Measurement of fetal head circumference

Name: Hoang Minh Chi
ID Number: BA12-030

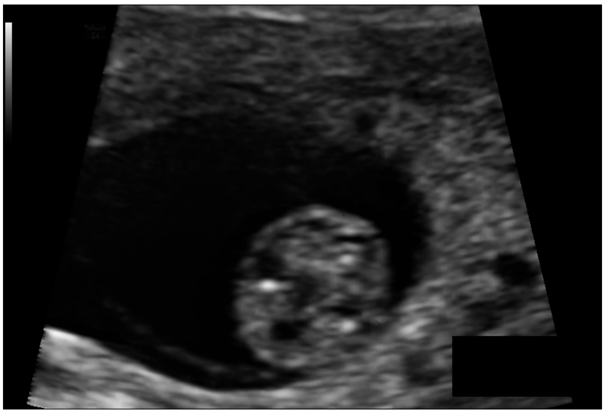
1 Introduction

In this assignment, I tried to implement a Vgg16&U-Net models with dice loss functions and dice coefficent to detect the head boundary in 2D ultrasound images, and then measure the head circumference.

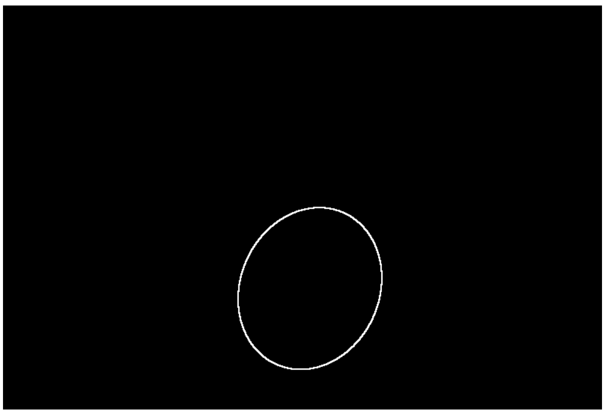
2 Data and Methodology

2.1 Overview Data

This study utilizes the HC18 Dataset from Grand Challenge, which includes 999 2D ultrasound images of fetal head and the corresponding groundtruth masks, whereas the test dataset consists of 335 2D ultrasound images of fetal head with no labels. The size of each 2D ultrasound image is 800 by 540 pixels with a pixel size ranging from 0.052 to 0.326 mm. The CSV file `training_set_pixel_size_and_HC.csv` includes the head circumference measurement (in millimeters) for each annotated HC in the training set. For training the model, 799 of the training dataset will be used for training, 100 for validation, and 100 for testing to evaluate the model.



