Project 3-1
Babywatcher Ultrasound Interpretation

Zain Farhah, Emery Karambiri, Liam Thomassen, Boris Borisov
Department of Data Science and Knowledge Engineering
Maastricht University
Maastricht, The Netherlands

***Abstract*—Pregnant mothers often go to ultrasound imaging for multiple reasons. One reason is to monitor the developing baby's health. Unfortunately, the ultrasound images are not self- explanatory and often the mothers need guidance by a medical expert to understand what they are seeing. In this project, an artificial intelligence program, which can recognize the body parts of a fetus, was implemented. The approaches used were the Histogram of Gradients (HOG) approach using Support Vector Machines (SVM) to classify the HOG features and a YOLO model. The SVM achieved an accuracy of 76.18% and the YOLO model achieved an accuracy of 97.99%. This study has shown that the YOLO model can achieve better results and should be used by parents to help them understand an ultrasound image.**

I. Introduction

According to the UN, 385 000 babies are born each day which leads to the fact that a lot of women are pregnant and most of these women want to have a great pregnancy experience. This experience includes visits to the doctor for the purpose of checking on the baby's health and even seeing the fetus using the ultrasound machine. Babywatcher offers scanners, that can be rented by parents, to perform their own ultrasound scans in the comforts of their own home, however since there are no medical experts present during these scans, interpreting these images can be challenging.
The main problem that this paper aims to answer is:
Is there a method that can effectively help a mother understand the ultrasound image of her baby?
Note that, understanding an image would mean to be able to recognize where the body parts of a baby are, if there exists any in an image.
Creating an image recognition AI to assist these parents in identifying regions of interest would ease the use of these scanners, however, this brings several challenges. As the final product is developed, problems are encountered and so the other research questions are:

- **How reliable is the labelled data given that it is labelled by the group members, who have no medical experience, and are not medical experts?**
- **How does noise and increased contrast affect the performance of each model? Can they be improved?**
- **Which model performs best in terms of precision, recall and F1 score?**

II. Related Work

The ultrasound image is the first time that the parents see their child. Black, Rita Beck et al. [1] stated that ultrasound technology allows the woman and her partner to "see" inside the womb. In that act of seeing, the fetus becomes a baby. The parents take away from the ultrasound a mental image built both from their memory of the machine-generated picture and their imagination. Ji,E. et al. [2] includes several questionnaires and interviews of women, which describes whether there is a difference between 2D and 3D Ultrasound screening. What is more, Rustico, M.A. et al. [3] explained that there is a 4D ultrasound in pregnancy and proves that the facial expressions and hand-to-mouth movements were twice as likely to be seen with 4D ultrasound rather than 2D ultrasound scanning. Overall, the study indicates that the addition of 4D ultrasound does not change significantly the perception that women have of their baby nor their antenatal emotional attachment compared with conventional 2D ultrasound.

A. Object Detection

The world is in a great need for accurate object detection algorithms as they would open a gateway to very advanced technology like reliable and safe self driving cars, and more responsive and human assisting devices. Due to this growing need, various techniques have been developed. Some approaches like, Discriminatively Trained Part Based Models, use the sliding window mechanism in which a window of a specific size is run over the whole image and then for each window, the detector is run to check if an object exists [4]. Other approaches like, R-CNN, first propose bounding boxes and then run the detectors on these boxes to check if an object exists within them [5]. The R-CNN belongs to the group of object detectors classified as two stage detectors which prioritize the accuracy of their final outcomes. On the other hand, there are one stage detectors like the YOLO object detector that prioritize inference speed but the outcomes are of good accuracy too. YOLO treats the object detection problem just like a regression problem in which a single neural network predicts bounding boxes and class probabilities from full images in one evaluation [6].

