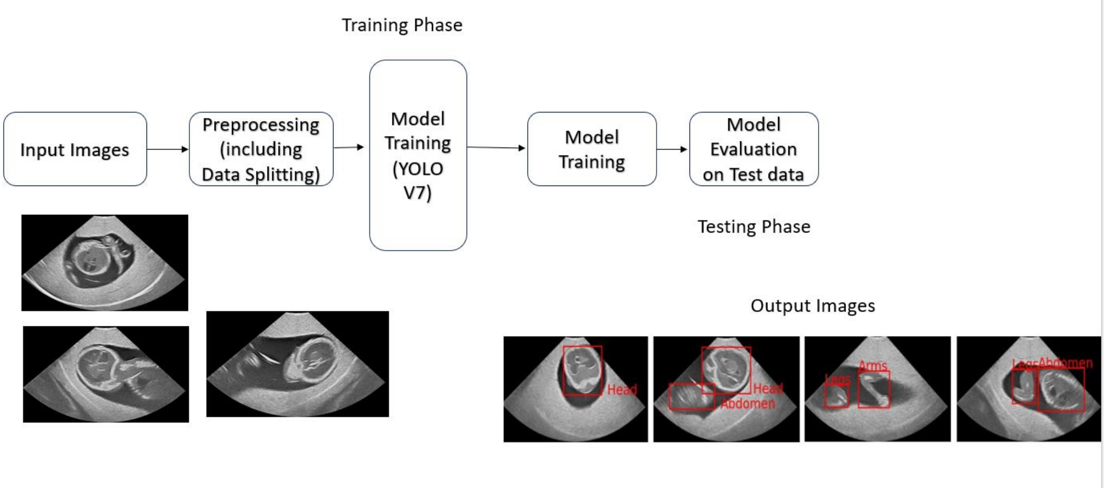
Detection of Anatomical Bounds from Fetus Ultrasound Images using YOLO architecture

Overview of the project:

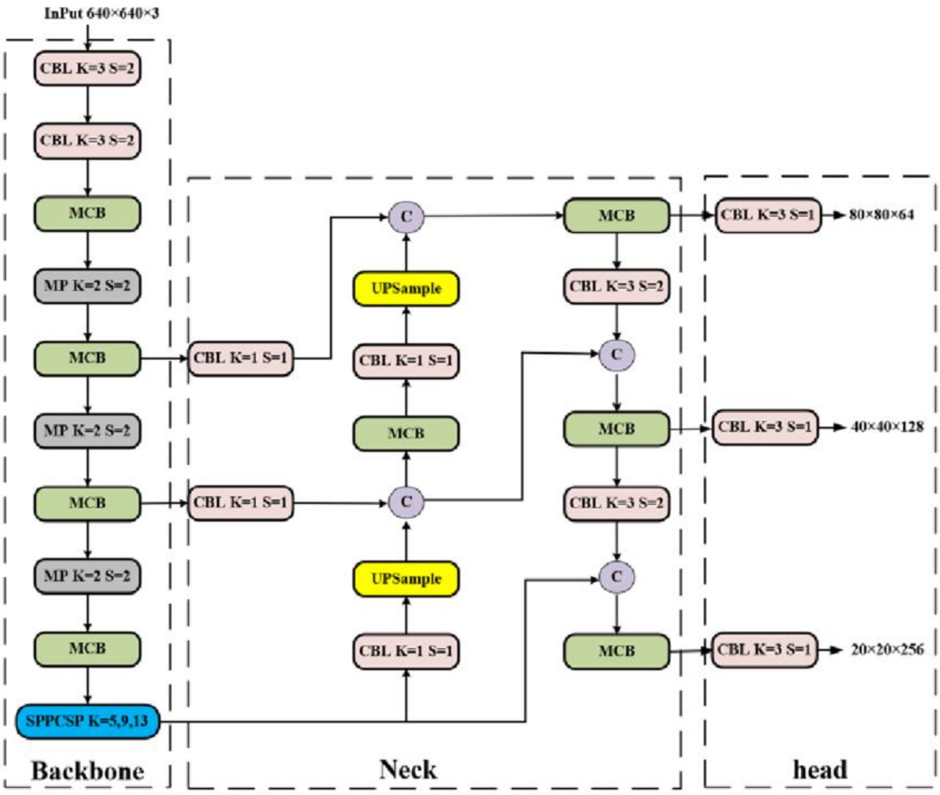
Ultrasound imaging is a key technique used to assess the growth, development, and overall well-being of a fetus throughout pregnancy. However, the complexity and subtlety of these images often require the expertise of specialized medical professionals for accurate interpretation. To enhance this process and explore possibilities for remote, at-home fetal monitoring, this project employs the YOLOv7-tiny model, tailored to identify and outline anatomical features in fetal ultrasound images. Fetus Phantom Ultrasound dataset (FPUS23), which is a comprehensive collection of 9455 images (size: 672*389 ,containing anatomical bounds for 4 classes heads, abdomen, arms, legs)specifically designed for fetal ultrasound analysis is considered as the dataset for the project.