(a) Ultrasound Image

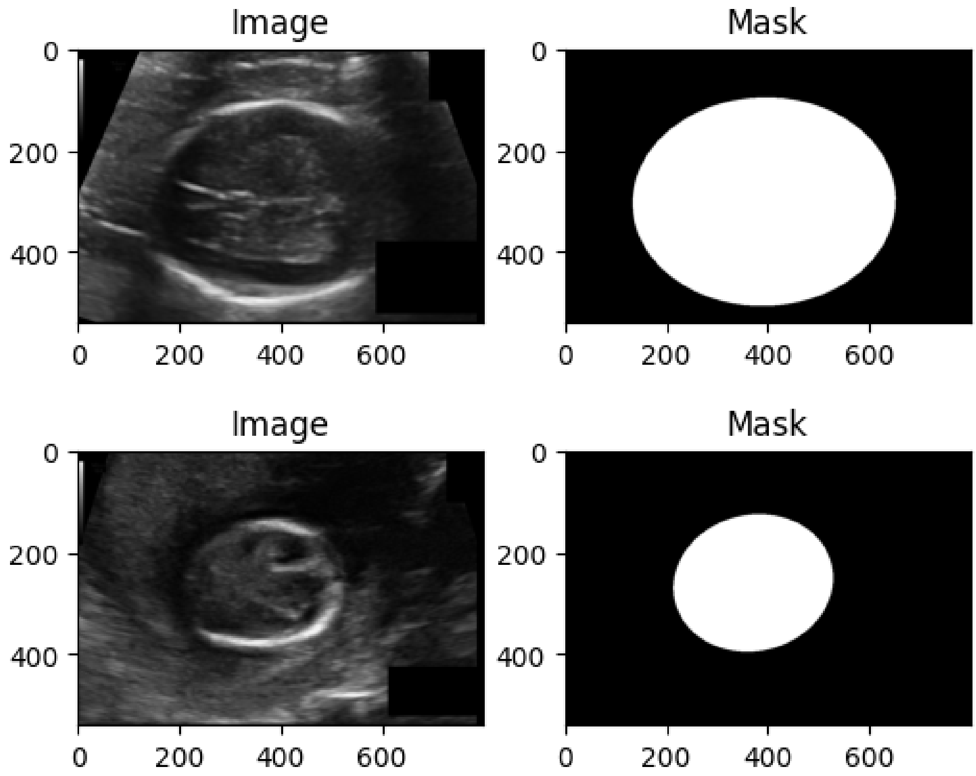


(b) Annotation of Image

Figuur 1: Example of Dataset

2.2 Data Preprocessing

Initially, I converted all the original annotated images into segmentation images, creating masks from the boundary contours of the head. Then, I resized the images to (256, 256), normalized the

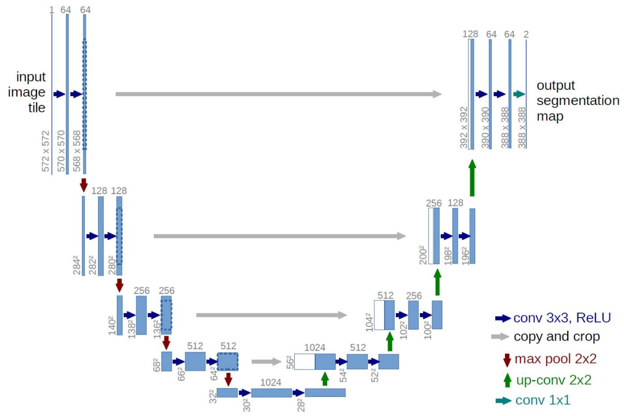


Figuur 2: Mask Of Image

pixels by dividing by 255, and converted them to the input type for TensorFlow with input shape (256, 256, 3) and output shape (256, 256, 1) with batch size = 36

2.3 Model Architecture

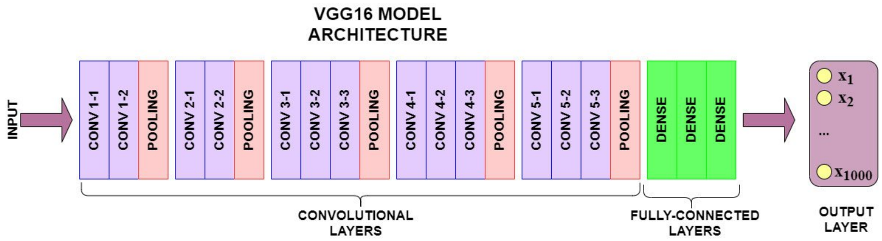
The VGG16 model, known for its powerful feature extraction capabilities, is combined with the U-Net architecture to create a robust framework for image segmentation tasks, leveraging the strengths of both deep learning models to achieve precise and efficient segmentation results.



Figuur 3: Unet Architecture

2.3.1 Encoder

The input to the model is an image with dimensions (256, 256, 3). Then, I use the pre-trained VGG16 model, removing the last layer and utilizing the weights trained on the ImageNet dataset.

