Ultrasound in Med. & Biol., Vol. 48, No. 4, pp. 663–674, 2022

0301-5629/$ - see front matter

https://doi.org/10.1016/j.ultrasmedbio.2021.12.006

● *Original Contribution*

AUTOMATIC PLACENTA LOCALIZATION FROM ULTRASOUND IMAGING IN A RESOURCE-LIMITED SETTING USING A PREDEFINED ULTRASOUND ACQUISITION PROTOCOL AND DEEP LEARNING

MARTIJN SCHILPZAND,[*,†,‡] CHASE NEFF,[†] JEROEN VAN DILLEN,[§] BRAM VAN GINNEKEN,[*] TOM HESKES,[‡] CHRIS DE KORTE,[†,¶] and THOMAS VAN DEN HEUVEL[*,†]

* Diagnostic Image Analysis Group, Department of Medical Imaging, Radboud University Medical Center, Nijmegen, The Netherlands; [†] Medical Ultrasound Imaging Centre, Department of Medical Imaging, Radboud University Medical Center, Nijmegen, The Netherlands; [‡] Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands; [§] Department of Obstetrics, Radboud University Medical Center, Nijmegen, The Netherlands; and [¶] Physics of Fluids Group, Technical Medical Center, University of Twente, Enschede, The Netherlands

(*Received* 7 *November* 2020; *revised* 22 *November* 2021; *in final form* 2 *December* 2021)

Abstract—Placenta localization from obstetric 2-D ultrasound (US) imaging is unattainable for many pregnant women in low-income countries because of a severe shortage of trained sonographers. To address this problem, we present a method to automatically detect low-lying placenta or placenta previa from 2-D US imaging. Two-dimensional US data from 280 pregnant women were collected in Ethiopia using a standardized acquisition protocol and low-cost equipment. The detection method consists of two parts. First, 2-D US segmentation of the placenta is performed using a deep learning model with a U-Net architecture. Second, the segmentation is used to classify each placenta as either normal or a class including both low-lying placenta and placenta previa. The segmentation model was trained and tested on 6574 2-D US images, achieving a median test Dice coefficient of 0.84 (interquartile range = 0.23). The classifier achieved a sensitivity of 81% and a specificity of 82% on a holdout test set of 148 cases. Additionally, the model was found to segment in real time (19 ± 2 ms per 2-D US image) using a smartphone paired with a low-cost 2-D US device. This work illustrates the feasibility of using automated placenta localization in a resource-limited setting. (E-mail: Martijn.sch@gmail.com) © **2021 The Author(s). Published by Elsevier Inc. on behalf of World Federation for Ultrasound in Medicine & Biology. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).**

Key Words: **Ultrasound, Placenta previa, Segmentation, Machine learning, Neural network, Computer-aided diagnosis, Resource-limited countries, Prenatal, Obstetrics.**

INTRODUCTION

Placenta previa is a maternal risk factor characterized by the placenta either partially or completely covering the endocervical os (Silver 2015). When the placenta does not cover the endocervical os but is located within a distance of 2 cm, it is defined as a low-lying placenta. Placenta previa and, to a lesser extent, low-lying placenta are associated with severe obstetric risks caused by blood loss in the third trimester and during delivery (Fan et al. 2017; Jansen et al. 2019). Obstetric ultrasound imaging is commonly used to detect placental position. Unfortunately, resource-limited countries often lack financial resources and trained sonographers to perform ultrasound examinations in rural areas (LaGrone et al. 2012).

Van den Heuvel (2019) proposed combining the usage of low-cost ultrasound equipment with software for automatic detection of maternal risk factors. Health care workers can be trained within a few hours to acquire ultrasound data with an obstetric sweep protocol (DeStigter et al. 2011). The acquired ultrasound data can be processed by a machine learning algorithm on a smartphone to automatically detect maternal risk factors. This system could be used to enable the referral of women with high-risk pregnancies to a hospital for safe delivery. In their work, Van den Heuvel et al. (2019) found that it is possible to automatically estimate gestational age, determine fetal

Address correspondence to: Martijn Schilpzand, Department of Medical Imaging, Radboud University Medical Center, Geert Grooteplein Zuid 10 6500 HB Nijmegen, The Netherlands. E-mail: Martijn.sch@gmail.com

presentation and detect twin pregnancies using the obstetric sweep protocol.

In the literature, there is relatively little published work on automatic placenta localization. There is a single study on automatic ultrasound assessment of placenta previa (Saavedra et al. 2020) in which a localization algorithm for identifying potential cases of placenta previa was proposed. However, the data set included only 10 study participants. Qi et al. (2017) reported that weakly supervised learning can be used to automatically localize anatomical structures in placenta ultrasound images. Several other studies have been conducted on automatic placental segmentation from ultrasound imaging. Looney et al. (2018) and Yang et al. (2019) attempted volumetric (3-D) placental segmentation, whereas Hu et al. (2019) considered 2-D placental segmentation. Hu et al. (2019) achieved a mean Dice score of 0.92 with a U-Net trained on 1364 2-D images acquired from 247 cases. The gestational age of these cases ranged from 8 to 34 wk.

In this study, we present a method that automatically detects low-lying placenta or placenta previa from ultrasound imaging with the use of an obstetric sweep protocol. This detection method was optimized to run on a smartphone and produce real-time segmentation to enable usage in remote areas. The main contribution of this work is to illustrate the feasibility of using the proposed automatic placenta localization method in a resource-limited setting.

METHODS

Our placenta localization method consists of two phases. In the first phase, a deep learning model with a U-Net architecture (Ronneberger et al. 2015) segments the placenta on 2-D ultrasound images. In the second phase, the placenta segmentation is used to classify cases as either normal placenta or low-lying placenta. For classification, placenta previa and low-lying placenta are grouped in the low-lying placenta class. To evaluate the feasibility of clinical application, the loading and inference times of the segmentation algorithm are examined on a smartphone.

Data acquisition

The data used in this study were acquired at St. Luke's Catholic Hospital, Wolisso, Ethiopia. As Ethiopia is a low-income country with the fourth-highest maternal mortality rate in 2017 (World Health Organization et al. 2019), the data were representative of the target population. A local gynecologist used the MicrUs Ext-1H with the C5-2R60S-3 transducer (Telemed, Vilnius, Lithuania) to obtain ultrasound data on 280 pregnant women. The average gestational age of these woman was 31 wk (range: 18–40 wk). The collection of the data used in this study was approved by the local ethics committee (ID Ref. No. BEFO/AHBTHQO/4004/1-20). All participants signed a written informed consent. For the image settings on the MicrUs Ext-1H, we used a gain of 81%, scanning depth of 15 cm, center frequency of 4 MHz, frame averaging of 4 and speckle reduction level 4 pureview; the Image Enhancement was set to method 3. The ultrasound data were acquired with the use of the obstetric sweep protocol (DeStigter et al. 2011), which consists of three transverse sweeps followed by three longitudinal sweeps over the abdomen (see Fig. 1). During a sweep, 20 2-D ultrasound images were acquired per second. The gynecologist was asked to perform each sweep in approximately 5 s, which corresponds to roughly 100 2-D ultrasound images per sweep and accumulates to 600 2-D images per woman. These images were exported in DICOM format using the Echo Wave II software (Telemed, Vilnius, Lithuania). For clarity, the 2-D ultrasound images will be referred to as frames in this work.