Approach:

The approach involves training the YOLOv7-tiny model from scratch (and also finetuning it using pretrained weights) on FPUS23 dataset, for objection detection of anatomical bounds. Extensive hyperparameter tuning is also performed to optimize model's performance, specific to Ultrasound images.

The architecture consists of lightweight backbone network of convolutional layers to extract hierarchical feature maps at 3 different scales. The model head then fuses these multi-scale features using a bottom-up path with upsampling and lateral connections. This allows objects to be detected at multiple scales by applying custom detection layers to the fused pyramid features. Finally, convolutional layers generate the output detection boxes along with class predictions and bounding box regressions to detect objects in the input image. The architecture is designed to be fast and efficient by using this feature pyramid network with a small, optimized backbone and detection head.



YOLO v7 -tiny model architecture

Additionally, the Yolov7-tiny model architecture is also modified with respect to depth and width of the layers to capture the intricate patterns in the data.

The performance of the enhanced YOLOv7 -tiny model against the baseline results obtained from a Faster R-CNN model, which was developed by the authors of the FPUS23 dataset.

Aspects of the project that are coded independently:

- Preparation of data using XML files and images (annotated using CVAT) into YOLO model training format.
- Preprocessing the data (Tested with various smoothing and noise removal filters such as gaussian filter, median filter and selected median filter based on performance)
- In modifying the YOLOv7-tiny architecture, specific changes were made to the depth and width multipliers of the model. The **depth_multiple** was set to 1.5, enhancing the model's depth, which allows it to learn a more complex hierarchy of features. Similarly, the **width_multiple** was also adjusted to 1.5, increasing the number of channels in each layer, which enables the model to process a wider array of information at once, to detect and differentiate between various anatomical structures within the images. The selection of a 1.5 multiplier for both depth and width is decided by the need to strike a balance between increased model complexity and computational efficiency.
- Hyperparameter tuning – As a part of this various parameters such as hsv, mixup and mosaic augmentation, momentum, translation, and learning rate were adjusted to check their impact on the performance.
- The experiments related to training of the YOLO model on FPUS23 is not yet explored by data science community.

Aspects of the algorithm used from online resources:

1. YOLOv7-tiny model architecture and YOLO model training and testing pipeline https://github.com/WongKinYiu/yolov7

Experimental Protocol:

Data Preparation: This involves converting the images and their corresponding annotations(marked using Computer Vision Annotation Tool – CVAT), which were initially in XML files, into a format compatible with the YOLO model, which usually include the coordinates of the centre of the bounding box, the width and height of the box, and the class label.

Data Preprocessing: As part of this, several filtering techniques were evaluated to enhance the quality of the images in the FPUS23 dataset. These included Averaging, Gaussian Blurring, Median Blurring, and Bilateral Blurring. After thorough testing and comparison of these methods, the Median filter was chosen as the most effective for preprocessing and these images are used for few sets of experiments. On the top pf this, several other preprocessing steps are handled by YOLO model based on the parameters set.

The dataset is then split into train, validation and test sets in 70:15:15 ratio for training.

A diverse range of experiments were carried out to thoroughly evaluate the performance of the YOLO model on the FPUS23 dataset, with different training approaches and configurations:

- **Using Pretrained Weights:** The objective was to assess how well the model could adapt its pre-learned features to ultrasound images. This model is trained for 77 epochs.
- **Training From Scratch**: To observe how the model learns and adapts when it is exclusively exposed to the FPUS23 dataset from the beginning of its training. This model is trained for 213 epochs
- **Training From Scratch with Modified Hyperparameters and Preprocessed Images with median blur:** The HSV image augmentation parameter was set to a value of 0, while other parameters like mixup and mosaic augmentation, momentum, translation, and learning rate were maintained at their original values, as adjustments to these parameters resulted in a decrease in performance. This model is trained for 75 epochs.
- **Using Pretrained Weights with Modified Hyperparameters and Preprocessed Images:** The above mentioned approach was also applied while utilizing pretrained weights.This model is trained for 109 epochs

For all the above training approaches, Mean Average Precision is considered as evaluation metric in addition with Precision and Recall.

