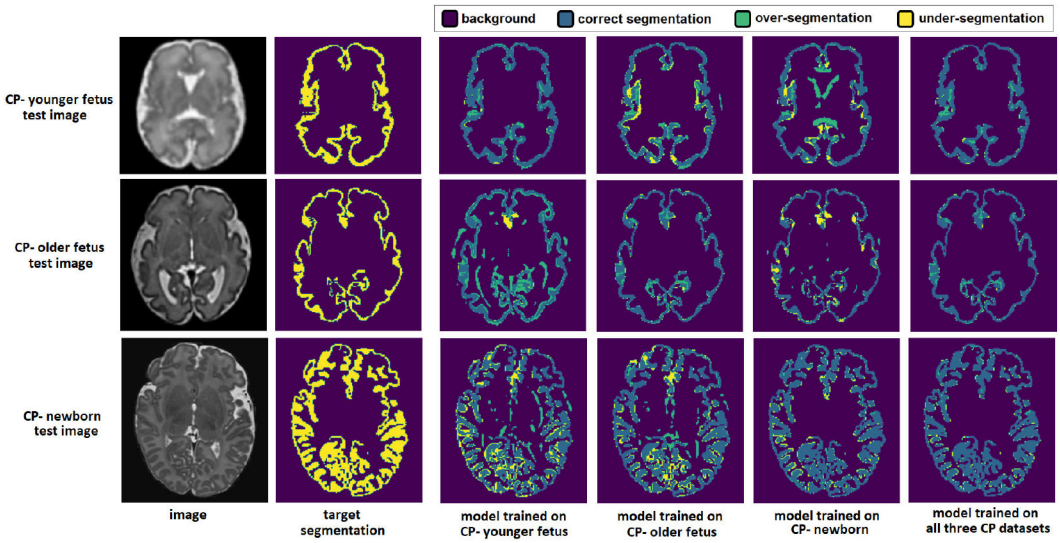

Fig. 6.

An example test image from each of the three brain cortical plate datasets, the corresponding ground truth segmentation, and segmentations produced by models trained on each of the three datasets separately and by a model trained on all three datasets. A model trained on each one of the datasets does not segment the other two datasets accurately. For example, a model trained in CP- younger fetus has large errors on CP- newborn test images. On the other hand, a model trained on all three training datasets accurately segments test images from all three datasets.

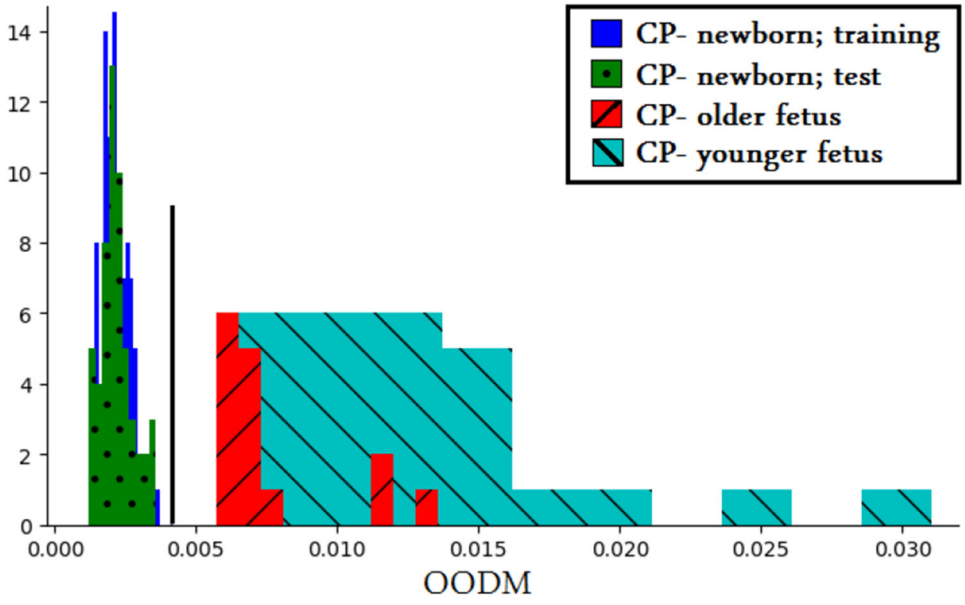

Fig. 7.

OODM Histograms (computed using Eq. (1)) for an experiment on cortical plate segmentation with the model trained on CP-newborn dataset. The threshold $\tau = 0.00358$ is marked with the vertical black line.

