

Accelerating 3D ultrasound brain imaging using machine learning

Final Report for Biomedical Engineering MEng Individual Project

Nicholas Ustaran-Anderegg

Department of Bioengineering, Imperial College London, UK

Supervisors: Dr Mengxing Tang and Mr Carlos Cueto Mondéjar

June 13, 2021

Word count: 4784

Abstract

Brain imaging for stroke diagnosis is typically performed using magnetic resonance imaging or computed tomography. These technologies both have significant disadvantages. Ultrasound imaging of the brain, previously considered impossible due to the skull, has been proposed based on full-waveform inversion. For full-waveform inversion, finite-difference methods are used to solve the acoustic wave equation efficiently. These finite-difference methods require the implementation of absorbing borders around a wavefield to prevent reflections, and are typically slow. Here, a convolutional LSTM neural network is trained on a dataset of simulated wavefields containing artificial skulls, using a perceptual loss function, to predict the edges of wavefields propagating to infinity to high accuracy. The network is coupled with a finite-difference solver to continually predict the edges of a wavefield for entire simulations, removing the need for absorbing borders. This results in a system that is more efficient, giving a computational time advantage over alternative methods. This acceleration of solving the wave equation results in faster ultrasound image reconstruction with FWI.

Introduction

Stroke is globally the second leading cause of death and the leading cause of adult-acquired neurological disability [1]. For each minute during a stroke, 1.9 million neurons, 14 billion synapses, and 12 km of myelinated fibres are destroyed [2]. Speed is critical, and a quick diagnosis of stroke can save lives and prevent disability by ensuring a patient receives the correct treatment [3]. This diagnosis is confirmed by imaging the brain, allowing a clinician to determine the cause, severity and location in the brain of the stroke. Currently, the two most commonly used brain imaging techniques for stroke diagnosis are magnetic resonance imaging (MRI) and computed tomography (CT) [3]. Both technologies are expensive and are not portable as they require large machinery. This requires that patients visit a hospital with access to these de-

vices to receive diagnosis, slowing down the process, and this access can be particularly limited in developing countries [4, 5]. Furthermore, MRI cannot be used with ferromagnetic implants, and CT uses harmful ionising radiation. These technologies therefore have considerable disadvantages, and there is a lack of, and a need for, a technology to better facilitate brain imaging for stroke diagnosis.

Traditionally, ultrasound imaging of the brain has been considered impossible due to the scattering and large reflections and refractions caused by the skull [6], which has a high density and speed of sound compared to its surroundings and contents. However, ultrasound brain imaging based on full-waveform inversion (FWI) poses a potential solution to this problem, allowing imaging through the skull, and has been proposed as a method to reconstruct clinical 3D ultrasound images [7, 6]. Unlike MRI and CT, ultrasound

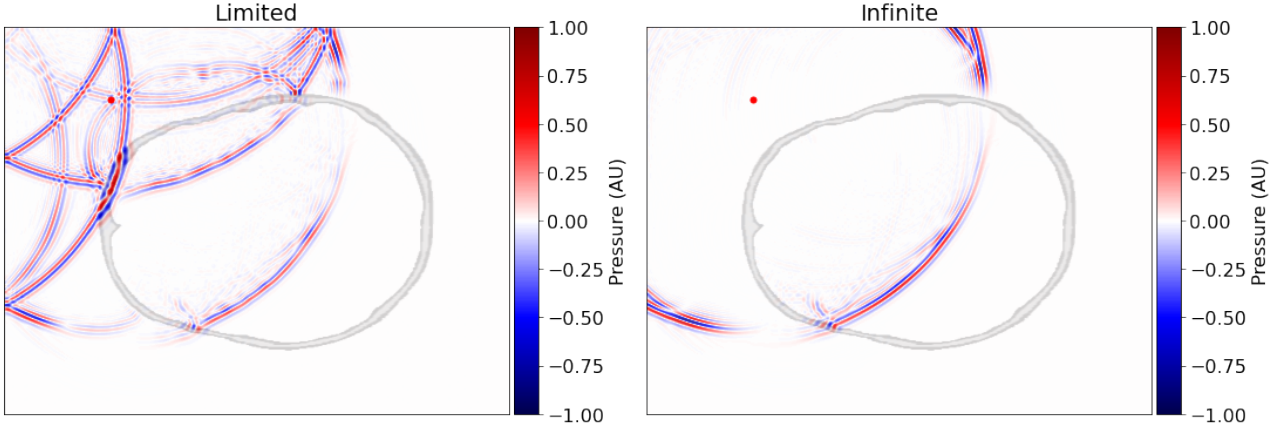


Figure 1: Comparison of behaviour of wavefield in model containing a skull, with and without propagations to infinity. The red dot in each image shows the location of the wave source, from which waves propagate. The Limited image shows a boundary placed directly around the region of interest, resulting in the reflection artefacts observed. The Infinite image demonstrates how the wavefield propagating to infinity should look. This ground-truth Infinite image was created by extending the boundaries of the wavefield simulation such that the edges were never reached by the propagating wave.