A medical expert (C.N.) annotated (manually segmented) the placenta on frame level using ITK-SNAP 3.6.0 (Yushkevich et al. 2016). Figure 2 illustrates an example of a frame with its corresponding placenta annotation. Because of time constraints, it was not possible to annotate the placenta for all 280 cases. Therefore, the data were randomly split into two sets. Set 1 consisted of 132 annotated cases and set 2 consisted of the remaining 148 cases. To reduce the duration of the annotation process, the expert annotated only one in every five frames that contains placenta. As only one in every five frames containing placenta is annotated, all frames within a range of five frames from an annotated frame are considered positive frames. The other frames are considered negative frames, that is, frames that do not contain placenta. Set 1 consists of 36,504 positive frames, 65,305 negative frames and 6574 annotated

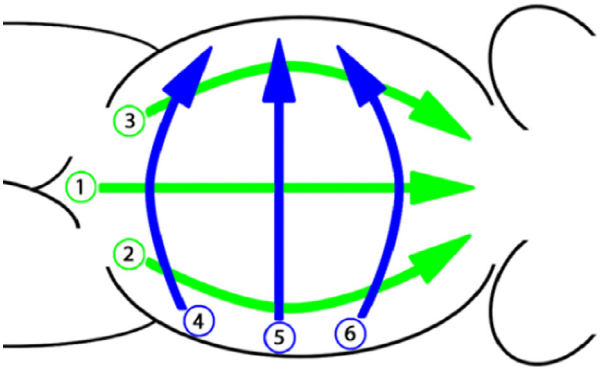

Fig. 1. Visualization of the obstetric sweep protocol. The numbers indicate the order in which the sweeps are performed. Reprinted, with permission, from Van den Heuvel (2019).

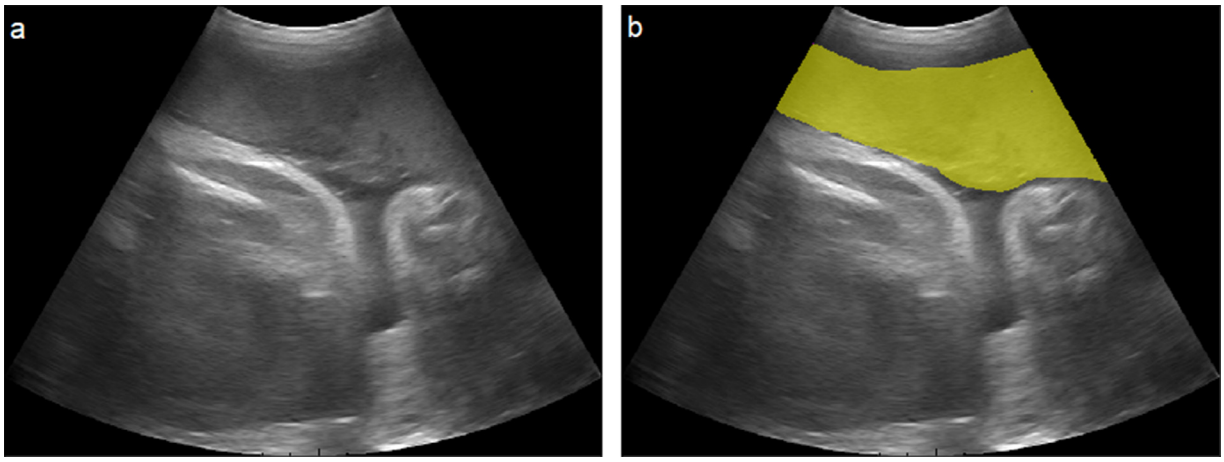

Fig. 2. Example of an ultrasound frame (a) and the same frame with the placenta in *yellow* (b). This frame originates from sweep 1, as illustrated in Figure 1. The images in this figure were created using the Python package Matplotlib (Hunter 2007).

frames. Figure 3 is a 3-D visualization of all annotated frames for one pregnant woman.

The expert also provided case labels for all 132 cases in set 1. The case labels are low-lying placenta (includes placenta previa), normal placenta or not assessable. A case was labeled not assessable when the quality of data was insufficient to determine the placenta location. The expert did not have any additional information and created the case labels based solely on the ultrasound data. Two observers (C.N. and J.v.D.) independently labeled all cases in set 2. The final ground truth case labels for set 2 were determined via a consensus meeting in which the two observers discussed the cases on which they initially did not agree. Table 1 outlines the class distribution for sets 1 and 2.

Phase 1: Placenta segmentation

Set 1 contains 6574 annotated frames. Two preprocessing operations were performed on these frames. First, the frames were masked and cropped to remove surrounding markers such as the ultrasound acquisition settings, the lookup table, and the ruler. Afterward, all pixel values were divided by 255, which scaled the values from 0 to 255 to floating point values ranging from 0 to 1.

A fully convolutional network with a 2-D U-Net architecture inspired by Ronneberger et al. (2015) was implemented for segmentation of the placenta. To optimize the performance and computational efficiency of the U-Net, four experiments were performed. First, the effect of scaling the size of input images and the number of model parameters was examined. Second, different padding strategies were evaluated. Third, an experiment was carried out to study the effect of batch size on performance. Lastly, negative frames were added to the training data to improve overall segmentation performance on a case level.