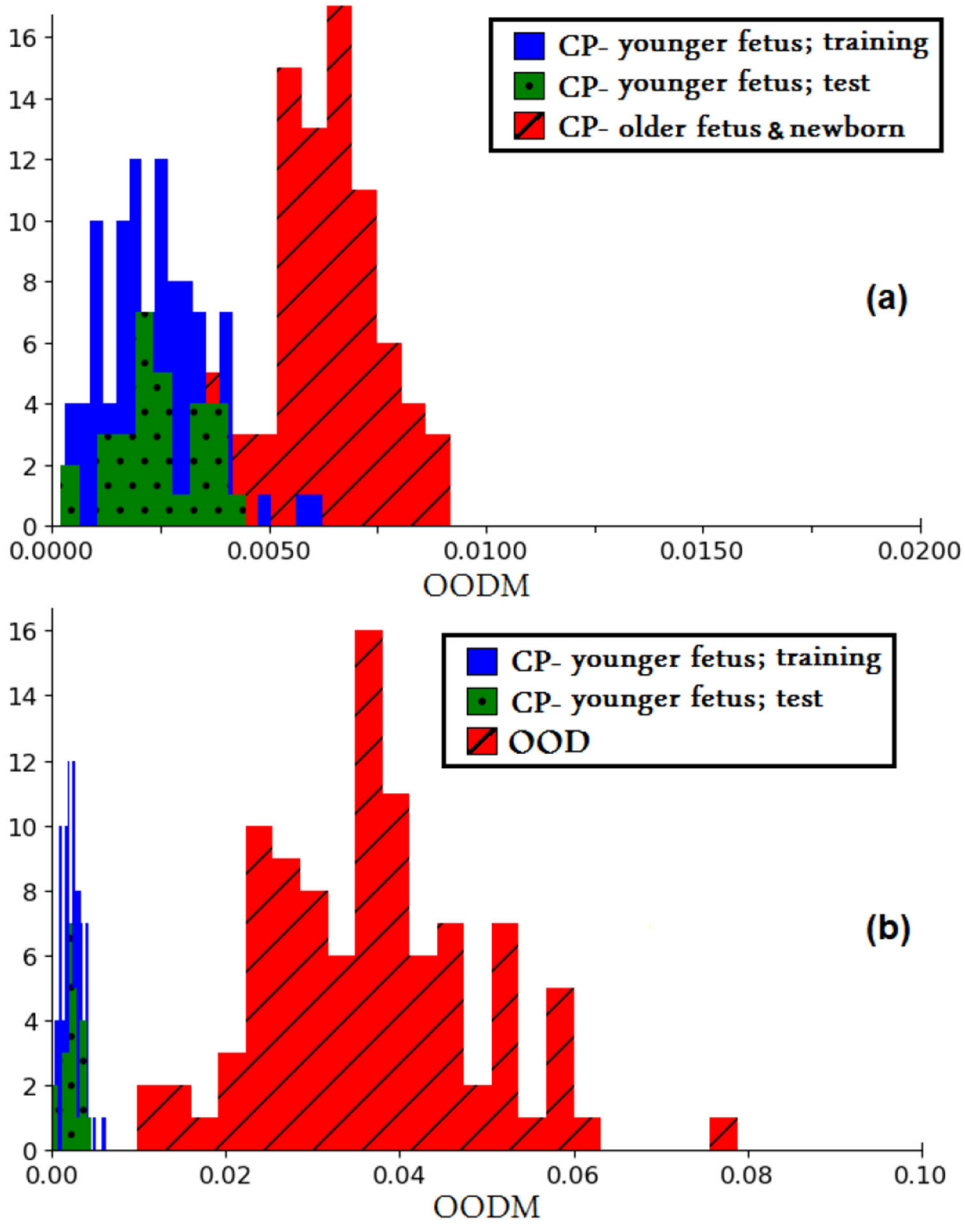

Fig. 8.

OODM Histograms for an experiment where the model was trained on CP-younger fetus. (a): OODM histograms for the two other CP datasets (CP- older fetus and CP-newborn). (b): OODM histograms for four other datasets (Heart, Liver-CT, Hippocampus, and Pancreas).

