README — Fetal Head Segmentation & Biometry Landmark Detection

Project Overview

This project focuses on fetal cranium segmentation from ultrasound images using a U-Net–based deep learning model.
The segmented mask is later used for ellipse fitting to derive **Biparietal Diameter (BPD)** and **Occipitofrontal Diameter (OFD)**.

The repository contains:

- Model weights
- Training & testing notebooks
- Dataset asset reference
- Final project report

Folder Structure

```
task_1_segmentation/
|
├── Model Weights/
|   └── hypothesis_1_full_saved_mode.pth
|
├── Python Script/
|   ├── Trainer.ipynb
|   ├── Tester.ipynb
|   └── Assets/
```

```
│       └── KaggleDatasetLink.txt
│
├── Report/
│   └── Report.pdf
│
└── README.md   (this file)
```

Directory & File Descriptions

Model Weights/

Contains trained segmentation model checkpoint.

File	Description
`hypothesis_1_full_saved_mode.pth`	Final trained U-Net model (PyTorch). Used for inference and validation.

Python Script/

Contains Jupyter notebooks for training and testing the model.

`Trainer.ipynb`

- Full training pipeline
- Dataset preprocessing

- Augmentations & normalization
- U-Net architecture
- Dice + BCE loss
- Model saving routine

Use when you want to **retrain or fine-tune** the model.

`Tester.ipynb`

- Loads trained model checkpoint
- Runs inference on validation/test images
- Converts outline masks → filled segmentation masks
- Computes Dice score
- Visualizes predictions & overlays

Use when you want to **evaluate or verify model performance**.

Assets/

Supporting references or metadata.

File	Description
`KaggleDatasetLink.txt`	Link to dataset used for training & validation.

Report/

File	Description
`Report.pdf`	Final project report including approach, methodology, experiments, metrics, and observations.

This document explains:

- Data preprocessing & augmentations
- Model architecture & hypotheses
- Performance metrics & validation analysis
- Future improvements & key takeaways

How to Use

Run Training

Open `Trainer.ipynb`, configure dataset path, and run all cells.

Run Evaluation / Testing

Open `Tester.ipynb`, load the `.pth` model, and execute:

- Dice score computation
- Sample predictions
- Mask visualization