Motivation

==========

- The motivation of this project is to be able to process the axial fetal images and get the coordinates of the 2 biometry points. The 2 biometry points are Biparietal Diameter(BPD) and Occipitofrontal diameter(OFD).
- BPD and OFD are 2 important landmarks in fetal axial images and are used to estimate gestational age and assess fetal growth and neurodevelopment, and is usually considered the first step in the detection of CNS(Central Nervous System) anomalies.
- Occipitofrontal line: line used in the measurement of head size and it extends from occiput to the frontal bone(the forehead).
- Biparietal line: it is used in assessing the size of the fetal head.
- My previous experience in medical imaging, including implementing an MRI module and analyzing cancer X-rays, had particularly fueled my enthusiasm for this project.
- Research Interest -My research interest is fueled by my commitment to ongoing education and staying updated with the latest developments in the Machine Learning field. For this particular project, I consulted a variety of research papers to understand the most recent models and architectural advancements relevant to the problem at hand. Some of which are:
    - Regressing Heatmaps for Multiple Landmark Localization using CNNs
    - Automatic Cephalometric Landmark Detection on X-ray Images Using a Deep-Learning Method
    - U-Net: Convolutional Networks for Biomedical Image Segmentation
    - Attention U-Net: Learning Where to Look for the Pancreas
- I also consulted some research papers for the specific task of understanding the problem statement, the meaning of the given biometry points and the whole importance of them.

Abstract
=======

Here I'll try and summarize the work I did for both the task and the results I accomplished at the end of it.
For task#1:

- In this case the major hurdle was processing the images. The given images in the dataset were of very varied sizes. And such large images could not be used for training purposes as that would require a lot of memory.
- Scaling the images itself wouldn't have been a very major issue except in this case, we needed to make sure that the aspect ratio of the image is being maintained. Because after resizing the images, the coordinates needed to be scaled accordingly so that they could be plotted on the image.

For task#2:

- Here I performed processing on image and mask both to resize them, as well as normalize them.
- The processed image and mask was then stored in a dictionary.
- I used the U-Net architecture to predict the segmented image.
- Ideally, I planned on getting the coordinates from the segmented image but this did not work since the final segmented image obtained was too noisy.
- This could be improved if we get better accuracy in segmentation.

Introduction
==========

For Task#1

1. Initially, after looking at the project, I decided to work with ResNET50 architecture as TransferLearning could greatly leverage the learning process. And also due to my past experience in detecting cancer images with TransferLearning, I believe, the given task could've been treated as a regression problem, where I'd pass the given coordinates as labels.
2. However, after going through several papers that dealt with medical imaging, I realized that U-Net would be a better architecture. So I converted the coordinates into heatmap and stored the heatmap as a label for training.
3. After the model was trained, I extracted the coordinates from the output heatmap and stored it in a cs file along with the image name.
4. The final variable "Result" is a pandas DataFrame which contains the name of the image from the test dataset and its corresponding predicted coordinates. This can be further downloaded if required.

Task#2:

1. For a segmentation based approach, After going through some research papers, I initially wanted to implement the activation gated version of U-Net architecture. But due to time shortage, I settled with basic U-Net architecture.
2. Here the input was the images that I'd processed and the ground truth values, the labels were processed masks.
3. The prediction output is segmented masks.
4. The final outcome would be to process these masks and extract coordinates from them.

Data PreProcessing/ Analysis

=========================

Task#1:

- First, I plotted the image along with its coordinates to get an idea about the training data, one analysis I made from this was that the training data was very inaccurate and had not been augmented properly.
- Also, I checked all the columns in the given csv file to make sure that there were no null values.
- Here data processing involved changing the size of image to (256,256). One important thing that I had to keep in mind was that the initial image was not square, so if the image had been hard coded to square shape, then aspect ratio of the image would’ve been distorted.
- Therefore, resizing of the image was done while implementing padding.
- The most difficult part of processing was getting the scaling factor right.
- Another processing step was scaling the coordinates.which had to keep the padding information so that even after resizing and scaling, the coordinates would match the image.

Task#2:

1. Here I plotted the masks (ground truth values) on the images to make sure that the given masks are a true representation of the given images. All the masks fit the image accurately.
2. Preprocessing involved, resizing both the image and the mask in the same manner.
3. Both the image and the mask were scaled to (256, 256) to make sure that even after resizing, the mask would fit the image properly.

Model Architecture:

==================

Task#1:

1. Initially I intended to use ResNET50 architecture and perform Transfer Learning. This would've been a relatively much simpler approach.
2. This is because I already had experience training the ResNET50 model on cancer images. In this case then, the problem could've been treated as a simple regression problem.
3. After going through the recommended research papers, I figured that U-Net would be a better architecture to go along with. Also, the research papers told me that rather than regressing absolute coordinates, regressing heatmaps for landmarks would be a better approach. This ideology was inspired by Pfister et al.
4. So, eventually, rather than directly predicting coordinates, I decided to represent landmark locations using heat maps since heat maps can represent the probability or likelihood of a landmark's presence at each pixel.

