A ConvLSTM-Based Approach to Nondestructive Ultrasound Molecular Imaging for Breast Cancer Detection

John Miller[1], Jihye Baek[2], Jeremy Dahl[2]

[1]Department of Computer Science, Grinnell College, Grinnell, IA, USA

[2]Department of Radiology, Stanford University, CA, USA

1. Introduction

With the prevalence of breast cancer, early diagnosis is important. The goal of this task is to utilize deep learning to accurately detect targeted microbubbles (TMBs) when the injected TMBs bind to cancer tissue.

Previous approaches to TMB identification with machine learning have involved the convolutional neural network (CNN) and the transformer [1, 2]. While both approaches accomplish the same task, they do so with different strengths and weaknesses. CNN is a lightweight approach of neural network architectures, thus a few convolutional layers can produce a probability map of MB presence, where each value represents the likelihood of a bound MB at a given pixel [2]. Training proceeds via backpropagation, and the process is repeated for each new input frame. Inference via CNN lasts only about 0.5 ms per frame. While swift, CNN produces noisy and inaccurate output due to its stochastic (time-independent) nature not lending itself well to a problem that requires both spatial and temporal awareness. The attention mechanism-based transformer, in comparison, is a mammoth interleaving of weights, embeddings, etc. that provides excellent probabilities, however, suffers in terms of efficiency when compared with the simpler CNN due to its longer process time and memory usage. While accuracy in detecting bound MBs is the primary objective, efficiency during both training and inference is also an important measure of overall model performance. Recurrent neural networks strike a strong balance, demonstrating strong results across both metrics. The Recurrent neural network, particularly the convolutional-Long Short-term Memory (LSTM) provides a hidden and cell state [3] (explicated further in subsequent sections) that allow for reasoning about bound-MB probability over time, while also providing the lightweight efficiency that makes an architecture like a CNN shine.

2. Materials and Methods

2.1: Materials and scans: This project utilized a dataset acquired for USMI by previous research [2] to minimize the number of mice and scans needed for research. Mice (n=14) were randomly divided into five groups to perform cross-validation via the 5 groups. Control/non-targeted MBs (NTMBs) were injected into the mice, followed by the injection of targeted MBs (TMBs) containing PDL1, B7-H3, and VEGFR2 antibodies. A Vevo2100 (FUJIFILM VisualSonics, Toronto, ON, Canada) was used for the ultrasound scans, which recorded 2D video scans to observe the tumor's largest cross-section over time and 3D sweeps to observe the tumor volume. The scans were performed at various points, namely before MB injection, after injection, before burst, and after burst. The pre-injection scans served as a negative control for training, and pre/post-burst scans were utilized to create DTE ground truth images for training.

2.2: Hyperparameters: Training is performed over 200 epochs using the 50-video master dataset, with 40 videos allocated for training and 10 reserved for validation. In each epoch, seven random patches ranging in random sizes from 64x64 to 256x256 are extracted across all frames of every video, serving as data augmentation to capture details in microbubble movement [2]. These same 7 patches are maintained across each epoch. Binary Cross Entropy was used as the loss function [2]. Tensor handling operations were performed with the library torch. The optimizer Adam was used, with a learning rate of 0.001.

2.3: Model mathematics and architecture: Following the general architecture put forth by the paper on precipitation nowcasting, the architecture combines convolutions and the original LSTM cell mechanics to take advantage of previous research that has shown the efficacy of the LSTM for sequence learning [2, 4]. From a high level, the end goal is to track changes in values across each frame which allow for more informed inferences as to the location of the bound microbubbles in the current frame. We now present our convolutional LSTM. The data pipeline begins with concatenating the 2-channel input frame with the 32-channel hidden state. This provides a tensor of shape [34 x 256 x 256].