imaging is inexpensive, highly portable, and emits no ionising radiation [8]. These advantages over MRI and CT have deep implications for stroke diagnosis. It is feasible that an ambulance could carry an ultrasound scanner on board and provide an immediate stroke diagnosis en route to a hospital. This would significantly reduce the time between stroke onset and treatment, giving a patient the best chance of survival and reducing the likelihood of permanent disability. Furthermore, the ability to image in 3D allows clinicians to fully understand spacial anatomic relationships and diagnose more accurately [9]. This would also have applications in head trauma, during surgery, and neurological monitoring.

FWI is an imaging technique developed in geophysics, widely used for imaging the Earth’s subsurface [10, 11]. FWI can achieve a high spatial resolution because it uses the full physics of wave propagation. This enables it to accurately model the effects of the skull on a wavefield. FWI seeks to find a numerical model that can reproduce experimental data by minimising the least-squares norm between predicted data $d(m)$ from the model and observed data d_{obs} , shown in Equation 1.

$$f(m) = \frac{1}{2} \|d_{obs} - d(m)\|_2^2 \quad (1)$$

This minimisation problem is solved using an adjoint formulation, which requires solving the wave equation twice at every iteration. As full wave equation propagations must be calculated at each iteration with FWI, this is a computationally expensive task, particularly for 3D cases. To solve these partial differential equations (PDEs) efficiently, finite-difference (FD) methods are used. Finite difference methods

solve PDEs numerically by approximating derivatives as linear combinations of function values. This transforms a differential equation into a system of algebraic equations that can be solved more easily [12].

Modelling wavefields that match experimentally seen behaviour entails ensuring that waves travelling away from the model vanish into infinity. However, numerical methods are defined over a finite domain and, if nothing is done, artificial reflections are generated at the borders when solving the PDE. The PDE being solved is the acoustic wave equation, shown in Equation 2.

$$\frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} - \rho \nabla \cdot \left(\frac{1}{\rho} \nabla p \right) = s \quad (2)$$

Reflection artefacts caused by the edges of a wavefield are shown in the Limited image in Figure 1. An ultrasound image produced with FWI from the Limited wavefield would be uninterpretable. To deal with these reflections, the computational domain is generally extended and some form of absorbing boundary condition is imposed on this region, such as a perfectly matched layer (PML) or a sponge layer [13]. The addition of this absorbing boundary implies increased computational cost. Increased computational cost results in increased computational time, which is undesirable for clinical ultrasound imaging, particularly for stroke diagnosis where speed is critical [2]. To achieve clinical applicability, this technique must therefore be made faster.

The sponge boundary [14] avoids apparent inward reflections by introducing large, absorbing padding areas around a wavefield model. The width of these borders is problem-specific, but is typically around 50 pixels. The borders are given a gradually enhanced

attenuation factor, from inner to outer, to reduce the intensity of wave propagations. When a wave travelling outwards eventually reaches the outer edge of the damping boundary belt, there is no reflection, and so to the region of interest in the centre, waves appear to propagate to infinity. The larger borders increase the computational domain for which FD methods must be used, resulting in a greater computational time for image reconstruction with FWI.

The PML [15] attenuates waves in a different way to the sponge boundary. It modifies the partial derivatives in the wave equation by introducing an imaginary part associated with an attenuation factor [13]. This requires less padding around the wavefield, meaning the computational domain for FD methods is smaller than that with a sponge layer, but the extra PDEs introduced in these regions significantly increase the computational cost.

Machine learning is widely used in research to speed up tasks by training a model to act as a universal mapping between input and output spaces. Here, machine learning methods are explored to reduce ultrasound image reconstruction time by reducing the computational space in which reconstruction takes place. A deep neural network (NN) is trained to predict the edges of the wavefield, such that waves appear to propagate to infinity, and thus removing the need for a computationally expensive absorbing boundary.

Creating a NN for this task was previously attempted by Graef in 2020 [16], to limited success. Graef used a dataset consisting of $[115 \times 23]$ 2D images of waves adjacent to edge of a wavefield, and employed a simple convolutional neural network (CNN) to predict wavefield edges adjacent to these input images. These wave images consisted of 5 $[23 \times 23]$ images concatenated together, forming larger 2D images, where each $[23 \times 23]$ image corresponded to an adjacent timestep backwards in time. Border predictions from Graef’s model are shown in Figure 2.