Scaling. In the first optimization experiment, the effect of scaling with respect to the size of the input frames and the model parameters was examined. The input frames were downsampled to reduce computation time and memory cost. Downsampling also avoids the need for a deeper U-Net, as it effectively increases the

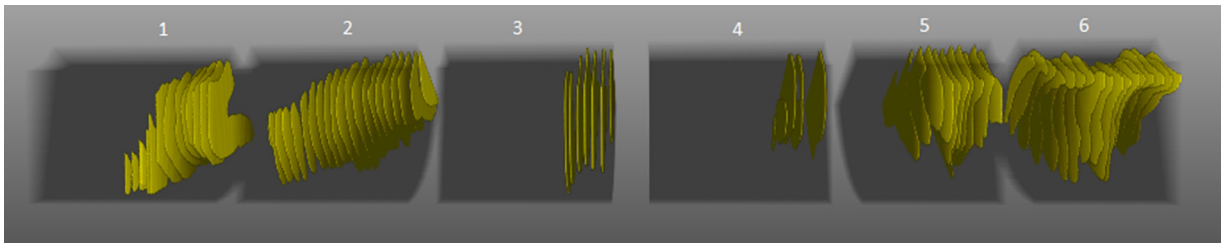

Fig. 3. Three-dimensional visualization of the placenta annotations for a single case. The *yellow* surfaces represent the annotations, and the *blocks of dark background* indicate the six sweeps. The numbering of these sweeps correspond to the numbers in Figure 1. The spaces between the blocks represent the window in which the transducer was lifted from the abdomen. These visualizations were made in MeVisLab 3.2 (MeVis Medical Solutions, Bremen, Germany).

Table 1. Class distribution of the placenta case labels for both sets 1 and 2

	Set 1	Set 2
Normal placenta	101	126
Low-lying placenta	27	16
Not assessable	4	6

receptive field of the convolutional kernels. The frames were downsampled by selecting pixels with a step size equal to the downsampling factor. Figure 4 visualizes the effect of different downsampling factors.

In a U-Net, the number of feature channels is doubled after each pooling layer and halved after each upsampling layer. Therefore, the number of model parameters can be derived from the number of output channels of the first convolutional layer (referred to in this work as model channels). The U-Net proposed by Ronneberger et al. (2015) has 64 model channels. Decreasing the number of model channels and thereby the model's parameters reduces computation time and memory cost. In this optimization experiment, all combinations of downsampling factors 1, 2, 4, 6 and 8 and model channels 8, 16, 32 and 64 were examined. The numbers of model parameters corresponding to these model channels are listed in Table 2. The batch sizes were selected as the largest power of 2 that could fit on the graphics processing unit (GPU) memory and, therefore, maximized the GPU capacity.

Padded convolutions. The U-Net introduced by Ronneberger et al. (2015) contains convolutional layers with no padding, referred to as unpadded convolutions. The unpadded convolutions decrease the size of the feature maps. Consequently, the output of the model is smaller than the input. To ensure that the output segmentation covers the entire frame, the input has to be heavily padded. Padded convolutions, sometimes referred to as same-padding convolutions, indicate that padding is added implicitly at layer level so that the feature map size is not decreased by a convolutional layer. As a result, the input and output of a model with padded convolutions have the same size, and heavy padding of the input can be avoided. This decrease in input size reduces the memory cost and the computation time of the algorithm.

Table 3 outlines the in- and output sizes for padded and unpadded convolutions for different downsampling factors. The output size of the unpadded convolutions and in- and output sizes of the padded convolutions are slightly larger than the frame size. This is to ensure that no padding is needed at the max-pooling layers. Both the ultrasound images and the placenta segmentations were zero-padded to their desired size. For comparison, the scaling experiment was repeated with the padded models, and these were trained with the same batch sizes as the unpadded models.

Batch size. The reduction in memory cost caused by the padded convolutions resulted in the possibility of selecting large batch sizes, especially for efficient models with a high downsampling factor and a small number of model channels. However, the models that were trained with these large batch sizes would either not converge or exhibited poor performance. Instead of maximizing GPU capacity, which resulted in large batch sizes, we evaluated batch sizes 4, 8, 16, 32 and 64 and selected the best performing batch size per model.

Negative frames. The models in the scaling, padding and batch size experiments were trained on the 6574 annotated positive frames. In the negative frame experiment, we aimed to improve generalization by training models on these annotated frames plus different percentages of randomly sampled negative frames. At the start of every epoch, these negative frames were added to the training data by randomly sampling from all negatives in the data set. The percentage of negative frames in the training data is referred to as the negative frame ratio. Experiments were performed with negative frame ratios of 0.0, 0.2, 0.4, 0.6 and 0.64. The latter value is equal to the negative frame ratio in set 1 and was chosen to examine a model that was trained with data representative of the original distribution.

Training and evaluation. The segmentation experiments were fivefold cross-validated to evaluate the generalization on all 132 cases in set 1. Three folds were used as training data, one as validation data and one as test data corresponding to a 60% train, 20% validation and 20% test split. During training of the segmentation model, the following parameters were kept constant for all experiments. The binary cross-entropy function was used as the loss function because this is a two-class segmentation problem. The sigmoid was used as the final activation function of the model. The output values of the model were binarized with a threshold at 0.5. The model weights were He-normal initialized (He et al. 2015). Adam (Kingma and Ba 2014) was used as the optimizer with an initial learning rate of 0.001. Training was stopped when there was no improvement in the validation Dice for 50 epochs. Finally, the same seed was set for all experiments to reduce variation and ensure reproducibility of the results. The models were implemented in Keras with Tensorflow 1.15.0 (Abadi et al. 2016) as the back end, and they were trained on a GeForce GTX 1080 Ti (Nvidia Corp., Santa Clara, CA, USA) graphics card.

