Label Efficient Localization of Fetal Brain Biometry Planes In Ultrasound Through Metric Learning

Yuan Gao[1](✉), Sridevi Beriwal[2], Rachel Craik[2,3], Aris T. Papageorghiou[2] and J.Alison Noble[1]

[1] Institute of Biomedical Engineering, University of Oxford, Oxford, UK
yuan.gao2@eng.ox.ac.uk
[2] Nuffield Department of Women's and Reproductive Health, University of Oxford, Oxford, UK
[3] King's College, London, UK

Abstract. For many emerging medical image analysis problems, there is limited data and associated annotations. Traditional deep learning is not well-designed for this scenario. In addition, for deploying deep models on a consumer-grade tablet, it requires models to be efficient computationally. In this paper, we describe a framework for automatic quality assessment of freehand fetal ultrasound video that has been designed and built subject to constraints such as those encountered in low-income settings: ultrasound data acquired by minimally trained users, using a low-cost ultrasound probe and android tablet. Here the goal is to ensure that each video contains good neurosonography biometry planes for estimating the head circumference (HC) and transcerebellar diameter (TCD). We propose a label efficient learning framework for this purpose that it turns out generalises well to unseen data. The framework is semi-supervised consisting of two major components: 1) a prototypical learning module that learns categorical embeddings implicitly to prevent the model from overfitting; and, 2) a semantic transfer module (to unlabelled data) that performs "temperature modulated" entropy minimization to encourage a low-density separation of clusters along categorical boundaries. The trained model is deployed on an Andriod tablet via TensorFlow Lite and we report on real-time inference with the deployed models in terms of model complexity and performance.

Keywords: Few-shot Learning · Portable Ultrasound · MobileNet.

1 Introduction

Fetal brain biometry measurements, such as estimation of the head circumference (HC) and transcerebellar diameter (TCD), are of great clinical importance to assess fetal growth. In this study, we investigate automated identification of the relevant fetal biometry planes, namely the transthalamic (TT) and transcerebellar (TC) planes of the fetal head. Of note, fetal brain videos used in this

study are acquired via a portable low-cost ultrasound probe and the solution is designed to be used by minimally trained healthcare staff. Such a system offers a number of benefits compared to a traditional ultrasound machine: it is affordable, flexible and, due to the limited training required, more scalable than traditional approaches. This may be of particular relevance to underserved regions where prenatal ultrasound may be limited or non-existent. However, one challenge this introduces is that video quality varies and the quality of the standardised biometry planes (TC and TT) may not be high. Other technical challenges presented by this application include that manual annotation of video is time-consuming and requires clinical expertise which is a scarce resource. Further, we are dealing with a dense labelling task (as shown in Fig.1) with a few annotated training samples-only two planes (one transcerebellar (TC) and one transthalamic (TT)) per video are annotated from which the TCD and HC are measured, respectively.

Contributions: To cope with these challenges (1) we propose a new quality control protocol by adopting and adjusting the standard criteria [1] for characterising fetal brain biometry planes; (2) we introduce a metric learning based method to automatically identify and index the TC and TT planes which builds a model by learning from a few annotations; (3) finally, we deploy the trained models within a prototype ultrasound video acquisition system on an Android tablet and investigate the trade-off between model complexity and performance.

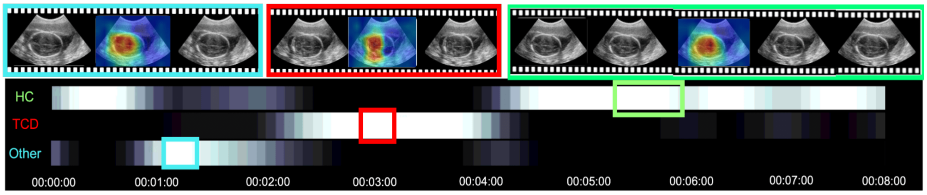


Fig. 1: Identifying and indexing good quality fetal brain biometry planes. Bottom "colorbars" depict the labelling confidence distributed over the whole video (8 seconds). The brighter the higher likelihood that a target plane is captured. HC: Planes good for measuring head circumference, TCD: Planes good for measuring transcerebellar diameter, Other: Planes not suitable for biometry.

