



HHS Public Access

Author manuscript

Med Image Anal. Author manuscript; available in PMC 2024 April 01.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Published in final edited form as:

Med Image Anal. 2023 April ; 85: 102731. doi:10.1016/j.media.2022.102731.

Learning to segment fetal brain tissue from noisy annotations

Davood Karimi^{a,*}, Caitlin K. Rollins^b, Clemente Velasco-Annis^a, Abdelhakim Ouaalam^a, Ali Gholipour^a

^aDepartment of Radiology, Boston Children's Hospital, Harvard Medical School, Boston, MA, USA

^bDepartment of Neurology, Boston Children's Hospital, Harvard Medical School, Boston, MA,

USA *this paper is about automatically segment different brain tissue even in
noisy image in dataset*

Abstract

Automatic fetal brain tissue segmentation can enhance the quantitative assessment of brain development at this critical stage. Deep learning methods represent the state of the art in medical image segmentation and have also achieved impressive results in **brain segmentation**. However, effective training of a deep learning model to perform this task requires a large number of training images to represent the rapid development of the transient fetal brain structures. On the other hand, manual multi-label segmentation of a large number of 3D images is prohibitive. To address this challenge, we segmented 272 training images, covering 19–39 gestational weeks, using an automatic multi-atlas segmentation strategy based on **deformable registration** and **probabilistic atlas fusion**, and manually corrected large errors in those segmentations. Since this process generated a large training dataset with noisy segmentations, we developed a novel **label smoothing procedure** and **a loss function** to train a deep learning model with **smoothed noisy segmentations**. Our proposed methods properly account for the uncertainty in tissue boundaries. We evaluated our method on 23 manually-segmented test images of a separate set of fetuses. Results show that our method achieves an average Dice similarity coefficient of 0.893 and 0.916 for the transient structures of younger and older fetuses, respectively. Our method generated results that were significantly more accurate than several state-of-the-art methods including nnU-Net that achieved the closest results to our method. Our trained model can serve as a valuable tool to enhance the accuracy and reproducibility of fetal brain analysis in MRI.

Keywords

fetal brain; tissue segmentation; noisy labels; deep learning

*Corresponding author: Tel.: +1-617-208-9736; fax: +1-617-730-0635.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

The authors do not have any conflicts of interest (financial or otherwise) to disclose.

The Dice Similarity Coefficient (DSC) — also called the Sørensen-Dice coefficient — is a statistical measure used to calculate how similar two sets are. It is widely used in image segmentation, clustering, and natural language processing to evaluate overlap between predicted results and ground truth.

Definition

For two sets A and B, the Dice coefficient is:

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

Intuition

- Measures overlap between two samples.
- 0 means no overlap.
- 1 means perfect overlap.

In Image Segmentation

A common application is comparing how well an algorithm segments an organ or object compared to manual (ground truth) segmentation.

If X = predicted pixels
and Y = ground-truth pixels:

$$DSC = \frac{2TP}{2TP + FP + FN}$$

Where

- TP = True Positives
- FP = False Positives
- FN = False Negatives

$$DSC = \frac{2 \cdot IoU}{1 + IoU}$$

- To do quantitative analysis (volumes, shapes, growth curves), you need to segment the brain into many tissue types: cortical plate, subplate, ventricles, white matter, deep nuclei, cerebellum, etc.
- Doing this manually in 3D for many fetuses is:
 - Extremely time-consuming (days per scan)
 - Hard, because fetal MRI has motion artifacts, partial volume effects, and rapidly changing anatomy with gestational age.

Main difficulty

- Deep learning needs lots of labeled images, but:
 - Getting perfect 3D multi-label segmentations is basically impossible at large scale.
 - Even "good" labels still have uncertain boundaries (especially for thin or poorly contrasted structures).
- So they accept that their labels will be noisy and uncertain near tissue boundaries, and try to explicitly model and use that uncertainty during training instead of ignoring it.

Core idea

1. Use an automatic atlas-based method to segment a large training set, then manually correct only big mistakes → results are noisy but cheap to obtain.
2. Ask experts to specify, for each tissue type, how uncertain its boundary is (in voxels).
3. Use this information to:
 - Create spatially varying "soft" labels near boundaries (label smoothing).
 - Design a loss function that treats certain and uncertain voxels differently, using a learned label transition matrix that describes how labels tend to get confused.
4. Train a 3D U-Net (via nnU-Net framework) on 272 noisy scans using this strategy.
5. Evaluate on 22 fetuses with carefully manually segmented ground truth and compare to many alternative methods.

They show that this approach beats nnU-Net and other strong baselines, and even outperforms a classical multi-atlas segmentation method. nihms-1861994

Because of brain development:

- Younger fetuses (<32 weeks): SP and IZ are separate labels.
- Older fetuses (≥32 weeks): SP + IZ are merged into WM.
- So they train two separate models:
 - Young: 33 tissue labels
 - Old: 31 tissue labels (plus background in both cases)

these are regions
in brain

How the test labels are created (high-quality "clean" labels)

For the 22 test scans:

1. Use a spatiotemporal fetal brain atlas (covering 19–39 weeks) as prior segmentations.
2. For each test fetus:
 - Register atlas images within ±1 week of that fetus's GA using diffeomorphic deformable registration.
 - Propagate the atlas labels to the subject.
3. Combine multiple warped labels using probabilistic STAPLE (a label-fusion method). nihms-1861994
4. Then expert annotators manually refine all labels thoroughly, in several rounds, until they are "clean".

This gives them very accurate ground truth for evaluation, but takes 4–10 days per scan.

How the training labels are created (noisy labels)

For the 272 training scans, they do a similar pipeline:

1. Same multi-atlas segmentation (registration + STAPLE).
2. Annotators correct only major errors, not fine details → ~2 hours per scan.

So the training labels:

- Are pretty good overall.
- But still have boundary noise, especially for structures that are hard to see or thin.

Crucially, annotators also provide for each tissue:

- A single number: boundary uncertainty, measured in voxels (0–4), e.g.:
 - Lateral ventricle: uncertainty = 0 voxels (sharp boundary).
 - Caudate nuclei: uncertainty = 2 voxels (blurry / uncertain edge).

This is the key input to their label smoothing method. nihms-1861994

1. What are atlas-based methods?

Atlas-based segmentation is a classical medical image segmentation technique used *before* deep learning became dominant.

What is an atlas?

An atlas is:

- A brain MRI image (or set of images)
- With a **manually created, high-quality segmentation**
- That represents a "typical" brain anatomy (or multiple anatomies at different ages)

In this paper, they use a **spatiotemporal fetal brain atlas** (19–39 gestational weeks), meaning many atlas brains at different ages. 

How atlas-based segmentation works

To segment a new fetal MRI:

1. **Pick several atlases close in gestational age.**
2. **Register (warp) each atlas image to the new fetal MRI.**
 - They use **diffeomorphic deformable registration** (non-rigid warping).
 - This aligns the atlas brain with the new fetal brain.
3. **Warp the atlas labels along with the images, producing several candidate segmentations.**
4. **Fuse the candidate labels into a final segmentation.**
 - They use **probabilistic STAPLE**, a label-fusion algorithm that **estimates the "most probable true segmentation"** by combining many atlas segmentations. 

Key idea

You "borrow" the labels from expertly annotated atlas brains by deforming them to match your target brain.

Limitations

Atlas-based segmentation works *reasonably well*, but:

- Registration is imperfect, especially for **complex or thin structures** (cortex, subplate, IZ).
- Atlases represent an *average morphology* → they **cannot model high inter-subject variability**.
- Fetal MRI has **motion artifacts** and rapidly changing anatomy, making registration harder.

3. What is the label transition matrix? Why is it needed?

This is the part you found confusing — let's make it very clear.

What problem does it solve?

Even after smoothing, labels near boundaries are still **noisy**:

- A voxel that *should* be cortex might be labeled subplate.
- A voxel that *should* be WM might be mislabeled as IZ.

These errors are **not random** — e.g., cortex is often confused with subplate, but almost never with ventricles.

The authors want the model to understand:

"**Some mistakes in the labels are expected and should not be penalized harshly.**"

What is a label transition matrix?

It is a matrix T of size $L \times L$ describing how labels get corrupted.

- **Rows (i) = observed noisy labels**
- **Columns (j) = true clean labels**

$$T_{ij} = \text{Prob}(\text{noisy label} = i \mid \text{true label} = j)$$

If cortex is often mislabeled as subplate,

$T[\text{subplate}, \text{cortex}]$ will be **high**.

If ventricles are almost never mislabeled as hippocampus,

$T[\text{hippocampus}, \text{ventricles}]$ will be **near zero**.

2. What are "soft labels near the boundaries"? (Label smoothing)

What is the problem?

Manual fetal brain labels — especially in noisy or low-resolution MRI — have **uncertain boundaries**.

Soft labels

A **soft label** assigns a **probability distribution** instead of a single hard class:

Example soft label for one voxel near an uncertain boundary:

Tissue	Probability
Cortex	0.6
Subplate	0.35
WM	0.05

This expresses: "We're 60% sure it's cortex, but it could be subplate."

What the authors do (spatially varying label smoothing)

They create soft labels **only near boundaries** and only according to **annotator-defined uncertainty**.

Annotators provide a number for each tissue (0–4 voxels) describing **how uncertain the boundary is**.

Example from the paper:

- Lateral ventricles: uncertainty = 0 voxels
- Caudate: uncertainty = 2 voxels 

Their smoothing algorithm:

1. For **each voxel**, check if it is **near a boundary**.
2. Determine the **boundary smoothing radius** using the **minimum uncertainty among the touching tissues**.
 - If one tissue is **extremely certain** (e.g., LV = 0), **no smoothing happens**.
3. Apply a **distance-weighted averaging** over the labels of nearby voxels.
 - **Closer voxels influence more.**
 - **Bigger uncertainty → bigger smoothing radius.**

The result is the smoothed probability Y .

Why do this?

Because forcing the model to **learn from hard, incorrect labels near uncertain boundaries** makes learning **unstable**.

Soft labels let the model learn "**the truth is ambiguous here**".

How is T used in the loss function?

The loss consists of two parts:

1. Certain voxels (unsmoothed, interior voxels):

- Use standard cross-entropy:
 $L(\hat{Y}, Y_s)$

2. Uncertain voxels (boundary voxels):

- Use a corrected loss:
 $L((T^T + \lambda I)^{-1} \hat{Y}, Y_s)$

The term $(T^T + \lambda I)^{-1}$ "de-noises" the prediction by accounting for the expected confusions.

This is called **backward loss correction**, originally from Patrini et al. (2017), and ensures training behaves as if the labels were clean. nihms-1861994

Intuition

If the model predicts:

- true tissue = cortex
- but observed noisy label = subplate

A standard loss would heavily penalize the model.

But with T:

- If $T[\text{subplate, cortex}]$ is high (common confusion)
- The correction ensures the model is not punished too much, because this mismatch is expected.

Thus the model focuses on learning real anatomy, not on fitting noisy boundaries.

Soft labels near boundaries (label smoothing)

- Replace hard labels with probability distributions only where boundaries are uncertain.
- Smoothing radius is determined by tissue-specific uncertainty.
- Result: The network learns that boundaries are ambiguous, preventing overfitting to noise. \leadsto * Important Point

Label transition matrix (T)

- Models how true labels are systematically corrupted into noisy ones.
- Used in the loss to correct predictions in uncertain regions.
- Prevents the model from being penalized for reasonable confusions.

Together, these two innovations (smoothing + T-corrected loss) enable deep learning to train on many noisy labels and still outperform all baselines.

↳ the main ideas of paper:

- 1) novel loss function for not penalizing heavily the confusion in labeling of some areas
- 2) soft labeling in boundaries base on tissue-specific uncertainty

3. In one sentence

The paper proposes a way to train a fetal brain segmentation network from many cheap, noisy labels by (1) turning hard labels into uncertainty-aware soft labels near tissue boundaries, and (2) using a noise-corrected loss based on a learned label transition matrix, and shows this beats state-of-the-art methods and classical atlas-based segmentation across >30 tissue classes and a wide gestational age range.

nihms-1861994

Multi-atlas segmentation is just the "many-atlases" version of atlas-based segmentation, used to make things more robust and accurate.

Let's build it up slowly.

1. Quick recap: what is an atlas?

An **atlas** = one MRI image + its expert segmentation.

For fetal brain here, they use a **4D atlas**: brains at different gestational ages, each with labeled tissues. nihms-1861994

Atlas-based segmentation of a new subject:

1. Register (warp) the atlas image to the subject image.
2. Apply the same warp to the atlas labels.
3. The warped labels become the predicted segmentation.

2. What makes it *multi-atlas*?

Instead of using one atlas, you use several:

1. Choose multiple atlases that are similar to the target (e.g., close gestational age).
2. Register each atlas to the target image separately (diffeomorphic registration here). nihms-1861994
3. Warp each atlas's labels → you now have multiple candidate segmentations of the same brain.
4. Fuse / combine those candidate labels voxel-wise using some label fusion method.

So for each voxel, you don't trust a single atlas; you aggregate the votes from many atlases.