The network’s prediction of the first border is qualitatively similar to the real target, but the subsequent three predictions are not accurate. The network is particularly poor at predicting the wavefield in the centre of the bottom two borders, where the wave displays ripples. Graef did not integrate the network with a FD solver to determine its suitability for use in accelerating ultrasound reconstruction [16], but it is unlikely that the network would have performed well given its performance on the test dataset.

In this report, deep learning models are investigated to accelerate the reconstruction of 2D wave propagations. Successful computation of 2D waves will demonstrate the feasibility of using this technique

for 3D wave propagations. This improves upon the work of Graef [16] by creating a more accurate and robust NN that can predict complex wave features, and which is fully integrated with a FD solver for use in ultrasound image reconstruction. The creation of the dataset, architecture of the NN and training methods are explained. Then, the performance of the model is examined via testing on previously unseen data and through integration with a FD solver. Finally, the model’s applicability to brain ultrasound imaging is discussed and comparisons are drawn with traditional techniques.

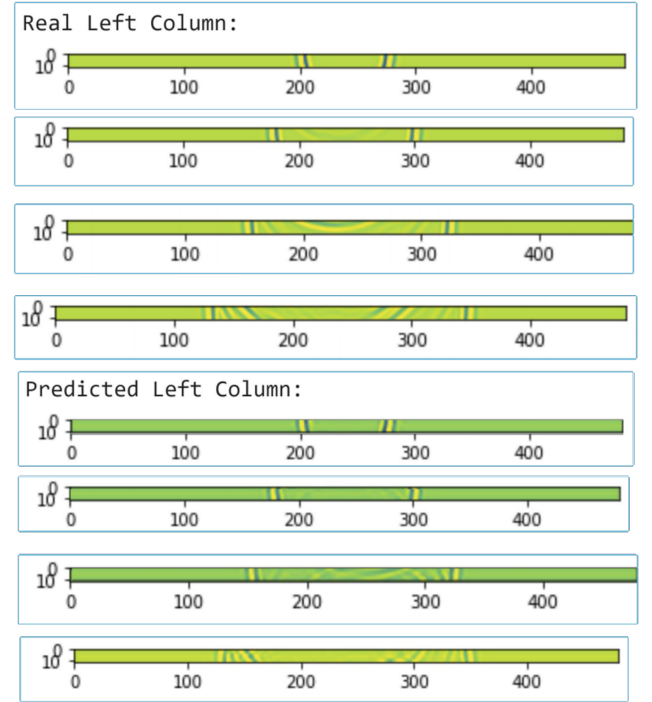


Figure 2: CNN wavefield border predictions by Graef [16]. The network is only able to predict one border that looks qualitatively similar to the real border.

Methods

The goal of accelerating the calculation of FD methods is, in turn, to accelerate FWI reconstruction of ultrasound brain images. The large absorbing borders traditionally implemented surrounding a wavefield are replaced by a border that is just 11 pixels wide using machine learning. A width of eleven pixels is chosen because FD stencils up to this order are used to solve the wave equation. In 2D, a simulation size of 502×372 pixels is used. With the addition of absorbing borders that are 50 pixels wide at each edge, the computational space for FDs to be calculated in is made 52% larger. This work is in 2D, but the results are applicable to 3D imaging, where similar techniques could be used to predict the borders of a 3D wavefield.