B. Object Detection on Ultrasound images

A novel method of ultrasound image captioning generation based on region detection was proposed in [7]. It works by simultaneously detecting and then encoding the focus areas of an ultrasound image. Then it utilizes an Long short-term memory (LSTM) RCNN to decode the encoding vectors. The method aims to generate annotation text information to describe the diseases content information in ultrasound images.

III. Methods

A. Data

For the purpose of this research, around 50,000 unlabelled images (fig:1) were provided. These ultrasound images were taken by parents who used the BabyWatcher scanner at home. Due to this fact some images contain noise or did not contain any valuable information to interpret i.e. an image taken by mistake or a very blurry image. Furthermore, parents are able to increase the contrast and zoom into the images which can cause interpretability to decrease. This meant that the team of students had to manually go through the images and select only the meaningful images. Unfortunately, this lead to the second issue which is the reliability of the labels produced.

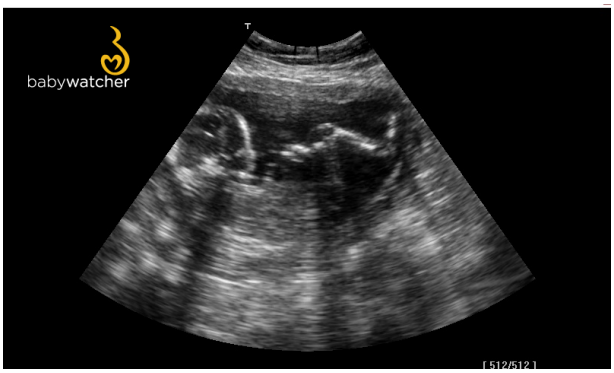


Fig. 1. Example of an ultrasound scan containing the fetus head and leg

1) Labelling the data: The data that was provided is a set of images each with a size of 800xK. As mentioned before, a team of four students had to label the data manually. Two different formats had to be produced, one for each model to be trained. The SVM needed a csv file with the attributes:

- Label
- X coordinate of the bounding box's center
- Y coordinate of the bounding box's center
- Width of the bounding box
- Height of the bounding box
- Name of the image inspected
- Width of image
- Height of image

The YOLO model needed .txt files for each image to be inspected. The .txt files included:

- Label
- (X coordinate of the bounding box's center) / (Width of the image)
- (Y coordinate of the bounding box's center) / (Height of the image)
- (Width of the bounding box) / (Width of image)
- (Height of the bounding box) / (Height of image)

The labelling was done using a free open source online labelling tool called makesense.ai [8].

2) Measuring the reliability of the data: As mentioned above, the data was labeled by students who have no medical background. Due to this issue, the reliability of the data used in this research can be questioned. In addition to that, even medical experts can sometimes disagree on some things. To measure the reliability of the data, the inter-rater reliability test was used. Inter-rater reliability is a measure of the agreement among different raters or annotators that are assessing the same data. The values range from 0 to 1, where 1 is perfect agreement within the group of raters and 0 is indistinguishable from a random baseline. The main assumption is that the more raters agree on a given rating or label, the higher the chance that it is correct.

B. Histogram of Gradients approach

1) Overview: This approach utilizes the Histogram of Gradients to extract feature descriptors from subsections of the image. Once we have these feature descriptors we need to classify them. To do this we'd need a classifier with high bias and low variance because we have a high number of features (e.g. 3600 for an image of size 128x128). This description fits with the Support Vector Machine.

The images are split up into smaller grids (typically 8 by 8) called cells. The gradient directions and gradient magnitudes are calculated for each of these cells. Once these are calculated, they magnitudes are split into bins (typically 9) based on what their direction is. This leaves us with with a vector of size 9 for each 8 by 8 block. Then we create blocks usually consisting of 2 by 2 cells, sum the vectors of cells that make up this block and normalize this summed up vector. It does this for every 2 by 2 cell block over the whole image, allowing for overlap. All the normalized vectors obtained are then concatenated together to create the HOG features of the given image. [16]

Considering we have an unbalanced data (1897 heads, 528 spines, 79 hands, 78 arms, 19 legs and 12 feet annotated in full data set), using multiply single class SVMs would be optimal, therefore we trained one SVM for each body part. The training is done by resizing all annotated bounding boxes to the same size (this is done to keep the same amount of HOG features) and computing the HOG features for each annotated body part. Then for each body part x, an SVM is trained using the ones annotated x as positive and all the rest as negative.