In this paper the fusion step is done with **probabilistic STAPLE**, which estimates the "most likely true label" from all atlas votes and also models which atlases are more reliable.

this prob Point ↳

- **Uncertainty estimates:** fusion methods like STAPLE can output probabilities instead of only hard labels.

For each fetal scan:

- Register several atlas images (within ± 1 week gestational age) to the fetus.
- Warp atlas labels.
- Fuse them with probabilistic STAPLE → get an automatic segmentation.

□ nihms-1861994

5. One-sentence summary

Multi-atlas segmentation = use many annotated example brains (atlases), warp each one to a new scan, and then fuse all their warped labels (e.g., with STAPLE) so that the final segmentation benefits from a consensus instead of relying on a single atlas.

2. As a baseline method

- They run **multi-atlas segmentation alone** (no deep learning) on the 22 test images and compare it to their proposed network.
- Multi-atlas does *okay*, but is significantly worse than their deep model, especially on complex structures like cortical plate, subplate, IZ, VZ, and WM.

□ nihms-1861994

They also explain why multi-atlas struggles here:

- Registration errors are large for thin, convoluted structures.
- Atlas shape can't fully represent **inter-subject variability**, especially in rapidly developing fetal brains.

□ nihms-1861994

Why is atlas-based segmentation limited?

The paper clearly notes two weaknesses:

- Registration is **imperfect**, especially for thin or complex structures (cortical plate, ventricular zone, etc.).
- Atlases represent an "**average shape**", so they cannot fully represent the large variability in fetal brain shapes.

How the paper does label smoothing

Their smoothing uses **expert-estimated boundary uncertainty** for each tissue (e.g., caudate nucleus = 2 voxels uncertainty). □ nihms-1861994

Algorithm summary (Section 2.2.1):

1. For each voxel, check a neighborhood of radius 4 (max uncertainty given by annotators).
2. If all voxels in that region have the same label → no smoothing (certain).
3. Otherwise:
 - Compute **boundary uncertainty r_u = minimum uncertainty of tissues in neighborhood.**
 - Smooth labels within a radius r_u using distance-weighted averaging.

This turns crisp boundaries into **gradual probability transitions**.

See Figure 1 on page 18, where the bottom row shows smooth probability heatmaps. □ nihms-1861994

- Far from any boundary → keep the original hard label.
- Near a boundary between tissues → mix information from nearby voxels to get probabilities.

For each voxel i :

1. Look at a **big neighborhood** around it, radius $R = 4$ voxels: call this set of neighbors i_R .
2. If all voxels in i_R have the same label → we consider the label at i to be **certain**:
 - No smoothing: $Y_S(i) = Y(i)$ (still a one-hot vector).

If not all labels are the same → voxel i is close to a boundary → go to the "otherwise" part.

first with radius 4 and 2nd with 28 uncertain
we would use r_u for computing soft label

how compute r_u <

we have got uncertainty for each region
and there are multiple labels around voxel
that it comes to

otherwise → so we chose minimum value of voxel uncertainty
so actually the most certain tissue win and specify the r_u

2. The "otherwise" part in plain words

We're now in a region where different tissues meet. The algorithm does:

1. Compute how uncertain this boundary is → a radius r_u .
2. Look only within a smaller radius r_u around voxel i .
3. Compute a weighted average of labels in that smaller neighborhood.
 - Closer neighbors influence more (via an exponential weight).
 - This average becomes the **soft label** at i .

Let's unpack each step.

2.1 Compute boundary uncertainty r_u

We have:

- $U(k)$ = **boundary uncertainty value** (in voxels) at voxel k
(This is derived from the tissue type at k : e.g., caudate = 2 voxels, LV = 0 voxels, etc.) □ nihms-1861994
- In the big neighborhood i_R , different labels → possibly different uncertainties.

The algorithm sets: ↳ initial neighborhood

voxel that we want
to calculate its
vector

$$r_u = \min_{k \in i_R} U(k)$$

So r_u is the minimum uncertainty value among all tissue types seen in that neighborhood.

2.2 Define the smaller patch i_{r_u}

Now we look at a smaller patch around i :

i is the voxel that we're making label vec $i_{r_u} = \{k \mid \|k - i\| \leq r_u\}$

This is all voxels whose distance to i is $\leq r_u$.

- If boundary is very uncertain (e.g., $r_u = 3$), we allow a bigger patch (more smoothing).
- If boundary is pretty clear (e.g., $r_u = 1$), we only look at very close neighbors.

2.3 Build the weight matrix W

For each voxel k in the small patch i_{r_u} , we assign a weight:

$$W_k \propto \exp\left(-\frac{\|k - i\|}{\tau}\right)$$

with *here, it only add weights to the surrounding voxels base on the thier distance* $\tau = r_u$

So:

- Closer voxels (small distance $\|k - i\|$) get larger weights.
- Further voxels get exponentially smaller weights.
- Larger $r_u \Rightarrow$ larger $\tau \Rightarrow$ slower decay \Rightarrow smoother average.

Then we normalize all weights so they sum to 1:

$$\sum_{k \in i_{r_u}} W_k = 1$$

So the weights form a proper local probability distribution over neighbors.

Now a case where both tissues are uncertain:

- Tissue C: uncertainty = 3
- Tissue D: uncertainty = 2

Around voxel i , you see both C and D.

Step 1 – r_u

$$r_u = \min(3, 2) = 2$$

So the boundary is somewhat uncertain, radius 2.

Step 2 – Small patch

Now i_{r_u} contains, say, 5 voxels: $i-2, i-1, i, i+1, i+2$.

Imagine their labels are:

```
text ⌂ Copy code
i-2: C
i-1: C
i : C
i+1: D
i+2: D
```

Step 3 – Weights

The closer to i , the larger weight:

- i : biggest weight
- $i \pm 1$: medium weights
- $i \pm 2$: smaller weights

After normalization, example weights (just to illustrate):

```
text ⌂ Copy code
W(i-2) = 0.10
W(i-1) = 0.20
W(i) = 0.40
W(i+1) = 0.20
W(i+2) = 0.10
```

Step 4 – Smoothed probabilities

- Total weight for C: $0.10 + 0.20 + 0.40 = 0.70$
- Total weight for D: $0.20 + 0.10 = 0.30$

So:

$$Y_S(i) = [C: 0.70, D: 0.30]$$

The voxel i becomes 70% C, 30% D, which nicely reflects it being at a fuzzy boundary.

Intuition:

- At a boundary between a very certain tissue and a very uncertain tissue, the more certain tissue "wins" and makes the boundary sharper.
- Example (like the paper's text):
 - Tissue A has uncertainty 2 voxels
 - Tissue B (e.g., ventricles) has uncertainty 0 voxels
 - The boundary "uncertainty" becomes 0, so we should not blur it.

That's why they use the minimum uncertainty, not max or average.

For each class l , the smoothed probability at voxel i is:

$$Y_S(i)[l] = \sum_{k \in i_{r_u}} W_k \cdot \mathbf{1}(Y^*(k) = l)$$

$Y^*(x) = l$ means (if we have Y with length equal to number of labels) we make element for class l equal to 1 and multiply it by its weight $\ln w_k$

The probability of label l at voxel i is the weighted fraction of neighbors in the small patch that had label l .

So if most neighbors within r_u are "cortex", but some are "subplate", you get a vector like:

$$Y_S(i) = [CP: 0.7, SP: 0.3, others: 0.0]$$

This is your soft label at voxel i .

3. A simple toy example

Let's do a 1D example with two tissues:

- Tissue A: uncertainty = 2 voxels
- Tissue B: uncertainty = 0 voxels

Imagine a line of voxels like this (labels):

```
text ⌂ Copy code
... A A A A A B B B ...
^
i = boundary voxel (currently labeled A)
```

Around voxel i , in radius R=4, you see both A and B, so you go to the "otherwise" branch.

Step 1 – Compute r_u

Look at uncertainties of A and B in this neighborhood:

- $U(A) = 2$
- $U(B) = 0$

Then:

$$r_u = \min(2, 0) = 0$$

So the boundary is considered fully certain, because B is very certain.

Step 2 – Small patch i_{r_u}

$$r_u = 0 \Rightarrow i_{r_u} = \{i\}$$

So the patch contains only voxel i itself.

Step 3 – Weights

There's only one voxel in the patch, so:

- $W_i = 1$

Step 4 – Smoothed label

$$Y_S(i) = Y(i) \quad (\text{still one-hot A})$$

So no smoothing happens at this boundary — it remains sharp.

That matches the text:

"at the locations where a tissue shares a boundary with the lateral ventricles (which have a boundary uncertainty of zero voxels), the boundary becomes unambiguous and no smoothing is performed."

4. Big picture of what this achieves

- **Inside tissues:**

The big neighborhood i_R is homogeneous \rightarrow no smoothing \rightarrow labels stay one-hot (certain).

- **At boundaries between uncertain tissues:**

- Compute a boundary uncertainty r_u .

- Mix neighbors within radius r_u according to their distance.

- Result: smooth probability transitions instead of hard jumps.

- **At boundaries with a very certain tissue (uncertainty=0):**

The min uncertainty becomes 0 \rightarrow no smoothing \rightarrow boundary remains sharp.

2.2 Novel method, part 1: spatially varying label smoothing

Why smoothing?

Segmentation labels are originally **hard one-hot**: each voxel is exactly one class.

But **near boundaries** (especially between tissues with fuzzy edges), that **hard label is not trustworthy**. Instead of pretending it's certain, they:

- Turn each voxel near boundaries into a **soft probability vector** over labels.
- Do this in a way that depends on **how uncertain each tissue's boundary is**.

2.3.1 Defining certain vs uncertain voxels *as if there is soft labeling*

Let M be a **binary mask**: *if open, it would set $m_{ij} = 1$, otherwise 0 and m is same size of masked image*

- $M(i) = 1$ if voxel i 's label changed during smoothing ($Y_S[i] \neq Y[i]$).
 - i.e. near uncertain boundaries.
- $M(i) = 0$ if smoothing did nothing (certain interior voxels).

This mask M is computed directly from Y and Y_S .

2.4 Network architecture & training

They use **nnU-Net** as the baseline framework:

- 3D U-Net architecture.
- Self-configuring patch size, batch size, preprocessing, augmentations, learning rate schedule, etc., as described in Isensee et al. (2021).

2.6 Evaluation: metrics & main results

Metrics per label:

- **DSC** (Dice Similarity Coefficient) – overlap.
- **HD95** – 95th percentile Hausdorff distance (boundary error).
- **ASSD** – average symmetric surface distance.

Then they report **mean ± std across labels** for each method, separately for younger and older fetuses (Table 1, on the last page).

3. In one sentence

The paper proposes a way to **train a fetal brain segmentation network from many cheap, noisy labels** by (1) turning hard labels into **uncertainty-aware soft labels near tissue boundaries**, and (2) using a **noise-corrected loss based on a learned label transition matrix**, and shows this beats state-of-the-art methods and classical atlas-based segmentation across >30 tissue classes and a wide gestational age range.

2.3.2 Label transition matrix T

T is an $L \times L$ matrix where:

- $T_{(i,j)} = \text{probability that true label } j \text{ is observed / flipped as noisy label } i$.

They estimate T empirically from 50 training images:

- Consider only voxels that:
 - Had original label j , and
 - Are **uncertain** ($M = 1$).
- Look at the smoothed probability distribution Y_S over classes for those voxels.
- Sum these probabilities to compute how much probability mass goes from j to each i .
- Normalise each column so it sums to 1 → T becomes a **left-stochastic matrix**.

On page 19, Fig. 2, they show heatmaps of T (after log transform) separately for younger and older fetuses. Bright diagonal means labels usually stay the same; small off-diagonals show typical confusion pairs.

2.3.3 Loss definition

Let:

- \hat{Y} = network prediction (probabilities per voxel)
- Y_S = smoothed "noisy" labels
- $L(\cdot, \cdot)$ = base loss = **cross-entropy**

They define:

- For **certain voxels ($M=0$)**: use standard cross-entropy $L(\hat{Y}, Y_S)$.
- For **uncertain voxels ($M=1$)**:
 - Following Patrini et al. (2017), they apply a **backward loss correction**:
 - Multiply \hat{Y} by $(T^T + \lambda I)^{-1}$ (with $\lambda=1$ for stability) before computing CE.
 - This approximately "undoes" the label noise described by T and recovers an **unbiased estimate of the loss w.r.t. true labels**.
 - The total loss is the sum of these two parts (equation (3)).

Intuition:

- For voxels near uncertain boundaries, the label in the training data is "noisy" in a structured way.
- T encodes which mislabelings are **plausible** (e.g. CP vs SP) and which are unlikely.
- The corrected loss **doesn't over-penalize** the network for predictions that are consistent with this uncertainty.

Some qualitative points:

- Generalized Dice and Focal loss often **completely miss one or two small structures** (empty predictions for those labels) – a problem when there are >30 classes.
- Training only on 22 clean subjects, even with transfer learning:
 - **Performs worse** than training on 272 noisy labels + their approach.
 - Shows that **more data with noise + good modeling beats tiny clean dataset** in this task.
- Standard label smoothing and SVLS both **hurt performance** vs nnU-Net baseline:
 - They don't use **tissue- and location-specific uncertainty**.
- UNet++ and DeepLab don't beat plain nnU-Net, confirming that **fancier architecture ≠ guaranteed better**.

