Segmentation of Fetal Adipose Tissue Using Efficient CNNs for Portable Ultrasound

Sagar Vaze and Ana I. L. Namburete(✉)

Institute of Biomedical Engineering, Department of Engineering Science, University of Oxford, Oxford, UK
ana.namburete@eng.ox.ac.uk

Abstract. Adipose tissue mass has been shown to have a strong correlation with fetal nourishment, which has consequences on health in infancy and later life. In rural areas of developing nations, ultrasound has the potential to be the key imaging modality due to its portability and cost. However, many ultrasound image analysis algorithms are not compatibly portable, with many taking several minutes to compute on modern CPUs.

The contributions of this work are threefold. Firstly, by adapting the popular U-Net, we show that CNNs can achieve excellent results in fetal adipose segmentation from ultrasound images. We then propose a reduced model, `U-Ception`, facilitating deployment of the algorithm on mobile devices. The `U-Ception` network provides a 98.4% reduction in model size for a 0.6% reduction in segmentation accuracy (mean Dice coefficient). We also demonstrate the clinical applicability of the work, showing that CNNs can be used to predict a trend between gestational age and adipose area.

1 Introduction

Ultrasound has the potential to be the key imaging modality in rural areas of developing nations, due to its low cost and portability. To complement this portability, there is a need for image analysis tools which are similarly mobile, allowing them to be implemented alongside the imaging itself in remote locations. The most practical mode of deployment would be an application on an iOS or Android device, such as a tablet or mobile phone, which typically come with hardware limitations such as smaller RAM and less powerful GPUs.

However, current ultrasound analysis techniques are not compatibly efficient, with many taking several minutes to run on modern CPUs [1]. Furthermore, most convolutional neural networks (CNNs) - which are currently state-of-the-art in image analysis tasks - require too much memory to deploy on a mobile phone or tablet. In response to this, we propose a novel CNN architecture, `U-Ception`, which uses *depth-wise separable convolutions* to analyze ultrasound images in a computationally efficient manner.

A. Melbourne and R. Licandro et al. (Eds.): DATRA/PIPPI 2018, LNCS 11076, pp. 55–65, 2018.
https://doi.org/10.1007/978-3-030-00807-9_6

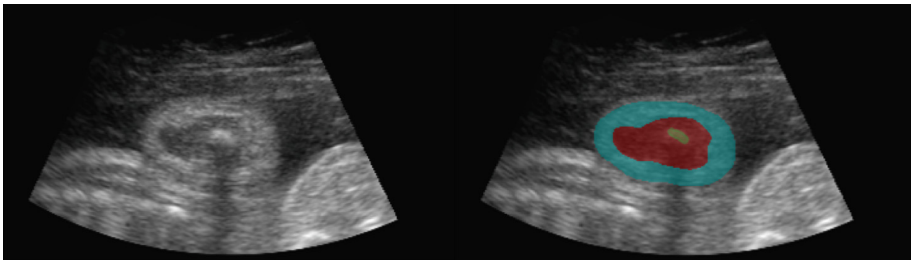


Fig. 1. Cross-sectional image of fetal a arm (left), with segmentation (right). Segmentation shows adipose tissue (of interest, blue), muscle (red), and humerus (green). (Color figure online)

The target application is the segmentation of fetal adipose tissue, as shown in Fig. 1. It has been shown that adipose mass has a 'pronounced sensitivity' to maternal - and thus fetal - nutritional state [2]. This is of special importance in the developing world, where 152 of the world's 155 million stunted under-five year olds reside [3]. Thus, the observation and control of fetal nourishment is crucial: developmentally, the most important time for proper nourishment is in the first 1000 days (from conception until the 2nd birthday), and catch-up growth in later childhood is 'minimal' [4].

This work presents the `U-Ception` network: a CNN designed for segmentation of adipose tissue in fetal ultrasound data. Firstly, in Sect. 2, this work summarizes previous efforts at fetal segmentation, and a number of popular methods which reduce neural network size. Section 3 will then describe the CNNs proposed for this segmentation challenge. The first - an adaptation of the popular 'U-Net' [5] - provides a baseline performance against which the reduced `U-Ception` architecture can be compared. Section 4 describes the experimental set-up, and Sect. 5 outlines our results, showing the similarity in performance between the adapted U-Net and `U-Ception` models.