This approach has five main steps:

1) **Image processing** First, we aim to reduce the size of the image by cropping out parts of the image that have no information, meaning the areas with only empty pixels. Given the database these areas would be the sides. Then we process the image as an attempt to reduce the noise using a Gaussian blur filter and final aim to enhance the edges by emphasizing differences in pixel intensities by squaring the image and interpolate the pixel intensities back to the valid range of [0, 255].
2) **Sliding window**
   Then we need to extract subsections of different sizes over the whole image. To do this we slide a window of a given size over the image, left to right, top to bottom. To achieve the different window sizes we scale the image, this technique is called pyramid representation [15]. All the windows are square to accommodate for long objects without needing to use more window sizes in different orientations. The sizes of the windows were selected based on the sizes of the objects that we want to analyse. Smaller body parts such as hands, feet, arms and legs can typically be found in the range [32, 100]. Whereas larger body parts such as heads, spines and legs can be found in by window sizes in the range [55 to 200]. The window sizes we decided to use were 32, 64, 96, 128, 160, 196 and 224 because they fit the body part size ranges. Using the windows we know *where* the object is.
3) **HOG**
   Then resize each window to the same size and we compute the HOG features for each window. The resizing is done so that we get the same number of HOG features for each window because otherwise the SVM will not accept the input.
4) **Classify**
   Next we pass the computed HOG features through each SVM to classify and classify the window from which the HOG features come from with the highest probability classification. The SVMs tell *what* the object is.
5) **Post-Classification pruning**
   And finally, once we have the windows that were classified as potentially having a body part we try to find the ones that are most likely to be correct. To do this we first drop those whose prediction probability falls below a certain threshold (we use 0.9). And for each window left, we get the how many times it's overlapped by another window classified the same and by how much and sum these together. We then modify the predicted probability for windows of classes that appear multiple times

using the formula:

$$w_{i,newprob} = w_{i,oldprob} + w_{i,overlap} * \frac{1 - w_{i,oldprob}}{n_{w_{i,class}}}$$

where $w_{i,overlap}$ is the sum of the overlap computed for window i, $w_{i,oldprob}$ is the probability predicted by the SVM for window i, $w_{i,newprob}$ is the modified predicted probability for window i and $n_{w_{i,class}}$ is the amount of windows found that are the same class as w_i We final select the one with the highest predicted probability for each class.

The algorithm is relatively slow and not applicable on video due to the amount of windows that get generated. Increasing the step size, reducing the amount of windows generated, would speed up the algorithm considering most of the time is spent on calculating the HOGs, however, this would affect the coverage of the image,thus reducing possible hits.

C. YOLO approach

1) Overview: YOLO is an object detection algorithm that uses neural networks to provide real time object detection. The main questions YOLO tries to find answers to is 'What is the object?' and 'Where is the object?'. It uses convolutional neural networks to detect the objects. The final output of the algorithm is the class probabilities of the detected image. YOLO was one of the chosen techniques to solve the main problem of this paper for 3 main reasons:

- **Accuracy**
  YOLO has proved that it could give results that have a high accuracy with very low errors. This is important when detecting the fetus's body parts to prevent the confusion of the parent. YOLO was compared to the fastest and most accurate detectors in [10] and it proved to be superior to them in terms of speed and accuracy.
- **Speed**
  YOLO provides real time object detection and this is an important factor for an eager parent waiting to understand the ultrasound image. It can process 45 frames per second [11].
- **Ability to Learn**
  YOLO has great learning capabilities and this is needed when trying to find the body parts in a fetus's ultrasound as the child in the womb can be in many different orientations. It outperforms the other top detection models because it can learn the generalizable representations of objects [14]. The algorithm used will have to label a body part regardless of the orientation of the image or the child.