1. Introduction

1.1. Background and motivation

Fetal magnetic resonance imaging (MRI) has emerged as an important and viable tool for assessing the development of brain in utero. It has enabled assessment of normal and abnormal brain growth trajectories in utero (Corbett-Detig et al., 2011). Moreover, fetal MRI may offer more accurate assessment and quantification of fetal brain development and degeneration when ultrasound images are inadequate (Hosny and Elghawabi, 2010; Weisstanner et al., 2015). Faster image acquisition methods (Yamashita et al., 1997) and superior super-resolution algorithms (Ebner et al., 2020; Kainz et al., 2015; Kuklisova-Murgasova et al., 2012; Gholipour et al., 2010) can now reconstruct high-quality 3D fetal brain images from stacks of 2D slices. These technical advancements have significantly improved the quality of fetal brain MRI. As a result, a growing number of works have successfully used fetal MRI to study various congenital brain disorders (Egaña-Ugrinovic et al., 2013; Mlczech et al., 2013). As the use of fetal MRI in clinical and research studies grows, quantitative image analysis methods become an urgent requirement. Automatic analysis methods can increase the speed, accuracy and reproducibility of quantification of fetal brain development. Accurate segmentation of the fetal brain into relevant tissue compartments is especially critical because many congenital brain disorders manifest themselves as changes in the size or shape of these tissues. Whereas manual segmentation is time-consuming and prone to high intra/inter-observer variability (Gousias et al., 2012), automatic segmentation promises high speed and reproducibility. As a result, in recent years there have been multiple efforts to segment different tissue compartments in the fetal brain.

review Paper about brain segmentation

Makropoulos et al. (2018) have reviewed automatic fetal and neonatal brain segmentation techniques in MRI. Here, we focus primarily on deep learning (DL) methods. Deep learning-based segmentation of adult brain into different tissue compartments has been successfully attempted by several studies in recent years (Dolz et al., 2018; Sun et al., 2019). Overall, they show that DL methods are capable of accurately segmenting brain into relevant tissue compartments. Comparatively, much fewer studies have targeted the fetal brain tissue segmentation. For fetal cortical gray matter segmentation, one study proposed obtaining cheap annotations using an automatic segmentation method originally designed for neonatal brains (Fetit et al., 2020). They used a human-in-the-loop method to refined those segmentations on selected 2D slices. A fully convolutional network (FCN) was trained using these annotations. Their method achieved an average Dice Similarity Score (DSC) of 0.76. Segmentation of cortical gray matter has been addressed by several other works. One work used a deep attentive FCN and reported a mean DSC of 0.87 (Dou et al., 2020), whereas another study proposed integrating a topological constraint into the training loss function and achieved a mean DSC of 0.70 (Dumast et al., 2021). Another study used the nnU-Net framework to segment the white matter, ventricles, and cerebellum in fetal brain MRI, achieving DSC values in the range 0.78–0.94 (Fidon et al., 2021). To reduce the impacts of motion artifacts and partial volume effects, Li et al. (2021) proposed a unified deep learning framework to jointly estimate a conditional atlas and predict a segmentation. The rationale for this approach is that the prior knowledge provided by the atlas can guide the segmentation where image quality is low. The idea of leveraging atlases to improve

human in loop

Human-in-the-loop (HITL) refers to a workflow where an automated system does the initial work, and then a human expert reviews, corrects, or guides the system, creating a loop between machine and human until the result is sufficiently good.

1. What usually happens in human-in-the-loop segmentation?

A typical HITL workflow in medical imaging looks like this:

1. Step 1 – Automatic segmentation

A preliminary segmentation is generated by an algorithm (e.g., atlas-based, deep learning, or a neonatal segmentation method as in the cited study).

2. Step 2 – Human correction

A radiologist or trained annotator:

- Checks specific slices
- Edits mistakes (e.g., missing regions, mislabeled boundaries)
- Deletes false positives, fills holes, fixes anatomical aliasing

3. Step 3 – Model updates or acceptance

- The corrections can be used to improve the model (i.e., retraining)
- Or simply accepted to produce a higher-quality final segmentation

4. Step 4 – Loop continues

If the algorithm is retrained on refined labels, the next automatic segmentation is slightly better → requires fewer corrections next time.

This is why it's called a *loop*.

3. What exactly happened in the cited study?

The paper refers to another work (Fetit et al., 2020), where:

- They used a neonatal brain segmentation tool to produce rough fetal segmentations.
- These segmentations were often wrong because neonatal models don't generalize perfectly to fetuses.
- Instead of manually segmenting entire 3D brains:
 - Annotators corrected only selected 2D slices that were most problematic.
 - Their corrections were used to refine or retrain the model.

Thus, HITL allowed them to cheaply obtain better labels for training a fetal cortical plate segmentation network.

- They use an atlas-based automatic segmentation to generate initial labels.
- Annotators only fix major errors, not all details.
(This is their own version of HITL.)
- They then use these *inexpensive but noisy labels* to train a deep learning model robustly.

deep learning-based segmentation has been explored in several other works (Oguz et al., 2018; Diniz et al., 2020; Karimi et al., 2018; Zeng et al., 2018; Karimi et al., 2019). A succession of two FCNs was proposed by Khalili et al. (2019), the first to extract the intracranial volume and the second to segment the brain tissue into seven compartments. This method achieved a mean DSC of 0.88. For segmenting the fetal brain into seven tissue compartments, another study used a single 2D UNet and achieved a mean DSC of 0.86 (Payette et al., 2020). Payette et al. (2021) compared a multi-atlas segmentation method with several DL methods for segmentation of fetal brain into seven tissue types and found that overall DL methods can achieve more accurate results.

1.2. Segmentation with noisy labels

Deep learning segmentation models, which represent the state of the art, require large accurately-labeled training datasets. Such datasets are especially difficult to come by in fetal MRI because the image quality is low and accurate multi-label segmentation of 3D images is very time-consuming. Despite recent progress in super-resolution reconstruction methods, 3D fetal MR images can suffer from residual motion and partial volume effects, making accurate delineation of tissue boundaries challenging and uncertain. Moreover, the fetal brain undergoes rapid and significant changes during the second and the third trimesters. Therefore, to develop an accurate DL model, training data should include a sufficiently large number of subjects at different gestational ages (GA) in order to fully capture the variability in the transient fetal brain structures.

need big dataset for each gestational age which is impossible and prohibitive

Because detailed manual segmentation of a large number of 3D fetal brain images is impossible or prohibitive, an alternative strategy would be to use less accurate annotations. Scenarios with weak, partial, or noisy labels are very common in medical image analysis. Hence, training of DL models with imperfect labels has been the subject of intense research in recent years (Cheplygina et al., 2019; Tajbakhsh et al., 2019; Rajchl et al., 2016). Such labels can often be obtained at low cost using automatic or semi-automatic methods. Song et al. (2020) have reviewed the state of the art methods for handling the label noise in DL. Karimi et al. (2020) present a survey that is more focused on medical image analysis applications, where the authors have identified six classes of methods for training DL models under strong label noise. Below, we describe two of the techniques that are more relevant to this work.

Loss function.—There have been many efforts to devise loss functions that are tolerant to label noise (Zhang and Sabuncu, 2018; Rusiecki, 2019). These loss functions typically tend to down-weight the penalty on data samples that incur very high loss values, under the assumption that those data samples are likely to have wrong labels. Another group of loss functions and training procedures are based on estimating and incorporating a label transition matrix (Patrini et al., 2017; Sukhbaatar et al., 2014). Label transition matrix $T \in \mathbb{R}^{L \times L}$ where L is the number of labels, is meant to describe how the correct labels are flipped into incorrect labels. If we denote the clean and noisy class probability vectors with, respectively, $p_c \in \mathbb{R}^L$ and $p_n \in \mathbb{R}^L$, then we have $p_n = T p_c$. Hence, $T_{i,j}$ is the probability that the correct label j is flipped to label i . Different approaches to estimating T and using it in

training DL classification models have been proposed in prior works (Thekumpampil et al., 2018; Bekker and Goldberger, 2016).

Label smoothing.—Label smoothing has been extensively used in image classification (Pereyra et al., 2017) as well as in natural language processing applications (Chorowski and Jaitly, 2016). However, very few studies have used label smoothing for segmentation applications. In fact, the standard label smoothing approach is unlikely to be suitable for segmentation applications. This is because, unlike classification where the whole image is represented with one probability vector, in segmentation a probability vector belongs to a single pixel/voxel and there are strong spatial correlations between the labels of nearby voxels. Standard label smoothing ignores those spatial correlations and essentially assumes that the probability that label k is flipped to label l $|k$ is the same for all l , which is an unrealistic assumption. One study suggested smoothing the object boundaries in training data in order to improve the uncertainty calibration of the trained model (Islam and Glocker, 2021). However, although they used the term “spatially-varying” to describe their method, they applied a fixed operation to all voxels. Another study proposed a label smoothing approach to improve the model uncertainty calibration for scene segmentation (Liu et al., 2021). In the context of image colorization, one study used label smoothing to achieve more accurate scene segmentation (Nguyen-Quynh et al., 2020). However, none of these studies have properly addressed the spatially-varying nature of boundary uncertainty in semantic segmentation.

(1)

Another challenge in fetal brain tissue segmentation is that it involves a large number of compartments that vary significantly in size. In this work, we aim to segment the fetal brain into more than 30 classes, where the volume of the smallest class is typically 10^4 times smaller than the volume of the largest class. This can present a significant challenge for some of the loss functions that are commonly used to train DL segmentation models.

unbalanced size
of compartment
is challenging
for loss function

The goal of this work is to develop methods for accurate fetal brain tissue segmentation in MRI. Most prior works have segmented only a single tissue (e.g., Dou et al. (2020); Fetit et al. (2020)) or have divided the fetal brain into a small number of tissue compartments (e.g., Payette et al. (2020); Fidon et al. (2021)). In this work, we consider more than 30 relevant and important tissue compartments. To capture the rapid brain growth in utero and the complex developmental trajectories of these tissues, we use a training dataset of 272 images covering the gestational age between 19 and 39 weeks. Instead of manually annotating this dataset, which would have been prohibitive, we use a combination of automatic atlas-based segmentation and manual correction of gross errors. Using the expert-estimated boundary uncertainty for different tissues, we develop novel methods for label smoothing and for training a DL model with the smoothed labels. We evaluate our trained model and compare it with several alternative methods on a set of manually-segmented test images. We show that our method achieves high segmentation accuracy and outperforms several state of the art methods.

1. What does "spatially-varying" mean?



"Spatially-varying" just means:

It changes from place to place in the image.

So something is not constant across the image; its value depends on **where** you are (which voxel).

Examples of spatially-varying things:

- Image intensity (brightness)
- Tissue type (cortex vs ventricle)
- Curvature of the cortex

Here, they are talking about **boundary uncertainty** being spatially-varying.

2. What is "boundary uncertainty"?

Boundary uncertainty = how unsure we are about exactly where the border between two tissues is.

3. Putting it together: "spatially-varying boundary uncertainty"

This means:

At different **locations** in the brain, the amount of uncertainty about the boundary is different.

Two key reasons:

1. Different tissue pairs

- CP-SP boundary is more uncertain than LV-WM boundary.
- So uncertainty is different in different regions.

2. Within the same tissue, some regions are easier or harder

- Parts of cortical plate in noisy regions or near motion artifacts are more uncertain.
- Deeper interior voxels are very certain.

4. Why is this a criticism of previous work?

Meaning:

- Previous label-smoothing methods for segmentation:
 - Either smooth everywhere the same way, or
 - Apply a fixed blur around all boundaries, regardless of which tissues are involved.
- They don't use the fact that some boundaries should be heavily smoothed (very uncertain) and some not smoothed at all (very certain).

In other words, earlier methods treat uncertainty as global and uniform, not location- and tissue-dependent.

5. What this paper does differently

This paper's method **explicitly models spatially-varying uncertainty**:

- Annotators give per-tissue boundary uncertainty (0–4 voxels).
- At each voxel:
 - Look at what tissues are nearby.
 - Derive a local uncertainty radius r_u from the *most certain* tissue.
 - Smooth within that radius, with distance-based weights.

One-sentence version

"Spatially-varying" here means boundary uncertainty is not the same everywhere in the image—it depends on where you are and which tissues meet there—and previous methods treated it as if it were uniform, while this paper designs label smoothing that *changes across space* to match that reality.

2. Materials and methods

2.1. Data and annotation procedures

Data from 294 fetuses with GA between 19.6 and 38.9 weeks (mean 30.6; standard deviation 5.3) were used in this study. These data were collected in studies approved by the institutional review board committee. Written informed consent was obtained from pregnant women volunteers who participated in research MRI scans for fetal MRI. All images were collected with 3-Tesla Siemens Skyra, Trio, or Prisma scanners using 18 or 30-channel body matrix coils via repeated T2-weighted half-Fourier acquisition single shot fast spin echo (T2wSSFSE) scans in the orthogonal planes of the fetal brain. The slice thickness was 2mm with no inter-slice gap, in-plane resolution was between 0.9mm and 1.1mm, and acquisition matrix size was 256×204 , 256×256 , or 320×320 . Volumetric images were reconstructed using an iterative slice-to-volume reconstruction algorithm (Kainz et al., 2015), brain extracted and registered to a standard atlas space in a procedure described in (Gholipour et al., 2017). The resulting 3D images had isotropic voxels of size 0.8mm. We selected 22 of these fetuses and set them aside as “test subjects” for final evaluation. The test subjects had GA between 23.3 and 38 weeks (mean 32.9; standard deviation 4.14). We used the remaining, completely independent, 272 subjects as “training subjects” to develop/train our methods and also to train the competing techniques.