2 Previous Work

The current state-of-the-art in fetal adipose segmentation is the feature asymmetry approach proposed by Rueda et al. [6]. Feature asymmetry is a phase-based method, which uses points of phase congruency at specific frequencies to build an edge map which is robust to changes in contrast. Other approaches to fetal ultrasound segmentation include the use of active contours [7,8], Hough transforms [9] and multi-level thresholding [1].

However, for most biomedical image segmentation, the most prevalent algorithms are convolutional neural networks (CNNs). An important CNN is the 'U-Net' [5], which has been used extensively in the biomedical field [10,11]. The network won the ISBI neuronal segmentation challenge in 2015 by a significant margin - despite a small training set of 30 images - by performing strong data

augmentation. It is adapted in this work: first to form a baseline for the application of CNNs to fetal adipose segmentation, and then as a guide for the proposed CNN with a smaller 'size'.

Network size is defined as its memory footprint, which is directly proportional to the number of its parameters, and is the main bottleneck in the application of CNNs on mobile devices. Mobile devices typically come with between 2 GB and 4 GB of RAM, with many modern networks (especially segmentation architectures) having hundreds of millions of parameters, with sizes nearing a gigabyte.

Numerous efforts have been made to reduce neural network size by efficiently storing these parameters. Wu et al. quantized the weights of the network, learning an optimal quantization codebook using K-Means clustering [12]. Huffman coding (a lossless method of compressing data) has also been used to efficiently store network weights [13].

Another method of building a smaller network is distillation [14]. Distillation is the process of using a larger network to train a smaller network, passing on the generalization ability of the large network.

A class of techniques seeks to factorize the convolutions in the networks, breaking them down into a number of steps. One example of this was suggested by Jin et al., which, instead of convolving feature maps with 3D tensors, decomposes the process into convolutions with three one-dimensional vectors [15].

This work uses *depth-wise separable convolutions* [16], which have been shown to provide high accuracy results in the 'Xception' classification network [17]. The latest model from Google DeepLab ('DeepLab v3+' [18]) adapts the Xception network for segmentation purposes. Depth-wise separable convolutions were chosen for this work as they factorize the 3D convolution in an intuitive fashion, breaking the process into spatial and channel-wise components (see Sect. 3.2).

3 Architecture Design

3.1 Adapted U-Net

The 'U-Net' [5] was first adapted to provide a baseline performance for neural networks in the context of fetal adipose segmentation. The encoder path of the network contains two convolutional layers (13×13 kernels, see Sect. 4) followed by max-pooling, repeated 4 times, resulting in a reduction of spatial channel dimensions by a factor of 16. With each down-sampling layer the number of feature channels is doubled, with 48 channels in the first layer, and 768 channels in the lowest. The decoder is symmetrical, but with up-sampling in place of max-pooling. The final layer is a 1×1 convolutional layer.

All convolutional layers were zero-padded, with all but the final layer using the ReLU non-linearity. The final layer uses a sigmoidal activation to map network predictions to values between 0 and 1, with scores close to 1 indicating a confident prediction of adipose tissue at a pixel location.

3.2 Reduced U-Net: `U-Ception`

This section describes the efforts made to reduce the number of parameters in the segmentation network, and hence the size of the model's parameter file. The proposed method uses *depth-wise separable convolutions*, which were used successfully in the 'Xception' network [17]. These convolutions were applied to the U-Net architecture, with the resulting architecture termed `U-Ception`.

The proposed architecture is essentially identical to the adapted U-Net, but with more feature channels per layer, and all convolutional layers replaced with separable convolutional layers. This modification leads to a drastic reduction in the network's parameter count, from **296 million** to **4.6 million** parameters. The architecture is detailed in Fig. 2.