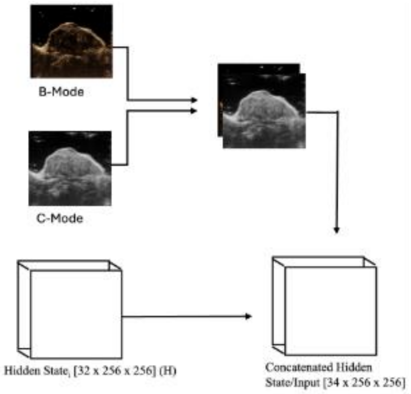
B-Mode
C-Mode
Hidden State, [32 x 256 x 256] (H)
Concatenated Hidden State/Input [34 x 256 x 256]

Figure 1: Input and hidden state concatenation

From this tensor, a 34 to 128 channel convolution is performed [3]. This means 128 weight matrices of shape 3x3x34 are applied to produce a value for each pixel, ultimately producing 128 channels as shown in Figure 2.

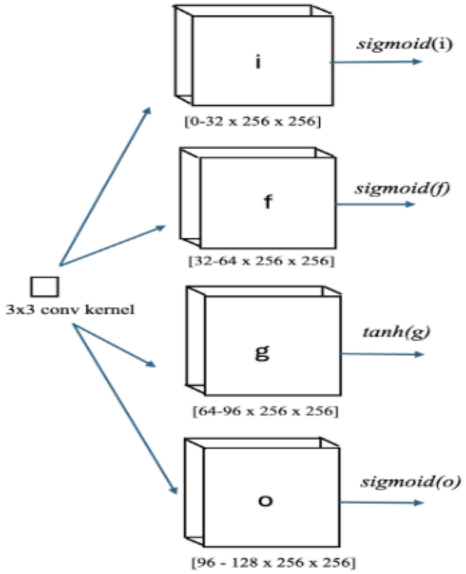

Figure 2: 34 channels to 128 channels convolutions

This provides a tensor of shape *[128 x 256 x 256]* to supply to the LSTM cell for cell update, as shown in Figure 2. These are the input (i), forget (f), candidate (g), and output (o) gates (Figure 2). This tensor is split into four equal chunks of 32 channels each, representing the pre-activation values for the four gates of the LSTM cell: the input gate, forget gate, candidate cell state, and output gate. Each pre-activation chunk is then passed through a specialized activation function (equations 1, 2, 3, 4).

$$\mathrm{i} = \sigma(i) \tag{1}$$

$$f = \sigma(f) \tag{2}$$

$$\mathrm{g} = \tanh(\mathrm{g}) \tag{3}$$

$$\mathrm{o} = \sigma(o) \tag{4}$$

We then update the hidden state and cell state for the next input frame (Hidden (i + 1) and Cell (i + 1)) with the following mathematics (equations 5 and 6):

$$\mathrm{Cell}_{i+1} = f \cdot c + i \cdot g \tag{5}$$

$$\mathrm{Hidden}_{i+1} = o \cdot \tanh(\mathrm{Cell}_{i+1}) \tag{6}$$

If the forget gate is small enough, much of the previous cell state will be “lost” as the product between itself and the previous cell state will be small. At the same time, the candidate gate can introduce new information, adding to the portion of the cell state that is preserved. The new cell state is used to update the hidden state, acting as a gatekeeper, where the o (output gate) determines the proportion of the long-term memory that is relevant to the current output. From this point in the data pipeline, a 32 to 1 channel convolution is performed on the updated hidden state to get model output (equation 7).

$$Y_{i,j} = \sum_{c=1}^{32} W_{cij} X_{c,i,j} \tag{7}$$

$$\text{output} = \sigma(Y) \qquad (8)$$

This provides a simple 1-channel output frame. To turn this output frame into probabilistic logits to indicate the presence of microbubbles, we perform sigmoid on the [1x256x256] tensor, where Y is this tensor (equation 8).

This produces a greyscale tensor with values between 0 and 1, representing the likelihood of a bound MB at each pixel. The Binary Cross Entropy loss is then computed, and the convolutional weights are updated via backpropagation. The key advantage of the ConvLSTM model is that temporal information is carried primarily by the hidden and cell states, rather than by introducing many additional trainable weights. This allows for less necessary updates during backpropagation, saving seconds and potentially minutes as the training dataset frame count grows.