Related Works: Recently, deep learning based approaches have shown success in recognition of fetal brain planes [2], [3], [4], [5] and [6]. Baumgartner et al.[2] use a variant of VGG-Net, called SonoNet, for classification of 13 standardised fetal biometry planes, including TC and TV planes in fetal brain which was improved by [3] by introducing an attention mechanism. Gao et al.[5] model multi-scale spatial-temporal attention for detecting and tracking fetal structures, including the fetal head, but different to [2] and [3], does not specifically look at standard fetal biometry planes. Other works, such as [4] and [6], localize fetal brain standard planes in 3D ultrasound. Y. Li et al. [4] uses a CNN to regress a rigid transformation iteratively for localising TC and TV planes in 3D fetal ultrasound. Dou et al.[6] propose to localize TT and TV planes in 3D fetal

ultrasound with a reinforcement learning framework, which can progressively interact with the volumes and modify the search trajectory towards the target planes. Yaqub et al. [7] consider automatic quality assurance of the TV plane when it has been acquired. In contrast to the above works considering standard fetal plane detection only, our concern is to detect planes and to explicitly verify that they are suitable for measurement. A further commonality of all the above methods is that they require supervision with a large amount of labelled training data. By contrast, we seek to learn how to select candidate image frames from a few annotated examples.

2 Method

We propose a semi-supervised learning framework, shown in Fig. 2, which consists of a feature extractor (CNN), a prototypical learning module and a semantic transfer module. We formulate the learning as a multi-way classification problem and the learned model is employed to automatically label unseen video frames in a dense manner. For each training iteration, we randomly sample a query set χ^Q and a small support set χ^S (few images per class) from n labelled frames, and an unlabelled set χ^U from m unlabelled frames (n$\ll$m).

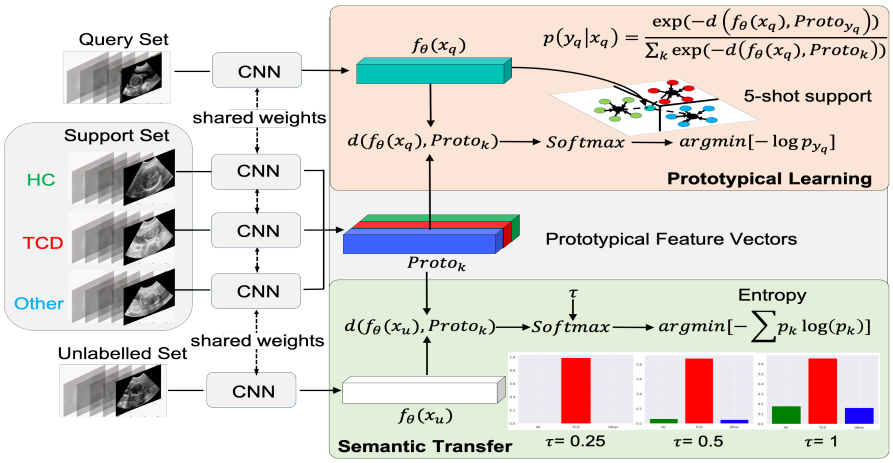


Fig. 2: Overview of our proposed learning framework. Notations: f_θ, feature descriptor; $d(\cdot)$, euclidean distance; k, class index; p, posterior probability; $Proto_k$, prototypes for k-th class; τ, temperature.

CNN Architecture: Considering the typical computational resources on a current consumer tablet, we employ current state-of-the-art light-weight deep CNNs, such as [10], [11] and [12], in this study. For instance, MobileNet [10] introduces depth-wise separable convolution which significantly reduces the computational cost and the number of trainable parameters of convolution layers. This is achieved by applying a channel-wise convolution first followed by 1×1 point-wise convolution to linearly combine the feature maps across the channels.