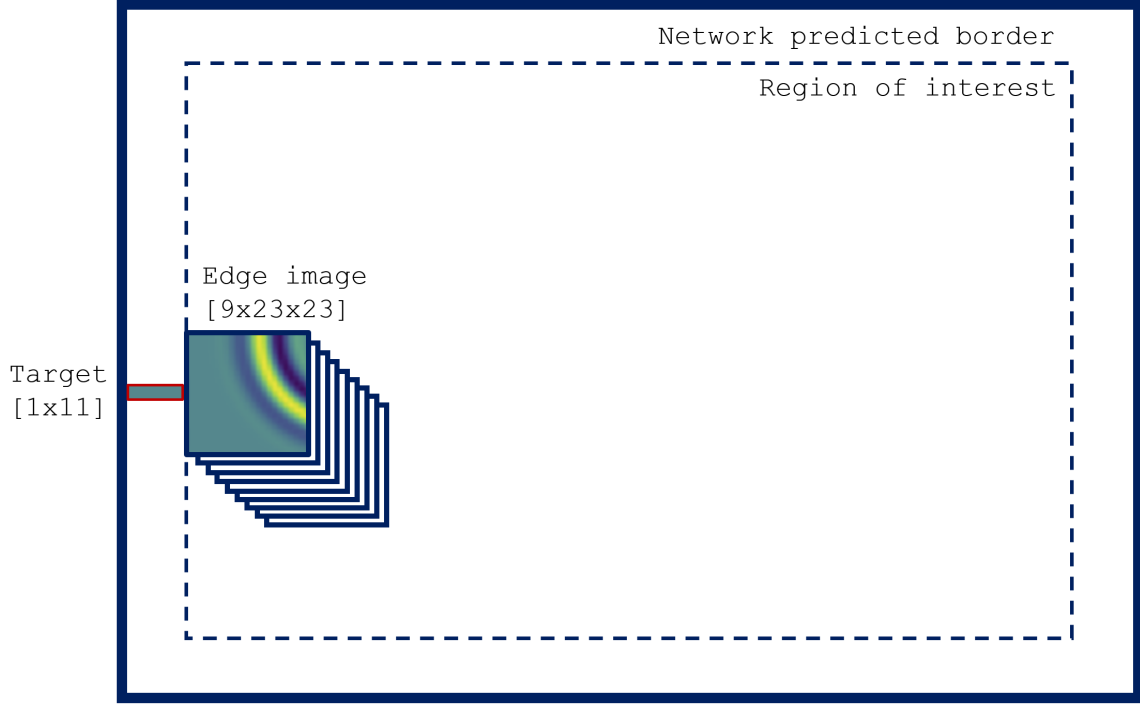


Figure 3: Diagram of the edge image and border slice target, which serve as input and output to the CNN respectively. These images are created by simulating wavefields using Devito, an FD solver.

To create smaller, more efficient borders, a neural network trained on wave data predicts a $[1 \times 11]$ border slice adjacent to a $[9 \times 23 \times 23]$ image of the 2D wavefield. Here, the depth of 9 in the third dimension corresponds to 9 previous timesteps, these being the current timestep t , $t - 6$, $t - 12$, $t - 18$, ..., $t - 48$. This is illustrated in Figure 3. Border predictions are made for all locations and substituted into the FD solver’s wavefield solution at each time step. Considering ultrasound frequency and the size of the wavefield in computational space, Nyquist sampling [17] dictates that the sampling frequency must be at least every 60 timesteps, and so creating network input images from these timesteps avoids aliasing. Samples are made up of images 6 timesteps apart, rather than from every timestep, because this provides greater computational efficiency, while still presenting the NN with enough information to predict borders accurately.

Devito, a framework primarily designed to create wave propagation kernels for use in seismic inversion problems [18], is used as the FD solver with which the NN is integrated. At each step in the time-stepping solver, using Devito to calculate the wavefield, the borders of the wavefield are replaced with those predicted by the NN, such that waves appear to propagate to infinity without reflection artefacts. 11 pixels is a great enough distance that any reflections calculated by Devito are overwritten by the network’s predictions.

Dataset

To create the dataset used by the neural network, a real skull model was warped randomly using a perspective transformation to create new models that qualitatively looked like real skulls, but were unique. Each skull model consists of an array, in which the value of each pixel defines the speed of sound at that location, and these are shown in Figure 4. Artificial skull models were used to create a dataset that would allow the NN to become general by having trained on a variety of different skulls. This is critical for use with a real ultrasound scanner, where every patient has a unique skull, and the network must therefore be able to cope with different anatomical shapes and sizes.

Using Devito, wavefields around the artificial skull models were simulated by propagating waves from a point source, representing an ultrasound transducer. These ground-truth simulations, used to construct the dataset, were created by extending each model in computational space so that the border of the computational domain was never reached during simulation by a propagating wave. This emulates the expected behaviour of the network.

To create the dataset, the $[9 \times 23 \times 23]$ network input images and $[1 \times 11]$ border targets were captured at various timesteps during each simulation with a random point source location and skull model. The dataset was normalised between -1 and 1 . This was