Data Source: Ultrasound Fetus Phantom Dataset (https://arxiv.org/pdf/2303.07852.pdf, https://github.com/bharathprabakaran/FPUS23)

Scale of Data: 9,455 labelled anatomical bounds(Train- 6618, Test-1418, Validation -1418)

Coding Resource Requirements:

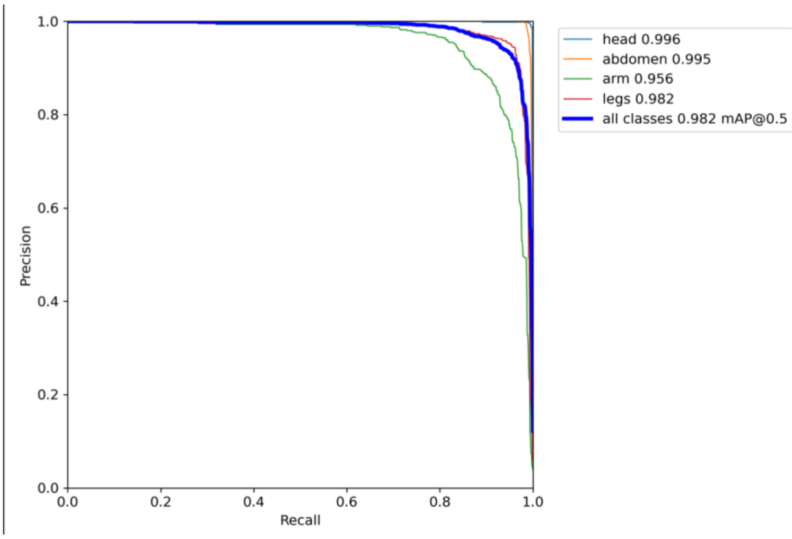
Libraries: PyTorch, OpenCV, Tensorflow, XMLtree

Success Definition: Accurate identification and demarcation of anatomical bounds in the ultrasound images and the model mAP should outperform the baseline model’s mAP

Results :

Using Pretrained Weights:

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	1419	3035	0.971	0.944	0.982	0.775
head	1419	663	0.996	0.994	0.996	0.872
abdomen	1419	972	0.995	0.986	0.995	0.843
arm	1419	740	0.936	0.851	0.956	0.667
legs	1419	660	0.957	0.944	0.982	0.718



PR Curve

The model achieves a high precision (P) across all classes, with the head detection being particularly high at 0.996.

Training From Scratch with default hyperparameters:

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	1419	3035	0.965	0.975	0.99	0.804
head	1419	663	0.99	0.998	0.997	0.89
abdomen	1419	972	0.99	0.992	0.997	0.862
arm	1419	740	0.933	0.934	0.977	0.713
legs	1419	660	0.948	0.976	0.99	0.749

Training From Scratch with Modified Hyperparameters and Preprocessed Images with median blur:

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95:
all	1419	3035	0.959	0.948	0.983	0.762
head	1419	663	0.989	0.976	0.996	0.866
abdomen	1419	972	0.99	0.988	0.995	0.831
arm	1419	740	0.921	0.881	0.96	0.655
legs	1419	660	0.935	0.945	0.981	0.697

Using Pretrained Weights with Modified Hyperparameters and Preprocessed Images with median blur:

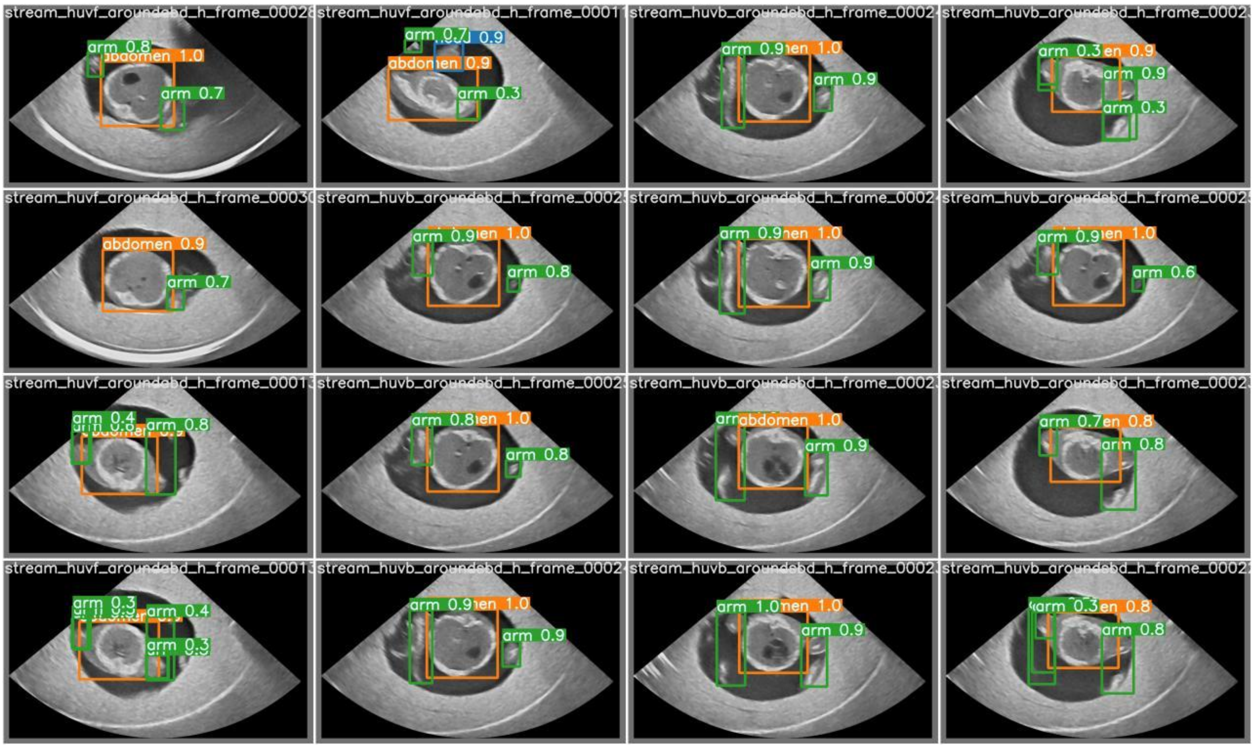
Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
all	1419	3035	0.963	0.96	0.987	0.785
head	1419	663	0.985	0.99	0.997	0.877
abdomen	1419	972	0.992	0.99	0.997	0.849
arm	1419	740	0.925	0.905	0.969	0.692
legs	1419	660	0.949	0.955	0.983	0.722

The results of the above experiments with respect to evaluation metric mAP seem to be same without any significant difference. However, with median blur preprocessing and hyperparameter change in effect, the training time gets significantly reduced with the model reaching its optimal performance more rapidly at around 31st epoch

```
  Epoch    gpu_mem       box       obj       cls     total    labels  img_size
 31/299        14G   0.03672   0.01059  0.008541   0.05585       104       672: 100% 104/10
             Class     Images     Labels         P         R     mAP@.5  mAP@.5:.95:
               all       1419       3035     0.937     0.909     0.961        0.66
```

whereas, it took around 70 epochs for the model which is run from scratch with the images which are not preprocessed and hyperparameter changes are not done.

The model trained from Scratch with default hyperparameters can be considered as best model based on the evaluation metric mAP.



Results from the best model on few test images is provided above.

Analysis:

When these results are compared with baseline results, with model from the paper by the authors of FPUS23 , with Faster RCNN and ResNet 34 as backbone, the model trained using YOLOv7 -tiny seems to performs better with mAP of 0.99

mAP	98.60%
mAR	84.40%
No. of Flops	64.40G
Memory (MB)	357.20

Anatomy Bounds - Faster-RCNN with ResNet34 backbone

Discussion and Lessons Learned:

- Some of the models are not trained for longer epochs because of resources constraints. But the training is done till proper accuracies are obtained.
- Given this, all the different models will outperform the baseline model when trained for longer epochs.
- The YOLOv7 seems to better because of its speed during inference and it can be easily integrated with Ultrasound scanner for faster processing of images and for instant results. The models can be tested on real world dataset from hospitals to check its effectiveness and more

advanced YOLO models such as YOLO-NAS and objection detection algorithms such as Detectron can be tested for their effectiveness as continuing research.

- The backbone of the YOLO model can be changed and can be tested as a future research item ,but this might be complex as it involves changing the YOLO model code as a whole completely.

Bibliography:

1. https://github.com/bharathprabakaran/FPUS23
2. "**FPUS23: An Ultrasound Fetus Phantom Dataset with Deep Neural Network Evaluations for Fetus Orientations, Fetal Planes, and Anatomical Features**" Bharath Srinivas Prabakaran, Paul Hamelmann, Erik Ostrowski, Muhammad Shafique. https://arxiv.org/abs/2303.07852
3. "**YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors**." Chien-Yao Wang, Alexey Bochkovskiy, Hong-Yuan Mark Liao. https://arxiv.org/abs/2207.02696
4. https://github.com/WongKinYiu/yolov7?tab=readme-ov-file

*All the plots and results related to corresponding experiments are attached with the project