OBUS – Multiple Gestation Model Architecture

This document describes the architecture used for the development of the Multiple Gestation model. See the GHL OBUS Github Repository [1] for an instantiation of this architecture along with more detail in its Readme file.

Design considerations

In the following sections, the terms "multiple gestation" and "twin" are used interchangeably for ease of reference. In actual practice, triplets and higher-order multiple gestations are extremely rare, and twins will represent the vast majority of the multiple gestation exams in most datasets.

Information about the relative position and count of fetal anatomical structures, such as the head, abdomen, and limbs, helps in predicting the number of fetuses. For example, if the model sees two heads, each in a different portion of one video, it is likely that these are the heads of two fetuses, and thus the model can predict that this is a multiple gestation pregnancy. However, this is not a reliable signal, because a single video may not capture, say, the heads of both fetuses, especially for exams in mid-to-late pregnancy because the fetuses are larger at that time, and the video may only show one of the fetuses.

Other signals that may indicate twins are the distinctive pattern of (portions of) two heads, two amniotic sacs, two placentas, or other 'duplicate' features seen in the same frame of the video. This is not a reliable signal either, since even in a twin pregnancy, a single blind sweep may not encounter any of these distinctive features, which are sparse and only seen in fleeting glimpses in some ultrasound sweeps.

In traditional supervised learning, every sample has an associated label and the back-propagation algorithm tries to push the output for every sample towards its target value. For the gestational age, fetal weight, and fetal presentation models developed based on the FAMLI dataset, a sample is a video. For example, for the fetal presentation model, the label would be 0 for every video from a cephalic case, and 1 for every video from a non-cephalic case. This video-level training approach gives excellent results for the models mentioned above, where practically every video contains evidence of the feature of interest.

Twin prediction based on single video training, however, is unlikely to work well because of the unreliability of the twin signals mentioned above. Traditional supervised learning would try to push the model to predict 1 for every video belonging to a twin pregnancy, whereas the majority of frames and videos in a twin pregnancy may not contain any evidence of

multiple fetuses. This would lead to a model that receives conflicting signals and it would be unable to learn a reliable pattern for multiple gestation pregnancies.

This is an example of a general machine learning problem known as multiple instance learning (MIL). In MIL, the model is trained on bags of instances, where each bag is labeled, but the individual instances in the bag are not labeled. The model learns to predict the bag label based on the instances in the bag, and it can be used to predict the label of new bags. In the case of twins, the bag is an exam, which contains multiple videos, and the bag label is whether the exam is from a twin pregnancy or not. There are a few choices for what is considered as an individual instance of the bag. They could be identified as frames, or videos, or short sequences of consecutive frames known as clips. The current repository implements frame-level MIL but provides support for clip-level and video-level MIL in the underlying code.

One limitation of these approaches is that while they aggregate the data from different videos and frames in an exam, they do not consider the temporal relationship between frames in the same exam. The ML group did some initial exploration of architectures that account for temporal relationships between frames as well as train weights at an exam level.

Model Architectures

The models follows a **Spatial → Temporal aggregation → Classifier** pipeline. Several architectures were explored for classification of multiple gestations, but three were fully developed, and two models are provided for final release:

- **CNN → BAA (Basic Additive Attention) → Classifier:** The CNN, BAA, and Classifier are trained concurrently at the video level, then evaluated at the exam level. We will refer to this as the "BAA" model.
- **Two Multiple Instance Learning (MIL) versions:**
    - **Frozen MIL (frMIL)** – a pre-trained CNN (from the BAA architecture above) is followed by an attention-based MIL network. The model is trained at the exam level, but only the MIL and final classifier weights are modified.
    - **Integrated MIL (iMIL)** – a CNN is followed by an attention-based MIL network followed by a classifier. The CNN is initialized with weights from the BAA architecture, but the CNN, MIL, and classifier weights are trained concurrently, at the exam level.