3. Results:

The model performance can be analyzed first to determine whether it is minimizing the loss function via backpropagation, as shown in Figure 3.

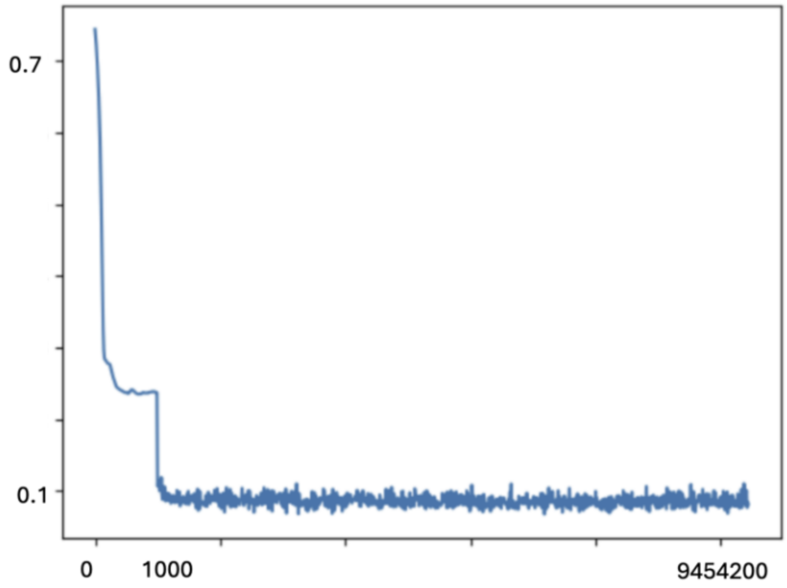

Figure 3: Loss from training frames 0-1000, average per-1000 frame loss from training frames 1000 through 9,454,200, making up roughly 9000 data points in addition to the leftmost 1000.

Here, the loss value for the first 1000 frames is shown (the first few videos), followed by the averaged loss value across every subsequent 1000 frames regardless of patch or frame. This allows for additional representative data points, as the per-epoch loss values only provide 200 data points for almost 10 million frames. We see a first-frame loss value north of 0.7, with a local minimum being found during the first patch of the first video around 0.25, and an eventual minimum found around 0.08. This indicates training reached its local (potentially global) minimum around 5 to 10 epochs. This highlights the ConvLSTM architecture [3], as a decrease in weights to update means there is a decreased likelihood of finding a sub-optimal local minimum during the gradient descent process.

A key qualitative measure of the model's success in suppressing noisy unbound signals while isolating bound MB signals is the visual "eye test" when compared against other models. The inference images were provided in Figure 4.