The main techniques that makes YOLO a good algorithm [13] for the application are :

- Residual Box: YOLO does not search for regions of interest in the input image, instead the input image is divided into various grids. Each grid is of size SxS and every grid cell is responsible to detect the objects that appear in them [12].
- Bounding Box Regression: Yolo uses this technique once to find the width, height, center and the class of a detected bounding box.
- Intersection Over Union: This technique measures how much of the predicted box is actually bounding the object to be detected by comparing it to the real box.
- Non Maximal Suppression: This ensures that bounding boxes with low class probabilities are eliminated.

D. User Interface

The User Interface was implemented using the Flask framework. This specific framework was used due as the back-end Python models had to be connected to HTML and Flask made this feasible. In addition to that, the aim was to create a user interface that would make it easy for a parent to navigate through the website.

The UI contains an instructions panel that describes the steps the user needs to go through in order to get an analysis of the target image. The steps are as follows:

1) Choose an image that you want to analyze
2) Click on the analyze button and wait for the result

If the image actually has body parts of a fetus, the result will be the same image but with bounding boxes around the detected parts. The boxes will have a label together with a confidence score. If the algorithms cannot detect anything, then the user will have to retry again with another image.

IV. Experiments

A. Performance of approaches

The experiments setup is as follows:

- Training set: Both approaches were trained on two data sets, one with 1000 images and the other with 2000 images. The data sets were all annotated by the group members and no medical experts. This means that the results of the experiments depend greatly on the reliability of the labels produced.
- Test set: The test set used for both approaches was made up of 300 images. Using these images, different sets were produced:
  - Original test set
  - Images with Gaussian Noise
  - Increased contrast x2
  - Increased contrast x5
  - Increased contrast x10

  The aim of experimenting on images with Gaussian noise is to see how the approaches will perform on noisy ultrasounds. The noise in ultrasounds is usually white specks that occur randomly in the image. In addition to that, the different contrasts were also included because in, the BabyWatcher product, the user can alter the contrast of an image and so it is important that the final approach selected does not really get affected by the varying brightness of an image.
  The main metrics used to measure the performances of the approaches were:
  - Precision: measures exactness
  - Recall: measures completeness
  - F1 Score: combines both recall and precision into one metric

  Due to data shortages, the experiments will aim to detect the most obvious features like the head and/or the spine.

B. Inter rater reliability using Jaccard similarity

For this experiment, the members of the group were all given 20 different images to label. The results from each annotator are then compared to see if they agree on the labels or not. After labeling, all the members' bounding boxes were cross examined using Jaccard similarity, this method works by examining the similarity between sets. The Jaccard distance is defined as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ and allowed us to calculate a coefficient to find a match. In this case, it was used to determine whether two members agreed or disagreed on the labeling of an image. Two Members having a Jaccard similarity coefficient above a certain threshold such as 50% (fig: 2) or 80% (fig: 3) or if both members labeled nothing in the image, counted as a match and was represented as a 1. If the threshold was not passed or if only one member labeled an object in the image, it counted as a disagreement and was represented as a 0. Afterwards the results for each image