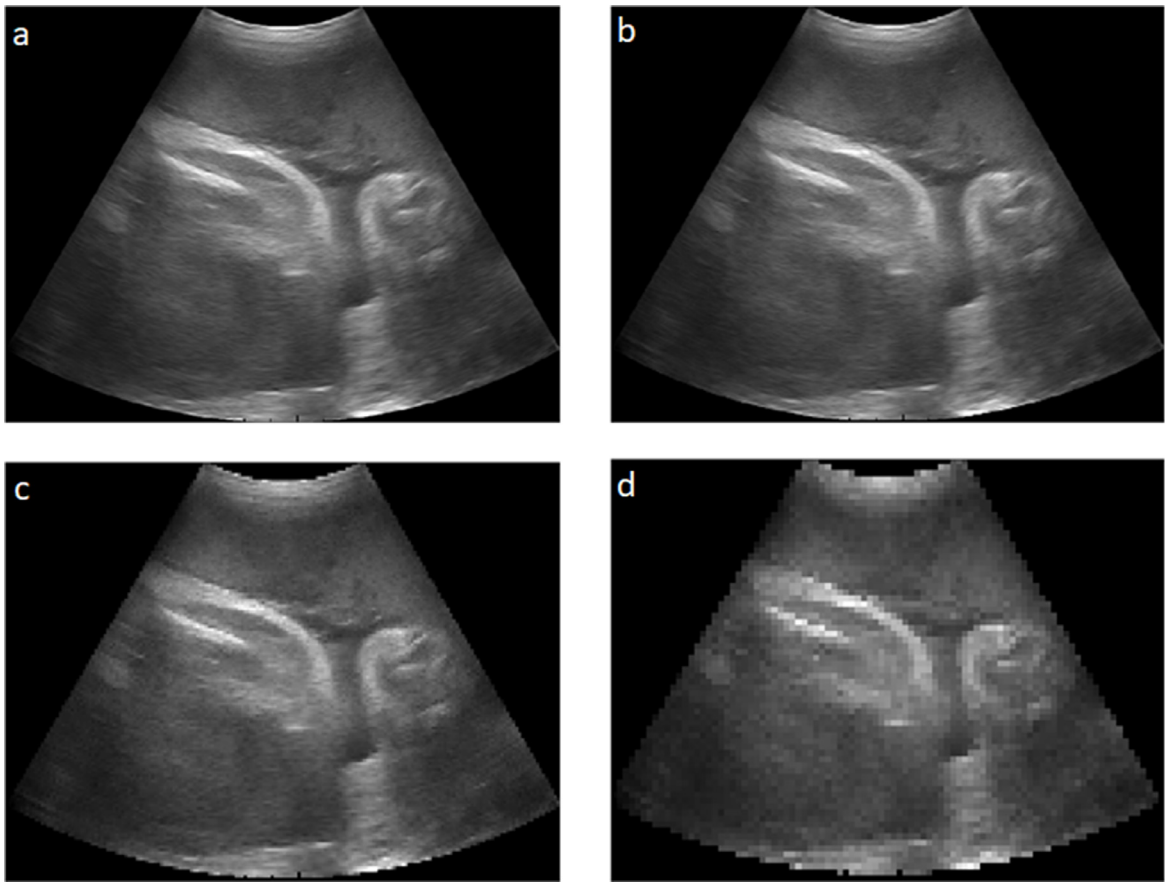

Fig. 4. Visualization of different downsampling factors. The downsampling factors with the corresponding dimensions from (a) to (d) are no downsampling (562 × 744), downsampling 2 (281 × 372), downsampling 4 (141 × 186) and downsampling 8 (71 × 93).

The Dice score (Dice 1945; Sørensen 1948) was used as the evaluation metric and is defined as

$$\text{Dice} = \frac{2|X \cap Y|}{|X| + |Y|} \quad (1)$$

For binary segmentation, X represents the segmentation, and Y the annotation or vice versa. The performance on a set of frames was represented by the median Dice and the interquartile range as the Dice scores were non-normally distributed according to the Shapiro-Wilk test (p value < 0.05) (Shapiro and Wilk 1965). After training, the models were finalized by loading the weights corresponding to the epoch with the highest validation Dice. The Dice score is not suited as a performance metric for negative frames as the numerator in eqn (1) for a negative frame is zero, and, if the model correctly segments no placenta, the denominator is zero as well. In other words, the Dice score of a negative

Table 2. Number of model parameters for a four-level deep 2-D U-Net for different model channels

Model channel*	Parameter
8	485,673
16	1,940,817
32	7,759,521
64	31,030,593

* Number of output channels of the first convolutional layer.

Table 3. Frame size and resulting input and output sizes of the 2-D U-Net for different DFs*

DF	Frame size	UC input	UC output	PC input/output
1	562 × 744	764 × 956	580 × 772	576 × 752
2	281 × 372	476 × 572	292 × 388	288 × 384
4	141 × 186	348 × 316	164 × 196	144 × 192
6	94 × 124	284 × 316	100 × 132	96 × 128
8	71 × 93	284 × 284	100 × 100	80 × 96

DF = downsampling factor; UC = unpadded convolution; PC = padded convolution.

* The input and output sizes in the third and fourth columns correspond to a U-Net with UCs, and the input and output sizes in the fifth column correspond to a U-Net with PCs.

frame will either be zero or undefined. To evaluate the performance on negative frames we introduce a quantity called false positives on negative frames (FPN). This quantity is calculated by taking the sum of the placenta predicted pixels in a negative frame and dividing this by the number of pixels in the frame. Finally, the mean FPN is taken for all negative frames in the data.

Phase 2: Placenta classification

The best performing segmentation model was used to segment all frames in a case to obtain case segmentations. The case segmentations were subsequently used as input for the placenta classifier. To perform cross-validation, five instances of the best performing segmentation model were trained in phase 1, each with a different combination of train, validation and test folds. Every case in set 1 was segmented by the instance for which that case belonged to the test data. For set 2, the mean prediction of the five cross-validation instances was used to create case segmentations. Set 1 was used to train the classifier, and set 2 was used as a holdout test set.

The cases that were labeled not assessable were excluded from both data sets, resulting in a training set of 128 cases and a test set of 142 cases.

Because of the small number of cases, a machine learning solution was not feasible. Instead, a case was classified based on its *n*th percentile of volume. In this context, the term *n*th percentile of volume was used to indicate the height below which n percent of the placenta volume resides. A case was classified as low-lying placenta if its *n*th percentile of volume was lower than a chosen threshold. This threshold was selected to maximize the sum of the sensitivity and specificity on set 1. With this threshold, the classifier was evaluated on set 2.

The *n*th percentile of volume was calculated from the first three longitudinal sweeps of the acquisition protocol. The number of placenta pixels in the frame represents the placenta volume of the frame. The total placenta volume of a case was considered to be the sum of the placenta volumes for the frames in the first three sweeps. The *n*th percentile is a value between 0 and 1. All frame indices in the sweeps were normalized between 0 and 1, so that the placenta volume below the *n*th percentile in the three sweeps combined summed up to n percent of the total placenta volume. The classifier was evaluated for the percentiles 1, 2.5, 5, 10, 25 and 50. As an example, Figure 5 illustrates the distribution of the placenta volume for both a normal placenta and a low-lying placenta and highlights their percentile values.