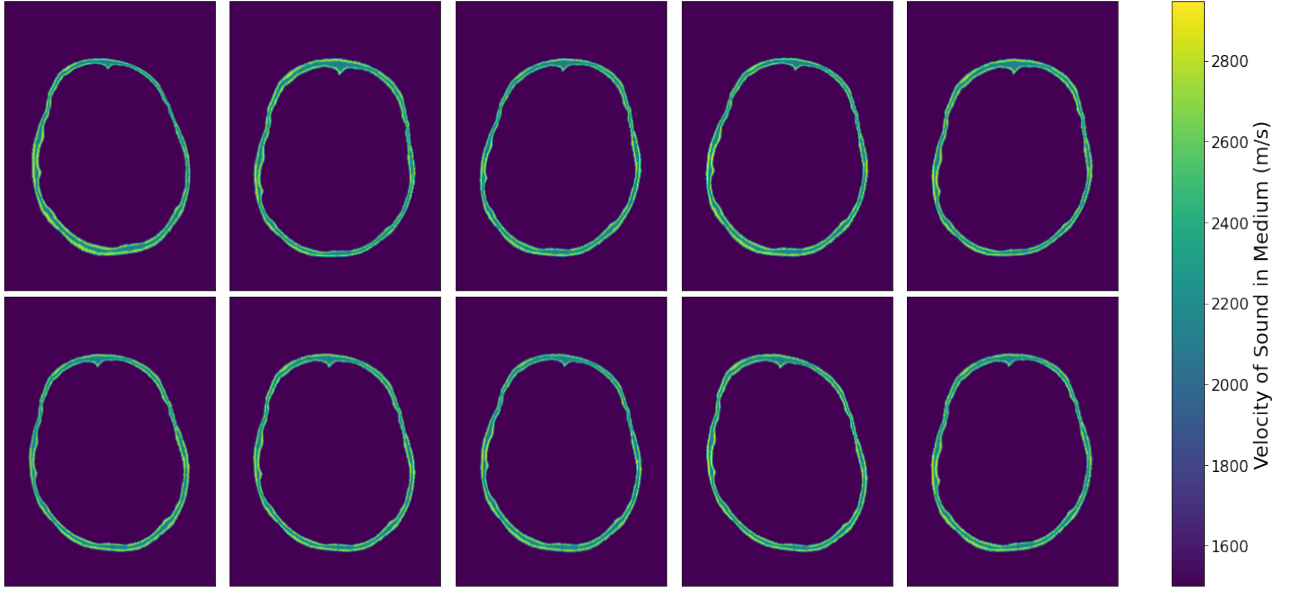


Figure 4: Artificial skulls created using a random perspective transformation of a real skull model. These models are similar, but unique, and resemble real skulls. These were used to simulate wave propagations and create a dataset for training the NN.

split into 80% for training, with a size of 138,000 samples, and 20% for testing, with a size of 42,000 samples. K-fold validation was used on the training dataset to ensure generalisation.

Neural Network Architecture

Dataset samples contain images that are correlated through time. CNNs are prevalent in research for their ability to learn from images [21, 22]. Deep CNNs are able to extract image features using convolution, and networks learn to optimise filters for extracting these feature maps. The translation of these convolutional filters allows a network to learn from spatial relationships between image pixel values as a filter is moved across an image.

The input images form a time sequence, and this can also be taken advantage of when designing the network architecture. Recurrent neural networks (RNNs) are specialised at learning from sequences [23], and Long Short-Term Memory (LSTM) is a class of RNN specialised at overcoming the vanishing gradient problem [24, 25]. The vanishing gradient problem arises during training, when gradients of the loss function approach zero as they are backpropagated through the network. As a result, synaptic weights in the early layers of a network are updated less, restricting learning. This is exacerbated by the depth of the network, which is especially relevant in the case of RNNs.

As opposed to a vanilla RNN [26], LSTM incorporates a memory cell, c_t . This allows the network to avoid the vanishing gradient problem because back-

propagated gradients become trapped in the cell, preventing them from vanishing too quickly [25]. The equations governing LSTM are displayed in Equation 3, where \circ denotes the Hadamard product. The information from an input is accumulated to the cell if the input gate i_t is activated, and the cell's previous state c_{t-1} is forgotten if the forget gate f_t is activated. The output gate o_t controls whether c_t is propagated to the final state h_t [20]. In a fully-connected LSTM, the input, cell output and states are all 1D feature vectors, meaning that inputs to an LSTM network must be 1D, and cannot encode spatial information.

$$\begin{aligned}
 i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
 h_t &= o_t \circ \tanh(c_t)
 \end{aligned} \quad (3)$$

The network architecture implemented to predict wave propagations to infinity is a Convolutional LSTM (ConvLSTM) [20] network. ConvLSTM takes full advantage of both the spatial information encoded in the dataset images, and its time sequence nature. This hybrid between a CNN and a RNN is well suited to image sequences. It is similar to LSTM, but is able to handle 3D tensors instead of 1D feature vectors. The ConvLSTM determines the future state of a certain pixel from the inputs and past states of its local neighbours [20]. This is achieved by replacing mul-