BAA model

The BAA architecture is similar to that used for gestational age and fetal weight estimation, except that the final stage is a classifier instead of a regressor. The architecture block diagram is shown in Figure 1.

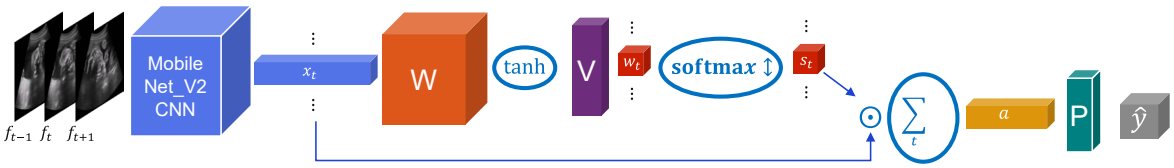

Figure 1. CNN→BAA→Classifier architecture for twin classification

This model is trained at the video-level, thus every video in an exam is pushed towards the target label for the entire exam. This has the drawback that not all videos or frames contain evidence for twins, as noted in the overview. Nevertheless, reasonable performance is obtained. This model mainly serves the purpose of pre-training the CNN weights for the frMIL model and providing the initial CNN weights for the iMIL model. This usage of the CNN weights from the BAA model in the other two models is depicted in Figure 2.

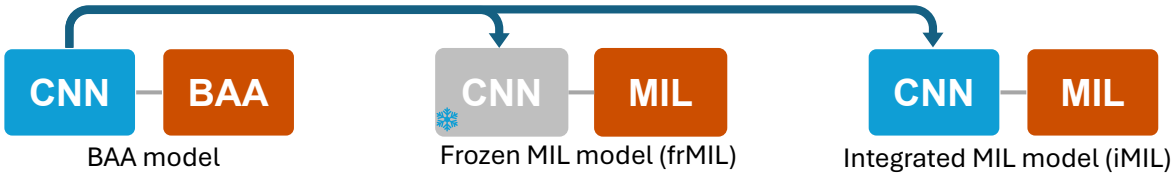

Figure 2. Use of CNN weights from the BAA model in frMIL and iMIL models.

At inference time, every frame of every blind sweep video in the exam folder can be processed by the model. This increases the likelihood that evidence for multiple gestation pregnancies will not be missed, which improves performance.

Frozen MIL (frMIL) model

The frozen MIL model is also a CNN→RNN→Classifier architecture, with the temporal aggregator being an MIL model, using the attention-based algorithm described by Ilse et al. [2]. The attention-based mechanism used in this MIL model is additive attention, but it has an enhancement compared to Basic Additive Attention. There are two parallel channels of attention being applied, where the second channel uses a different activation function (sigmoid vs hyperbolic tangent) and acts as a gating mechanism to the first channel. This permits the attention mechanism to respond to more nuanced sets of patterns compared to simple additive attention. The architecture is depicted in Figure 3.

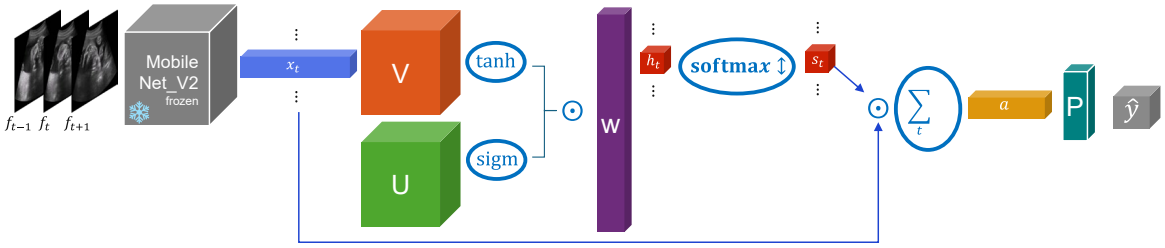

Figure 3. MIL architecture with CNN having frozen weights from CNN-BAA model and two independent attention channels, where the "U" channel gates the "V" channel.