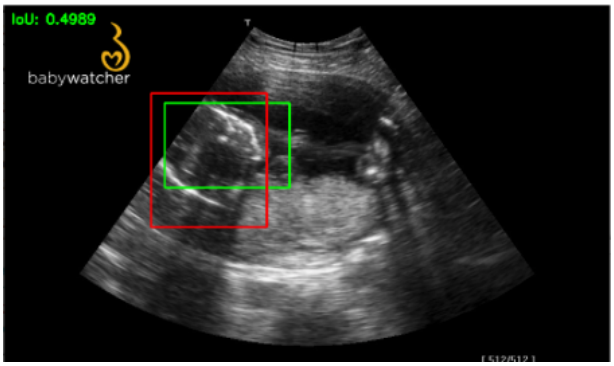


Fig. 2. Threshold = 0.5

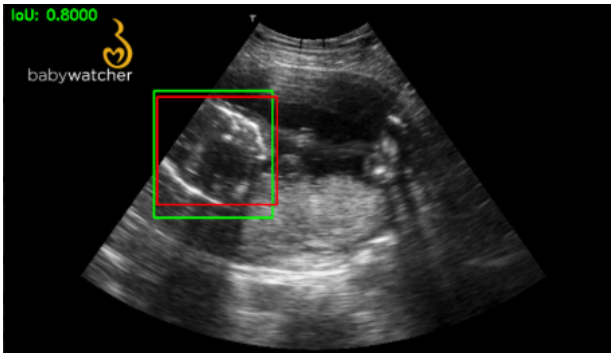


Fig. 3. Threshold = 0.8

was summed up and normalized to find the percentage of agreement for the 20 images.

V. Results and Discussion

A. HOG

1) Parameter tuning: In order to improve the HOG approach we tested different computing the HOG features using a different settings when computing the HOG features:

- 128x128 window size , 16x16 cell size, 2x2 block size: Base line
- 128x128 window size , 16x16 cell size, 4x4 block size: Modifying block size
- 128x128 window size , 8x8 cell size, 2x2 block size: Modifying cell size
- 96x96 window size , 16x16 cell size, 2x2 block size: Modifying window size

	128 16 2	**128 16 4**	**128 8 2**	**96 16 2**
Head precision	71.74%	76.99%	64.48%	72.00%

TABLE I
Precision of HOG+SVM models trained using different HOG computing settings

When testing different HOG computing parameters, the SVMs were re-trained using the same parameters.

We can see that using a cell size of 16 and using blocks with 4 cells as HOG settings we get the best results. We will keep those settings for further testing. Then, we tested skipping the post-classification pruning step explained in the overview and directly taking the maximum of predicted probabilities for each body part classification, and also tested to see how skipping the image processing would change the results:

When testing without image-processing, the SVMs were re-trained as well without image-processing to keep it fair.

	Without processing	**Without pruning step**	**With pruning**
Head precision	70.91%	76.67%	76.99%

TABLE II
Precision of HOG+SVM models tested by omitting certain steps

We can see that skipping the image processing step reduces the precision of the algorithm significantly. This might be due to the fact that the image processing step reduces a lot of the noise in the image which makes the appearance of the body parts a lot more varied. We also notice that skipping the pruning step does decrease the resulting precision compared, however not by much.

To run the final test, we decided to compute the HOG features using a cell size of 16, blocks of size 4. We also kept using the image-processing and the pruning step.

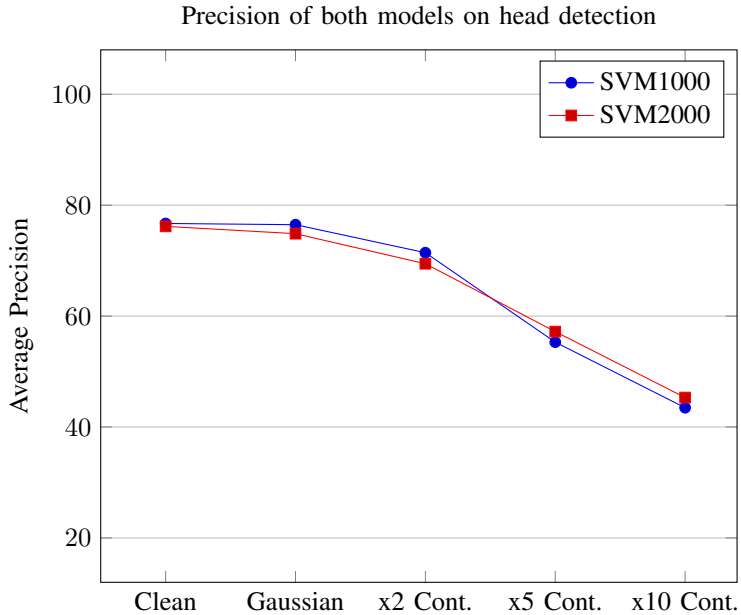
2) Precision Tests: .

