Segmentation-Based Fetal Head Biometry Using Ellipse Fitting

Author: Fathima Nashiba M
Discipline: Biomedical Engineering
Framework: PyTorch
Execution Mode: CPU-only
Date: 31 December 2025

---

a. Overall Approach

This work formulates fetal head biometry estimation as a segmentation-driven geometric analysis problem. The pipeline is structured into two sequential stages. The first stage focuses on accurate segmentation of the fetal cranial boundary from ultrasound images. The second stage derives biometric parameters—Biparietal Diameter (BPD), Occipitofrontal Diameter (OFD), and Head Circumference (HC)—by fitting an ellipse to the segmented skull contour.

Ultrasound imaging presents significant challenges due to speckle noise, weak and incomplete boundaries, and intensity variations across scans. Rather than directly regressing biometric measurements, the skull was first isolated via segmentation. This mirrors clinical practice, where sonographers visually trace the skull contour before performing measurements. Once a clean segmentation mask is obtained, ellipse fitting provides a robust, interpretable, and clinically meaningful geometric approximation.

A U-Net–based architecture was selected due to its proven effectiveness in biomedical image segmentation, particularly for boundary-sensitive structures. To better capture both fine skull edges and global head shape, a deeper four-level encoder–decoder U-Net was implemented.

All experiments were conducted on a **CPU-only system**, without GPU acceleration. Consequently, the pipeline was designed to balance computational feasibility and segmentation accuracy. Input images were resized to **384 × 384**, batch size was limited to **8**, and training epochs were selected conservatively. Despite these constraints, the complete pipeline—from raw ultrasound image to final biometric overlay—was successfully implemented and validated.

b. Dataset Description

The dataset consists of grayscale fetal ultrasound head images accompanied by corresponding **binary segmentation masks** representing the annotated cranial boundary. Ultrasound images inherently exhibit:

- Speckle noise
- Low contrast between skull and surrounding tissue
- Scanner-dependent intensity variation
- Thin and partially missing skull boundaries in challenging cases

A total of **622 images** were available and split into training and validation sets using an **80:20 ratio**. Care was taken to ensure that data augmentation was applied **only to the training set**, preventing information leakage into validation.

c. Data Preprocessing and Augmentation

Preprocessing was critical to improve skull visibility and stabilize segmentation learning.

The following steps were applied consistently:

- **Contrast Limited Adaptive Histogram Equalization (CLAHE):**
  Enhanced local contrast and emphasized hyperechoic skull boundaries without excessive noise amplification.
- **Gaussian Blur (3 × 3):**
  Reduced speckle noise while preserving edge continuity, limiting false-positive boundary detections.
- **Percentile-Based Intensity Normalization (2nd–98th):**
  Clipped extreme intensity values and normalized remaining intensities to improve consistency across scans.

Segmentation masks were binarized and resized using **nearest-neighbor interpolation** to preserve label integrity.

Data Augmentation (Training Only)

- Random horizontal flips
- Random rotations within −10° to +10°

These augmentations reflect realistic fetal head orientations encountered during scanning, improving generalization without introducing unrealistic distortions.

d. Model Variants and Experimental Evaluation

Multiple configurations were evaluated to assess the contribution of preprocessing, augmentation, and architectural depth. All experiments were executed on CPU.

Experiment 1: Baseline U-Net

- Shallow U-Net architecture
- Binary Cross-Entropy + Dice loss
- No preprocessing or augmentation

This configuration struggled with noisy predictions. Segmentation masks were fragmented, and ellipse fitting often failed, indicating that raw ultrasound images were insufficient for stable learning.

Experiment 2: Preprocessing + Augmentation

- CLAHE, Gaussian blur, and normalization
- Flip and rotation augmentation

This setup showed noticeable improvement. Skull boundaries became more continuous, and ellipse fitting began producing plausible measurements. However, under-segmentation persisted in low-contrast regions.

Experiment 3: Final Selected Model

- Deeper four-level U-Net
- **Focal Loss + Dice Loss**
- Learning-rate scheduler (ReduceLROnPlateau)
- Full preprocessing and augmentation pipeline

Focal loss effectively addressed severe class imbalance by emphasizing harder-to-classify boundary pixels. The deeper architecture captured both global head geometry and fine boundary details. This configuration produced the most stable masks and reliable biometric estimates and was selected as the final approach.

e. Model Architecture

A U-Net–based architecture was employed with modifications to improve boundary detection while remaining computationally feasible for CPU-only training.