We manually segmented the test images in detail. To speed up the process, we first generated automatic segmentation for each subject with a multi-atlas segmentation method using a publicly available atlas (Gholipour et al., 2017). This is a four-dimensional (i.e., spatio-temporal) atlas that covers the GA range between 19 and 39 weeks at one-week intervals. For each test fetus, we registered atlases that were within one week GA of the fetus using a diffeomorphic deformable registration algorithm. We then used the probabilistic Simultaneous Truth and Performance Level Estimation (STAPLE) algorithm (Akhondi-Asl and Warfield, 2013) to fuse the segmentations. Then, experienced annotators carefully refined all labels in several rounds until the segmentations were consistent and free of any non-trivial errors. This was a laborious effort, which required 4–10 days of work for each scan. We used these manual segmentations as “ground truth” to test our method and competing techniques.

*jitterable registration
then Probabilistic
estimation of label*

We then segmented the 272 training scans using a similar two-step approach, but with one major difference. Specifically, in the manual refinement step the annotators only corrected major errors, which on average required approximately two hours of work for each scan that had major errors. This was done because manually segmenting all 272 images with the same level of detail as done for the test images would have been impossible given the annotators’ time. Furthermore, in order to account for the potential errors and uncertainties in these segmentations, we asked the annotators to specify the degree of uncertainty in the boundary of each tissue type. Given that all images had the same spatial resolution, this uncertainty was expressed in terms of the number of voxels. The boundary uncertainty specified by the annotators varied considerably for different tissues. For example, for lateral ventricle the boundary uncertainty was 0 voxels, meaning that the boundary was generally unambiguous, while for the caudate nuclei it was 2 voxels.

Labels considered in this study included the following: hippocampus (HP)[†], amygdala (AM)[†], caudate nuclei (CD)[†], lentiform nuclei (LN)[†], thalamus (TH)[†], corpus callosum (CC), lateral ventricles (LV)[†], brainstem (ST), cerebellum (CR)[†], subthalamic nuclei (SN)[†], hippocampal commissure (HC), fornix (FN), cortical plate (CP)[†], subplate zone (SP)[†], intermediate zone (IZ)[†], ventricular zone (VZ)[†], white matter (WM)[†], internal capsule (IC)[†], CSF, and ganglionic eminence (GE)[†]. A † next to a label in this list indicates that separate components in the left and the right brain hemispheres were considered for that tissue type. In the rest of this paper, we use the acronyms defined above to refer to these tissues and structures. Following the construction of the atlas (Gholipour et al., 2017), there is one age-dependent difference in tissue labels. Specifically, fetuses that are younger than 32 weeks GA have separate SP and IZ labels, whereas for fetuses that are 32 weeks GA and older these two tissue types are merged as a single label: WM. As a result, younger fetuses have 33 tissue labels, whereas older fetuses have 31 labels (in addition to the background label). Therefore, for our method and also for all competing methods, we trained two separate models for the two fetal age groups.

2.2. Development of a DL-based segmentation method

2.2.1. Label smoothing.—In order to account for the inherent and unavoidable uncertainty in the tissue boundaries, we propose a spatially-varying label smoothing method. Our label smoothing method is presented in Algorithm 1. It is based on the fact that label uncertainty is limited to tissue boundaries and depends on the tissues that meet at the boundary. Specifically, our label smoothing strategy is based on the observation that, as confirmed by our annotators, the boundary uncertainty is dictated by the more certain tissue. For example, consider a boundary where one of the adjoining tissues has an uncertainty of two voxels (less certain) but the other has an uncertainty of zero voxels (more certain). The boundary will have an uncertainty of zero because the tissue with more certain boundary resolves the ambiguity. *→ the reason why we use min for calculating r_v*

because we get r_v base on the min of tissue in surrounding and then weight them base on neurons

Let us denote the expert-provided semi-automatic segmentation with $Y \in \mathbb{R}^{N \times L}$, where N is the number of voxels and L is the number of labels. Note that we can also write

$Y \in \{0, 1\}^{N \times L}$ because Y is a hard (0 or 1) label. For each voxel, Y is a one-hot probability vector e_k (which equals 1 at location k and 0 elsewhere), where k is the indicated tissue label for that voxel. We define $Y^* \in \mathbb{R}^N$ as $Y^* = \text{argmax}_{i \in [1, L]}(Y)$; in other words

$Y(i) = e_k \Rightarrow Y^*(i) = k$. We denote with $U \in \mathbb{R}^N$ the tissue boundary uncertainty map. U is obtained from Y by simply setting $U(i)$ to the boundary uncertainty of the tissue label for $Y(i)$. If, for example, tissue label for $Y(i)$ is caudate nuclei (CD), then $U(i) = 2$ because the boundary uncertainty for CD is two voxels. We use i^r to denote all voxels that are within a distance r from voxel i ; in other words $i^r = \{k, \|k - i\| \leq r\}$. Throughout this paper, $\|\cdot\|$ denotes the ℓ_2 -norm. Also note that all voxel indices are in fact 3D indices (i.e., they have xyz elements) and, hence, $\|k - i\|$ is a distance in \mathbb{R}^3 . However, we use single letters for indices in order to simplify the notation. Finally, we use $Y(i^r)$ to denote the “patch” of Y centered on voxel i with a radius r . *→ part of voxels around voxel i used for calculate r_v*

Algorithm 1: The proposed segmentation label smoothing algorithm.

Input: hard segmentation labels $Y \in \mathbb{R}^{N \times L}$,
tissue boundary uncertainty map $U \in \mathbb{R}^N$,
and upper bound on tissue boundary
uncertainty R .

Output: smoothed segmentation labels $Y_S \in \mathbb{R}^{N \times L}$.

Initialize: $Y_S = 0 \in \mathbb{R}^{N \times L}$.

for $i \in [1, N]$ do

- if $\text{std}[Y(i^R)] = 0$ or $\min[U(i^R)] = 0$ then
- | $Y_S(i) = Y(i)$;
- else
- | $r_u = \min[U(i^R)]$;
- | $W_k \propto \exp(-\|k - i\|/r_u) \quad \forall k \in i^u$;
- | $Y_S(i)[l] = \sum_{i^u} W \odot \mathcal{P}(Y(i^u) = l) \quad \forall l \in [1, L]$;

weights around voxel i are changed empirically for each label in R

so if we calculate y_s

Now, given manual segmentation labels, $Y \in \mathbb{R}^{N \times L}$, we would like to compute smoothed labels, $Y_S \in \mathbb{R}^{N \times L}$, that account for uncertain tissue boundaries. To do this for voxel i ,

we first consider i^R , where $R = 4$ is the upper bound of boundary uncertainty reported by our annotators for all tissues. If $Y(i^R)$ is homogeneous, that is, $\text{std}[Y(i^R)] = 0$, it means that voxel i is far from tissue boundaries since all voxels in i^R have the same label. Otherwise, voxel i is close to a boundary. In that case, we compute the boundary uncertainty as $r_u = \min[U(i^R)]$, i.e., the minimum of the uncertainty of the tissues in i^R .

This is done following the justification provided above since the tissue with the lower uncertainty determines the uncertainty of the boundary. We then use a weighted average of tissue probabilities in i^u to compute the smoothed class probability vector for this voxel.

Specifically, we use a weight matrix $W_k \propto \exp(-\|k - i\|/\tau) \quad \forall k \in i^u$, which gives higher weights to voxels that are closer to voxel i . We have found that the kernel width, τ , should depend on the patch size, which is in turn related to the boundary uncertainty. This also makes intuitive sense because for more uncertain boundaries (larger r_u) a larger kernel width should be used to achieve a higher degree of smoothing. In the experiments reported in this paper we simply set $\tau = r_u$, which we found empirically to work well. In order to ensure that the class probability vector for each voxel sums to one, W_k is normalized to sum to one. For each class label l , the smoothed probability at voxel i is then computed as:

$$Y_S(i)[l] = \sum W \odot \mathcal{P}(Y(i^u) = l), \quad (1)$$

where \odot denotes element-wise (Hadamard) product and the summation is carried out over all voxels in the patch i^u . Figure 1 shows example label smoothing results generated with our proposed method. These examples show that our method smooths the boundary of each tissue/label based not only on the boundary uncertainty of that tissue but also on the boundary uncertainty of the adjoining tissues. For example, at the locations where a tissue shares a boundary with the lateral ventricles (which have a boundary uncertainty of zero voxels), the boundary becomes unambiguous and no smoothing is performed. \rightarrow if adjoining tissues are certain boundary would be certain

both tissue and adjoining tissue are account for boundary uncertainty

2.2.2. Loss function.—We use a loss function that treats certain and uncertain regions differently. We use M to denote a binary mask that shows voxels with uncertain (smoothed) labels. M is easily obtained as voxels where the maximum class probability in Y_S is not equal to one, or equivalently, as voxels whose labels are altered in the process of label smoothing. In other words:

$$(Y_s[i] \neq Y[i]) \Rightarrow M[i] = 1. \quad (2)$$

Then, our loss function is:

$$Loss(\hat{Y}, Y_S) = \sum_{M=0} \mathcal{L}(\hat{Y}, Y_S) + \sum_{M=1} \sum_l T^{-T} \mathcal{L}(\hat{Y}, Y_S), \quad (3)$$

where T^{-T} is the inverse of transpose of T , \hat{Y} is the segmentation map predicted by our DL model, and \mathcal{L} is the base loss function, which we choose to be the cross-entropy. It has been shown that multiplication with T^{-T} , for the uncertain boundary voxels in the second loss term above, leads to an unbiased minimizer (Patrini et al., 2017). In other words, the minimizer of the corrected loss function is the same as the minimizer obtained with clean (true) labels. Even though in this work T^T was non-singular, instead of T^{-T} we used $(T^T + \lambda I)^{-1}$ with $\lambda = 1$ as suggested in prior works (Patrini et al., 2017) because it resulted in faster and more stable training.

*to prevent from
singular matrix
which are non-inversible
that label is
for class j in real
but there's Probable
by $P(Y_S = i)$ in the
soft label that
we have calculate*

We computed the label transition matrix T empirically from our training data. Specifically, after applying Algorithm 1 on the training labels, we computed T from 50 of the training images as $T_{i,j} = \sum_{Y=j \cup M=1} \mathcal{P}(Y_S = i)$, where summation is performed over all images. This empirical estimation is based on the standard definition of the label transition matrix. We normalized each column of T to sum to unity because it should be a left stochastic matrix. Figure 2 shows our estimated T for younger and older fetuses.

2.2.3. Implementation and experiments.—As the baseline for implementation and comparison of different methods (described below) we used the nnU-Net framework (Isensee et al., 2021). nnU-Net is considered to be the state of the art in medical image segmentation. In particular, on 53 different segmentation tasks, nnU-Net has shown that with proper selection of the training pipeline settings, standard 3D U-Net architectures (Çiçek et al., 2016) can match or outperform more elaborate network architectures. Hence, we adopt nnU-Net’s network architecture (i.e., a 3D U-Net) and follow its training and inference strategies. We assess the effectiveness of our proposed methods by comparing against the methods below.

*they use some loss
function of NN U-Net
which is Dice & crossentropy*

- nnU-Net. We followed the methods and settings in Isensee et al. (2021). In particular, we used the default loss function, which is the sum of cross-entropy and Dice.
- We tried three alternative loss functions. These included 1) Generalized Dice (Sudre et al., 2017) which has been proposed to improve the segmentation accuracy when the target objects are small or suffer from severe class imbalance, 2) Focal loss (Lin et al., 2017), which has been devised to address the extreme class imbalance by down-weighting the impact of structures that are easier to segment, and 3) improved Mean Absolute Error (iMAE) (Wang et al., 2019), which has been proposed for training DL models under strong label noise.