	Head	**Arms**	**Hands**	**Spine**	**Leg**	**Feet**
Clean Images	76.70%	0.00%	0.00%	18.52%	0.00%	0.00%
Gaussian Noise	76.48%	0.00%	0.00%	17.20%	0.00%	0.00%
x2 Contrast	71.43%	0.00%	0.00%	14.22%	0.00%	0.00%
x5 Contrast	55.30%	0.00%	0.00%	10.87%	0.00%	0.00%
x10 Contrast	43.48%	0.00%	0.00%	0.0%	0.00%	0.00%

TABLE III
Precision of HOG approach trained on 1000 images

	Head	**Arms**	**Hands**	**Spine**	**Leg**	**Feet**
Clean Images	76.18%	0.00%	0.00%	19.59%	0.00%	0.00%
Gaussian Noise	74.86%	0.00%	0.00%	18.87%	0.00%	0.00%
x2 Contrast	69.44%	0.00%	0.00%	17.41%	0.0%	0.00%
x5 Contrast	57.20%	0.00%	0.00%	13.89%	0.00%	0.00%
x10 Contrast	45.31%	0.00%	0.00%	0.0%	0.00%	0.00%

TABLE IV
Precision of HOG approach trained on 2000 images



Tables III and IV both show how the models are affected when the images have noise or the contrast is changed. Looking at the performance of SVM1000 and SVM2000 we can see how the model that has been trained on more images has a slightly precision. This is because it has seen more images and was able to learn different features. So, SVM2000 is better than SVM1000. They also show that gaussian noise does not affect either models that much.

B. YOLO

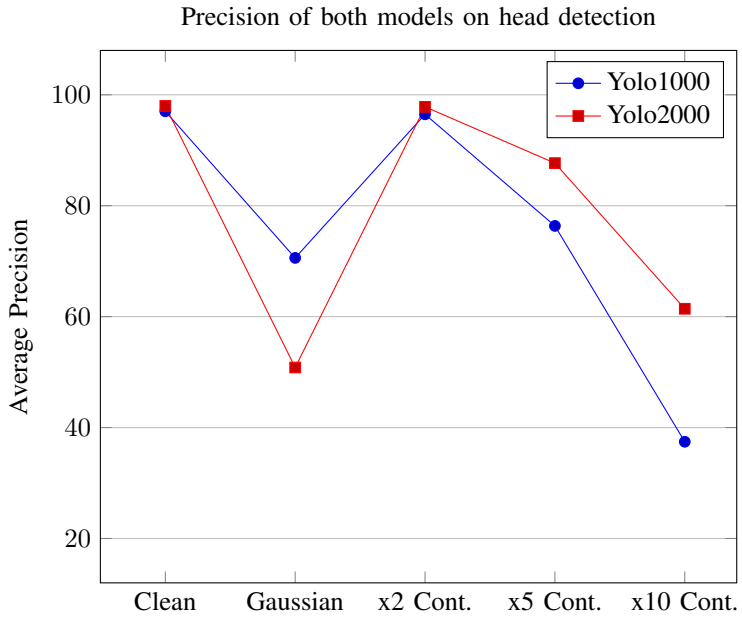
Tables V and VI both show how the YOLO models are affected when the images have noise or the contrast is changed.