Key architectural features include:

- Four encoder–decoder levels for multi-scale feature extraction
- Double convolution blocks with Batch Normalization
- Skip connections to preserve spatial resolution
- Sigmoid output layer for binary segmentation

The increased depth enhanced the model's ability to capture elongated and curved skull contours without excessive computational overhead.

f. Loss Function and Training Strategy

The fetal skull occupies a small fraction of the image, leading to pronounced class imbalance. To address this, a composite loss function was used:

1. **Dice Loss**
   Promotes spatial overlap between predicted and ground-truth masks.
2. **Focal Loss**
   Focuses learning on difficult boundary pixels and improves thin-structure segmentation.

The final loss was defined as the sum of Dice and Focal loss.

Training configuration:

- Optimizer: Adam
- Initial learning rate: 1e−3
- Batch size: 8
- Epochs: 60
- Learning-rate scheduler: ReduceLROnPlateau (based on validation Dice score)

g. Training and Validation Performance

```
Using device: cpu
🔍 Found 622 image files (Augment: True)
First few: ['000_HC.png', '001_HC.png', '002_HC.png', '003_HC.png', '004_HC.png']
🔍 Found 622 image files (Augment: False)
First few: ['000_HC.png', '001_HC.png', '002_HC.png', '003_HC.png', '004_HC.png']
Train batches: 63 | Val batches: 16

🚀 Training started

Epoch [ 1/60] | Train Loss: 0.9825 | Val Loss: 0.9685 | Val Dice: 0.0379 <- BEST
Epoch [ 2/60] | Train Loss: 0.9478 | Val Loss: 0.9783 | Val Dice: 0.0704 <- BEST
Epoch [ 3/60] | Train Loss: 0.9017 | Val Loss: 0.8933 | Val Dice: 0.1263 <- BEST
Epoch [ 4/60] | Train Loss: 0.8689 | Val Loss: 0.8684 | Val Dice: 0.1540 <- BEST
Epoch [ 5/60] | Train Loss: 0.8573 | Val Loss: 0.8760 | Val Dice: 0.1605 <- BEST
Epoch [ 6/60] | Train Loss: 0.8432 | Val Loss: 0.8402 | Val Dice: 0.1742 <- BEST
Epoch [ 7/60] | Train Loss: 0.8428 | Val Loss: 0.8446 | Val Dice: 0.1796 <- BEST
Epoch [ 8/60] | Train Loss: 0.8394 | Val Loss: 0.8343 | Val Dice: 0.1865 <- BEST
Epoch [ 9/60] | Train Loss: 0.8390 | Val Loss: 0.8267 | Val Dice: 0.1896 <- BEST
Epoch [10/60] | Train Loss: 0.8281 | Val Loss: 0.8254 | Val Dice: 0.1908 <- BEST
Epoch [11/60] | Train Loss: 0.8323 | Val Loss: 0.8339 | Val Dice: 0.1922 <- BEST
Epoch [12/60] | Train Loss: 0.8281 | Val Loss: 0.8327 | Val Dice: 0.1869
Epoch [13/60] | Train Loss: 0.8262 | Val Loss: 0.8167 | Val Dice: 0.2005 <- BEST
Epoch [14/60] | Train Loss: 0.8304 | Val Loss: 0.8188 | Val Dice: 0.1974
Epoch [15/60] | Train Loss: 0.8191 | Val Loss: 0.8365 | Val Dice: 0.1840
Epoch [16/60] | Train Loss: 0.8152 | Val Loss: 0.8126 | Val Dice: 0.2055 <- BEST
Epoch [17/60] | Train Loss: 0.8107 | Val Loss: 0.8069 | Val Dice: 0.2110 <- BEST
Epoch [18/60] | Train Loss: 0.8052 | Val Loss: 0.8130 | Val Dice: 0.2060
```