- Author Manuscript*
- **Training on clean labels.** In this approach, instead of training on the 272 images with noisy segmentations, we used leave-one-out cross-validation to train and test using the 22 images with highly accurate labels, which we have called “test subjects” so far. Of those 22 subjects, 8 were younger fetuses (GA < 32) and 14 were older fetuses (GA > 32). Therefore, for younger fetuses each time we trained the model on 7 images and tested on the remaining image, and for older fetuses each time we trained on 13 of the images and tested on the remaining image. In this approach, each of the 22 images was used as a test image in exactly one of the experiments. Therefore, the test set for this experiment was the same as for the other methods. Since the number of training images in this approach is small, we performed this experiment both without and with transfer learning, which is a common method for dealing with limited training data (Cheplygina et al., 2019; Karimi et al., 2021). For transfer learning, we used 400 subjects from the Developing Human Connectome Project dataset (DHCP) (Hughes et al., 2017; Cordero-Grande et al., 2016) for pre-training. To the best of our knowledge, this is the most similar public dataset for the application considered in this work. It includes T2 images and tissue segmentation maps with 87 labels for newborns with GA in the range 29–45 weeks. For transfer learning, we pre-trained the network using this dataset. We then replaced the last network layer (i.e., the segmentation head) with a new head to match the number of labels considered in this work (34 for the younger fetuses and 32 for the older fetuses). We fine-tuned the pre-trained network on our data following the same leave-one-out cross-validation approach described above. We performed this separately for younger and older fetuses. Moreover, as in all experiments except for UNet++ and DeepLab mentioned below, the network architecture was the same 3D U-Net from the nnU-Net framework.
 - *↪ less count of labels* **Standard label smoothing.** Following the standard label smoothing approach (Szegedy et al., 2016; Pereyra et al., 2017; Müller et al., 2019), we set $Y_s[i] = (1 - \alpha) Y[i] + \alpha/L$ for every voxel, except for the background voxels. We use $\alpha = 0.1$, which previous studies have shown to be a good setting (Pereyra et al., 2017; Müller et al., 2019).
 - **SVLS** (Islam and Glockner, 2021). SVLS is a label smoothing method, recently proposed for medical image segmentation.
 - **UNet++**. In order to also investigate the potential impact of network architecture, we compared with UNet++ (Zhou et al., 2018). This is a more elaborate nested U-Net architecture that has been proposed specifically for medical image segmentation, which claims to be better than the standard U-Net.
 - **DeepLab**. DeepLab is a popular deep learning model for semantic segmentation (Chen et al., 2017). The novelty of the network architecture is the use of atrous convolutions. Furthermore, a Conditional Random Field is used to improve the resolution of the segmentation predictions.

Note that in all of the above approaches, except for UNet++ and DeepLab, we followed the same nnU-Net framework for the choice of network architecture and training settings. We

1. What are they doing with T in the first place?

Recall: for voxels with uncertain labels, they apply a **backward loss correction**:

- In theory, you should multiply by $(T^T)^{-1}$ to "undo" the effect of label noise.
- That's the clean theory in Patrini et al. (2017): use the exact inverse of T^T .

But in practice, they instead use:

$$(T^T + \lambda I)^{-1} \quad \text{with } \lambda = 1$$

This is like a **regularized inverse** of T^T .

3. What does adding λI do?

They use:

$$(T^T + \lambda I)^{-1}, \quad \lambda = 1$$

Think of λI as a **small cushion on the diagonal**:

- It increases every eigenvalue of T^T by λ .
- So very small eigenvalues (e.g., 0.0001) become at least 1.0001.
- When you invert, you **avoid enormous numbers**.

This is a form of **Tikhonov regularization / damping**.

Effects:

1. Better conditioning

The matrix is more "well-behaved" numerically. Inversion doesn't create huge values.

2. Less noise amplification

Since you're not exactly undoing T , but a slightly "shrunk" version of it, you don't amplify tiny errors or noise in T as strongly.

3. Smoother gradients

The gradients backpropagated through this term are:

- more moderate in size,
- less erratic across updates.

This is what they mean by "more stable training": the loss decreases more smoothly, gradients don't explode, and optimization is easier.

$$T_{i,j} = \sum_{\{Y=j \cup M=1\}} \mathcal{P}(Y_s = i)$$

1. The Left Side: $T_{i,j}$

This represents a single entry in the transition matrix T at row i and column j .

- **Meaning:** It represents the probability that a voxel with the **true anatomical label** j might be observed (or mislabeled) as **label** i due to noise or uncertainty. ☀
- **High Value:** If $T_{i,j}$ is high (e.g., Cortex and Subplate), it means these two tissues are often confused near boundaries.
- **Low Value:** If $T_{i,j}$ is near 0 (e.g., Ventricle and Hippocampus), it means these tissues are rarely confused. ☀ ☀

3. Summary Example

Imagine the algorithm is calculating the transition chance from **Cortex (j)** to **Subplate (i)**:

1. It finds all voxels originally labeled **Cortex** that sit on an uncertain boundary ($M = 1$).
2. For each voxel, it checks the smoothed probability: "How much does this look like **Subplate**?" (e.g., 0.3).
3. It adds 0.3 to the total sum for $T_{\text{Subplate, Cortex}}$.
4. Finally, the columns are **normalized to sum to 1**, creating a **valid probability matrix**. ☀

Why do they do this? By building this matrix, they can tell the loss function: "Don't punish the model too hard if it predicts **Subplate** when the label is **Cortex**, because we know these frequently get confused.". ☀ ☀

2. Why is inverting T^T directly a problem?

Even if T^T is **non-singular** (invertible), it can still be **ill-conditioned**:

- Some eigenvalues of T^T might be **very small**.
- When you invert it, those tiny eigenvalues become **very large**.
- That means some directions get **huge scaling**.

What does that do in training?

- The corrected prediction $(T^T)^{-1}\hat{Y}$ can have **very large or very unbalanced values**.
- Loss and gradients can:
 - fluctuate wildly,
 - **explode in some directions**,
 - make optimization **zig-zag or diverge**.

So although mathematically correct, $(T^T)^{-1}$ can be **numerically nasty**.

4. Why can it also be **faster**?

"Faster" here is empirical, but the intuition:

- With $(T^T)^{-1}$:
 - Gradients might be huge or swing around → optimizer takes tiny effective steps or keeps overshooting.
 - You may need a smaller learning rate to avoid divergence.
- With $(T^T + I)^{-1}$:
 - Gradients are more controlled → optimizer can use a **larger effective step size**.
 - Convergence is smoother → fewer epochs to reach a good solution.

So "faster" doesn't mean the matrix inversion itself is faster (that's negligible, you compute it once). It means:

The training optimization converges quicker and with fewer numerical issues.

5. Intuitive analogy

Think of $(T^T)^{-1}$ as a very aggressive correction:

- It tries to perfectly undo the noise model.
- But if your estimate of T is even slightly off, you can **overcorrect** and create big errors.

Adding λI is like saying:

"I trust my noise model T , but not **infinitely**. I'll correct it, but with a safety margin."

That safety margin:

- Reduces overcorrection,
- Makes gradients less extreme,
- Which in practice → **more stable and often faster training**.

A. The Condition: $\{Y = j \cup M = 1\}$

The summation is performed **only** over voxels that meet two specific criteria simultaneously:

1. $Y = j$: The original "noisy" label of the voxel is tissue j . ☀ ☀
2. $M = 1$: The voxel is marked as "**uncertain**" (part of a fuzzy boundary). The mask M equals 1 only if the label changed or became "soft" during the smoothing process. ☀

Note: In this context, the symbol \cup is used to denote the set of voxels where the condition "Label is j " AND "Mask is 1" applies. ☀ ☀

B. The Value: $\mathcal{P}(Y_s = i)$

For every voxel selected by the condition above, they look at the **smoothed soft label** (Y_s).

- Y_s is a probability distribution (e.g., 70% Cortex, 30% Subplate).
- $\mathcal{P}(Y_s = i)$ is the specific probability assigned to **label** i at that voxel. ☀

refer to Isensee et al. (2021) for the details of this framework. For UNet++, we followed the settings of the original paper (Zhou et al., 2018). Furthermore, as mentioned above, for all methods except for “Training on clean labels”, we had 272 training images. For each method, we first selected a good initial learning rate using a subset of 100 images. We then trained the model using the selected initial learning rate on all 272 images. We used the “poly” learning rate decay as in Isensee et al. (2021). All training and test runs were performed using TensorFlow 1.14 under Python 3.7 on a Linux computer with an NVIDIA GeForce GTX 1080 GPU. The source code, trained model, and sample image data and segmentation labels for this work have been made publicly available at https://github.com/bchimagine/fetal_tissue_segmentation.

3. Results and Discussion

Table 1 shows the summary of the segmentation accuracy results in terms of Dice Similarity Coefficient (DSC), 95 percentile of the Hausdorff Distance (HD95), and Average Symmetric Surface Distance (ASSD). Our method has achieved the best results in terms of all three criteria. To determine the statistical significance of the differences, we performed paired t-tests to compare our method with every competing method in terms of these three criteria. These tests showed that, with a *p*-value threshold of 0.001, our method achieved significantly higher DSC and significantly lower HD95 and ASSD than all other methods, both for younger and older age groups.

In terms of all metrics, nnU-Net was the second best method after our proposed method. As we mentioned above, we used the default loss function, which is the sum of Dice and cross-entropy. Compared with this default loss function, Generalized Dice and Focal Loss performed poorly because they systematically missed one or two of the structures. That is, with Generalized Dice and Focal Loss, the network output for one or two of the labels was empty. The missed structure(s) changed with network weight initialization, but they were usually the smaller structures such as amygdala, caudate, or subthalamic nuclei. Overall, in this application with more than 30 labels we have found that loss functions based on Dice do not perform well. This may be due to the fact that the overall loss is the sum of the loss on individual labels and as the number of labels increases the relative contribution of each of the labels to the total loss becomes smaller. Since the Dice is limited to the range [0,1], the worst-case effect of a label on the total loss is $(1/L)$, where L is the number of labels. In our application with $L \approx 33$, the effect of completely missing one of the labels is only 3%. As a result, training is prone to ignoring one the labels entirely and proceed to reduce the overall loss by improving the segmentation of the other labels.

Compared with the Generalized Dice and the Focal Loss, the iMAE loss performed comparatively better and always segmented all the labels. However, it did not perform as well as nnU-Net’s default loss. The iMAE loss has been proposed to strike a balance between the Mean Absolute Error (MAE) and cross-entropy. Although some studies in image classification have shown improved accuracy with MAE and iMAE when label noise is high, our results show that in the application considered in this study they do not lead to the highest segmentation accuracy. Another limitation of the iMAE loss is much longer training time, as we discuss further below.

Training on clean labels was not effective, evidently because of the much reduced number of training images. Although this approach used more accurate manual labels for training, it was limited to far fewer training images (7 or 13 training in leave-one-out cross-validation experiments, compared with 272 training images for the other methods in Table 1).

7-13 image could be enough for segmentation but not for rapid change of fetal brain

Although in some applications 7–13 training images may be adequate to achieve high segmentation accuracy (Karimi et al., 2021), the application considered in this work is especially challenging due to the rapid fetal brain development and significant changes in the brain size and shape. Because of the rapid developments in the shape and complexity of structures such as the cortical gray matter, much larger numbers of training images are needed to allow the network to effectively learn these structures across the gestational age. As mentioned in the Methods section above, this experiment was performed both without and with transfer learning. As shown in Table 1, there was a consistent but small improvement in segmentation accuracy due to transfer learning. We used paired t-tests to assess the statistical significance of these differences. The tests showed a significant reduction in HD95 ($p < 0.001$) for both younger and older fetuses, although no significant differences ($p \approx 0.16 – 0.35$) in DSC or ASSD were found for either younger or older fetuses. The results presented for transfer learning in Table 1 were obtained by fine-tuning all layers of the pre-trained network. We experimented with other transfer learning approaches such as shallow fine-tuning (Tajbakhsh et al., 2016; Karimi et al., 2021) but did not achieve better results. The results of these experiments suggest that, in the application considered in this work, training with a small number of manually labeled images cannot achieve the same level of segmentation accuracy as training with a much larger number of training images with less accurate labels.

Standard label smoothing and SVLS did not work well and achieved worse accuracy metrics than nnU-Net. Standard label smoothing treats all voxels, including voxels that are far from any tissue boundary, in the same way. The SVLS method only smooths the boundary voxels, but it uses a label smoothing approach that does not take into account the actual tissue-dependent boundary uncertainties. The results obtained with these two methods show that un-informed or spatially uniform label smoothing is not useful for segmentation with noisy labels. Finally, UNet++ and DeepLab did not perform as well as nnU-Net, confirming the arguments presented by Isensee et al. (2021) that more elaborate network architectures and post-processing operations do not necessarily lead to better results.

Figure 3 shows more detailed label-specific DSC comparison of our method with nnU-Net, which was better than the other competing methods in terms of the overall segmentation accuracy, as shown in Table 1. For structures that have separate left and right labels, we have combined the two labels into one label in order to produce a less cluttered plot. On younger fetuses, our method achieved significantly higher DSC than nnU-Net on all structures ($p < 0.001$), except for CC, where the difference was not significant. For older fetuses, our method achieved significantly higher DSC on 14 of the structures ($p < 0.001$), while on the remaining four structures (LN, TH, WM, and CSF) although the mean DSC for our method was higher, the differences were not significant. Figure 4 shows similar plots for HD95 and ASSD. For HD95, paired t-tests showed that our proposed method achieved significantly ($p < 0.001$) lower errors than nnU-Net on all structures except for LN and CR on older fetuses.

For ASSD, our method achieved significantly ($p < 0.001$) lower errors on all structures except for CR on older fetuses.

Figure 5 shows two example segmentation maps, for one younger fetus and one older fetus, predicted by our method and nnU-Net. Both our proposed method and nnU-Net succeed in segmenting different structures with good accuracy. Nonetheless, the segmentations produced by the proposed method were consistently superior. The segmentation results produced by nnU-Net often included clearly visible errors that were not present in the segmentations produced with the proposed method.