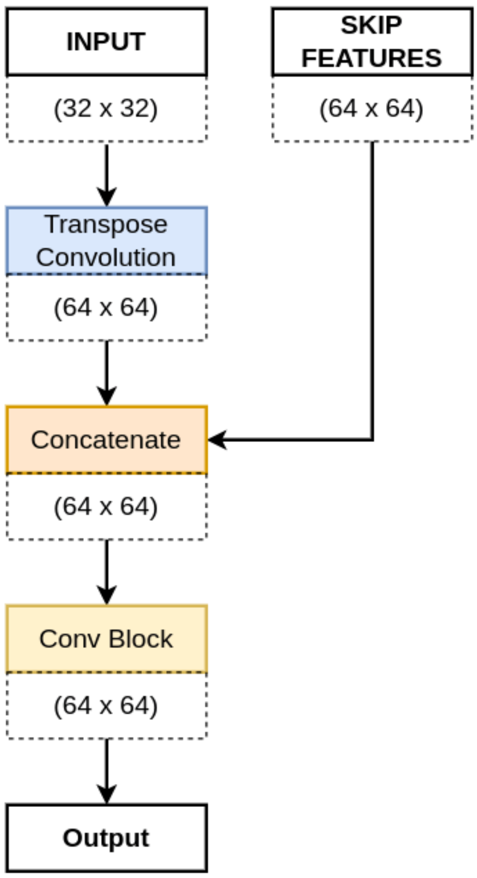


Figuur 4: Vgg16 Architecture

Next, I will extract the necessary feature maps from specific layers of the pre-trained encoder. These features will serve as both the skip connections and the output of the encoder.
I have 4 skip connections from pre-trained encoder corresponding to 4 output layers of vgg16 which are `block1_conv2`, `block2_conv2`, `block3_conv3`, `block4_conv3`.

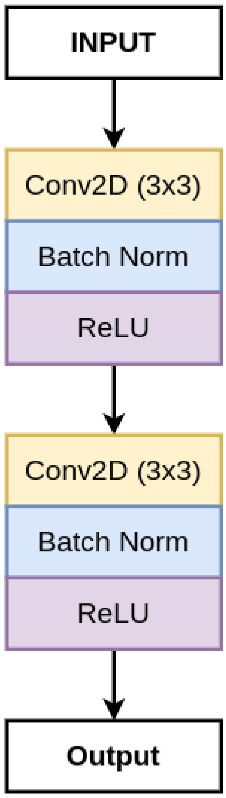
2.3.2 Decoder

The input to the Decoder part is `block5_conv3`. I have built the decoder network consisting of four decoder blocks with the structure of 2×2 Transpose Convolution layer followed by the skip connection taken the VGG16 pre-trained encoder. Next, it is followed by a convolution block.



Figuur 5: Decoder Block Structure

In Convolution block, I have 3 layers which consists of two 3×3 convolution layers. Each convolution layer is followed by a batch normalization layer and a ReLU activation function.



Figuur 6: Convolution Block Structure

2.4 Loss Function

I utilize the Dice loss formulation during the training process for segmenting the fetal head mask and MSE(Mean Squared Error) for calculating Head Circumference.
Dice Loss is a loss function used in image segmentation that measures the similarity between the predicted and ground truth regions, with the goal of maximizing their overlap.

$$DiceLoss(y, p) = 1 - \frac{2yp + 1}{y + p + 1} \tag{1}$$

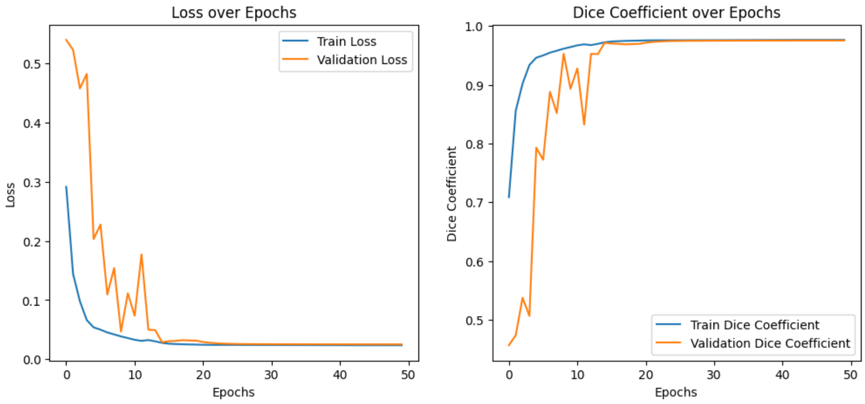
Mean Absolute Error (MAE) is a loss function that measures the average absolute difference between predicted and actual values.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