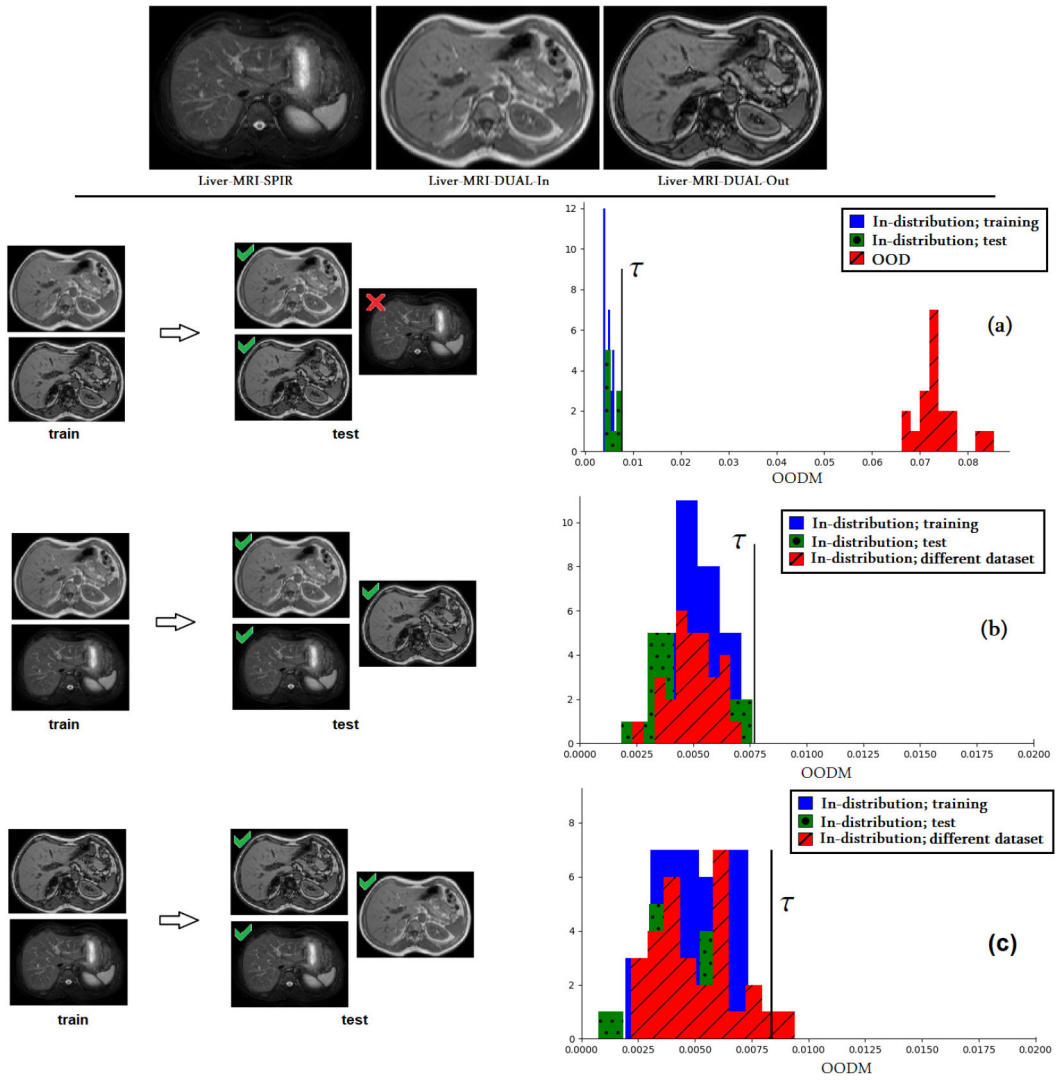

Fig. 9.

TOP: Sample images from the three liver MRI datasets. **BOTTOM:** The results of OOD detection experiment when different pairs of these three datasets are used for training. In the left section, green ✓ and red ✗ symbols, respectively, denote success and failure at test time. **(a)** The model was trained on Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out datasets. The OOD data included Liver-MRI-SPIR dataset, on which the model failed at test time. OODM perfectly separated the OOD data from in-distribution data. **(b)** Liver-MRI-SPIR and Liver-MRI-DUAL-In datasets were used for training. At test time the model accurately segmented Liver-MRI-DUAL-Out dataset (DSC= 0.886). OODM values for Liver-MRI-DUAL-Out are distributed similar to the training data. **(c)** Liver-MRI-SPIR and Liver-MRI-DUAL-Out were used for training. Note that the scales of the horizontal axes in (b) and (c) are different from (a).

