Ultrasound Sound Imaging with YOLO

Name: Sai Asish Samineni **PSID:** 2088595 **e-mail:** ssaminen@cougarnet.uh.edu
Name: Sammy Tawakkol **PSID:** 1665545 **e-mail:** stawakko@cougarnet.uh.edu
Name: Jaya Lakshmi Nandini Mangipudi **PSID:** 2080214 **e-mail:** jmangipu@cougarnet.uh.edu

Abstract:
One of the most used imaging modalities for medical diagnostics is ultrasound. With its distinctive benefits, deep learning has recently had considerable success in computer vision. Deep learning has also been utilized for its enormous potential around medical US image analysis, and more and more researchers are using it for CAD systems. One of the most used imaging modalities for medical diagnostics is ultrasound (US). Real-time, cheap cost, non-intrusive nature, and ease of operation are its benefits. However, it also has several distinct drawbacks, such as severe artifacts and noise and a great reliance on the medical professional's experience. Numerous computer-aided diagnostic (CAD) systems have been created to address the limitations of ultrasound diagnosis and aid doctors in increasing the precision and effectiveness of diagnosis. High-energy pulses are used by medical ultrasound scanners to scan the human body. The radiation force caused by the impact of such pulses on an object can cause the object to vibrate, creating a focused, loud sound that is audible. Here we are using the YOLO dataset.

Introduction:
Ultrasound imaging is the most common practice in medicine. It is used to perform cross-sectional diagnostics. It is most widely used because of its portability and real-time display. Comparing Ultrasound to other technologies, ultrasound [2] gives us the benefit of low cost and convenience. Ultrasound also has its fair share of disadvantages such as artifacts and Noise.

Ultrasound imaging is a clinical technique, which utilizes high-recurrence sound waves to create dynamic visual pictures inside the body. Ultrasound imaging works in Realtime and can help the assessment, determination, and treatment of illnesses in various circumstances. The advantages we have for the ultrasonic sound is

- Safer than X-ray and CT scan
- Painless procedure
- Widely used technology
- Can be implemented to various methods

The fundamental task of the Computer Vision task is object detection. Localizing, identifying and segmentation are key areas in object detection. The traditional computer vision for object detection uses a slightly different approach. Their assumptions are different. Traditional methods are used to consider thresholding, subtraction, gradient, or shape-based features. YOLO is a popular data set that has proven many uses in the real world. More importantly, this one predicts the bounding boxes and probabilities from the image in one round. Due to this reason, the detection of the baby in ultrasound imaging is quicker and helps in guiding the doctors.

Literature survey:
In this paper, they were able to gesture the database of the HMI based on the B-mode ultrasound. In that paper, they achieved the desired accuracy and showed a good amount of database collection.

This paper has concluded that YOLO can be trained on the full images, The loss functions have been used to train them, unlike the classifier-based models. They were able to push the state of the art in real-time. In the paper, they got the dataset and trained the B-node neural network to reconstruct according to the echogenicity of the medium, which has the other name brightness mode. Object detection in medical imaging has been considered as the segmentation problem, The pixel-wise annotations are used to train and identify the object. Most advances include the DL approach, unlike the segmentation-based approach. YOLOv7 has outclassed all the existing object detectors in speed and accuracy ranging from 5 fps to 160 fps with the best accuracy.

Roboflow is a cloud-based workflow management system. The roboflow provides the pipelines. Usually, most pipelines are process-oriented but the roboflow is a little different as it uses the data as the metric. Data processing, algorithmic development, back testing, and application adaption are unified under one framework. Due to this the maintainability and reusability parallel is possible.

Methods:

We would like to implement the YOLO for the detection of the Ultrasound image. The Yolo will be used to get the weights and labels as this library will help in detecting the baby. Instead of using the neural network which will take a little more time to process. Tkinter will be used to create the graphical interface where we can upload the photos and get the results.

Training the dataset is one of the crucial factors in any machine-learning model. Training the custom data set is one of the difficult tasks. Training the model involves many steps.

- Install YOLOv7 dependencies
- Load custom dataset from Roboflow in YOLOv7 format
- Run YOLOv7 training
- Evaluate YOLOv7 performance
- Run YOLOv7 inference on test images

Initially, we will clone and download the requirements for the YOLO v7 by WongKinYiu. The next step would be to export our custom data set which is a custom YAML file. Once we are done with the exporting we will begin our training. The epochs have been modified to best fit our dataset and get the best accuracy possible. Once the training is done, we are left with the testing phase, this step will help in deciding whether the model is good enough or not.

Fig: Output after the evaluation

Implementation:

The YOLO family has been growing at a quick pace since the YOLO v5. The Implementations start with the installation of the YOLOv5 environment. In our code, we cloned the YOLOv7 repository [1] by WongKinYiu. After installing the environment, we labeled the unlabelled data. Then defining the model configuration and architecture.