MobileNetV2 [11] improves on the MobileNet design by introducing inverted bottleneck layers and residual connection between the bottlenecks. The inverted bottleneck layers project features to a higher dimensional manifold that [11] argues prevents information loss from ReLU activation and the residual blocks facilitate the gradient back-propagation through a deep network. MobileNetV3 [12] upgrades the inverted bottleneck layers by introducing the squeeze and excitation in the residual layer which explicitly models channel interdependences and re-weights the feature importance.

Learning from Prototypes: As we only have a small number of labelled frames, the deep CNNs mentioned above are prone to overfitting when explicitly learning the classification. Instead, we sample few-shot examples (a support set) from each class and feed them through a CNN f_θ first to compute categorical prototypes [9]. Given a support set of N_S frames $x_s \epsilon \chi^{S_k}$ from the k^{th} class, prototype $Proto_k$ is computed as the mean of the embedded D-dimensional feature vectors $f_\theta(x_s)\epsilon\mathbb{R}^D$:

$$Proto_k = \frac{1}{N_S} \sum_{x_s \epsilon \chi^{S_k}} f_\theta(x_s) \tag{1}$$

Then, we sample a query set of N_Q labelled frames (larger than the support set). For each query point (x_q, y_q), where $x_q \epsilon \chi^Q$ and $y_q \epsilon \{1, ..., K\}$ (K=3 in this study), we measure its similarity to each prototype by the Euclidean distance and produce a probability distribution over all classes with a distance based softmax. We can then calculate the metric based cross-entropy loss:

$$L_{metric}(\chi^S, \chi^Q) = -\frac{1}{N_Q} \sum_{\{x_q, y_q\} \epsilon \chi^Q} log \frac{exp(-\left\| f_\theta(x_q) - Proto_{y_q} \right\|^2)}{\sum_{k=1}^{K} exp(-\left\| f_\theta(x_q) - Proto_k \right\|^2)} \tag{2}$$

With the loss, we introduce the constraint that data from the same class should be similar in the embedding space. Also, we find that this loss can provide a guidance to stabilize the unsupervised semantic transfer (described next).

Unsupervised Semantic Transfer We define a new semantic transfer objective, $L_{ST}(\chi^U)$, which transfers information from the prototypes produced above to unlabelled datapoints by minimizing the entropy of a temperature-tuned softmax. Entropy minimization has been widely used for unsupervised [15], [14] and semi-supervised [16] learning by encouraging low density separation between classes. At each training epoch, we sample an unlabelled set χ^U of N_U frames ($N_U = N_Q$), then for each datapoint in the unlabelled set $x_u \epsilon \chi^U$ we again compute the Euclidean distance between its feature embedding and each categorical prototype. Our semantic transfer loss is then defined as:

$$L_{ST}(\chi^U) = -\frac{1}{N_U} \sum_{x_u \epsilon \chi^U} \sum_{k \epsilon \{1...K\}} \tau^{-1} P(x_u, Proto_k) log \tau^{-1} P_\tau(x_u, Proto_k) \tag{3}$$

Where $P(x_u, Proto_k)$ is a softmax function that generates a probability distribution over all classes based on: $-\left\| f_\theta(x_u) - Proto_k \right\|^2$ and τ is the temperature

of the softmax. We then tune the softmax temperature. As shown in Fig.2, the distribution becomes one-hot when the temperature approaches 0, whereas becomes more uniformly distributed when increasing the temperature. Intuitively, a small temperature encourages each unlabelled frame to be very similar to one class, whereas a larger temperature will allow it to be similar to multiple classes.