The frozen MIL model is trained at the exam level. Only the MIL and classifier weights are modified. The MIL temporal aggregator ensures that samples (frames) that do not contain evidence of twins are downweighted by the gated attention mechanism, thus avoiding label conflicts and the concomitant loss of performance. Because the model is trained in batches, all samples (exams) must have the same number of instances, e.g., the bag_size. And because the frame features are computed in advance and stored on disk, memory usage is minimal compared to models with video inputs. This allows setting the bag_size to a large number, say 1,000, which decreases the likelihood of missing any piece of multiple gestation evidence. During inference, the number of frames sampled per exam can be arbitrarily large or set to all frames in the exam. Also, Matern sampling is used to avoid frames that are too close to each other being sampled (as would happen with pure Poisson sampling).

The code for the implementation of the frMIL was based on converting a Keras MIL package [3] to PyTorch. It must be stated, however, that a deviation from the equations described in Ilse et al. was noticed in the Keras code and was fixed in our PyTorch implementation. Ultimately, this model is not being released in the code repository because its performance lags the more powerful integrated MIL model described next.

Because the frame embeddings from CNN→BAA are computed and stored in advance, the inference time for the frMIL is very low. This makes it possible to use the frMIL to quickly explore the configuration space for the MIL architecture. The optimal architecture can then be adopted for both frMIL as well as the integrated MIL model, which we discuss next.

Integrated MIL (iMIL) model

The Integrated MIL model architecture is the same as the Frozen MIL architecture, although its code implementation is completely different, having been coded from scratch by the GHL ML team. All the weights in the model, namely the CNN, MIL, and Classifier weights are learned simultaneously and trained at an exam level. The CNN weights, however, are first initialized to the BAA model's CNN weights, as this was found to be necessary to accelerate learning and result in better final performance. The iMIL architecture is shown in

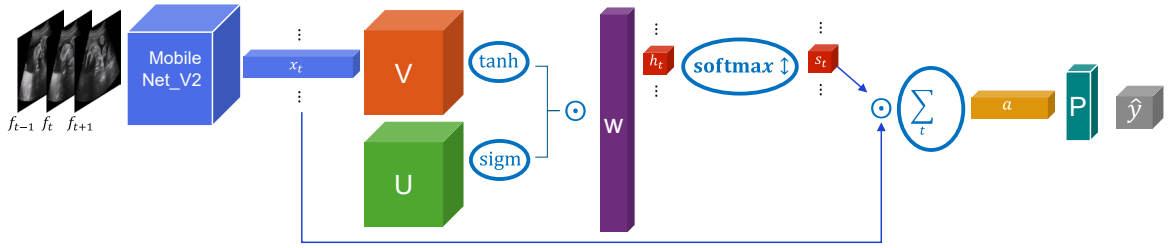

Figure 4. iMIL architecture for twin classification, where CNN weights are fine-tuned along with MIL and classifier weights being learned during training.

Figure 4.

During inference, the number of frames sampled per exam can be arbitrarily large or set to all frames in the exam. But in practice, the number of frames used must be limited to constrain the computational time on an edge device.

Training

During training of the models, exams were duplicated for two purposes: (1) to balance singleton vs multiple gestation exam counts; and (2) to maintain a rough balance between counts in a coarse, four-level gestational age binning scheme. This ensured that the algorithm was trained such that singletons and multiple gestations were equally represented as well as the different gestational age brackets. Data augmentation was applied to provide diversity among the duplicated exams.

References

[1] "GHL OBUS GitHub Repository," 2025. [Online]. Available: https://github.com/Global-Health-Labs/OBUS-GHL-DEV.

[2] J. T. M. W. M Ilse, "Attention-based deep multiple instance learning," *arXiv,* p. arXiv:1802.04712v4, 2018.

[3] M. Jaber, "Classification using Attention-based Deep Multiple Instance Learning (MIL)," *Keras.io,* p. https://keras.io/examples/vision/attention_mil_classification/, 2021.