3 Result

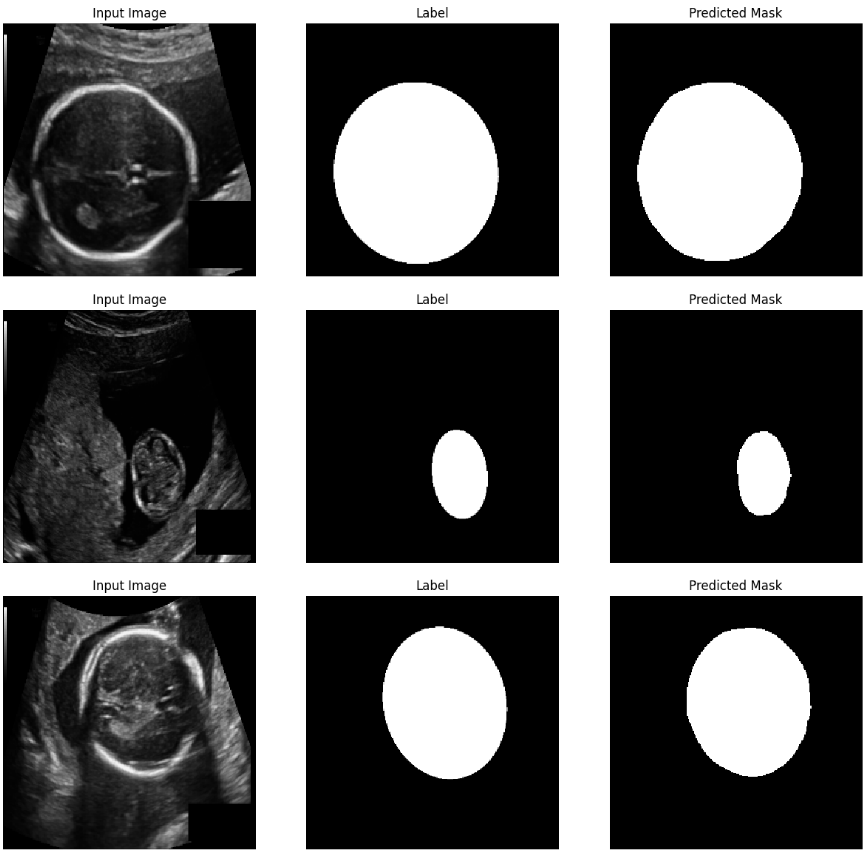
3.1 Segmenation of fetal mask head

After training for 50 epochs with a learning rate of 1e-3, the Dice Loss for both the training and test sets dropped significantly from epoch 13, from 0.5 to 0.03. The model performs well with a Dice coefficient of 0.96 on the training set and also performs excellently on the validation set with a Dice coefficient of 0.95.



Figuur 7: Dice Coefficient and Loss in train val dataset

Here is the result of the model running on the test set as shown in the image below.



Figuur 8: Result with the Image Label Predicted

3.2 Head Circumference

After obtaining the predicted mask images from the model, I applied various image processing techniques, such as identifying the largest contour in the image. Then, I extracted the major and minor axis lengths of the ellipse and multiplied them by the pixel size of each image to calculate the head circumference.
The length of an ellipse denoted HC is approximated by the Ramanujan approximation method. To calculate the perimeter of the ellipse, I used Ramanujan's approximation formula:

$$\mathrm{HC} = \pi \left(3(a+b) - \sqrt{(3a+b)(a+3b)}\right) \tag{3}$$

Dice Coefficient	MAE	PMAE
0.9624	2.7396	1.57%

Tabel 1: Results On Test Set Of Segmentation And Head Circumference