Joint Learning In addition to the above, we introduce another fully-connected layer that maps the feature vectors of query frames to class scores directly. This serves as an auxiliary classifier, trained with a cross-entropy (CE) loss, allowing us to investigate the interactions between direct learning and metric learning. To mitigate overfitting, we introduce training signal annealing (TSA) [17] to the cross-entropy loss:

$$L_{CE}(\chi^Q) = -\frac{1}{N_Q} \sum_{\{x_q, y_q\} \epsilon \chi^Q} [-I\{P_\theta(y_q|x_q) < \eta_t\}] \, log P_\theta(y_q|x_q) \tag{4}$$

Where $I\{.\}$ is the indicator function and $P_\theta(y_q|x_q)$ is the probability of x_q belonging to the class y_q. Specifically, the example (x_q, y_q) does not contribute to the loss function if the model predicted probability surpasses a threshold η_t, at training step t. We set $\eta_t = exp((\frac{t}{T} - 1) * 5) * (1 - \frac{1}{K}) + \frac{1}{K}$ that corresponds to the exponential schedule in [17] realising most of the supervised signal at the end of training. Intuitively, this is to prevent model from overfitting too quickly by penalizing over confident prediction in the early stage of training. Finally, our model jointly optimizes over the objective function as follows:

$$L(\chi^S, \chi^Q, \chi^U) = L_{ST}(\chi^U) + \alpha L_{metric}(\chi^S, \chi^Q) + \beta L_{CE}(\chi^Q) \tag{5}$$

where the hyperparameters α and β determine the influence of the metric learning and the direct learning, respectively.

3 Experiments and Results

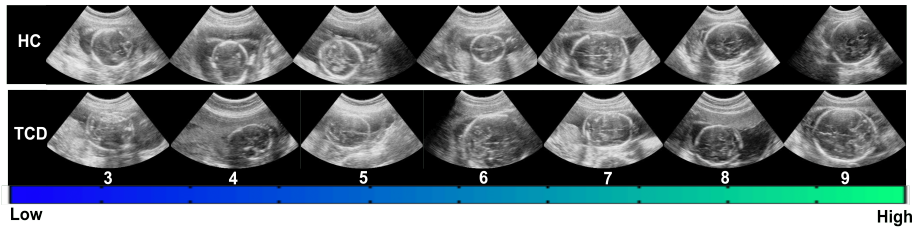


Fig. 3: Quality assessment. Scoring a **TCD** plane [1-9]: image horizontal*[1], 30% magnification†[1], symmetrical hemispheres†[1], CSP‡(clear:[2], suspected:[1]), thalami‡(clear:[2], suspected:[1]), cerebellum edge‡(clear:[2], unclear:[1]). For a **HC** Plane [1-9]: all same as TCD except for No cerebellum visible†[1], HC oval*[1]. *: new criteria; †: existing criteria; ‡: revised criteria.

Datasets and Annotation The ultrasound videos used in this study are acquired by a wireless portable probe (Konted GEN 1 C10R) with which we acquire

a set of three short fetal brain videos (8sec/video) for each subject. The goal is to identify the TT plane and TC planes that have good enough quality to measure the HC and TCD, respectively. However, we found that the standard criteria [1] for scoring plane quality do not apply well to our cases; for example, the appearance of key anatomies, such as cavum of the septum pellucidum (CSP), thalamus and cerebellum can be difficult to see. Therefore, we developed a revised criteria scoring system more customized to the appearance of typical video frames, as described in Fig. 3. Two experienced sonographers annotated three frames only per video (TCD, HC and background) and scored them. For training, we choose the annotated frames having a quality score ($\geqslant 6$). This resulted in: 301 HC frames, 208 TCD frames, 120 Other (background) frames and 33,957 unlabelled frames from 441 training videos (147 subjects). For testing, a further 60 videos (20 subjects) were manually labelled frame by frame over the whole video for HC, TCD and background.