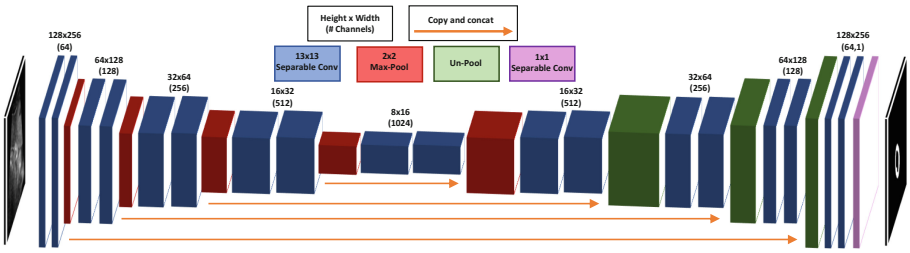


Fig. 2. `U-Ception` architecture

Depth-Wise Separable Convolutions: Regular convolutional layers in CNNs involve convolutions with three-dimensional kernels. Two of these dimensions are spatial, and are responsible for combining data from a single channel, in a similar manner to convolution filters in classical image processing. The third dimension, however, is responsible for combining information from all of the feature channels, such that new feature maps can be produced. The number of parameters in the convolution tensor for a layer, therefore, can be described by Eq. 1. Here, K is the spatial dimension of the square filter, N is the number of input channels, and M the number of output channels. Note that K^2N parameters are required to compute each of the M output feature maps. The process is illustrated in Fig. 3(a).

$$n_{parameters} = K^2NM \qquad K, N, M \in \mathbb{Z}^+ \tag{1}$$

The idea behind depth-wise separable convolutions is to separate the convolutions in the spatial dimensions and the channel dimension. First, one feature map is calculated per input channel by spatially convolving each input channel with a single filter. Next, the output is fed to a regular convolutional layer with $1 \times 1 \times N$ kernel size, so the information across input channels can be combined. In this way, the multiplicative interaction between the N input channels and M

output channels is not scaled by the squared spatial kernel size, K^2. The number of parameters in this new layer is described by Eq. 2.

$$n_{parameters} = K^2 N + \bar{K}^2 NM = K^2 N + NM \qquad K, N, M \in \mathbb{Z}^+ \qquad (2)$$

Note that the variable $\bar{K} = 1$ is introduced to illustrate that the second stage of convolutions is identical to a regular convolutional layer with a spatial kernel size of 1. Also, in some implementations, a channel multiplier C_m is introduced such that, in the spatial convolution stage, C_m intermediate feature maps are produced per input channel. This would scale the number of parameters in the depth-wise separable layer by C_m. In this work, a channel multiplier of 1 is used. The process is shown in Fig. 3(b).

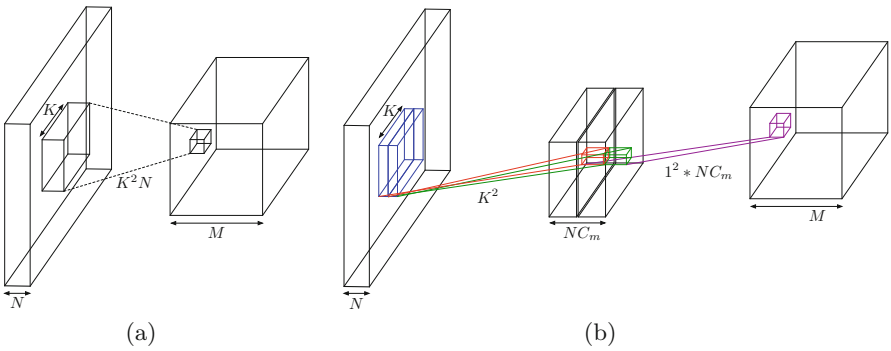


Fig. 3. (a) Regular convolutional layer. Here convolution occurs with a tensor with square spatial dimensions K, and depth equal to the number of input channels, N. Each of the M filters requires K^2N parameters. (b) Separable convolutional layer. Here each input feature channel is convolved separately with C_m tensors with depth of 1 and spatial dimensions of K. The resulting feature maps are convolved with M tensors of depth NC_m and spatial dimensions of 1. In this work, layers with $C_m = 1$ are used.

4 Experimental Setup

4.1 Fetal Dataset

Data for this task was collected as part of the INTERGROWTH-21st Project, with 324 3D ultrasound volumes of healthy fetal arms acquired. From each volume, five 2D slices were extracted perpendicular to the humerus and annotated by one of three experts, delineating the adipose tissue, as shown in Fig. 4. The images were collected with a Philips HD9 ultrasound machine (resolution of 0.99 mm per voxel), with the subjects' gestational ages ranging from 17 to 41 weeks. The dataset is a larger sample of that used by Rueda et al. [6] (the previous effort at fetal adipose segmentation).