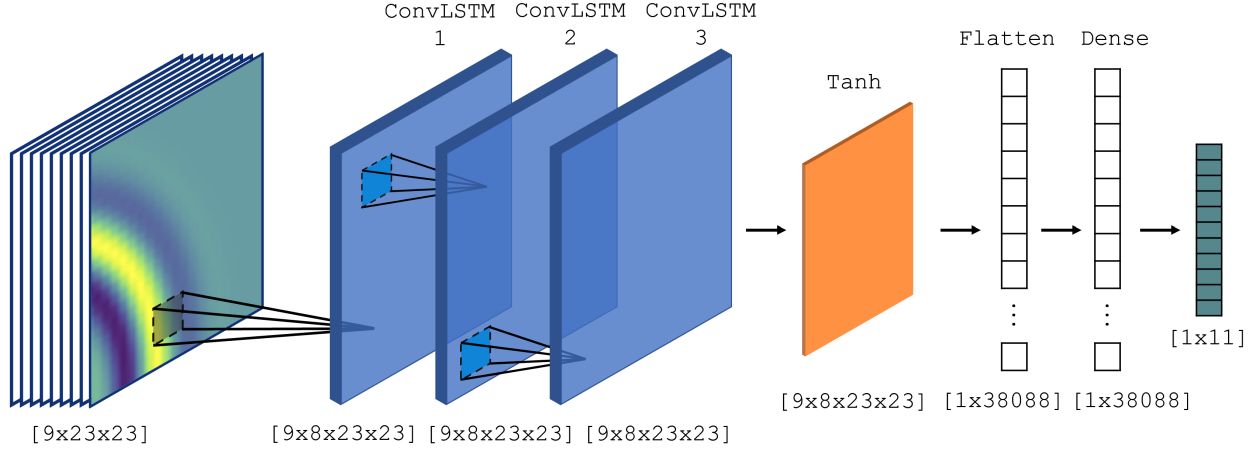


Figure 5: Deep ConvLSTM network architecture. This network was implemented in PyTorch [19] and uses stacked ConvLSTM [20] layers to take advantage of the spatial and sequential nature of the network input images.

tiplication in the LSTM equations with a convolution operation $*$ in transitions, as shown in Equation 4.

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
 C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
 H_t &= o_t \circ \tanh(C_t)
 \end{aligned} \quad (4)$$

ConvLSTM can be used as a building block [20] and stacked in layers to create a deep network. The network architecture used consists of 3 stacked ConvLSTM layers, with each layer passing its cell state C_t and hidden state H_t to the next layer. Each ConvLSTM layer has 8 convolutional features, followed by a \tanh activation layer, a flatten layer, and a fully-connected dense layer to produce the necessary $[1 \times 11]$ output. \tanh was chosen as the activation function because the targets are normalised between -1 and 1 . The nature of the large, varied training dataset meant that dropout layers were not required to prevent overfitting. The network was implemented in Python using PyTorch [19], and its architecture is summarised in Figure 5.

Training

During training, a traditional per-pixel loss function is replaced by a perceptual loss. Here, the outputs of the ConvLSTM network and the real targets are passed through a pretrained CNN, and the loss is calculated between the feature maps of the CNN on these two inputs. This allows the transfer of semantic knowledge of shapes learned by the pretrained network, whose weights are not updated, enabling faster learning and

better reconstruction of fine details [27]. As used by Johnson et al. in 2015 [27], when using perceptual losses for super-resolution, VGG16 [28] is used as the pretrained network. VGG16 is available on PyTorch [19] as a pretrained model, and this was implemented during training.

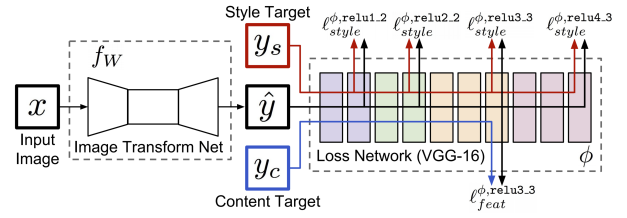


Figure 6: System overview of model used by Johnson et al. [27]. The same VGG16 [28] `relu2.2` layer features are used to determine perceptual loss when training the ConvLSTM network.

VGG16 was proposed in 2014, is trained on the ImageNet dataset [29], which contains over 14 million images, and was submitted to the ImageNet Large Scale Visual Recognition Challenge [30]. This perceptual loss network was selected because it had been trained for weeks [28] on a large, varied dataset [29], meaning it would be well suited to detect useful wave features, such as the shape of a curved wavefront, and transfer this knowledge to the ConvLSTM network. Johnson et al. used feature maps at the `relu2.2` layer of VGG16 [27], and this is similarly used here because the semantic knowledge contained at this level provides a balance between high and low level convolutional features. Figure 6 shows the implementation used by Johnson et al. when using VGG16 [28] for perceptual losses, where the `relu2.2` layer is the third convolutional layer followed by a ReLU activation. This layer is immediately preceded by a maxpooling layer.