Smartphone implementation

We evaluated the loading and inference time of the best segmentation model on a OnePlus 7T (OnePlus Technology Co. Ltd., Shenzhen, China) smartphone with a Qualcomm Snapdragon 855 Plus processor that ran on Android, version 10. The loading and inference time were evaluated over 10 runs and are represented by the mean and standard deviation. The segmentation

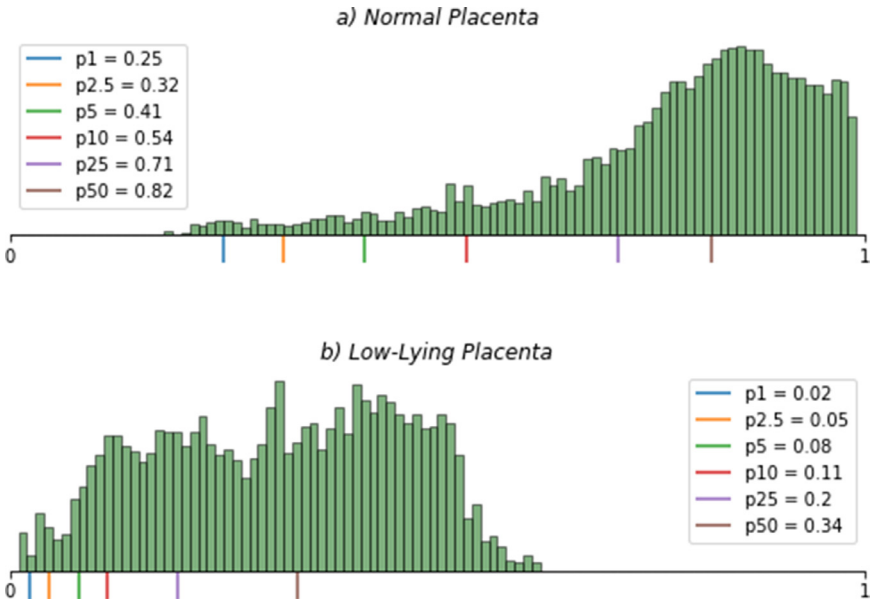

Fig. 5. Placenta volume distribution for a normal placenta (a) and a low-lying placenta (b) with their corresponding percentile values. Placenta volume distribution is determined by the placenta segmentation of the first three transverse sweeps and is normalized between 0 and 1. The range from 0 to 1 is a representation of the space from the pubic bone to the breast bone.

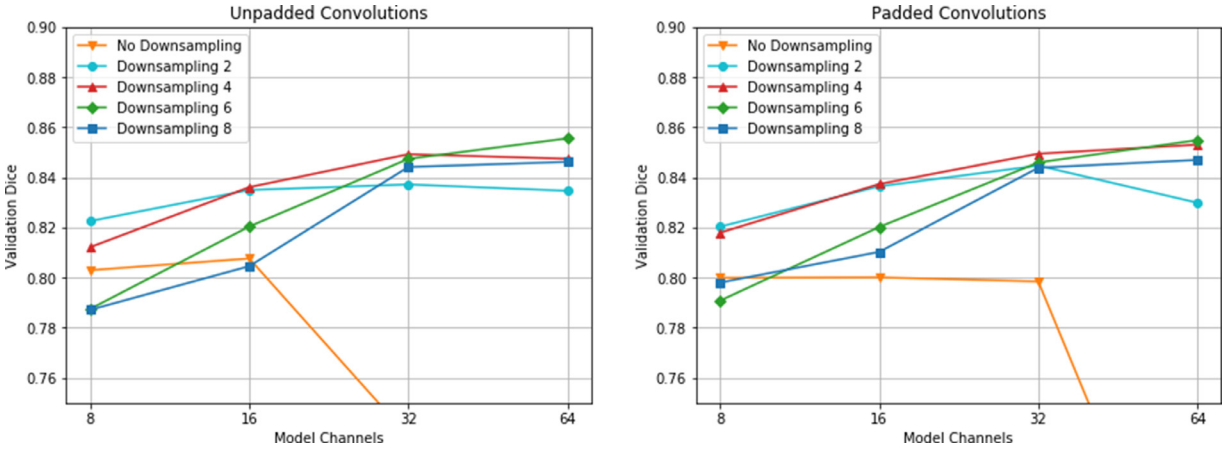

Fig. 6. Validation Dice scores for the scaling experiment for the models with unpadded and padded convolutions.

model was converted to Tensorflow Lite 1.13.1 to load it on an Android-based smartphone.

RESULTS

Phase 1: Placenta segmentation

Scaling. The results of the scaling experiment for both padded and unpadded models are illustrated in Figure 6. With the exception of no downsampling, all models perform within a Dice score range of 0.78 to 0.86. No downsampling for the unpadded models had Dice scores of 0.74 and 0.35 for 32 and 64 model channels, respectively. For the padded models, no downsampling scored a Dice of 0.64 for 64 model channels.

Padded convolutions. Figure 6 illustrates the effect of the two different padding strategies. The difference in segmentation performance between the padded and unpadded models is negligible for downsampling factors 4, 6 and 8, while the computation time was significantly reduced.

Batch size. The results in Figure 7 indicate that selecting the optimal batch size significantly reduces the difference in segmentation performance between the considered models. The difference between the worst and best performing models in Figure 7 is 0.020 Dice. The best performing model achieved a Dice score of 0.855 ± 0.162. The optimal batch sizes are outlined in Table 4 in comparison to the batch sizes used for the

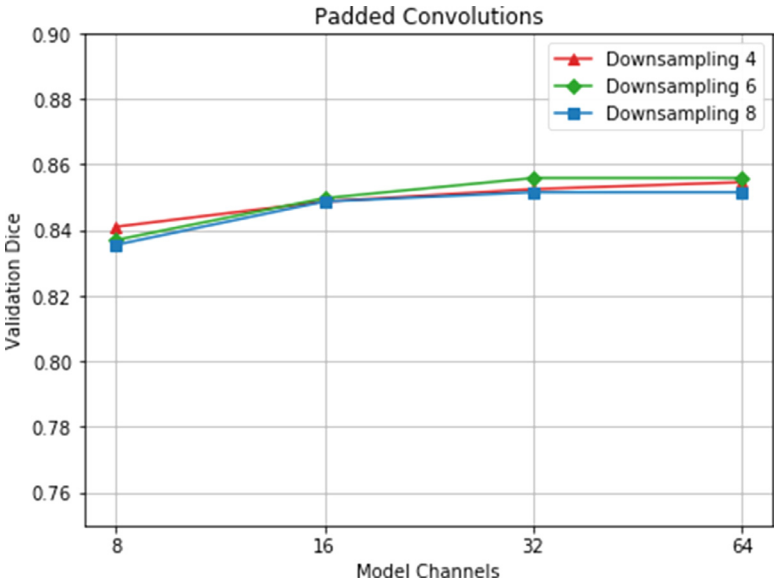

Fig. 7. Scaling experiment with padded models and batch sizes, which were optimized per model. The corresponding batch sizes are given in Table 4.

Table 4. The two batch size strategies for the scaling experiment*

Strategy	Modelchannels	Downsampling factor				
		1	2	4	6	8
Max GPU	8	16	64	128	256	256
	16	8	32	64	128	128
	32	4	16	32	64	64
	64	2	8	16	32	32
Optimal	8	—	—	4	4	4
	16	—	—	8	4	4
	32	—	—	16	8	16
	64	—	—	32	8	16

GPU = graphics processing unit.