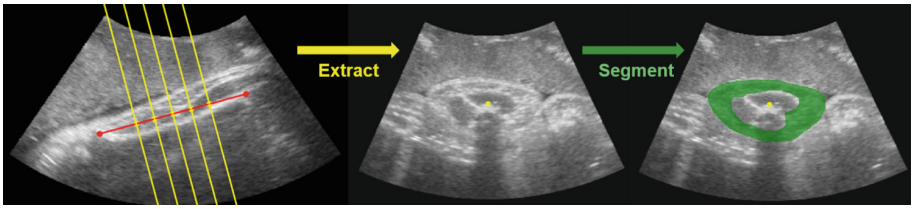


Fig. 4. Extraction and segmentation of 2D slices from an ultrasound volume. Left-most image shows a sagittal view of the fetal humerus. Red points show humerus end points and yellow lines indicate slice planes. Intersections of the red and yellow lines - the yellow points - show centers of extracted slices. Also shown is one extracted slice (middle) and its segmentation (right). (Color figure online)

The dataset was divided with an 80–20 split into folds for training and testing respectively. The training set was further broken down, with 20% of the 2D slices used as a validation set, on which network hyper-parameters were tuned. The training, validation and test sets had 1100, 270 and 340 slices respectively, and all slices were resized to 128×256 pixels.

4.2 Implementation Details

Both CNNs were optimized by maximizing the following function:

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y}, \hat{\boldsymbol{y}}) = -\lambda_1 ||\boldsymbol{\theta}||_2 + \sum_{i=1}^{n} IoU(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}) + \lambda_2 h(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}) \tag{3}$$

Here, $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ represent the manual training labels and the network predictions respectively, while $\boldsymbol{\theta}$ represents the CNN parameters. IoU represents the intersection-over-union score of the manual labels and the network predictions, with λ_1 signifying the weight decay strength. The function h represents a boundary regularizer, which explicitly penalizes incorrect network predictions at the adipose boundaries.

Optimization was done with stochastic gradient descent, reducing the learning rate by a factor of 10 every 15 epochs. Both networks were implemented using Keras (TensorFlow backend), with training done on an NVIDIA Quadro P5000 GPU. Interestingly, independent optimization of both the adapted U-Net and the `U-Ception` models showed that both networks had identical optimal hyper-parameter settings. An initial learning rate of 1×10^{-2} was used, with λ_1 (weight decay) set to 1×10^{-2}, and λ_2 (boundary regularizer strength) to 1×10^{-3}. Furthermore, batch normalization was used on all layer inputs, and dropout regularization was used on the input layer and lowest layer ($p = 0.2$ and $p = 0.5$ respectively) as in the original U-Net. Kernels of size 13×13 were used in both networks to deal with the large areas of adipose discontinuity in the images (a product of ultrasound shadows).

5 Results and Discussion

This section compares the performances of the regular convolution U-Net and `U-Ception` networks on a held-out test set of 340 slices (extracted from 68 volumes).

Sample qualitative results are shown in Fig. 5. It can be seen that the networks generally capture the adipose tissue well, with both learning to predict closed-ring segmentations, even in the presence of adipose signal occlusion (for instance, the vertical shadow below the humerus). Failure modes are also shown (Dice coefficient < 0.5), with failure occurring in the presence of a sparse signal (Example 6), or when the target slice has many distracting shapes (Example 7).

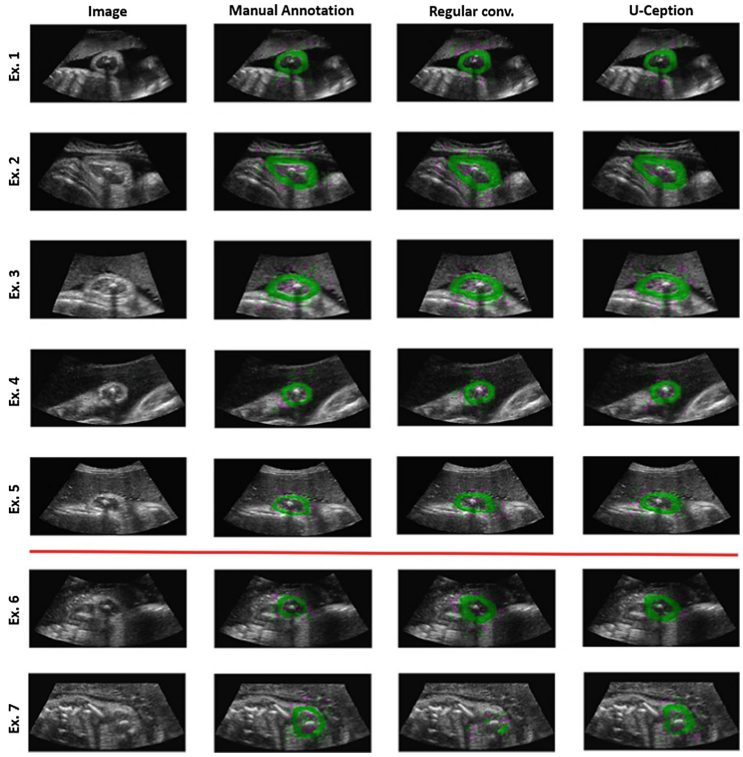
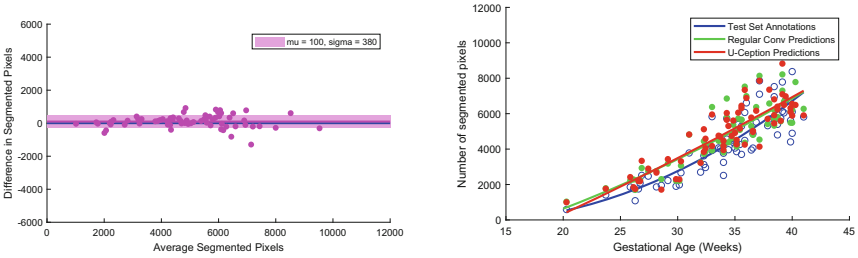


