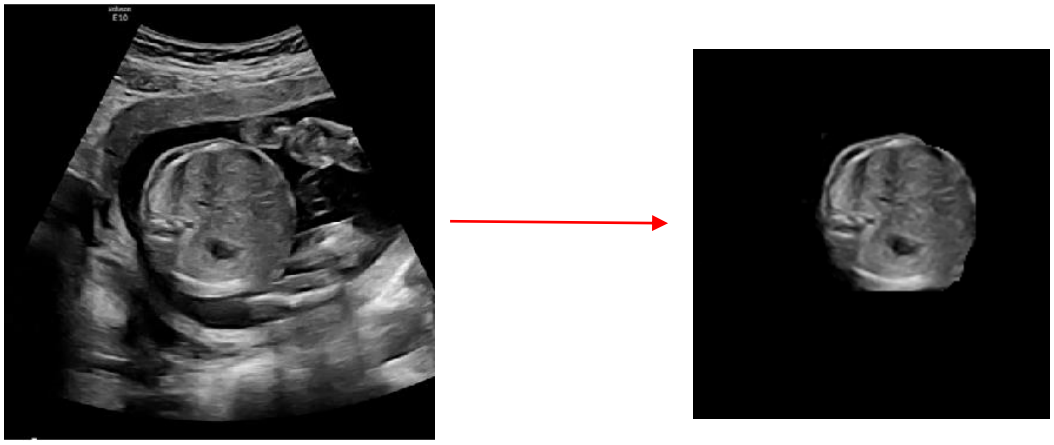
Solution: The idea is to build a deep learning model for classification (0,1,2, 3...) which is the grading of the bowel echogenicity. The given images contain a lot of areas and regions other than the bowel. Those areas are uncorrelated for the grading, so they might make it hard for the training to learn the grade or might affect the learning (i.e. the grading might be decided based on those uncorrelated areas). So to prevent this I suggest some preprocessing to prepare input data which contain the most the region of interest which is the bowl.

Here is the flow of preprocessing:

1. Let's first get rid of the biggest uncorrelated region which is everything outside of the abdomen body. i.e. developing an algorithm that take the given image as input and output an image contains the abdomen and zero outside the abdomen. For example:

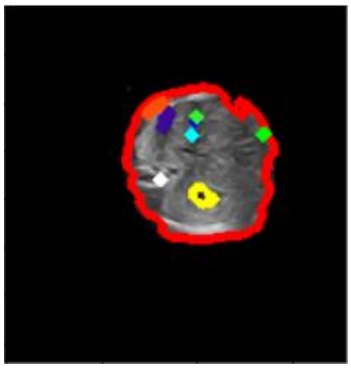


Input output

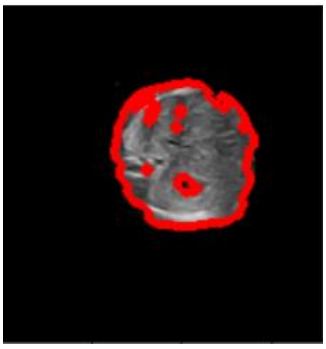
To implement this algorithm we can use a pre-trained network to identify the abdomen as here: https://github.com/junqiangchen/HC18-Automated-measurement-of-fetal-head-circumference. Or we can build and train a neural network on an available labeled open source datasets we can find it here: https://zenodo.org/record/1327317#.XtJJFWgzbNw. Because the low capacity of internet and devices which I have meantime, I implemented a neural network using fast.ai trained on the labeled data in the link. And I believe that training in the data available in the lab can give much better performances. After I got the mask (the segmentation) on the abdomen area I give zero in the original image for pixels outside of the mask. Applying this I get the output in the above example.

2. After I got the output in step 1 I extract the contours of the image using opencv2 library in python. I take the contour with the largest area size, because this contour will contain the bowel and ignore the stomach and the bones because they are other contours with smaller area size. I transform this counter to a patch and such patch will

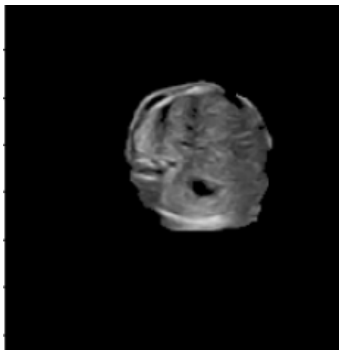
be the input for grading classification neural network model. This is the contours of the output image in the example in step 1:



And the Contour with largest area will be in**: (everything in shape that marked red) and we take all the pixels inside we get *:



**



*

I would try to build a deep learning classification model taking as inputs the images of *. If later I will see the results of the training still not good. I would do more another step of processing on *. I will take the pixel inside the contour in ** put them in a patch or an array and use this for training input then see if I get better results.

Another solution (in this step) that require less clinical efforts to train I would suggest is to look at the histogram of *. And give the grade according to the distribution of the histogram. Either by building a function that give grade zero if most of the pixels falls around zero in the histogram and 6 if the most pixels fall around 255, and a grade between 0 to 6 if the distribution almost uniform distributed (this can be determined by looking at the mean and std of the distribution). Or by building and training a small classification machine learning model that take the histogram the input and output the grade.

Note: If we have labeled data for the stomach and the bones as in the included example in the question. I then can train in step 1 adding segmentation also for the bones and the stomach, have their masks and use them to remove those areas as I did for the removing everything outside the abdomen. This also will give us better data to train on.