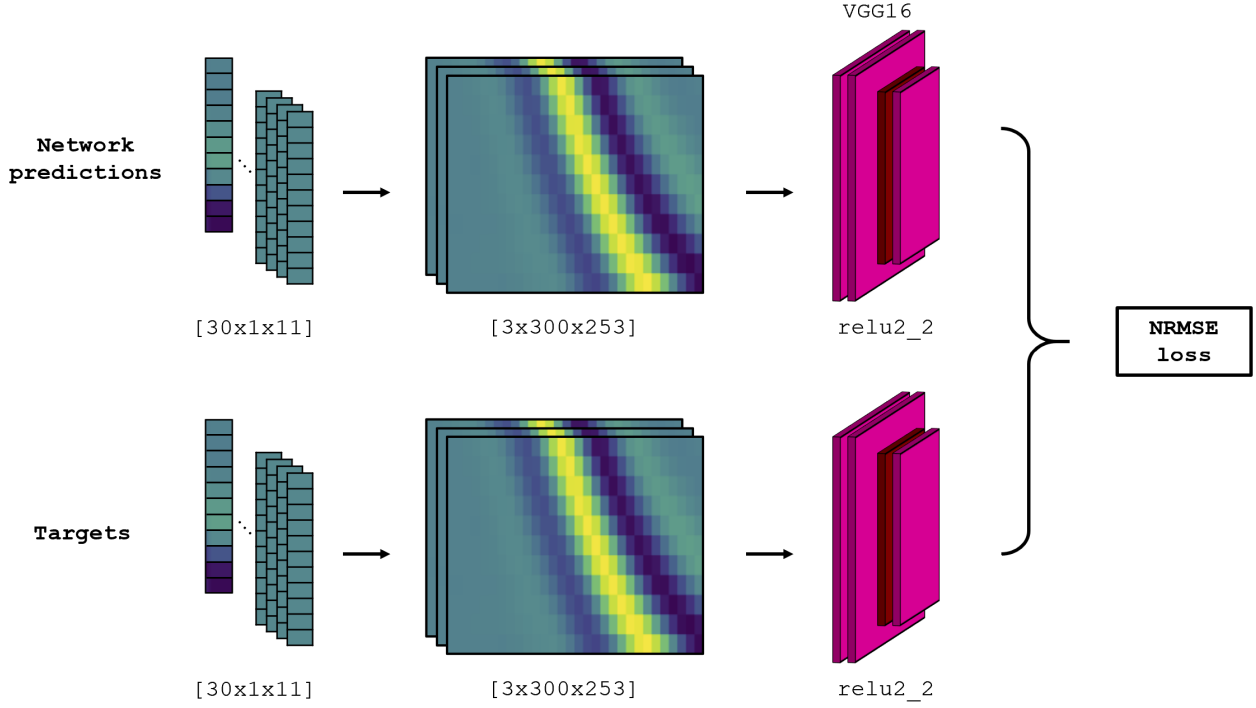


Figure 7: Flowchart for calculating perceptual loss. Fuschia VGG16 layers represent convolution followed by a ReLU activation, the dark red layer is a maxpooling layer. Each batch of network outputs and targets is concatenated and expanded into a larger image, and the NRMSE loss is found between the feature maps of the pretrained VGG16 [28] network.

A learning rate of 0.0001 and a batch size of 30 were used during training. Although batches were presented to the network in a random order, each batch contained ordered samples that were spatially adjacent when the dataset was created by capturing images of simulated wavefields. This technique was used so that the 30 outputs of the ConvLSTM network and targets per batch, each of size $[1 \times 11]$ could be concatenated into a larger prediction of a section of a border of size $[30 \times 11]$. These two reconstructed images were expanded by repeat-interleaving to a size of $[3 \times 300 \times 253]$ to create a larger image that resembled a border section. These larger images were passed to the loss network, and the normalised root-means-square error (NRMSE) was calculated as the loss between VGG16 [28] relu2_2 feature maps. This is demonstrated in Figure 7.

Evaluation

The NN created is tested on its ability to predict wave behaviour. A quality model can predict the borders of wavefield propagating to infinity that are similar to those calculated using traditional FD techniques. The network’s ability to predict these is tested in multiple ways. Firstly, the perceptual loss will be used as a difference measure between network outputs and actual wavefield targets during training. A test dataset

that has not been trained on is used to determine the test perceptual loss, a quantifiable measure of the network’s performance. Minimal test error is desirable to ensure the network can continually produce accurate predictions when integrated with Devito.