Fig. 5. Sample results from the CNNs on a held-out test set. Failure modes are shown in Examples 6 and 7. Note that a disproportionate number of failure modes are shown.

A chi-square test was performed on the Dice coefficients produced by the U-Net and `U-Ception` models on the test set. It was found that there is no statistically significant difference between the `U-Ception`'s Dice distribution and that of the regular U-Net ($p \approx 1.00$). Though this p value is high, it is perhaps unsurprising given the visual similarities of the two models' results (see Fig. 5).



(a) (b)

Fig. 6. (a) Bland-Altman plot of the number of segmented pixels in `U-Ception` model against regular convolution model. (b) Adipose area trend (in number of segmented pixels) with respect to gestational age using test set volumes.

Further insight into performance of the models can be gained by inspecting Bland-Altman plots of the number of segmented pixels in both models' predictions. Figure 6(a) shows a plot of the `U-Ception` architecture compared against the regular convolution U-Net. This diagram suggests strongly that there is little difference in the predictions of the two models; it shows very tight standard deviation bounds on the difference between the number of segmented pixels, and a similarly small mean difference ($\mu = 100, \sigma = 380$).

An example of the clinical applications of the algorithms is given in Fig. 6(b), where the trend of adipose area against gestational age is given. Here, the trends computed using the manual annotations and CNN results are given for all volumes in the test set. The similarity between the manual and CNN trends is evident, as is the similarity of the trends between the CNNs.

5.1 Comparison with Previous Work

The results of this work are quantitatively compared with the previous efforts by Rueda et al. [6] in Table 1. Here, the accuracy (sensitivity and specificity) and Dice are detailed. To contextualize the results, it should be noted that the images fed to the algorithm by Rueda et al. were heavily cropped to contain only the area of interest. This makes the task of adipose localization easier, contributing to the higher mean Dice coefficient achieved by the previous work. It also increases the foreground-to-background ratio, contributing to the higher specificities achieved by the CNNs. It should also be noted that a larger evaluation set was used in this work - 340 slices, in contrast to the 81 slices in the work by Rueda et al. - contributing to the larger standard deviations in our results.

Nonetheless, the classical algorithm outperforms the CNNs in terms of Dice coefficient, while the CNNs achieve better results with respect to both accuracy metrics. Also, the `U-Ception` network gives a small but not statistically significant compromise in performance when compared against the regular convolution U-Net.

Table 1. Comparison of method by Rueda et al. [6] against the proposed CNNs ($\mu \pm \sigma$).

	Sensitivity (%)	Specificity (%)	Dice (%)
Rueda et al.	87.30 ± 3.84	97.05 ± 1.17	$\mathbf{87.11 \pm 2.60}$
Regular conv.	$\mathbf{88.29 \pm 12.15}$	$\mathbf{98.85 \pm 0.75}$	80.89 ± 13.75
`U-Ception`	87.45 ± 13.30	98.71 ± 0.83	80.25 ± 11.50

5.2 Comparing Algorithm Efficiencies

Table 2 summarizes the two model sizes and prediction times of both networks on a CPU and on a variety of portable devices. The CPU prediction times are averaged over 100 samples. The times shown compare favorably with those required for classical techniques - many of which take several minutes to run on a modern CPU [1].