Downloading the custom-formatted data is an essential part of the algorithm. The improper dataset will cause the model to fail, and some might need more work in the future which isn't an ideal solution for the project.

Here is the sample code we used to custom download the dataset.

```
# REPLACE with your custom code snippet generated above

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="N2GBxqH6yLH28EowHWIS")
project = rf.workspace("ultrasoundbabydataset").project("ultrasound-baby-project")
dataset = project.version(3).download("yolov7")
```

The Epochs and batch have been adjusted to get the maximum result for the algorithm. Detect.py code has been used to evaluate and re-parameterization. This will complete the process and we can obtain the desired results.

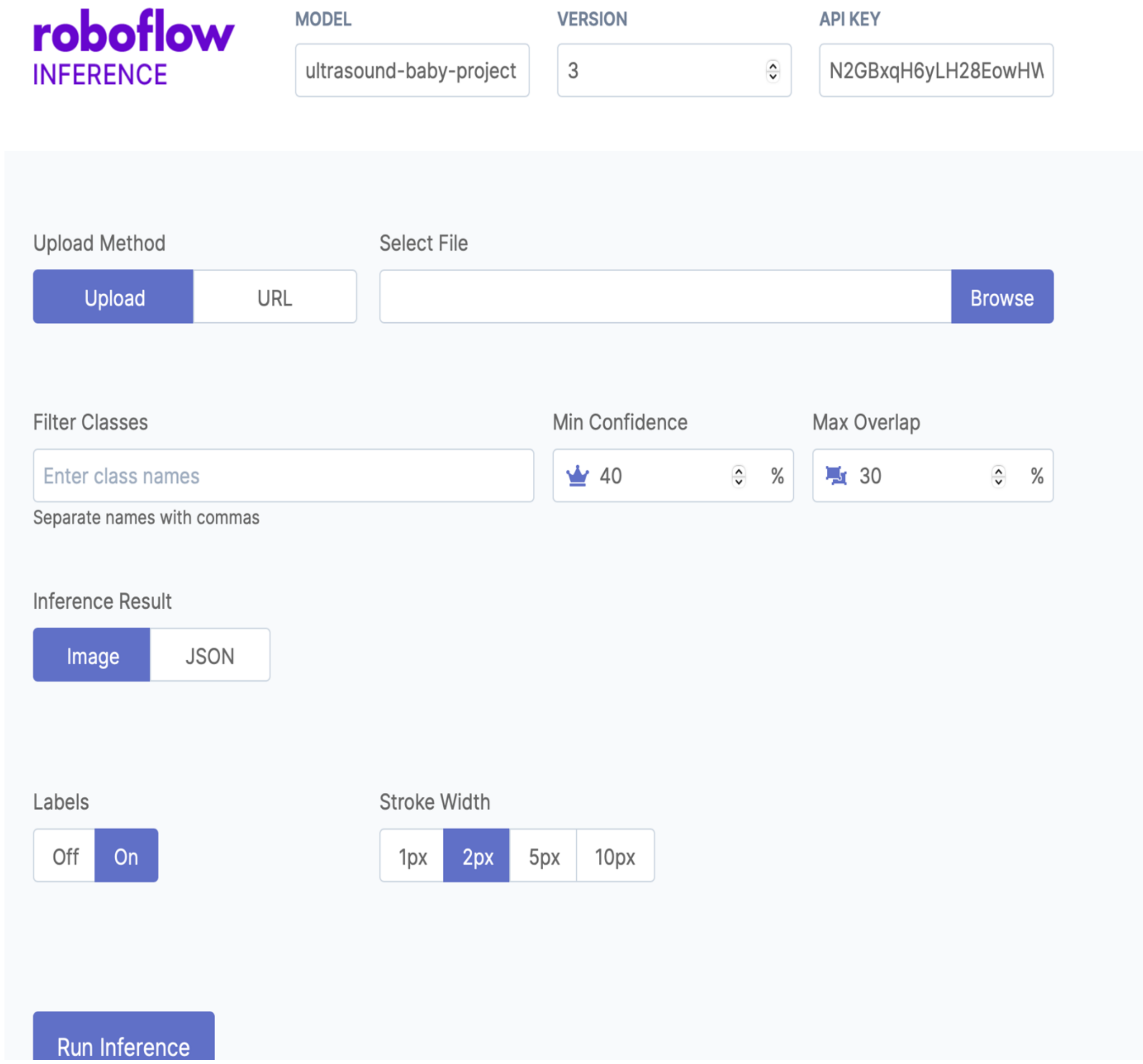
The main form of implementation came through using RoboFlow, an online website specifically tailored to creating, training, and annotating datasets or models. Sammy previously worked with implementing YOLOv5 on a different project but could not find a good dataset containing just feral images in an ultrasound. Not only did Roboflow handle storing the resulting model and weights in their database, but also allowed him to annotate images he collected to be used in the model. The code snippet above is the access key for that specific repository, allowing for everything to be downloaded into the notebook at once. After training, it shows in the Roboflow portal that one can now test the model via detection on a separate link. https://detect.roboflow.com/?model=ultrasoundbabyproject&version=3&api_key=N2GBxqH6yLH28EowHWIS

This is all done through a Notebook provided by Roboflow, which streamlines the process, as YOLO generally has issues with every update or package (CUDA, device settings, computer-related issues)

To use, follow the link to access the online GUI. Proceed to type 'baby' in the class name filter, and adjust confidence/overlap accordingly (set it low, as the model is not perfect). Pick any image, preferably one with a baby in ultrasound, and upload. With run inference, you'll be able to see if it is detected. Most confidence levels of detection are very low (0-0.15).