	Head	Arms	Hands	Spine	Leg	Feet
Clean Images	97.02%	15.62%	6.53%	67.65%	1.61%	0.00%
Gaussian Noise	70.59%	0.00%	0.66%	2.24%	0.00%	0.00%
x2 Contrast	96.48%	7.41%	4.39%	52.65%	1.02%	0.00%
x5 Contrast	76.36%	4.71%	0.24%	2.97%	0.63%	0.00%
x10 Contrast	37.44%	0.04%	0.08%	0.14%	0.00%	0.00%

TABLE V
PRECISION OF YOLO MODEL TRAINED ON 1000 IMAGES

	Head	Arms	Hands	Spine	Leg	Feet
Clean Images	97.99%	30.48%	9.37%	71.47%	8.33%	0.00%
Gaussian Noise	50.83%	0.00%	0.00%	5.31%	0.00%	0.00%
x2 Contrast	97.83%	21.96%	16.10%	63.30%	12.50%	0.00%
x5 Contrast	87.68%	18.45%	3.66%	17.19%	5.56%	0.00%
x10 Contrast	61.42%	2.91%	0.52%	3.22%	0.00%	0.00%

TABLE VI
PRECISION OF YOLO MODEL TRAINED ON 2000 IMAGES



As explained above for the SVM, YOLO2000 outperforms YOLO1000 as it was trained on more images so it has a higher precision as seen in the graph above. However, we can see that the Gaussian noise affects it a lot.

C. Comparing YOLO and the SVM

To compare the performance of both approaches, we have calculated other metrics like the recall and F1 score. Here, only the models trained on 2000 images were compared to each other so, YOLO2000 and SVM2000. In addition to that, the metrics were only calculated for the head body part as this is the label we are most confident of. The total number of heads in the test set was 343.

	Precision	Recall	F1 Score
Yolo2000	0.97	0.99	0.98
SVM2000	0.76	0.98	0.86

TABLE VII
COMPARING THE PERFORMANCE OF THE SVM AND YOLO

D. Inter rater reliability test

Matching Threshold	Head	Spine	Feet	Leg	Hand
50%	91.67%	61.67%	96.67%	78.33%	88.33%
80%	42.50%	57.50%	96.67%	78.33%	88.33%

TABLE VIII
PERCENTAGE OF AGREEMENT BETWEEN ALL MEMBERS PER FEATURE

Using a matching threshold of 50% results in a high percentage of agreement across all five features, however as can be seen on table VIII changing the threshold from 50% to 80% only impacted the agreement percentage of the head and spine. This is because the head and spine were the most frequently detected body parts on the ultrasound images, adjusting the threshold to 80% meant the overlap between the labels must be very precise. Especially the decrease in percentage of agreement on the head shows that often members find the head but have slightly different labels, As an example see image (fig: 2). For the spine this is usually not the case, the percentage of agreement about the spine is lower at 61.67% however changing the matching threshold does not influence the percentage of agreement much.
Increasing the threshold did not effect the feet, legs or hand, this resulted from counting no labels of a body part by both members as a match. This match is not influenced by the changing threshold and as such does not change the percentage of agreement. Although the percentage of agreement for these three features is high, this is caused by the members often not detecting any of these features in the ultrasound image. The reliability for the leg, hands and feet is low and explains why SVM and YOLO perform poorly on these body parts.

VI. CONCLUSION

Babywatcher is product that enables the parents to make an ultrasound image of their baby without visiting the doctor. The main aim of the paper was to assist users in finding the fetus's body parts while looking at the ultrasound image.
The paper shows two different approaches that can be used to solve this problem. Both approaches were compared and the best performing one was chosen to be in the final product. The models were tested on clean, noisy and images of different contrasts.
In conclusion, the models performed best when trained on more data and when tested on the clean images. Their performance worsens as the contrast increases and as noise gets added. The YOLO model outperformed the SVM model when their head detection precision was compared.
In addition to that, as mentioned before, the data was labeled by us, students with no medical background, and this made us doubt the reliability of the data that we have labeled. Due to this, an inter rater reliability test was performed which measured the reliability of our labelled data. This test showed that the head and spine labels were the most reliable and this explained why the models performed best on these body parts. To support this, we think that they were the most obvious body parts to see in an ultrasound so we were confident when labelling them.
To conclude the paper in one sentence, the YOLO model outperformed the SVM with regards to different metrics and the most reliable labels produced by us were the head and spine labels due to them being the most obvious body parts in an ultrasound image.