Figure 6 shows example segmentations for specific structures, which allow us to better visualize the segmentations errors and highlight under-segmentations and over-segmentations separately. Both our proposed method and nnU-Net segment most structures with good accuracy. Our method produces visibly superior segmentations on almost all structures and has less under-segmentation and less over-segmentation. Some of these structures, such as the cortical plate, have complex 3D geometries that are very difficult to segment manually. Although the additional errors in nnU-Net's output compared with our proposed method may seem small, errors of this magnitude can make analyses which rely on consistent topology impossible and may necessitate long hours of manual correction. Our method produces better segmentations that can reduce the required manual corrections and enhance the accuracy and reproducibility of automatic analysis pipelines.

All methods considered for comparison above are based on fully convolutional networks. Further comparison with a non-DL method can be instructive and useful. To this end, we compared our proposed method with atlas-based segmentation, which is a popular classical method (Cabezas et al., 2011; Aljabar et al., 2009). We used a multi-atlas segmentation (MAS) method similar to that described in Section 2.1, without the manual refinement, to segment the 22 test images. MAS approach segmented a target image via diffeomorphic deformable registration of at least 3 atlases to the target image, followed by label fusion using probabilistic STAPLE (Akhondi-Asl and Warfield, 2013). The mean DSC of MAS segmentations on younger and older fetuses were, respectively, 0.875 ± 0.102 and 0.874 ± 0.114 . These values were significantly ($p < 0.001$) lower than those achieved by the proposed method presented in Table 1. The only structure that MAS segmented slightly more accurately than the proposed method was the cerebellum on the younger fetuses (mean DSC of 0.924 for MAS versus 0.921 for the proposed method that was statistically not significant, $p = 0.30$). For all other structures, the proposed method achieved more accurate segmentation that were mostly statistically significant.

The difference between the proposed method and MAS were largest for more convoluted structures such as cortical plate (CP), subplate zone (SP), intermediate zone (IZ), ventricular zone (VZ), and white matter (WM). We show examples of the segmentations produced with the proposed method and MAS in Figure 7. For complex structures such as CP the segmentations produced by MAS show large errors. There are at least two causes for these errors. First, the registration between the atlas and the target images is never perfect, and the registration error is especially larger at the locations of thin and complex structures such as CP. Registration is difficult in fetal MRI due to stronger partial volume effects.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

and reduced spatial resolution. Second, there is significant inter-subject variability in the local shape of these structures, and atlases cannot fully represent this variability. Therefore, regardless of the registration error, the segmentation accuracy of atlas-based methods is limited by the fact that an atlas can only represent the “average shape” and fails to capture the inter-subject variability. Deep learning methods avoid these pitfalls because they learn the complex relationship between the image intensity maps and the target segmentation map directly from the subject training data, rather than inferring it from an atlas.

The training time for our proposed method and all other deep learning methods was 30 hours. The only exception was training with the iMAE loss that required 100 hours. The inference time for a test image was approximately 4–6 seconds, depending on the brain size with larger brains taking closer to 6 seconds. For MAS, non-rigid registration of the atlas images to the target image took approximately 12 minutes and the label fusion with the probabilistic STAPLE took approximately 13 minutes, for a total of approximately 25 minutes.

4. Conclusions

As the diagnostic quality of fetal MRI improves and its applications grow, accurate quantitative analysis methods are increasingly needed to take full advantage of this imaging technique. Training DL models to address the needs of this application is challenged by the typically low image quality and by difficulty of obtaining large labeled datasets. Our study is a step forward in improving the accuracy and reproducibility of fetal MRI analysis as it enables accurate segmentation of the brain into more than 30 relevant and important tissue compartments. To address the main challenges outlined above, we used a semi-automatic method to obtain segmentation labels on our training images. This enabled us to segment a large number of training images with reasonable annotator time. We developed a novel training method based on a label smoothing strategy that accounted for tissue boundary uncertainties. This enabled us to account for the label noise in a systematic and principled manner in our model training. Our evaluations showed that our method produced significantly more accurate segmentations than the state of the art. The improved segmentation accuracy offered by our proposed method can translate into more accurate and more reproducible quantitative assessment of fetal brain development. It can also lead to substantial reductions in the annotator time because manual segmentations are very time-consuming. Finally, our methods may be useful in many similar setting in medical image segmentation where boundary uncertainties are important.

Acknowledgments

This project was supported in part by the National Institute of Biomedical Imaging and Bioengineering, the National Institute of Neurological Disorders and Stroke, and the Eunice Kennedy Shriver National Institute of Child Health and Human Development of the National Institutes of Health (NIH) under award numbers R01EB031849, R01EB032366, R01HD109395, R01NS106030, K23NS101120, and R01NS121334; in part by the Office of the Director of the NIH under award number S10OD0250111; in part by the National Science Foundation (NSF) under award 2123061; in part by a research award from the Additional Ventures; in part by NVIDIA corporation through the Applied Research Accelerator Program and the Academic Hardware Grant Program; and in part by a Technological Innovations in Neuroscience Award from the McKnight Foundation. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the NIH, NSF, Additional Ventures, or the McKnight Foundation.

$$A = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.6 & 0.1 \\ 0.0 & 0.5 & 0.5 \end{bmatrix}$$

A left stochastic matrix (also called a row stochastic matrix) is a square matrix where:

- All entries are non-negative (≥ 0)
- Each row sums to 1

Check:

- Row 1: $0.7 + 0.2 + 0.1 = 1$
- Row 2: $0.3 + 0.6 + 0.1 = 1$
- Row 3: $0.0 + 0.5 + 0.5 = 1$

Formal definition:

Let A be an $n \times n$ matrix.

A is left stochastic if:

1. $a_{ij} \geq 0$ for all i, j
2. $\sum_{j=1}^n a_{ij} = 1$ for each row i

Common uses:

- Markov chains: Each entry a_{ij} represents the probability of transitioning from state i to state j in one step.
- Random walks: Describes probabilities of moving between nodes in a network.
- PageRank algorithm: Used to rank web pages based on link structures.

Comparison with related terms:

- **Right stochastic matrix (column stochastic)**: Each column sums to 1.
- **Doubly stochastic matrix**: Both rows and columns sum to 1 (and all entries non-negative).

In the context of matrices, an **entry** (also called an **element** or **component**) refers to a single number or value located at a specific row and column position within the matrix.

We used anatomical MRI scans from the DHCP in a transfer learning experiment. This dataset is provided by the developing Human Connectome Project, KCL-Imperial-Oxford Consortium funded by the European Research Council under the European Union Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. [319456]. We are grateful to the families who generously supported this trial.

References

- Akhondi-Asl A, Warfield SK, 2013. Simultaneous truth and performance level estimation through fusion of probabilistic segmentations. *IEEE transactions on medical imaging* 32, 1840–1852. [PubMed: 23744673]
- Aljabar P, Heckemann RA, Hammers A, Hajnal JV, Rueckert D, 2009. Multi-atlas based segmentation of brain images: atlas selection and its effect on accuracy. *Neuroimage* 46, 726–738. [PubMed: 19245840]
- Bekker AJ, Goldberger J, 2016. Training deep neural-networks based on unreliable labels, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 2682–2686.
- Cabezas M, Oliver A, Lladó X, Freixenet J, Cuadra MB, 2011. A review of atlas-based segmentation for magnetic resonance brain images. *Computer methods and programs in biomedicine* 104, e158–e177. [PubMed: 21871688]
- Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL, 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 834–848. [PubMed: 28463186]
- Cheplygina V, et al., 2019. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis* 54, 280–296. [PubMed: 30959445]
- Chorowski J, Jaitly N, 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*
- Çiçek Ö, Abdulkadir A, et al., 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 424–432.
- Corbett-Detig J, et al., 2011. 3d global and regional patterns of human fetal subplate growth determined in utero. *Brain Structure and Function* 215, 255–263. [PubMed: 21046152]
- Cordero-Grande L, et al., 2016. Sensitivity encoding for aligned multishot magnetic resonance reconstruction. *IEEE Transactions on Computational Imaging* 2, 266–280.
- Diniz JOB, Ferreira JL, Diniz PHB, Silva AC, de Paiva AC, 2020. Esophagus segmentation from planning ct images using an atlas-based deep learning approach. *Computer Methods and Programs in Biomedicine* 197, 105685. [PubMed: 32798976]
- Dolz J, Desrosiers C, Ayed IB, 2018. 3d fully convolutional networks for subcortical segmentation in mri: A large-scale study. *NeuroImage* 170, 456–470. [PubMed: 28450139]
- Dou H, et al., 2020. A deep attentive convolutional neural network for automatic cortical plate segmentation in fetal mri. *IEEE transactions on medical imaging* 40, 1123–1133.
- Dumast P.d., et al., 2021. Segmentation of the cortical plate in fetal brain mri with a topological loss, in: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis. Springer, pp. 200–209.
- Ebner M, et al., 2020. An automated framework for localization, segmentation and super-resolution reconstruction of fetal brain mri. *NeuroImage* 206, 116324. [PubMed: 31704293]
- Egaña-Ugrinovic G, et al., 2013. Differences in cortical development assessed by fetal mri in late-onset intrauterine growth restriction. *American journal of obstetrics and gynecology* 209, 126–e1.
- Fetit AE, et al., 2020. A deep learning approach to segmentation of the developing cortex in fetal brain mri with minimal manual labeling, in: Medical Imaging with Deep Learning, PMLR. pp. 241–261.
- Fidon L, et al., 2021. Distributionally robust segmentation of abnormal fetal brain 3d mri, in: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis. Springer, pp. 263–273.

- Gholipour A, et al., 2010. Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain mri. *IEEE transactions on medical imaging* 29, 1739–1758. [PubMed: 20529730]
- Gholipour A, et al., 2017. A normative spatiotemporal mri atlas of the fetal brain for automatic segmentation and analysis of early brain growth. *Scientific reports* 7, 1–13. [PubMed: 28127051]
- Gousias IS, et al., 2012. Magnetic resonance imaging of the newborn brain: manual segmentation of labelled atlases in term-born and preterm infants. *Neuroimage* 62, 1499–1509. [PubMed: 22713673]
- Hosny IA, Elghawabi HS, 2010. Ultrafast mri of the fetus: an increasingly important tool in prenatal diagnosis of congenital anomalies. *Magnetic resonance imaging* 28, 1431–1439. [PubMed: 20850244]
- Hughes EJ, et al., 2017. A dedicated neonatal brain imaging system. *Magnetic resonance in medicine* 78, 794–804. [PubMed: 27643791]
- Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH, 2021. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods* 18, 203–211. [PubMed: 33288961]
- Islam M, Glocker B, 2021. Spatially varying label smoothing: Capturing uncertainty from expert annotations, in: *International Conference on Information Processing in Medical Imaging*, Springer, pp. 677–688.
- Kainz B, et al., 2015. Fast volume reconstruction from motion corrupted stacks of 2d slices. *IEEE transactions on medical imaging* 34, 1901–1913. [PubMed: 25807565]
- Karimi D, Dou H, Warfield SK, Gholipour A, 2020. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis* 65, 101759. [PubMed: 32623277]
- Karimi D, Samei G, Kesch C, Nir G, Salcudean SE, 2018. 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* 13, 1211–1219. [PubMed: 29766373]
- Karimi D, Warfield SK, Gholipour A, 2021. Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. *Artificial Intelligence in Medicine* 116, 102078. [PubMed: 34020754]
- Karimi D, Zeng Q, Mathur P, Avinash A, Mahdavi S, Spadiner I, Abolmaesumi P, Salcudean SE, 2019. Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound images. *Medical Image Analysis* 57, 186 – 196. URL: <http://www.sciencedirect.com/science/article/pii/S1361841519300623>, doi:10.1016/j.media.2019.07.005. [PubMed: 31325722]
- Khalili N, et al., 2019. Automatic brain tissue segmentation in fetal mri using convolutional neural networks. *Magnetic resonance imaging* 64, 77–89. [PubMed: 31181246]
- Kuklisova-Murgasova M, et al., 2012. Reconstruction of fetal brain mri with intensity matching and complete outlier removal. *Medical image analysis* 16, 1550–1564. [PubMed: 22939612]
- Li L, Sinclair M, Makropoulos A, Hajnal JV, David Edwards A, Kainz B, Rueckert D, Alansary A, 2021. Cas-net: Conditional atlas generation and brain segmentation for fetal mri, in: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*. Springer, pp. 221–230.
- Lin TY, Goyal P, Girshick R, He K, Dollár P, 2017. Focal loss for dense object detection, in: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.
- Liu B, et al., 2021. The devil is in the margin: Margin-based label smoothing for network calibration. *arXiv preprint arXiv:2111.15430*
- Makropoulos A, et al., 2018. A review on automatic fetal and neonatal brain mri segmentation. *NeuroImage* 170, 231–248. [PubMed: 28666878]
- Mlczoch E, et al., 2013. Structural congenital brain disease in congenital heart disease: results from a fetal mri program. *European Journal of Paediatric Neurology* 17, 153–160. [PubMed: 22944287]
- Müller R, et al., 2019. When does label smoothing help? Advances in neural information processing systems 32.