TABLE I

Summary of the information on the datasets used in this study.

name	modality	organ	image resolution (mm)	data size	source
CP- younger fetus	T2 MRI	brain cortical plate	0.80	27	In-house (Boston Children's Hospital)
CP- older fetus	T2 MRI	brain cortical plate	0.80	15	In-house (Boston Children's Hospital)
CP- newborn	T2 MRI	brain cortical plate	0.80	400	[51]
Liver-CT	CT	liver	0.80 0.80 1.0	19	[52]
Liver-MRI-SPIR	MRI	liver	1.5 1.5 8	20	[53]
Liver-MRI-DUAL-in	MRI	liver	1.5 1.5 8	20	[53]
Liver-MRI-DUAL-out	MRI	liver	1.5 1.5 8	20	[53]
KiTS	CT	kidney	1.0, 0.8, 0.8	300	[54]
LiTS	CT	liver	80 0.80 1.0	130	[55]
Heart	MRI	left atrium	1.25 1.25 1.37	20	https://decathlon-10.grand-challenge.org/
Prostate	MRI	prostate	0.6 0.6 3.6	32	https://decathlon-10.grand-challenge.org/
Pancreas	CT	pancreas	0.8 0.8 2.5	281	https://decathlon-10.grand-challenge.org/
Hippocampus	MRI	hippocampus	1.0	260	https://decathlon-10.grand-challenge.org/
Spleen	CT	spleen	0.8 0.8 5.0	41	https://decathlon-10.grand-challenge.org/

TABLE II

Results of an experiment to compare multi-task learning with training separate models for each dataset. For each of the twelve datasets, the statistically significant differences between standard training and multi-task training have been marked using bold type.

Training method	Data	DSC	HD95 (mm)	ASSD (mm)	ECE	MCE
Training a separate model for each dataset	CP- younger fetus	0.89 ± 0.05	0.82 ± 0.04	0.24 ± 0.06	0.09 ± 0.02	0.21 ± 0.06
	CP- older fetus	0.82 ± 0.06	1.01 ± 0.14	0.35 ± 0.08	0.12 ± 0.07	0.28 ± 0.11
	CP- newborn	0.90 ± 0.04	0.83 ± 0.04	0.22 ± 0.07	0.07 ± 0.03	0.20 ± 0.10
	Heart	0.89 ± 0.06	8.0 ± 7.2	2.00 ± 2.20	0.17 ± 0.05	0.32 ± 0.09
	Hippocampus	**0.88 ± 0.04**	**1.04 ± 0.25**	**0.42 ± 0.06**	**0.11 ± 0.03**	**0.20 ± 0.05**
	Prostate	0.85 ± 0.06	12.7 ± 8.5	3.6 ± 4.4	0.26 ± 0.07	0.37 ± 0.10
	Liver-CT	0.96 ± 0.02	5.00 ± 1.99	1.46 ± 0.36	0.10 ± 0.02	0.24 ± 0.05
	Liver-MRI-SPIR	0.90 ± 0.03	25.1 ± 11.0	4.70 ± 1.51	0.18 ± 0.03	0.32 ± 0.05
	Liver-MRI-DUAL-in	0.89 ± 0.05	11.0 ± 3.11	4.02 ± 0.70	0.11 ± 0.03	0.18 ± 0.03
	Liver-MRI-DUAL-out	0.89 ± 0.04	10.1 ± 2.87	4.00 ± 0.54	0.10 ± 0.02	0.14 ± 0.03
	Pancreas	0.80 ± 0.03	9.0 ± 2.81	3.45 ± 0.92	0.08 ± 0.02	0.14 ± 0.06
	Kidney	0.91 ± 0.03	8.1 ± 3.01	0.81 ± 0.34	0.09 ± 0.03	0.16 ± 0.05
Training a single model for all datasets	CP- younger fetus	0.89 ± 0.04	0.82 ± 0.01	0.23 ± 0.06	**0.07 ± 0.01**	**0.18 ± 0.04**
	CP- older fetus	**0.84 ± 0.02**	**0.88 ± 0.13**	0.34 ± 0.06	**0.09 ± 0.02**	**0.21 ± 0.06**
	CP- newborn	0.90 ± 0.04	0.82 ± 0.03	**0.17 ± 0.05**	0.07 ± 0.03	**0.15 ± 0.09**
	Heart	0.88 ± 0.08	8.2 ± 7.5	2.14 ± 2.16	**0.11 ± 0.03**	**0.21 ± 0.07**
	Hippocampus	0.87 ± 0.03	1.25 ± 0.20	0.51 ± 0.06	0.14 ± 0.02	0.26 ± 0.05
	Prostate	**0.89 ± 0.05**	**5.12 ± 2.55**	**1.88 ± 0.50**	**0.21 ± 0.05**	**0.30 ± 0.10**
	Liver-CT	0.96 ± 0.02	**4.18 ± 1.07**	**1.30 ± 0.26**	**0.07 ± 0.02**	**0.16 ± 0.05**
	Liver-MRI-SPIR	**0.93 ± 0.03**	**10.5 ± 4.08**	**3.59 ± 1.60**	0.18 ± 0.05	0.33 ± 0.06
	Liver-MRI-DUAL-in	0.89 ± 0.04	**7.0 ± 3.01**	3.93 ± 0.50	**0.08 ± 0.04**	**0.14 ± 0.04**
	Liver-MRI-DUAL-out	**0.92 ± 0.03**	**7.3 ± 2.66**	3.95 ± 0.44	**0.08 ± 0.02**	**0.09 ± 0.03**
	Pancreas	**0.82 ± 0.03**	**7.3 ± 2.50**	3.48 ± 0.88	**0.06 ± 0.03**	**0.10 ± 0.06**
	KiTS	0.91 ± 0.03	8.1 ± 2.95	0.80 ± 0.24	**0.06 ± 0.02**	**0.11 ± 0.03**