Task#2:

1. Since here we had to follow a segmentation approach, I immediately decided to go with U-Net as U-Net has shown outstanding performance in various biomedical segmentation tasks, outperforming existing method s in competitions like the ISBI challenge for neural structure segmentation
2. But after going through the linked research papers, I realized that there exists an enhancement of the standard U-Net architecture, the Attention U_net model which integrates a novel component called the AttentionGate (AG).
3. AGs are designed to help the network focus on relevant regions of an input image, enhancing the model's ability to detect structures of varying shapes and sizes.

4. However, I couldn't implement this because of shortage of time, and had to eventually settle with U-Net.
5. One point to note is that traditional U-Net has 30 million parameters, but by reducing the depth of my architecture, I managed to get 1.9 million parameters.

Experimentation Setting
======================

Role 1 & Role 2 :

1. In case 1 I was dealing with the prediction of heatmaps and in case 2 I was predicting segmented images.
2. Therefore in both cases I used MeanSquaredError Loss as MSE is commonly used in regression problems where the goal is to predict continuous values.
3. In my case, predicting the coordinates of specific points in an image is a regression task. Hence MSE seemed appropriate.
4. For an optimizer, I decided to go with SGD as SGD with momentum and learning rate decay can help to accelerate training and lead to better final performance. Also, SGD has been very successful in training models for a wide range of problems.
5. For the scheduler I decided to go with ReduceLROnPlateau and set it in a way that the learning rate would decrease if the test loss doesn't decrease for 20 continuous steps.
6. Batch_size : initially I tried using a batch size of 64, but got memory issues and had to set this value to 32.
7. Epochs: For starters I decided to go with 50 epochs , which could be changed later if necessary.
8. Accuracy - in the test function for, I decided to not go along with general accuracy calculation as accuracy is based on the predictions's argmax, which is typically for classification tasks. In my regression task, where the output is a heatmap, accuracy calculated in this way might not be very meaningful.
9. So although it's not very common , I decided to go with Pixel_wise accuracy using which we could compute the percentage of pixels that match between the predicted and actual heatmaps. This requires thresholding the heatmap to create binary maps.

10. I decided to not put pixel accuracy in the training function as pixel accuracy might not be as informative if the heatmap has a lot of background area compared to the area of interest, as it could be biased towards correctly predicting background pixels.
11. In role2, I decided to use IOU(image overlap) as we're dealing with segmentation problems. IOU measures the overlap between the predicted segmentation and the ground truth . It's a relation between the intersection and the union of the 2 areas.

Hypothesis Tried

===============

Unfortunately, due to lack of time, I couldn't experiment with even half of the ideas I had for the model.
And therefore I intend on putting all the possibilities in the FutureWork section.

Key Findings

============

I visualized the image that I got from model prediction and noticed that U-Net indeed gives unparalleled accuracy and performance with such a small number of epochs.

Future Work
==========

I had a lot of things in mind, but due to shortage of time, I couldn't apply several things, and hence I’m listing them here.

Task#1:

1. From reading one of the research papers linked, I found out about SpatialConfiguration-Net, which consists of a three block architecture. This architecture combines local appearance of landmarks with the spatial configuration to all other landmarks. Initially, I’d wanted to implement this architecture, but due to the significant amount of time required in understanding and then implementing the architecture, I had to settle with the U-Net architecture. So, If time permits, I’d implement this architecture.
2. With the same architecture that I'd used, U-Net. I'd have tried different criterion, Binary Cross Entropy Loss(BCE) with logits, different possible optimizers - such as Adam, different measures for Accuracy apart from PixelWise measurement.
3. Due to shortage of time, I couldn't implement any image augmentation technique. But With the current architecture, I'd really liked to try several image augmentation techniques from the albumentations library like Elastic Distortions(high priority), CutOut and AugMix
4. Also, even though I said that regressing the heatmaps rather than regressing the coordinates directly seems like a better approach, I'd still like to work with ResNET50 architecture and regress the coordinates.
   This is because ResNET50 is a very efficient architecture and gives the best accuracy in its class with minimum parameters and can be optimized and fine-tuned for a given problem.

Task#2:

1. As I stated earlier, from my paper reading, I'd realized that Attention Gated U-Net architecture would be more suitable for the current situation, so I'd implement that.
2. Also, apart from the above methods of changing optimizer, criterion, and adding image augmentation techniques, I'd also be really interested in experimenting with different segmentation techniques like semantic segmentation, instance segmentation , panoptic segmentation.
3. And also, SAM(segment-anything-model) by Meta-AI. I believe this could have improved the accuracy a lot if implemented properly.