- Nguyen-Quynh TT, Kim SH, Do NT, 2020. Image colorization using the global scene-context style and pixel-wise semantic segmentation. *IEEE Access* 8, 214098–214114.
- Oguz BU, Wang J, Yushkevich N, Pouch A, Gee J, Yushkevich PA, Schwartz N, Oguz I, 2018. Combining deep learning and multi-atlas label fusion for automated placenta segmentation from 3dus, in: *Data Driven Treatment Response Assessment and Preterm, Perinatal, and Paediatric Image Analysis*. Springer, pp. 138–148.
- Patrini G, et al., 2017. Making deep neural networks robust to label noise: A loss correction approach, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1944–1952.
- Payette K, Kottke R, Jakab A, 2020. Efficient multi-class fetal brain segmentation in high resolution mri reconstructions with noisy labels, in: *Medical Ultrasound, and Preterm, Perinatal and Paediatric Image Analysis*. Springer, pp. 295–304.
- Payette K, et al., 2021. An automatic multi-tissue human fetal brain segmentation benchmark using the fetal tissue annotation dataset. *Scientific Data* 8, 1–14. [PubMed: 33414438]
- Pereyra G, et al., 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*
- Rajchl M, Lee MC, Oktay O, Kamnitsas K, Passerat-Palmbach J, Bai W, Damodaram M, Rutherford MA, Hajnal JV, Kainz B, et al., 2016. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE transactions on medical imaging* 36, 674–683. [PubMed: 27845654]
- Rusiecki A, 2019. Trimmed robust loss function for training deep neural networks with label noise, in: *International Conference on Artificial Intelligence and Soft Computing*, Springer, pp. 215–222.
- Song H, et al., 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*
- Sudre CH, et al., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, in: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, pp. 240–248.
- Sukhbaatar S, et al., 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*
- Sun L, et al., 2019. A 3d spatially weighted network for segmentation of brain tissue from mri. *IEEE transactions on medical imaging* 39, 898–909. [PubMed: 31449009]
- Szegedy C, et al., 2016. Rethinking the inception architecture for computer vision, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.
- Tajbakhsh N, Jeyaseelan L, Li Q, Chiang J, Wu Z, Ding X, 2019. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *arXiv preprint arXiv:1908.10454*
- Tajbakhsh N, et al., 2016. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE transactions on medical imaging* 35, 1299–1312. [PubMed: 26978662]
- Thekumpampil KK, Khetan A, Lin Z, Oh S, 2018. Robustness of conditional gans to noisy labels. *Advances in neural information processing systems* 31.
- Wang X, Kodirov E, Hua Y, Robertson NM, 2019. Derivative manipulation for general example weighting. *arXiv preprint arXiv:1905.11233*
- Weissstanner C, et al., 2015. Mri of the fetal brain. *Clinical neuroradiology* 25, 189–196. [PubMed: 26063004]
- Yamashita Y, et al., 1997. Mr imaging of the fetus by a haste sequence. *AJR. American journal of roentgenology* 168, 513–519. [PubMed: 9016238]
- Zeng Q, Samei G, Karimi D, Kesch C, Mahdavi SS, Abolmaesumi P, Salcudean SE, 2018. Prostate segmentation in transrectal ultrasound using magnetic resonance imaging priors. *International journal of computer assisted radiology and surgery* 13, 749–757. [PubMed: 29589259]
- Zhang Z, Sabuncu M, 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems* 31.
- Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J, 2018. Unet++: A nested u-net architecture for medical image segmentation, in: *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, pp. 3–11.

Our aim is automatic fetal brain tissue segmentation in MRI.
We generate noisy labels on a larger number of images using an atlas-based method.
We develop methods for training a deep learning model with these noisy labels.
On manually labeled images, our method achieves superior segmentation accuracy.

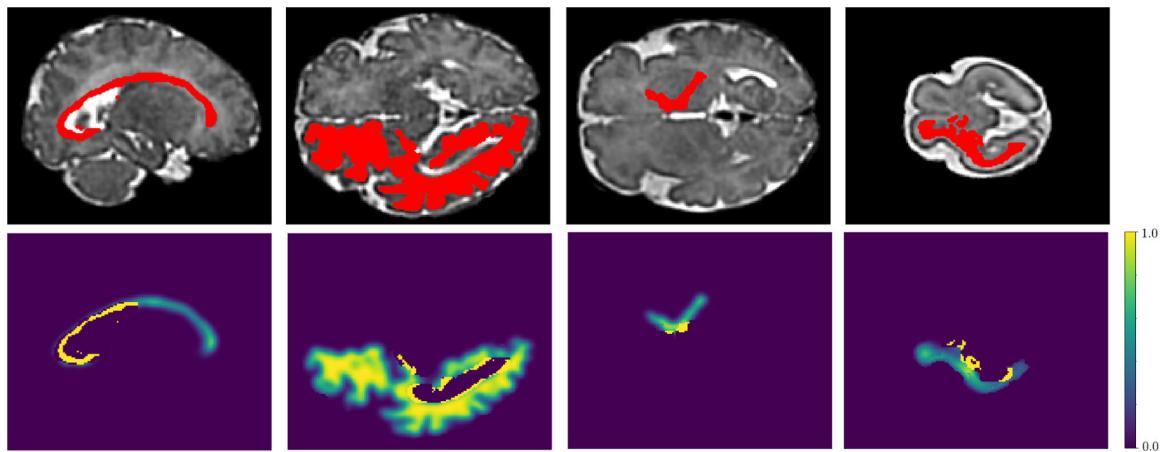
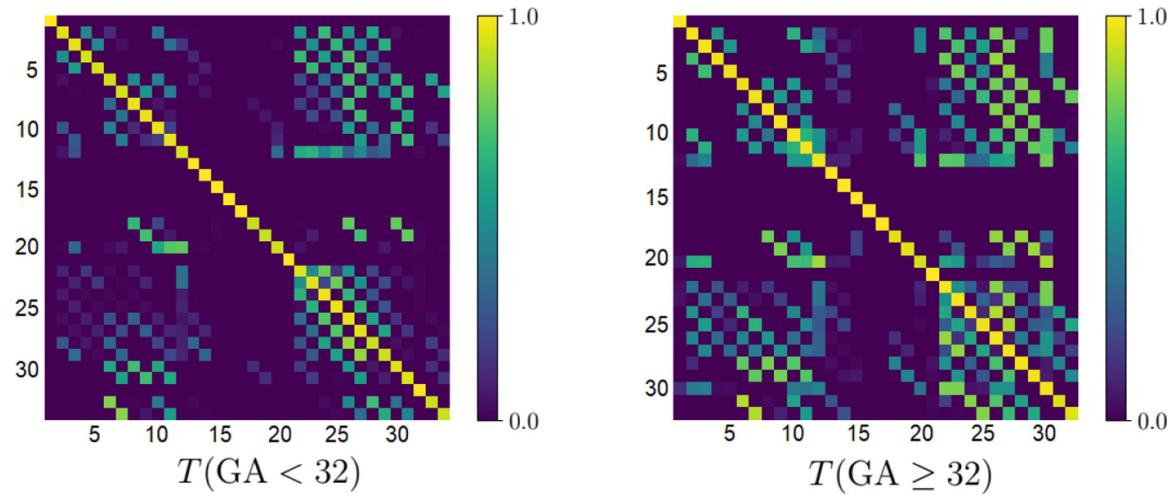


Fig. 1.

Examples of **label smoothing** performed by our proposed method (Algorithm 1). Note that for each image our algorithm performs the smoothing simultaneously on all labels, but here we show a single label in each example for better visualization. In each of the examples presented here, the top image shows the original label obtained via multi-atlas segmentation followed by manual correction of large errors, super-imposed on the T2 image. The lower images show the smoothed label.

**Fig. 2.**

Label transition matrices, T , for the younger (left) and older (right) age groups. We have used a log transformation, $\log(T + 0.001)$, to display these matrices in order to better highlight the smaller off-diagonal elements. Label numbers have been shown on the rows and columns of the matrices. For younger fetuses the label names are as follows (acronyms have been defined in the text in Section 2.1). 1: background, 2: HP left, 3: HP right, 4: AM left, 5: AM right, 6: CD left, 7: CD right, 8: LN left, 9: LN right, 10: TH left, 11: TH right, 12: CC, 13: LV left, 14: LV right, 15: ST, 16: CR left, 17: CR right, 18: SN left, 19: SN right, 20: HC, 21: FN, 22: CP left, 23: CP right, 24: SP left, 25: SP right, 26: IZ left, 27: IZ right, 28: VZ left, 29: VZ right, 30: IC left, 31: IC right, 32: CSF, 33: GE left, 34: GE right. For older fetuses, the first 23 labels are the same as those for younger fetuses, and the remaining labels are as follows. 24: VZ left, 25: VZ right, 26: WM left, 27: WM right, 28: IC left, 29: IC right, 30: CSF, 31: GE left, 32: GE right.

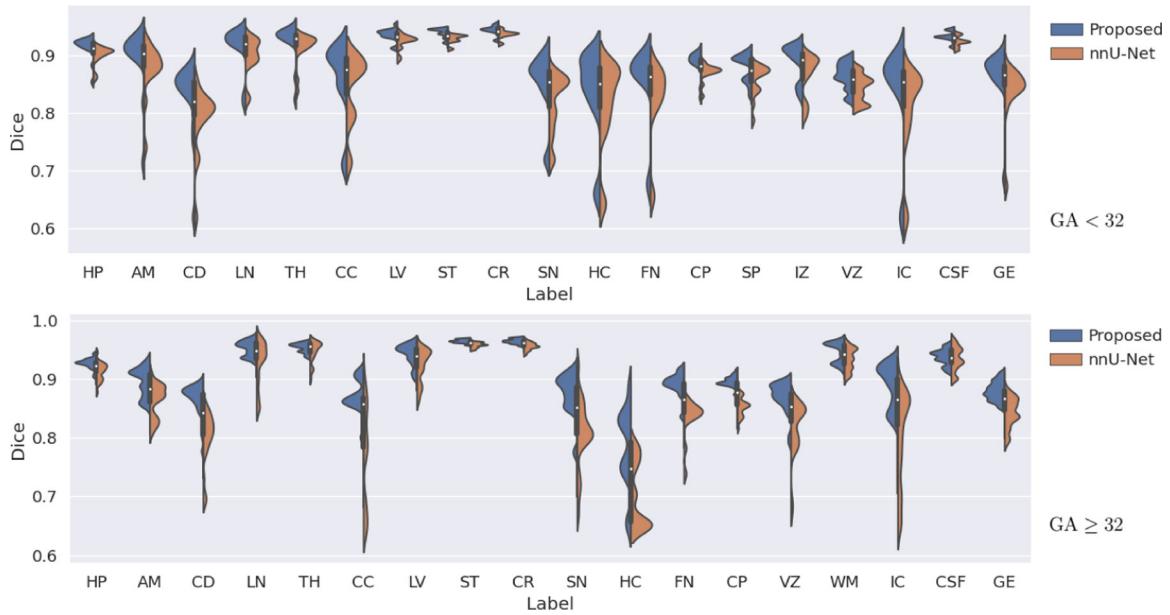
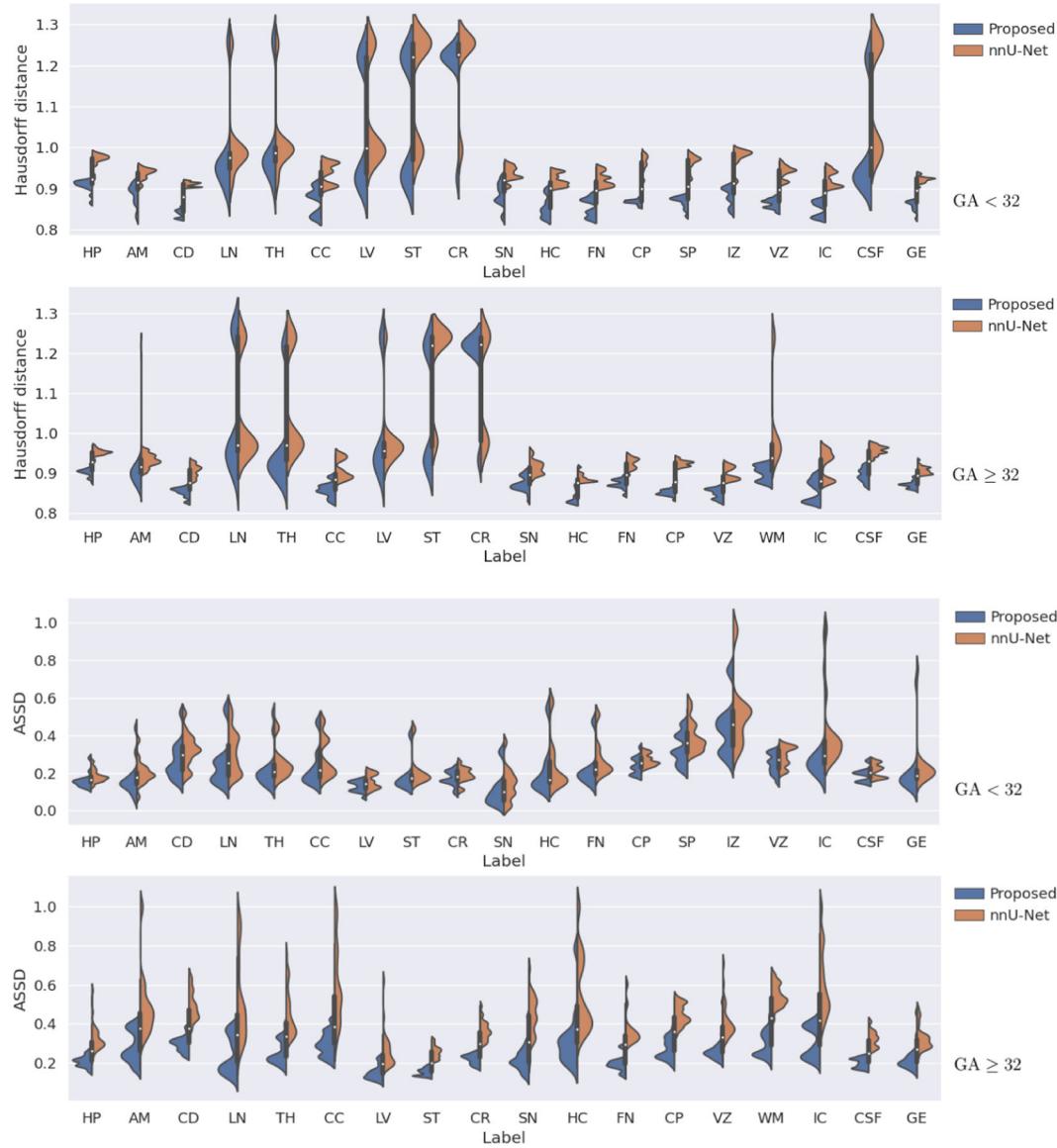


Fig. 3.