Implementation Details Firstly, a MobileNetV1 [10] was trained as a baseline model with CE loss only, on all labelled training data (629 images). Secondly, we added TSA to the CE loss (Eq. 4) to derive a second model. Finally, we introduced the L_{metric} (Eq.2) and the L_{ST} (Eq.3) for joint learning and the number of support, query and unlabelled images was chosen as N_S=15 (5 images/class), N_Q=45 (15 images/class) and N_U=45, respectively for each training iteration. We adjusted the signal weights α and β in this way: firstly, we set α to 1, and gradually reduce β from 1 to 0.5, 0.1 until 0 (no CE signal); then, we do the opposite and set β to 1, and gradually reduce the α to 0.5, 0.1 until 0 (no metric learning signal). We set the temperature $\tau = 0.5$ in Eq.3 and the total number of training iterations (T) to 20,000. All models were trained using an Adam optimizer and with learning rate 10^{-4} and all the experiments were conducted with the TensorFlow 1.14.0. We also experiment with MobileNetV2 [11] and V3 [12] and study the trade-off between performance and capacity of models by adjusting the width multiplier (0.25, 0.5, 0.75 and 1 used). Finally, we deploy the trained models onto a Huawei tablet (MediaPad M5 lite10) via the TensorFlow Lite API to study its inference in terms of computational cost and performance. We firstly freeze the trained models and save the frozen graphs as .pb files. Then we convert the .pb files into .tflite for on device deployment.

Categories \ Models	Baselines		α:β 1:1	α=1			β=1		
	CE	CE_{TSA}		β=0.5	β=0.1	β=0	α=0.5	α=0.1	α=0
TCD	0.459	0.547	0.673	0.735	**0.746**	0.740	0.639	0.614	0.611
HC	0.595	0.643	0.792	0.859	**0.897**	0.889	0.784	0.754	0.742
mAP	0.525	0.618	0.778	0.834	**0.869**	0.837	0.783	0.729	0.713

Table 1: Performance measures over different learning configurations on the test-set (CNN Backbone: MobileNetV1, Width multiplier:1).

Effect of Metric Learning We evaluate the models using Average Precision (AP) measured on frame level labels. For each class, we count a correct detection

if it is a positive prediction with confidence above a certain thresholds (ranging from 0.1 to 0.9). We report AP for the individual classes as well as mean Average Precision (mAP) in Tab.1. We found directly learning with CE loss can result in poor generalization to test data as indicated by the baseline model (CE) which is the worst among all models. Note that there is an improvement over all metrics after applying TSA to the CE loss but it is marginal. Moreover, we found that L_{metric} (Eq.2) plays a crucial role in improving model generalization. When applying a full metric learning signal (i.e. fixing α=1), all metrics increase when reducing the contribution of the cross-entropy loss (i.e. reducing β) (Eq. 5).

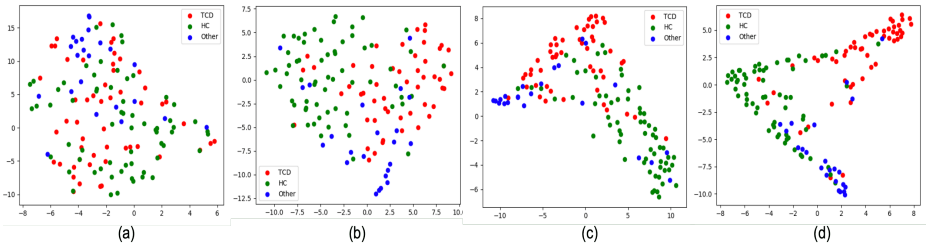


Fig. 4: t-SNE embeddings of global features (MobileNetV1 backbone) on test frames. (a):baseline (CE) (b):baseline (CE_{TSA}) (c): joint learning (α=1, β=0.5) (d): joint learning (α=1, β=0.1)