The maximum GPU batch sizes maximized the GPU memory capacity for the unpadded models and were used for both the padded and unpadded models in Figure 6. Under the optimal strategy are the batch sizes that were optimized for the padded models and used in Figure 7.

results in Figure 6. No downsampling and downsampling 2 were excluded from the batch size experiment as they were inefficient and did not outperform the higher downsampling factors.

Negative frames. The effect of adding negatives to the training data was evaluated with the best performing model, which had the following hyperparameters: downsampling factor 6, 32 model channels, padded convolutions and batch size 8. Figure 8 illustrates the effect of the negative frame ratio on segmentation performance. The results on the test set are also included in the figure as this was the last optimization step. The lowest FPN was achieved with a negative frame ratio of 0.6 with a reduction of 2.1% in both validation and test Dice scores. Figure 9 is a visualization of the effect of training on a data set with negative frames for a case segmentation. A negative frame ratio of zero resulted in many incorrect placenta predictions. These negative predictions were greatly reduced for the model trained with a negative frame ratio of 0.6. Because of its low FPN, the model with a negative frame ratio of 0.6 was considered the best and final model. It achieved a Dice score on the test set of 0.84 ± 0.23.

Phase 2: Placenta classification

The results of the placenta classification on the train set are visualized in the receiver operating characteristic (ROC) curves in Figure 10. The highest area under the ROC curve (AUC) was achieved by p25 (25th percentile). The figure also indicates the optimal threshold for each classifier. The values for these thresholds with their corresponding sensitivity and specificity values are outlined in Table 5 alongside the results on the test set. Notably, p1 achieved high train and test sensitivities of 0.96 and 0.94, respectively, while scoring lower on specificity. p2.5 reached the highest accuracy of 0.82 and scored high on both sensitivity and specificity on the test set.

Smartphone implementation

The smartphone loaded the best model in 371 ± 25 ms, and the time it took to segment and output a bitmap for a single frame was 19 ± 2 ms.

DISCUSSION

In this study, we illustrated the feasibility of automatic detection of low-lying placenta or placenta previa in a resource-limited setting. For this task, a deep learning algorithm was introduced that can be deployed on a smartphone and was trained on ultrasound data that were acquired with a standardized acquisition protocol. The algorithm consisted of two parts: placenta segmentation and placenta classification. The best performing placenta segmentation

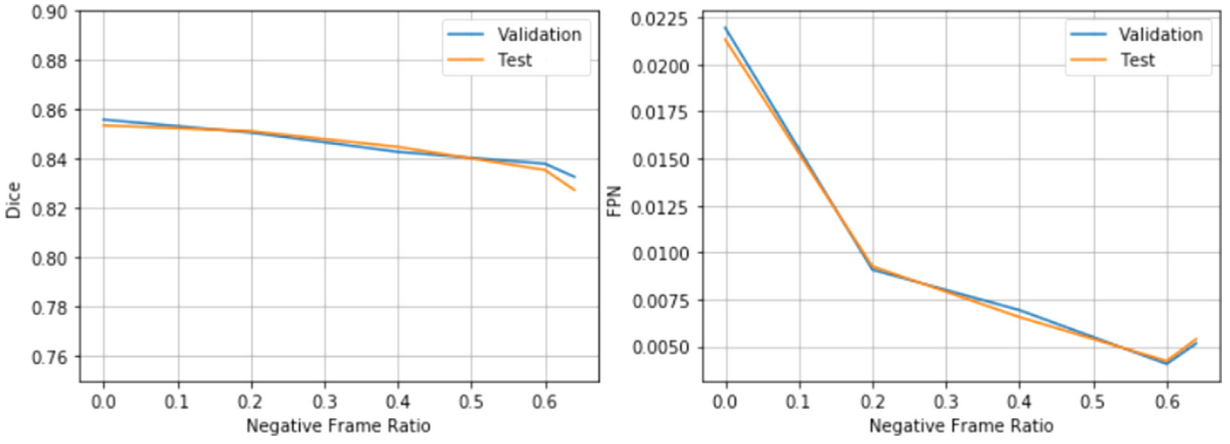

Fig. 8. Segmentation performance in Dice scores (left) and FPN (right) of the best performing model for different negative frame ratios. FPN = false positives on negative frames.

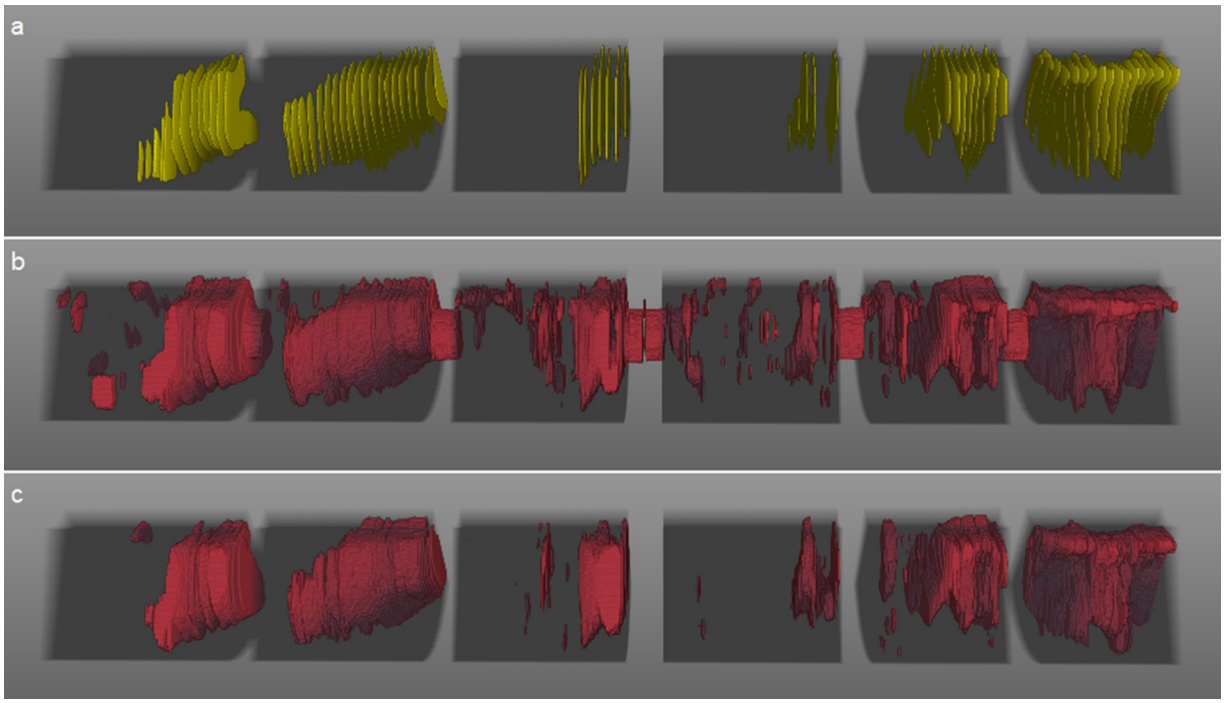

Fig. 9. Three-dimensional placenta visualization of the effect of different negative frame ratios for a single case. Here, (a) is the placenta annotation, and (b) and (c) are model predictions corresponding to negative frame ratios of 0.0 and 0.6, respectively.

model obtained a median Dice score of 0.84 on the test set. The placenta classifier achieved a sensitivity of 81% and a specificity of 82% on a holdout test set.