TABLE III

Results of experiments on cortical plate segmentation. We compare three different transfer learning approaches with our proposed method of multi-task learning with heterogeneous data. We ran paired t-tests between the four results (three transfer learning trials and multi-task learning trial), separately for each of the three datasets. Statistically better results ($p = 0.01$), were marked with bold type.

Training/fine-tuning data	Test data	DSC	HD95 (mm)	ASSD (mm)	ECE	MCE
Train on CP- younger fetus	CP- younger fetus	0.90 ± 0.03	0.82 ± 0.03	0.21 ± 0.03	0.07 ± 0.02	0.20 ± 0.04
↳ Fine-tune on CP- older fetus	CP- older fetus	0.80 ± 0.05	1.05 ± 0.22	0.40 ± 0.12	0.13 ± 0.06	0.31 ± 0.07
↳ Fine-tune on CP- newborn	CP- newborn	0.92 ± 0.07	0.80 ± 0.03	0.19 ± 0.02	0.07 ± 0.02	0.19 ± 0.02
Train on CP- older fetus	CP- older fetus	0.82 ± 0.04	1.10 ± 0.21	0.37 ± 0.08	0.10 ± 0.04	0.22 ± 0.10
↳ Fine-tune on CP- younger fetus	CP- younger fetus	0.90 ± 0.04	0.88 ± 0.04	0.22 ± 0.04	0.10 ± 0.03	0.17 ± 0.06
↳ Fine-tune on CP- newborn	CP- newborn	0.92 ± 0.03	0.89 ± 0.04	0.18 ± 0.02	0.07 ± 0.01	0.17 ± 0.03
Train on CP- newborn	CP- newborn	0.92 ± 0.03	0.84 ± 0.01	0.18 ± 0.03	0.06 ± 0.03	0.12 ± 0.04
↳ Fine-tune on CP- younger fetus	CP- younger fetus	0.90 ± 0.03	0.85 ± 0.03	0.22 ± 0.04	0.08 ± 0.02	0.19 ± 0.03
↳ Fine-tune on CP- older fetus	CP- older fetus	0.81 ± 0.05	0.96 ± 0.18	0.34 ± 0.14	0.16 ± 0.05	0.34 ± 0.05
Train a single model for all datasets	CP- younger fetus	0.90 ± 0.02	0.81 ± 0.01	**0.18±0.03**	**0.05±0.03**	**0.13±0.08**
	CP- older fetus	**0.85±0.03**	**0.90±0.16**	**0.30±0.05**	**0.04±0.02**	**0.09±0.05**
	CP- newborn	0.92 ± 0.02	0.81 ± 0.02	**0.16±0.02**	**0.03±0.01**	**0.07 ±0.02**

IEEE Trans Artif Intell. Author manuscript; available in PMC 2024 April 01.

TABLE IV

Comparison of multi-task learning with mixup on cortical plate segmentation. Bold type indicates statistically better results at $p = 0.01$.