To train and run the model yourself, run the notebook attached. You might receive a prompt to “restart the runtime” to fully update the libraries and train the model. Once finished, you can return to the GUI and try to do image detection

Here is the screenshot of the interface.

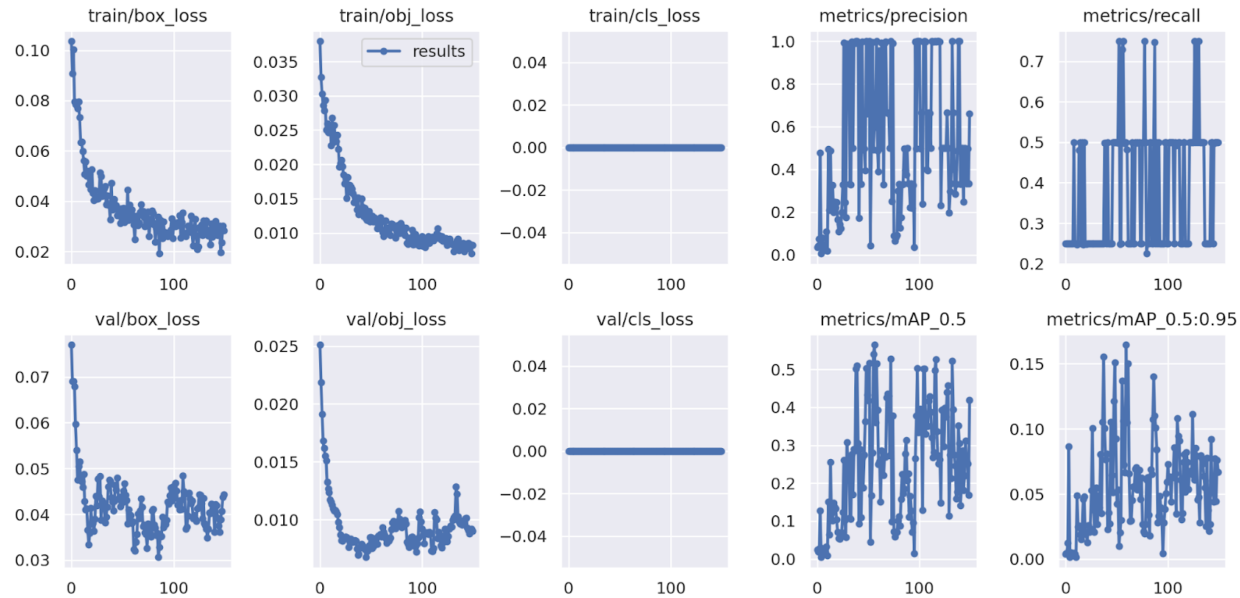


Results:

The model can detect images but at a very low confidence rate. This is unfortunate. However, there are some reasons for this:

There was no proper dataset on feral planes specifically showcasing the baby. Most had to be scraped from Google Images and annotated for YOLOv7 use by ourselves. Furthermore, with the low amount of images, there were only 130-150 in total after preprocessing and augmentation, which was not fully guaranteed for feature detection.

The model's training results came with 56.5% mAP, 59.8% precision, and 75% recall.



Conclusion:

Our model was able to detect the baby. Unfortunately, our model falls short in terms of confidence in the various categories. We believe the model has a lot of capabilities with further improvement in the model, training, and detection the model will reach its dependable state.

References:

[1]https://github.com/WongKinYiu/yolov7
[2]https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9395635
[3]Divvala,Santhosh., Redmon,Joseph.(2016,june, 26).You Only Look Once: Unified, Real-Time [4]Object Detection.https://ieeexplore.ieee.org/document/7780460
[5]Xia,Wei.,WeiYe,Lin.,Liu,Hong-Hai.(2017,July,9).Agesture database of B-mode ultrasound-based human-machine interface. https://ieeexplore.ieee.org/document/8107752
[6]Brattain, Laura.,Brian, Telfer.,(2019,April,01).Machine Learning for Medical Ultrasound: Status, Methods, and Future Opportunities. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5886811/
[7]https://proceedings.mlr.press/v164/lin22c.html
[8]https://ieeexplore.ieee.org/document/9050094

GitHub Link: https://github.com/saiasishsamineni/MedicalImaging