Fetal Biometry Challenge – Landmark Detection

1.Overview

This module addresses fetal head biometry estimation through landmark detection on ultrasound images. The task focuses on predicting key anatomical landmarks corresponding to Biparietal Diameter (BPD) and Occipitofrontal Diameter (OFD) using a U-Net–based heatmap regression approach.

Instead of directly regressing landmark coordinates, each landmark is encoded as a 2D Gaussian heatmap, allowing the model to learn spatial probability distributions. This strategy improves robustness in ultrasound images, which are affected by speckle noise, weak edges, and intensity variations.

All experiments were conducted on a CPU-only system using PyTorch, with careful architectural and training choices to balance accuracy and computational efficiency.

2.Key Results

- **Final training MSE:** ~0.001 after 30 epochs
- **Dataset size:** 622 fetal head ultrasound images
- **Average predicted biometrics (full dataset):**
    - BPD ≈ 120 pixels
    - OFD ≈ 140 pixels
- **Estimated relative error:** ~4% (based on predicted landmark distances)

The results indicate stable landmark localization and anatomically plausible biometric estimates across the dataset.

3.Methodology Summary

- **Model:** U-Net with skip connections for spatial precision
- **Input size:** 256 × 256 grayscale ultrasound images
- **Output:** 4 heatmaps (two landmarks for BPD, two for OFD)
- **Landmark encoding:** Gaussian heatmaps for robust localization
- **Measurement:** Euclidean distance between predicted landmark pairs

The heatmap-based formulation proved more reliable than direct coordinate regression, especially for noisy ultrasound data.

4.Folder Structure

```
task_1_landmark/
│
├── Model Weights/
│   └── unet_epoch_30.pth
│
├── Python Script/
│   ├── Trainer.py
│   └── Tester.py
│   ├── Additional script to run the experiment/
│             ├──Dataset.py
│             ├──UNet.py
│
├── Results/
│   ├──HC_results.csv
│   ├──taskA_predictions.csv
├── Report /
│   ├──Report.pdf
└── README.md
```

5.How to Run

1. Update Paths

Edit the following variables in `Trainer.py` and `Tester.py`:

- `CSV_PATH`
- `IMAGE_DIR`
- `MODEL_PATH` (for testing)

2. Training

```
cd code
python Trainer.py
```

- Saves model weights after each epoch
- Approx. **2 minutes per epoch** on CPU
- Default training: **30 epochs**

3. Testing / Inference

```
python Tester.py
```

- Runs inference on the full dataset
- Saves predictions to `taskA_predictions.csv`
- Prints BPD and OFD values per image

Example output:

```
000_HC.png → BPD: 120.45 px | OFD: 140.23 px
...
```

6.Dependencies

Install CPU-only dependencies using:

```
pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cpu
pip install opencv-python numpy pandas
```

Required libraries:

- PyTorch (CPU)
- OpenCV
- NumPy
- Pandas

7.Training Notes

- **Batch size:** 8 (CPU constrained)
- **Optimizer:** Adam
- **Loss function:** Mean Squared Error (MSE) on heatmaps
- **Training behavior:**
  Loss decreased smoothly from ~0.118 to ~0.001 with aligned training and validation trends, indicating stable convergence without overfitting.

8.Experiments Conducted

- **Baseline:** Direct landmark regression (MSE ≈ 0.05)
- **With resizing:** Improved stability (MSE ≈ 0.02)
- **Final approach:** Heatmap regression + U-Net (MSE ≈ 0.001)

The final configuration demonstrated the best balance between localization accuracy and computational feasibility.

9.Limitations and Future Work

- No GPU acceleration (limited batch size and epochs)
- No explicit data augmentation
- Sub-pixel localization not explored

Future improvements:

- GPU-based training with augmentation
- Integration with segmentation-based HC estimation
- Boundary-aware or uncertainty-aware landmark modeling

10. Notes

- **PEP-8 compliant** implementation; all code written from scratch using PyTorch. No external notebooks or Colab-based training was used.
- **CPU-only execution**: Batch size = 8, input resized to **256×256** to balance localization accuracy and computational efficiency. Training was completed in **30 epochs**, as loss convergence stabilized beyond this point without signs of overfitting (training and validation MSE curves closely aligned).
- **Heatmap-based landmark regression** significantly improved robustness compared to direct coordinate regression, especially in noisy ultrasound regions. Gaussian heatmaps helped preserve spatial uncertainty and enabled more stable convergence.
- **Experimental progression**:
    - Baseline CNN (direct regression): ~0.05 MSE
        -
            - Image resizing & normalization: ~0.02 MSE
    - Final U-Net + heatmap regression: **~0.001 MSE**
- **Inference outputs** were saved as CSV files containing landmark coordinates and derived BPD/OFD measurements, enabling easy quantitative evaluation and traceability across samples.