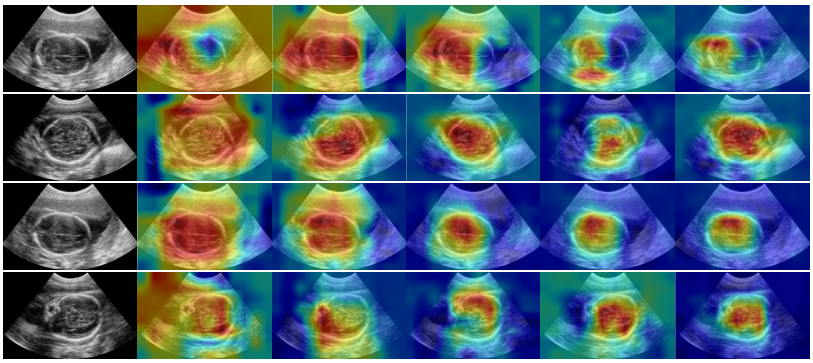


Fig. 5: Examples of TC (**Row 1 and 2**) and TT (**Row 3 and 4**) planes captured by different models. **Column 1**: example clips. **Column 2 to 6**: corresponding CAMs obtained from CE, CE_{TSA}, joint learning ($\alpha = 1$, $\beta = 1$), joint learning ($\alpha = 1$, $\beta = 0.5$), joint learning ($\alpha = 1$, β=0.1), respectively.Best viewed in Adobe Reader. All videos should play automatically.

The best performance is achieved when β equals to 0.1. whereas, when applying a full cross-entropy signal (i.e. fixing β=1), the performance of models overall is less superior than for models trained with a full metric learning signal.

Of note, all metrics drop in value when gradually reducing the contribution of the metric learning loss from 0.5 to 0.1 to 0. We also found, overall, that there is a higher AP for the HC than TCD and this may be because some low quality TCD frames (with unclear cerebellar edges) are prone to be confused as a HC frame. We have also performed t-SNE (t-Stochastic Neighbour Embedding) on the test dataset, summarized in Fig.4: feature embedding (feature vectors after global pooling) visualisation. We found that the baseline model (CE) is severely overfitted as depicted in Fig. 4(a), that categorical clusters are not formed at all and datapoints are distributed randomly. Of note that in Fig.4(b) a separation between classes can be seen after applying TSA to the CE loss. However, there is still no clear separation between the classes and all datapoints are still clustered together. In contrast, categorical clusters in Fig.4 (c) and (d) are formed well with clear separation between the classes with the joint learning. We also observe that the stronger the metric learning signal, the stronger the "forces" that push the clusters away from each other.

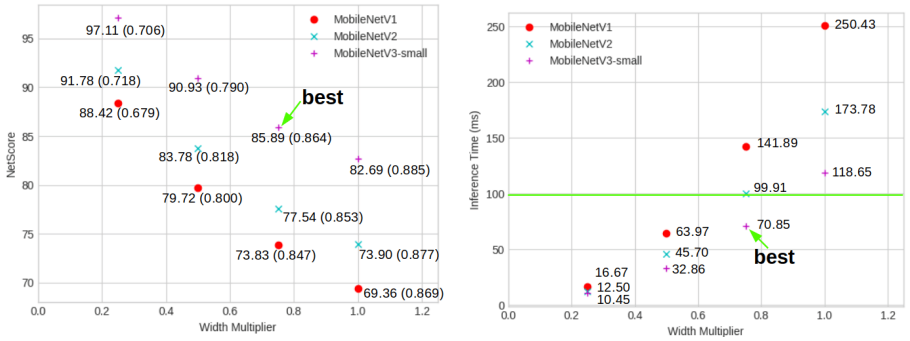
In Fig.5 , we give examples of HC and TCD images that have been identified by different models and class activation mappings (CAMs). We linearly combine the feature maps before global pooling with score mapping weights to produce frame-wise class activation maps. We found that CAMs produced by CE and CE_{TSA} are extremely random indicating that the models do not learn where the discriminative features are. However, the joint learning models produce more discriminative CAMs and learn to highlight a variety of key anatomical structures, such as cerebellum on TC planes and thalamus on TT planes.

On-Device Inference To evaluate models' on device performance, as shown in Fig.6a, we estimate the NetScore metric $\Omega = 20log_{10}(mAP^{\gamma}p^{-\delta}c^{-\epsilon})$ [18] which measures model efficiency as a trade-off between mean average precision (mAP, percent), number of trainable parameters (p, millions) and number of float operations (FLOPs, billions). We set γ, δ and ϵ to be 2, 0.5 and 0.5 respectively, based on [18]. We construct smaller and less computationally expensive models by a width multiplier [10]. Of note, although mAP drops continuously when networks become thinner, higher netscores are achieved because models become more compact and efficient.