Phase 1: Placenta segmentation

Figure 6 illustrates that the performance of the no-downsampling models drops significantly for larger

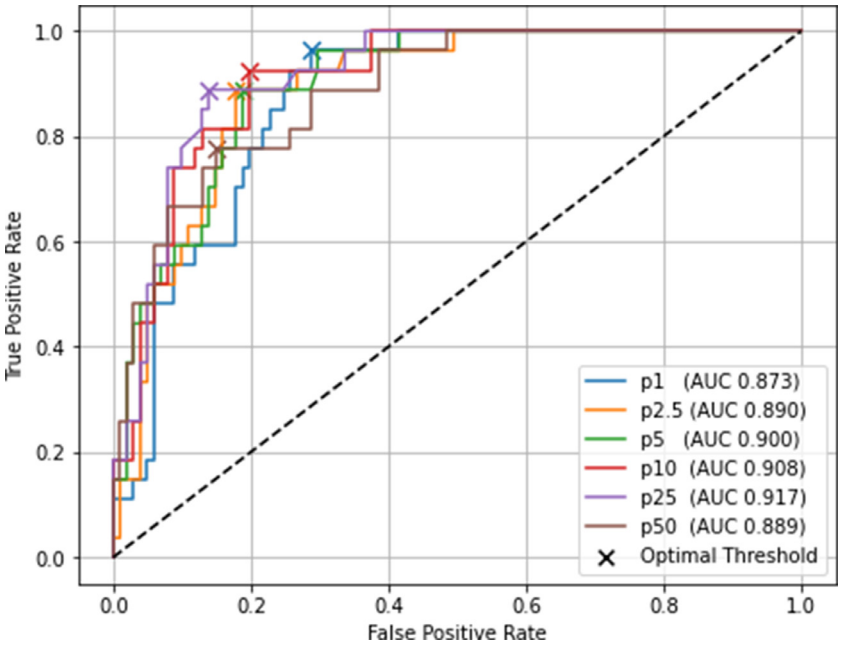

Fig. 10. Receiver operating characteristic curves with their optimal thresholds for the placenta classifier at different percentiles. The optimal threshold was defined as the threshold that maximizes the sum of sensitivity and specificity. AUC = area under the receiver operating characteristic curve.

Table 5. Classifier performance on the train and test sets for different percentiles and their optimal thresholds

Percentile	Threshold	Train set (set 1)		Test set (set 2)	
		Sensitivity	Specificity	Sensitivity	Specificity
1	0.11	0.96	0.71	0.94	0.74
2.5	0.15	0.89	0.82	0.81	0.82
5	0.21	0.89	0.81	0.81	0.8
10	0.27	0.93	0.80	0.75	0.74
25	0.34	0.89	0.86	0.63	0.79
50	0.47	0.78	0.85	0.63	0.79

model channels. This indicates that the combination of large input data and a large number of model parameters does not perform well on this data set. The training curves revealed that these models did not always converge. The no-downsampling models were also very inefficient and therefore excluded from further experiments alongside downsampling 2 models. As the difference in the remaining downsampling factors was negligible between the two graphs in Figure 6, the padded models are favored because of the increase in computational efficiency. Figure 7 illustrates that the performance for the different models in the scaling experiment can be equalized by selecting the optimal batch size. This indicates that the results in Figure 6 were influenced by the varying batch sizes shown in the max GPU section of Table 4. Table 4 indicates that the small batch sizes seem to perform better for downsampling factors 4, 6 and 8. This also explains the increase in performance for these downsampling factors at higher model channels in Figure 6, where the maximum GPU capacity batch sizes were used.

Figure 8 illustrates that the validation and test results follow the same pattern and are almost equal. This indicates that the model generalizes well and that no overfitting occurred on the validation data. Before adding negative frames, the segmentation model would show unexpected artifacts, such as predicting placenta when the transducer made no contact with the abdomen. This is visible in Figure 9b, which illustrates that the model predicts placenta between the sweeps. Training with only frames with placenta present created the bias that every frame must contain placenta. These unwanted artifacts were completely removed at a negative frame ratio of 0.6.

In this study, only the best performing model was used to create the case segmentations for the classifier. However, Figure 7 illustrates that the differences in Dice between the best performing model and the other models are relatively small. A more efficient model might result in similar classification performance while achieving faster inference time on a smartphone.

In comparison to other literature, Hu et al. (2019) achieved a mean Dice of 0.92 on 1364 2-D images acquired from 247 cases. However, their data was acquired with high-end ultrasound equipment, resulting in arguably better data quality. Furthermore, we trained the final models on negative frames, which greatly reduced false positives but led to a decrease in Dice score. Lastly, the segmentation model in this work was largely optimized for efficiency over performance, so that it could be deployed on a smartphone in a clinical setting. In contrast to Hu et al. (2019), our aim was not to achieve the best segmentation Dice score but to create an efficient method for automatic placenta localization.

Phase 2: Placenta classification

For classification of the placenta, low-lying placenta and placenta previa were grouped. This choice was made with regard to the data quality. The acquisition protocol that was used did not always show the location of the cervix, which made it impossible to make a distinction between a low-lying placenta and placenta previa. However, because a low-lying placenta can also lead to complications, the system would still identify potentially high-risk pregnancies without this distinction.

A machine learning approach was avoided for classification because of the small number of cases in the data set. The percentile classifier achieved a sensitivity and a specificity greater than 0.8 on the test set for p2.5 and p5 with the limited amount of data. The p1 classifier achieved a higher test sensitivity of 0.94; however, note that the small number of positive test cases makes this metric volatile.

The percentile-based classifier presented in this study did not use data from the longitudinal sweeps (sweeps 4−6 in Fig. 1). These data could carry important information to improve classification of the placenta. Implementing a classifier that uses the data from the longitudinal sweeps could be promising for future work.