Training/fine-tuning data	Test data	DSC	HD95 (mm)	ASSD (mm)	ECE	MCE
Multitask learning	CP- younger fetus	0.90 ± 0.02	0.81 ± 0.01	0.18 ± 0.03	**0.05±0.03**	**0.13±0.08**
	CP- older fetus	0.85 ± 0.03	0.90 ± 0.16	0.30 ± 0.05	**0.04±0.02**	**0.09±0.05**
	CP- newborn	0.92 ± 0.02	0.81 ± 0.02	0.16 ± 0.02	**0.03±0.01**	**0.07±0.02**
mixup	CP- younger fetus	0.89 ± 0.02	0.81 ± 0.01	0.17 ± 0.03	0.08 ± 0.03	0.18 ± 0.09
	CP- older fetus	0.84 ± 0.03	0.95 ± 0.25	0.34 ± 0.07	0.08 ± 0.04	0.14 ± 0.06
	CP- newborn	0.92 ± 0.02	0.84 ± 0.06	0.20 ± 0.04	0.06 ± 0.02	0.12 ± 0.03

TABLE V

OOD detection accuracy in an experiment where in-distribution data came from CP-younger fetus, CP-older fetus, Prostate, Heart, Liver-CT, Liver-MRI-SPIR, Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out datasets, and OOD data came from Pancreas, Hippocampus, and Spleen datasets.

Method	accuracy	sensitivity	specificity	AUC
Proposed method	1.00	0.98	1.00	0.98
UNC-Dropout	0.55	0.48	0.63	0.62
DkNN	0.76	0.67	0.82	0.79
UNC-Ensemble	0.84	0.87	0.71	0.82
Outlier exposure	0.83	0.80	0.86	0.80
Lee-2017	0.76	0.69	0.78	0.77
ODIN	0.70	0.61	0.68	0.67
Mah-Dist	0.74	0.66	0.80	0.77

TABLE VI

OOD detection accuracy in an experiment on CP segmentation. The model was trained on CP-newborn. CP-younger fetus and CP-older fetus are used as OOD.

Method	accuracy	sensitivity	specificity	AUC
Proposed method	1.00	1.00	1.00	1.00
UNC-Dropout	0.57	0.54	0.68	0.67
UNC-Ensemble	0.80	0.77	0.81	0.79
Outlier exposure	0.82	0.77	0.81	0.80
Lee-2017	0.70	0.68	0.71	0.73
ODIN	0.67	0.68	0.68	0.70
Mah-Dist	0.80	0.72	0.78	0.77

TABLE VII

OOD detection accuracy in an experiment where the model was trained on CP-younger fetus. As OOD, we used images two dataset groups: CP datasets (CP-older fetus and CP-newborn) and non-CP datasets (Heart, Liver-CT, Hippocampus, and Pancreas).

test data	Method	accuracy	sensitivity	specificity	AUC
CP-older fetus and CP-newborn	Proposed method	0.90	0.91	0.87	0.90
	UNC-Dropout	0.60	0.61	0.60	0.67
	UNC-Ensemble	0.84	0.82	0.74	0.81
	Outlier exposure	0.80	0.86	0.80	0.81
	Lee-2017	0.75	0.78	0.71	0.78
	ODIN	0.64	0.65	0.68	0.65
	Mah-Dist	0.67	0.70	0.60	0.64
Heart, Liver-CT, Hippocampus, and Pancreas	Proposed method	1.00	1.00	1.00	1.00
	UNC-Dropout	0.64	0.60	0.78	0.65
	UNC-Ensemble	0.80	0.88	0.71	0.76
	Outlier exposure	0.83	0.83	0.85	0.85
	Lee-2017	0.76	0.70	0.80	0.77
	ODIN	0.70	0.71	0.70	0.73
	Mah-Dist	0.78	0.77	0.78	0.75

TABLE VIII

OOD detection accuracy in an experiment where the model was trained on Liver-MRI-DUAL-In and Liver-MRI-DUAL-Out datasets. Images from Liver-MRI-SPIR dataset are OOD.

Method	accuracy	sensitivity	specificity	AUC
Proposed method	1.00	1.00	1.00	1.00
UNC-Dropout	0.64	0.61	0.60	0.65
UNC-Ensemble	0.82	0.81	0.84	0.84
Outlier exposure	0.80	0.84	0.75	0.82
Lee-2017	0.77	0.78	0.74	0.78
ODIN	0.60	0.55	0.62	0.59
Mah-Dist	0.66	0.74	0.60	0.70
DkNN	0.59	0.59	0.56	0.57