VII. FUTURE WORK

Future work for this work can be done on both the labelling of the data and on the YOLO model itself.

A. Data

Data labeling proved to be a challenging part of this project, in the future having reliably labeled training and testing data would benefit future studies greatly, This labeling should be done by or alongside trained professionals who can easily identify body parts of the fetus. Having this data available will make future models more robust and reliable. It will also allow YOLO to detect harder body parts like the nose, mouth, eyes and more, while also increasing its

precision. It would also be interesting to know which trimester an image was taken in. This would allow us to know what body parts we can expect when trying to detect.

B. YOLO

Regarding the model, it's hyper parameters can be fine-tuned and experimented with. This can allow us to produce more reliable results. It will also be interesting to test it on videos as this will offer the users of BabyWatcher a better experience. YOLO will be great for such application as it can provide real time detection. Finally, we can make use of the general context of an image so for instance, if a head is found then we can make the model look for the spine around or close to the head as this is it's usual position. This would increase the robustness of the model.

REFERENCES

[1] Black, R.B. (1992). Seeing the baby: The impact of ultrasoundtechnology. Journal of Genetic Counseling, 1(1), 45-54.

[2] Ji, E.-K., Pretorius, D.H., Newton, R., Uyan, K., Hull, A.D., Hollenbach, K. and Nelson, T.R. (2005), Effects of ultrasound on maternal-fetal bonding: a comparison of two- and three-dimensional imaging. Ultrasound Obstet Gynecol, 25: 473-477. https://doi.org/10.1002/uog.1896

[3] Rustico, M.A., Mastromatteo, C., Grigio, M., Maggioni, C., Gregori, D. and Nicolini, U. (2005), Two-dimensional vs. two- plus four-dimensional ultrasound in pregnancy and the effect on maternal emotional status: a randomized study. Ultrasound Obstet Gynecol, 25: 468-472. https://doi.org/10.1002/uog.1894

[4] P.F.Felzenszwalb,R.B.Girshick,D.McAllester,andD.Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation.

[6] J. Redmon, S. Divvala, R. Girshick, A. Farhadi.You Only Look Once: Unified, Real-Time Object Detection.

[7] Xianhua Zeng, Li Wen, Banggui Liu, Xiaojun Qi, Deep learning for ultrasound image caption generation based on object detection, Neurocomputing, Volume 392, 2020, Pages 132-141, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2018.11.114

[8] https://www.makesense.ai/

[9] Hallgren KA. Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial. Tutor Quant Methods Psychol. 2012;8(1):23-34. doi:10.20982/tqmp.08.1.p023

[10] Bochkovskiy, A., Wang, C. and Liao, H., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection.

[11] Chablani, M., 2017. YOLO — You only look once, real time object detection explained. [online] Medium. Available at: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006

[12] Gupta, M., 2020. YOLO — You Only Look Once. [online] Medium. Available at: https://towardsdatascience.com/yolo-you-only-look-once-3dbdbb608ec4

[13] Karimi, G., 2021. Introduction to YOLO Algorithm for Object Detection. [online] Engineering Education (EngEd) Program — Section. Available at: https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/

[14] Medium. 2018. Overview of the YOLO Object Detection Algorithm. [online] Available at: https://odsc.medium.com/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0

[15] Navneet Dalal,Bill Triggs. Histograms of Oriented Gradients for Human Detection. Available at: http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[16] Satya Mallick. 2016. Histogram of Oriented Gradients explained using OpenCV. Available at: https://learnopencv.com/histogram-of-oriented-gradients/