Smartphone implementation

For the algorithm to be applicable in a clinical setting, the computation time should be within a reasonable time frame. With the use of MicrUs Pro C60S (Telemed, Vilnius, Lithuania), ultrasound data can be acquired and transferred to a smartphone at 20 frames/s. The time to process and segment these images on a smartphone with

the presented segmentation model is 19 ms. In other words, the smartphone can segment more than 50 frames/s. As a result, it can produce real-time segmentation. After the acquisition protocol, the placenta classification is done instantly as it is a relatively simple computation. This demonstrates the feasibility of the algorithm in a clinical setting.

Clinical implication

Van den Heuvel (2019) found that it is possible to automatically estimate gestational age, determine fetal presentation and detect twin pregnancies using the same acquisition protocol. Combining these algorithms with the algorithm introduced in this study would result in a system that can automatically detect multiple maternal risk factors. This detection system coupled with trained health care workers can potentially enable affordable widespread ultrasound screening in resource-limited countries. It can assist in identifying high-risk pregnancies and refer them to the hospital for a conventional ultrasound examination by an experienced sonographer. In a resource-limited setting, it is important to consider patient management and the possibility for medical care after detection of maternal risk factors. This goes beyond the scope of this study; however, detection could be the first step in improving obstetric care in resource-limited countries.

Study limitations

The number of cases, especially low-lying cases, was relatively small for placenta classification. As a result, the classification metrics become volatile to minor changes in the threshold. Additional cases with case labels would be required to further improve and robustly evaluate the percentile classifier. Additional data could also open up the possibility of using a machine learning classifier. Another limitation of this study is the inconsistency in the scanned area during the acquisition protocol. Because the protocol consists of freehand sweeps, the abdominal area that was scanned differed per case, which slightly skewed the placenta volume distribution. Ideally, there would be a reference point (*e.g.,* the cervix) for all cases to resolve this issue. This would be challenging given that this reference point must also be automatically detected.

CONCLUSIONS

We developed the first algorithm that automatically detects low-lying placenta or placenta previa from ultrasound data. The data set that was used in this study was acquired in Ethiopia using a standardized acquisition protocol and low-cost equipment. The algorithm consisted of two phases: placenta segmentation and placenta classification. The segmentation model was optimized for efficiency and performance and can produce real-time frame segmentation on a smartphone. The difference in performance between the best performing and more efficient models could be heavily reduced by selecting the optimal batch size. The best performing model achieved a median Dice score of 0.84 (interquartile range = 0.23) on the test set. The classifier achieved a sensitivity of 81% and a specificity of 82% on a hold-out test set consisting of 148 cases. We found that the segmentation model was able to perform real-time segmentation on a smartphone, which further illustrates the feasibility of using the algorithm in a resource-limited setting.

Acknowledgments—This study did not receive any outside funding.

Conflict of interest disclosure—The authors have no conflicts of interest to declare.

REFERENCES

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow IJ, Harp A, Irving G, Isard M, Jia Y, Józefowicz R, Kaiser L, Kudlur M, Levenberg J, Mane D, Monga R, Moore S, Murray DG, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker PA, Vanhoucke V, Vasudevan V, Vióegas FB, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467 [cs DC], 2016.

DeStigter KK, Morey GE, Garra BS, Rielly MR, Anderson ME, Kawooya MG, Matovu A, Miele FR. Low-cost teleradiology for rural ultrasound. 2011 IEEE Global Humanitarian Technol Conf. 290–295.

Dice LR. Measures of the amount of ecologic association between species. Ecology 1945;26:297–302.

Fan D, Xia Q, Liu L, Wu S, Tian G, Wang W, Wu S, Guo X, Liu Z. The incidence of postpartum hemorrhage in pregnant women with placenta previa: A systematic review and meta-analysis. PLoS One 2017;12:1–15.

He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. IEEE Int Conf Comput Vis 2015;1502:1026–1034.

Hu R, Singla R, Yan R, Mayer C, Rohling RN. Automated placenta segmentation with a convolutional neural network weighted by acoustic shadow detection. Annu Int Conf IEEE Eng Med Biol Soc 2019;6718–6723.

Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng 2007;9:90–95.

Jansen CHJR, Kleinrouweler CE, van Leeuwen L, Ruiter L, Limpens J, van Wely M, Mol BW, Pajkrt E. Final outcome of a second trimester low-positioned placenta: A systematic review and meta-analysis. Eur J Obstet Gynecol Reprod Biol 2019;240:197–204.

Kingma D, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.

LaGrone LN, Sadasivam V, Kushner AL, Groen RS. A review of training opportunities for ultrasonography in low and middle income countries. Trop Med Int Health 2012;17:808–819.

Looney P, Stevenson GN, Nicolaides KH, Plasencia W, Molloholli M, Natsis S, Collins SL. Fully automated, real-time 3D ultrasound segmentation to estimate first trimester placental volume using deep learning. JCI Insight 2018;3 e120178.

Qi H, Collins S, Noble A. Weakly supervised learning of placental ultrasound images with residual networks. Med Image Underst Anal 2017;723:98–108.

Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. Med Image Comput Assist Interv 2015;234–241.

Saavedra AC, Arroyo J, Tamayo L, Egoavil M, Ramos B, Castaneda B. Automatic ultrasound assessment of placenta previa during the third trimester for rural areas. IEEE Int Ultrason Symp 2020;1–4.

Shapiro S, Wilk M. An analysis of variance test for normality (complete samples). Biometrika 1965;52:591–611.

Silver RM. Abnormal placentation: Placenta previa, vasa previa, and placenta accreta. Obstet Gynecol 2015;126:654–668.

Sørensen T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Biol Skr 1948;5:1–34.

Van den Heuvel TLA. Automated low-cost ultrasound: Improving antenatal care in resource-limited settings. Ph.D. thesis, Radboud University, Nijmegen, 2019.

Van den Heuvel TLA, Petros H, Santini S, de Korte CL, van Ginneken B. Automated fetal head detection and circumference estimation from freehand ultrasound sweeps using deep learning in resource-limited countries. Ultrasound Med Biol 2019;45:773–785.

World Health Organization. UNICEF, UNFPA, World Bank Group, the United Nations Population Division. Maternal mortality: Levels and trends: 2000 to 2017. Geneva: WHO; 2019.

Yang X, Yu L, Li S, Wen H, Luo D, Bian C, Qin J, Ni D, Heng PA. Towards automated semantic segmentation in prenatal volumetric ultrasound. IEEE Trans Med Imaging 2019;38:180–193.

Yushkevich PA, Gao Y, Gerig G. ITK-SNAP: An interactive tool for semiautomatic segmentation of multi-modality biomedical images. 38th Annu Int Conf. IEEE Eng Med Biol Soc (EMBC) 2016;3342–3345.