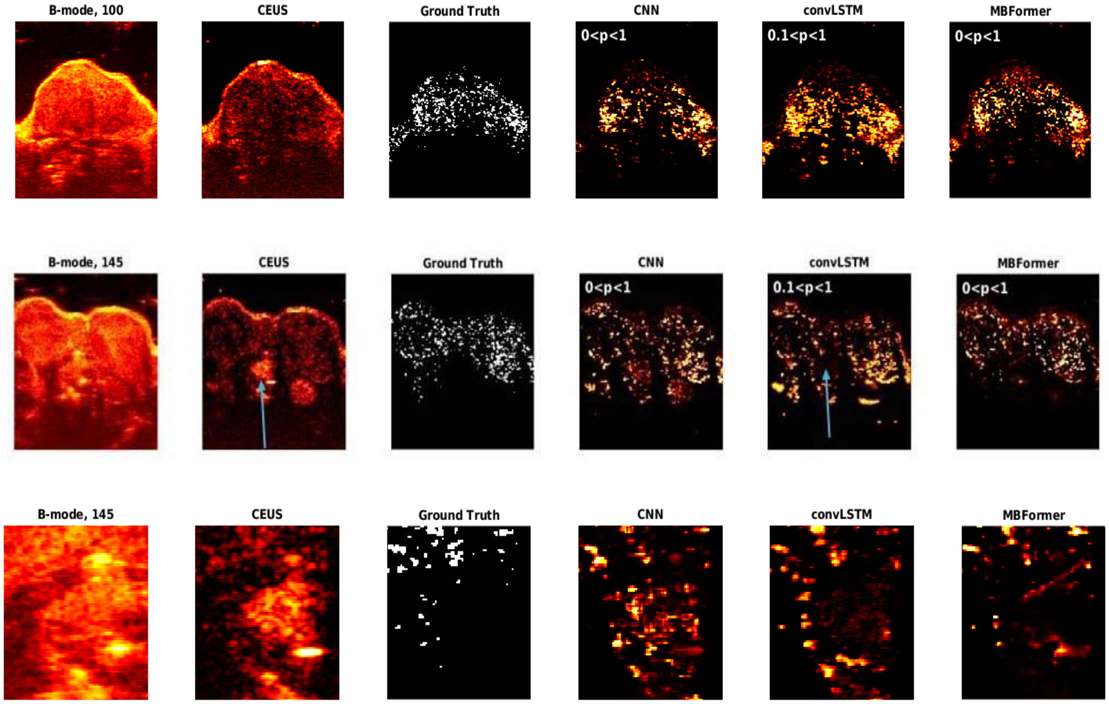

Figure 4: a sample inference frame from each model for comparison

As shown in Figure 4, one key suppression that the model architecture is tasked with is not demonstrating false positives in inference for the false positive brightness in the heartbeat of the mouse (seen to the left with the blue arrows). The time-unaware CNN is unable to identify that the spatio-temporal pattern of the heart isn't indicative of breast cancer tissue with bound MBs. In improvement upon the pure CNN, the ConvLSTM identifies that this tissue isn’t confirmative with bound MB behavior based on both GT data and previous hidden and cell state. Due to this, it predicts low probability values (likely between 30% and 0%) for this region of the mouse, preventing a false positive signal. In addition to this, the indicators of bound MB’s (true positives) align well with the best performing model of previous research [2], the MBFormer (our transformer-based model, microbubble former) model. ConvLSTM showed comparable suppression of false positive signals with MBFormer, while showing superior suppression compared to CNN.

The final metric of model performance is inference speed. Table 1 illustrates how the ConvLSTM achieves inference speeds comparable with the convolutional neural network.

Model	Per-frame inference time (mean)
CNN	0.8 ms
ConvLSTM	0.5 ms
Transformer:MBFormer	59.8 ms

Table 1: Per-frame inference times for each model

4. Discussion, Conclusion, and Implications

From both quantitative metrics and qualitative observations (Figure 4 and Table 5), we conclude that the ConvLSTM demonstrates strong efficacy in identifying bound MBs while simultaneously suppressing spurious or noisy signals from circulating unbound MBs. This effectiveness arises from the model's recurrent design, in which hidden and cell states track spatial–temporal patterns across frames, allowing it to leverage contextual information that simpler architectures fail to capture. Beyond accuracy, the ConvLSTM also provides advantages in efficiency: its relatively lightweight parameterization and reliance on recurrent state updates lead to faster training and inference times compared to deeper transformer-based approaches. In the big picture, these characteristics make the ConvLSTM a practical and reliable framework for ultrasound molecular imaging tasks where accuracy and speed are required.

5. References:

[1] D. Hyun, L. Abou-Elkacem, R. Bam et al., "Nondestructive detection of targeted microbubbles using dual-mode data and deep learning for real-time ultrasound molecular imaging," IEEE transactions on medical imaging, vol. 39, no. 10, pp. 3079-3088, 2020.

[2] J. Baek, D. Hyun, A. Natarajan et al., "Nondestructive ultrasound molecular imaging based on a neural network approach utilizing post-processed ultrasound images." pp. 1-3.

[3] X. Shi, Z. Chen, H. Wang, et al., (2015, September 19). *Convolutional LSTM network: A machine learning approach for precipitation nowcasting*. arXiv.org. https://arxiv.org/abs/1506.04214

[4] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP, pages 1724– 1734, 2014.