Figure 1: Training and validation Dice loss and Dice score across epochs, demonstrating stable convergence of the segmentation model under CPU-only constraints.

```
(skinenv) C:\Users\BME\Desktop\Task 2 segmentation>python Tester.py
□ Found 622 image files (Augment: False)
First few: ['000_HC.png', '001_HC.png', '002_HC.png', '003_HC.png', '004_HC.png']

□ TEST RESULTS (First 10 Samples)

Sample 0 (000_HC.png): BPD: 291.36px | OFD: 143.20px | HC: 682.60px
Sample 1 (001_HC.png): BPD: 35.40px | OFD: 7.45px | HC: 67.30px
Sample 2 (002_HC.png): BPD: 69.97px | OFD: 52.92px | HC: 193.03px
Sample 3 (003_HC.png): BPD: 132.94px | OFD: 112.25px | HC: 385.15px
Sample 4 (004_HC.png): BPD: 221.09px | OFD: 148.65px | HC: 580.79px
Sample 5 (005_HC.png): BPD: 63.07px | OFD: 24.04px | HC: 136.84px
Sample 6 (006_HC.png): BPD: 168.18px | OFD: 126.95px | HC: 463.59px
Sample 7 (007_HC.png): BPD: 28.60px | OFD: 5.42px | HC: 53.45px
Sample 8 (008_HC.png): BPD: 63.55px | OFD: 18.43px | HC: 128.77px
Sample 9 (009_HC.png): BPD: 88.59px | OFD: 58.19px | HC: 230.58px

Avg (10 samples): BPD: 116.27px | OFD: 69.75px | HC: 292.21px
□ Task B complete! Check test_overlays/ for visuals.
```

Figure 2: Predicted biometric measurements (BPD and OFD) for ten representative test ultrasound images obtained from ellipse fitting on the segmented fetal head.

h. Inference and Biometry Estimation

During inference, predicted segmentation masks were thresholded and processed using contour detection. The largest valid contour was selected, and an ellipse was fitted using OpenCV's fitEllipse function.

From the fitted ellipse:

- **BPD** was estimated from the minor axis
- **OFD** from the major axis
- **HC** using an ellipse-based circumference approximation

For qualitative validation, overlay images showing predicted masks and fitted ellipses were saved. These visualizations confirmed strong alignment between the fitted ellipse and the skull boundary when segmentation was accurate.

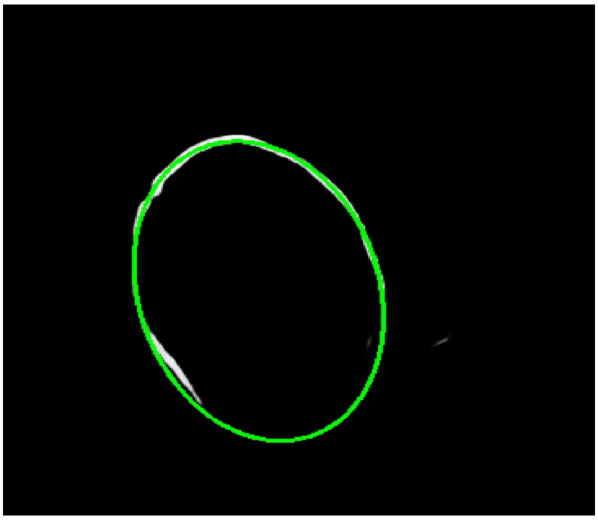

Figure 3*:* Predicted segmentation mask (white) overlaid with fitted ellipse (green) for fetal head biometry estimation.

i. Limitations and Future Improvements

The primary limitation of this work was the absence of GPU acceleration, which restricted:

- Training duration
- Batch size
- Extent of architectural experimentation

With additional computational resources, future improvements could include:

- GPU-based training with extended schedules
- Tversky or boundary-aware loss functions
- Morphological or CRF-based post-processing
- Ensemble models for improved robustness

j. Key Takeaways

This work highlights the importance of domain-specific preprocessing in ultrasound image analysis. Model depth alone was insufficient—contrast enhancement and noise suppression were essential for success. Focal loss effectively handled class imbalance, while ellipse-based biometry provided an interpretable and clinically meaningful measurement strategy.

Overall, the solution prioritizes robustness, interpretability, and reproducibility **over** brute-force computation, making it well-suited for real-world biomedical applications under limited computational resources.