Table 2. Model sizes of both networks, as well as prediction times on a range of hardware. Note that the regular convolution model was too large to deploy on the mobile devices.

	Model size	i5-4200M CPU	Google Pixel 2	Samsung Galaxy S5	Samsung Galaxy Tab A
Regular conv.	1.10 GB	15.8 s	N/A	N/A	N/A
`U-Ception`	18 MB	2.7 s	≈4 s	≈8 s	≈11 s

The table also shows that, with modern hardware, even the large network can make a prediction in reasonable time (15.8 s), as the number of FLOPs required for a forward pass rises only linearly with the number of parameters in the network. Thus, the main bottleneck in implementation of these networks on a mobile device is clarified: the size of the weight file. Typically, TensorFlow stores each parameter as a 32-bit float, meaning a network with 20 million parameters will have a weight file of approximately 75 MB in size. The regular convolution U-Net has **296 million** parameters, with the resulting weight file taking **1.10 GB** on disk. The `U-Ception` architecture requires only **4.6 million** parameters, with a weight file of **18 MB**. Thus the model provides a reduction in both weight file size and parameter count of **98.4%**, while achieving similar performance on the test set (with a 0.6% compromise in mean Dice coefficient).

6 Conclusion

This work proposes an end-to-end framework for the semantic segmentation of fetal adipose tissue using convolutional neural networks. Furthermore, a highly efficient novel network architecture - `U-Ception` - is proposed, using depth-wise separable convolutions to reduce model parameter count. It is shown that the `U-Ception` architecture's performance is statistically equivalent to that of the regular convolution U-Net, with the benefit of a 98.4% reduction in model size.

Acknowledgements. The authors are grateful for support from the Royal Academy of Engineering under the Engineering for Development Research Fellowship scheme, and the INTERGROWTH-21st Consortium for provision of 3D fetal US image data.

References

1. Rueda, S., et al.: Evaluation and comparison of current fetal ultrasound image segmentation methods for biometric measurements: a grand challenge. IEEE Trans. Med. Imaging **33**(4), 797–813 (2014)
2. Symonds, M.E., Mostyn, A., Pearce, S., Budge, H., Stephenson, T.: Endocrine and nutritional regulation of fetal adipose tissue development. J. Endocrinol. **179**(3), 293–299 (2003)
3. UNICEF: Joint Malnutrition Estimates 2017 - UNICEF Data and Analytics
4. Lloyd-Fox, S., et al.: Functional near infrared spectroscopy (fNIRS) to assess cognitive function in infants in rural Africa. Nat. Sci. Rep. **4**, 1–8 (2014)
5. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
6. Rueda, S., Knight, C.L., Papageorghiou, A.T., Alison Noble, J.: Feature-based fuzzy connectedness segmentation of ultrasound images with an object completion step. Med. Image Anal. **26**(1), 30–46 (2015)
7. Chalana, V., Winter, T.C., Cyr, D.R., Haynor, D.R., Kim, Y.: Automatic fetal head measurements from sonographic images. Acad. Radiol. **3**(8), 628–635 (1996)
8. Pathak, S.D., Chalana, V., Kim, Y.: Interactive automatic fetal head measurements from ultrasound images using multimedia computer technology. Ultrasound Med. Biol. **23**(5), 665–673 (1997)
9. Lu, W., Tan, J., Floyd, R.: Automated fetal head detection and measurement in ultrasound images by iterative randomized hough transform. Ultrasound Med. Biol. **31**(7), 929–936 (2005)
10. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
11. Ravishankar, H., Venkataramani, R., Thiruvenkadam, S., Sudhakar, P., Vaidya, V.: Learning and incorporating shape models for semantic segmentation. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 203–211. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_24

12. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: CVPR (2016)
13. Han, S., Mao, H., Dally, W.J.: Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. In: ICLR (2016)
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: NIPS Deep Learning Workshop (2015)
15. Jin, J., Dundar, A., Culurciello, E.: Flattened convolutional neural networks for feedforward acceleration. In: ICLR (2015)
16. Sifre, L., Mallat, S.: Rigid-motion scattering for image classification. Ecole Polytechnique, CMAP. Ph.D. thesis (2014)
17. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: CVPR (2016)
18. Chen, W., et al.: Compressing neural networks with the hashing trick. ICML (2015)