Comparison of our method and nnU-Net in terms of DSC for different structures on younger (top) and older (bottom) fetuses.

**Fig. 4.**

Comparison of our method and nnU-Net in terms of HD95 (the top two plots) and ASSD (the bottom two plots) on younger and older fetuses. The GA group for each plot is shown on the right of that plot.

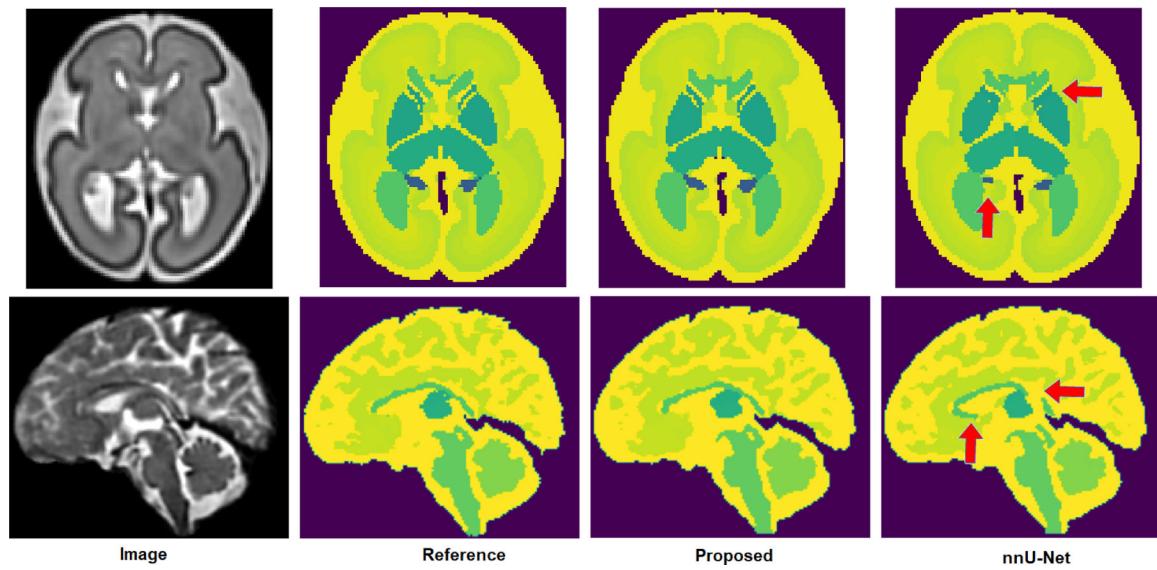


Fig. 5.

Two example segmentation maps predicted by our proposed method and nnU-Net. Red arrows point to some of nnU-Net's segmentation errors. The top example is for a younger fetus (GA= 26.7 weeks), whereas the bottom example is for an older fetus (GA= 38 weeks).

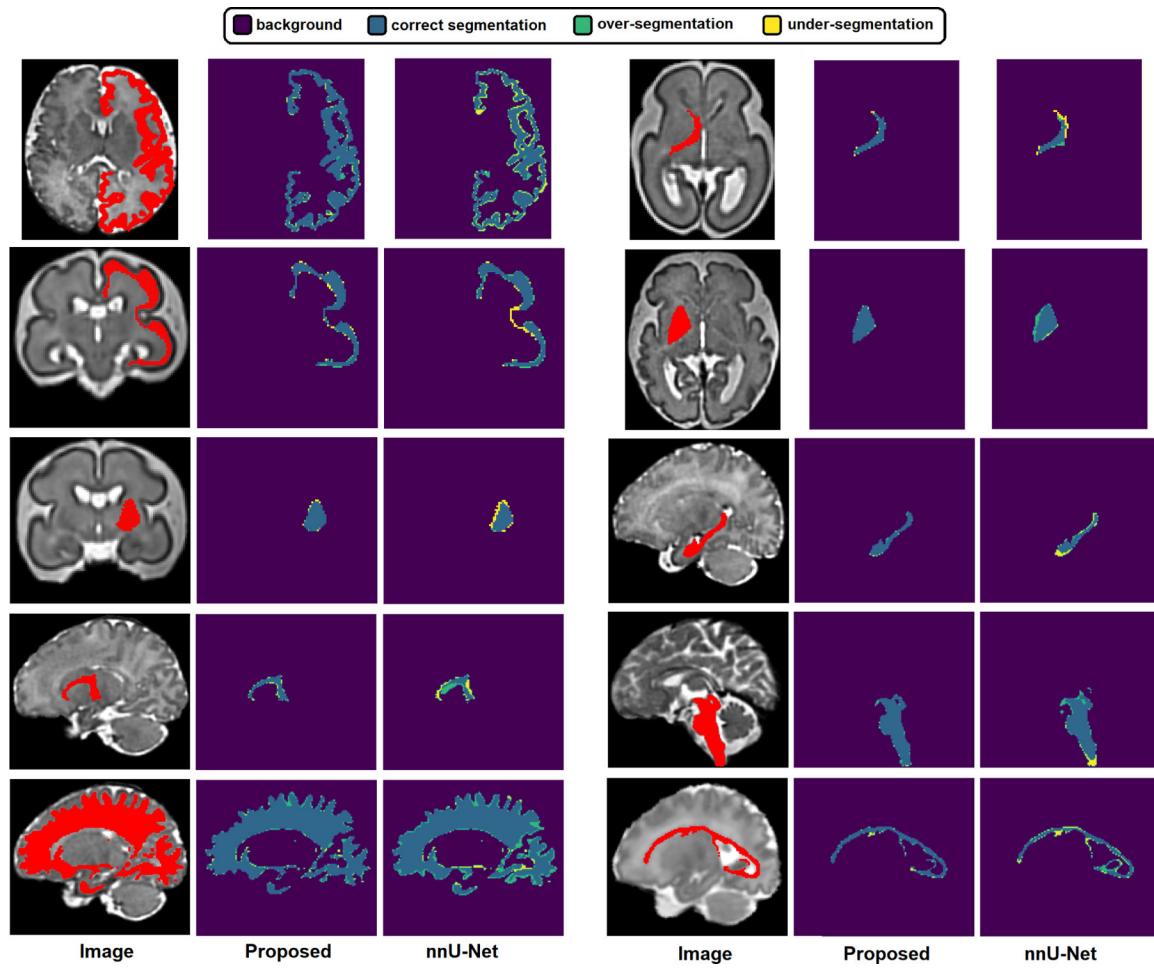


Fig. 6.

Example segmentations predicted by our proposed method and nnU-Net. Each of the ten examples shows one isolated structure and highlights correct segmentations, over-segmentations and under-segmentations.

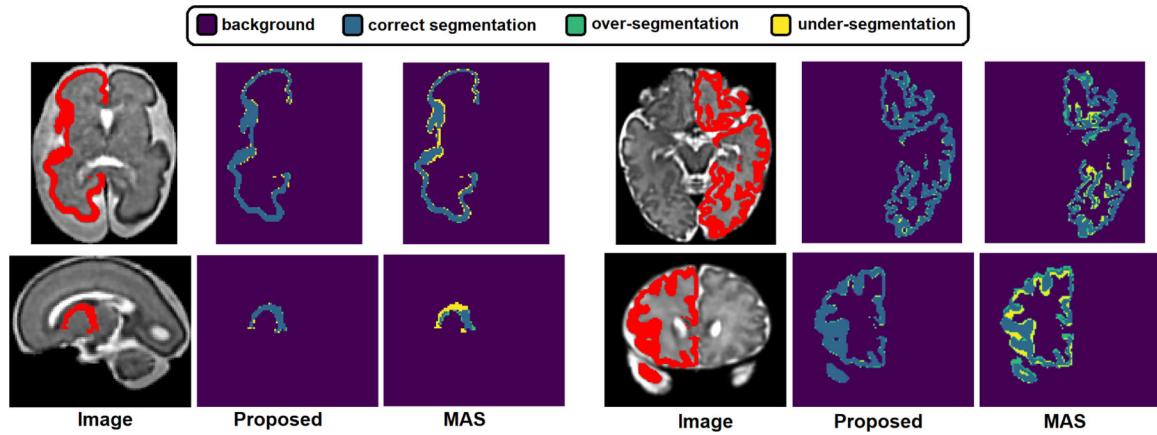


Fig. 7.

Example segmentations predicted by our proposed method and multi-atlas segmentation (MAS). Each of the four examples in this figure shows one isolated structure and highlights correct segmentations, over-segmentations and under-segmentations.

Table 1.
Segmentation accuracy metrics presented separately for younger and older fetuses. The metrics were computed separately for each label; this table presents mean \pm standard deviation over all labels. Best results for each metric are in bold. We used paired t-tests to compare our proposed method with every other method. Asterisks in this table denote significantly better results for the proposed method than all other methods

(at a significant threshold of $p < 0.001$). In the Method column in this table, T.L. stands for transfer learning.

Dataset	Method	DSC	HD95 (mm)	ASSD (mm)
Younger fetuses	nnU-Net	0.872 ± 0.063	0.99 ± 0.11	0.26 ± 0.14
	Generalized Dice	0.845 ± 0.087	1.09 ± 0.12	0.32 ± 0.26
	Focal loss	0.839 ± 0.080	1.15 ± 1.20	0.30 ± 0.19
	iMAE	0.865 ± 0.075	1.06 ± 0.15	0.26 ± 0.17
	Training on clean labels (without T.L.)	0.863 ± 0.068	1.09 ± 0.14	0.28 ± 0.16
	Training on clean labels (with T.L.)	0.866 ± 0.062	1.03 ± 0.13	0.27 ± 0.17
	Standard label smoothing	0.833 ± 0.084	1.08 ± 0.17	0.34 ± 0.21
	SVLS	0.843 ± 0.074	1.07 ± 0.17	0.30 ± 0.18
	DeepLab	0.851 ± 0.072	1.11 ± 0.15	0.30 ± 0.15
	UNet++	0.866 ± 0.060	1.02 ± 0.14	0.27 ± 0.15
Older fetuses	Proposed method	$0.893 \pm 0.066^*$	$0.94 \pm 0.13^*$	$0.23 \pm 0.13^*$
	nnU-Net	0.896 ± 0.066	0.98 ± 0.11	0.36 ± 0.12
	Generalized Dice	0.866 ± 0.070	1.16 ± 0.11	0.46 ± 0.15
	Focal loss	0.861 ± 0.068	1.16 ± 0.16	0.42 ± 0.16
	iMAE	0.880 ± 0.064	1.09 ± 0.17	0.41 ± 0.20
	Training on clean labels (without T.L.)	0.877 ± 0.073	1.12 ± 0.14	0.40 ± 0.18
	Training on clean labels (with T.L.)	0.880 ± 0.070	1.04 ± 0.15	0.40 ± 0.20
	Standard label smoothing	0.853 ± 0.071	1.16 ± 0.12	0.39 ± 0.23
	SVLS	0.856 ± 0.077	1.10 ± 0.13	0.37 ± 0.27
	DeepLab	0.865 ± 0.074	1.21 ± 0.19	0.43 ± 0.26
	UNet++	0.885 ± 0.070	1.08 ± 0.16	0.38 ± 0.23
	Proposed method	$0.916 \pm 0.059^*$	$0.94 \pm 0.13^*$	$0.25 \pm 0.09^*$