Moreover, we also report inference times, as shown in Fig.6b, for a single frame on the Huawei MediaPad equipped with KIRIN659 SoC. We found the thinner the networks, the faster the inference they achieved. As our videos are recorded at a rate of 10 frames/seconds (100 ms/frame), the models below the green line in Fig.6b achieve real time inference. Considering jointly the NetScore, mAP and speed, the MobileNetV3-small with a width multiplier 0.75 (as indicated by green arrows) is the best and achieves comparable mAP (0.864) to a full capacity MobileNetV1 (0.869) but more than triple the inference speed (70.85ms compared to 250.43ms).

4 Conclusion

We have proposed a new quality control protocol for characterizing fetal brain biometry planes acquired by a low-cost portable probe. We have also demon-



(a) (b)

Fig. 6: Performance-model complexity trade-off. (a) NetScore (mAP). (b) Inference Time. All models trained with joint learning configured ($\alpha = 1$,$\beta = 0.1$).

strated the success of learning from few labelled data and the metric learning module in this framework is the key to improve the model generalization. Finally, we deployed the trained models onto a mid-end consumer tablet and have studied the trade-off between models' performance and capacity.

Acknowledgements. We acknowledge the ERC (ERC-ADG-2015 694581, project PULSE) the EPSRC (EP/GO36861/1, EP/MO13774/1) and the NIHR Biomedical Research Centre funding scheme.

References

1. T. I. S. of Ultrasound in Obstetrics and Gynecology.: Sono-graphic examination of the fetal central nervous system: guidelines for performing the basic examination and the fetal neurosonogram. Ultrasound in Obstetrics and Gynecology, 2007.
2. Baumgartner, C.F., Kamnitsas, K., Smith, S., Koch, L.M., Kainz, B., Rueckert,D., "SonoNet: Real-Time Detection and Localisation of Fetal Standard Scan Planes in Freehand Ultrasound," In IEEE TMI, 2017.
3. J. Schlemper, O. Oktay, L. Chen, J. Matthew, C. Knight, B. Kainz, B. Glocker, D. Rueckert., "Attention-Gated Networks for Improving Ultrasound Scan Plane Detection," In MIDL, 2018.
4. Y. Li, B. Khanal, B. Hou, A. Alansary, J. J. Cerrolaza, M. Sinclair, J. Matthew, C. Gupta, C. Knight, B. Kainz, D. Rueckert, "Standard Plane Detection in 3D Fetal Ultrasound Using an Iterative Transformation Network," In MICCAI, 2018.
5. Y. Gao and J.A. Noble, "Learning and Understanding Deep Spatio-Temporal Representations from Free-Hand Fetal Ultrasound Sweeps," In MICCAI, 2019.
6. H. Dou, X. Yang, J. Qian, W. Xue, H. Qin, X. Wang, L. Yu, S. Wang, Y. Xiong, P.A. Heng, D. Ni, "Agent with Warm Start and Active Termination for Plane Localization in 3D Ultrasound," In MICCAI, 2019.
7. M. Yaqub et al, "A deep learning solution for automatic fetal neurosonographic diagnostic plane verification using clinical standard constraints," In Ultrasound in medicine & biology, 2017.
8. Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. "Matching networks for one shot learning," In NIPS, 2016.

9. J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In NIPS, 2017.
10. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Sandler, M., et al.: MobileNetV2: inverted residuals and linear bottlenecks. In CVPR (2018)
12. Andrew Howard et al. Searching for mobilenetv3. arXiv preprint arXiv:1905.02244, 2019.
13. X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. arXiv:1707.01083, 2017.
14. Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In NIPS, 2016.
15. Gintautas Palubinskas, Xavier Descombes, and Frithjof Kruggel. An unsupervised clustering method using the entropy minimization. In AAAI, 1999.
16. Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. In NIPS, 2004.
17. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848 (2019).
18. Wong, A.: NetScore: towards universal metrics for large-scale performance analysis of deep neural networks for practical usage. arXiv:1806.05512 (2018).