The most important test of the NN’s ability to predict wavefield borders is a comparison of the performance of the network, integrated with Devito [18], with the ideal infinite wave behaviour, when calculating wavefields at each timestep for an entire simulation. Multiple wavefields are simulated, and the network is tested on how it continually produces borders around them that do not result in reflections or other artefacts. The difference between ground-truth wavefields, calculated by extending the computational domain, and those predicted using FD methods with the learned model are compared using per-pixel error. Again, a minimal error is desirable, as this will indicate the accuracy of the network during an entire simulation. This evaluation is performed on a skull model and source location that have never been seen by the network.

Results

Training for 225 epochs took approximately 18 hours on a single Nvidia Tesla P100-PCIE-16GB GPU [31],

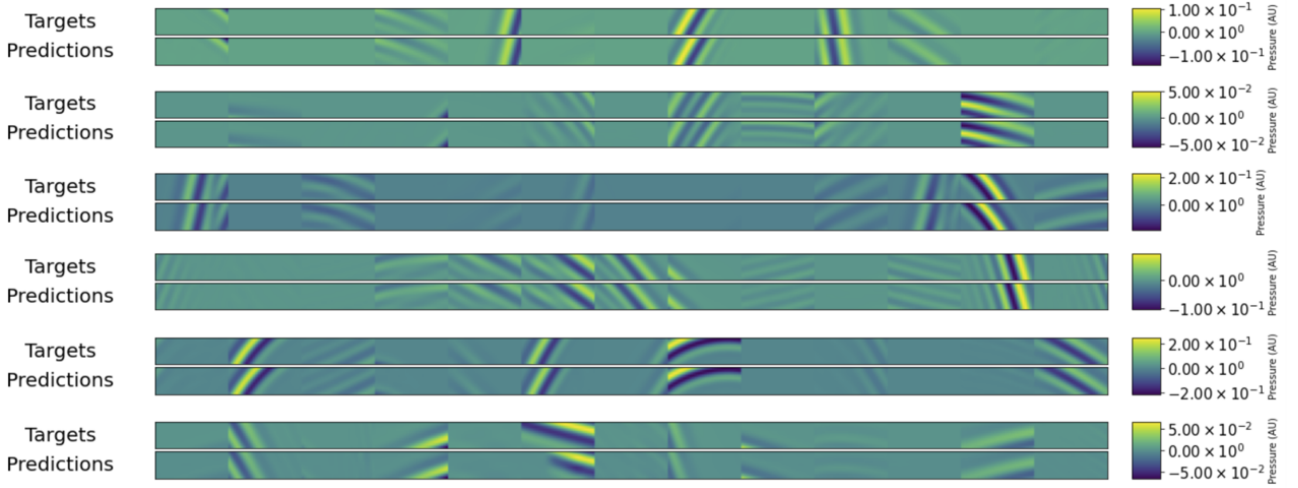


Figure 8: Example ConvLSTM network predictions from test dataset. The test dataset is randomly shuffled in batches of 30 samples to calculate perceptual loss. The network is able to predict different target borders that are qualitatively accurate.



Figure 9: ConvLSTM network predictions of an entire wavefield border at different timesteps. The predicted borders are all qualitatively similar to the real target borders. In comparison with the results produced by Graef, shown in Figure 2, the ConvLSTM network predicts wavefield borders that are significantly closer to the true wavefield.

using a hosted runtime with Google Colab [32]. Examples of test dataset predictions are shown in Figure 8. These are qualitatively similar to the targets, and the perceptual loss on the test dataset was 2.029×10^{-4} . The per-pixel NRMSE loss on the test dataset was 3.501. It was found that the network was able to learn best from a dataset made up of wavefield simulations using 10 different skull models, with randomly located point sources.

Predictions of entire borders can be seen in Figure 9. These show how the network can predict waves propagating to infinity which are developed to different extents. The model’s ability to predict the complex, rippled centres of borders after a wave has passed through them is vital for continued, accurate integration with Devito.

The network was integrated with Devito, and its

predictions of the wavefield borders were substituted in at each timestep. This was repeated for the duration of five wavefield simulations, using random source coordinates each time. An example of different timesteps during a simulation can be seen in Figure 10. The error, represented as percentage error in Figure 10, increases with each timestep as inaccurate predictions are propagated through the wavefield. However, these errors are always low and do not significantly affect the quality of the solution of the wave equation. The average per-pixel error, calculated as the NRMSE between all the Infinite and Network Predictions images over all simulations is 8.251×10^{-4} . The average percentage error between the Network Predictions images and Infinite images is small, at $6.61 \times 10^{-6}\%$. During each forward pass of the network, to predict borders of an entire wavefield, 1.09×10^{11} floating point oper-

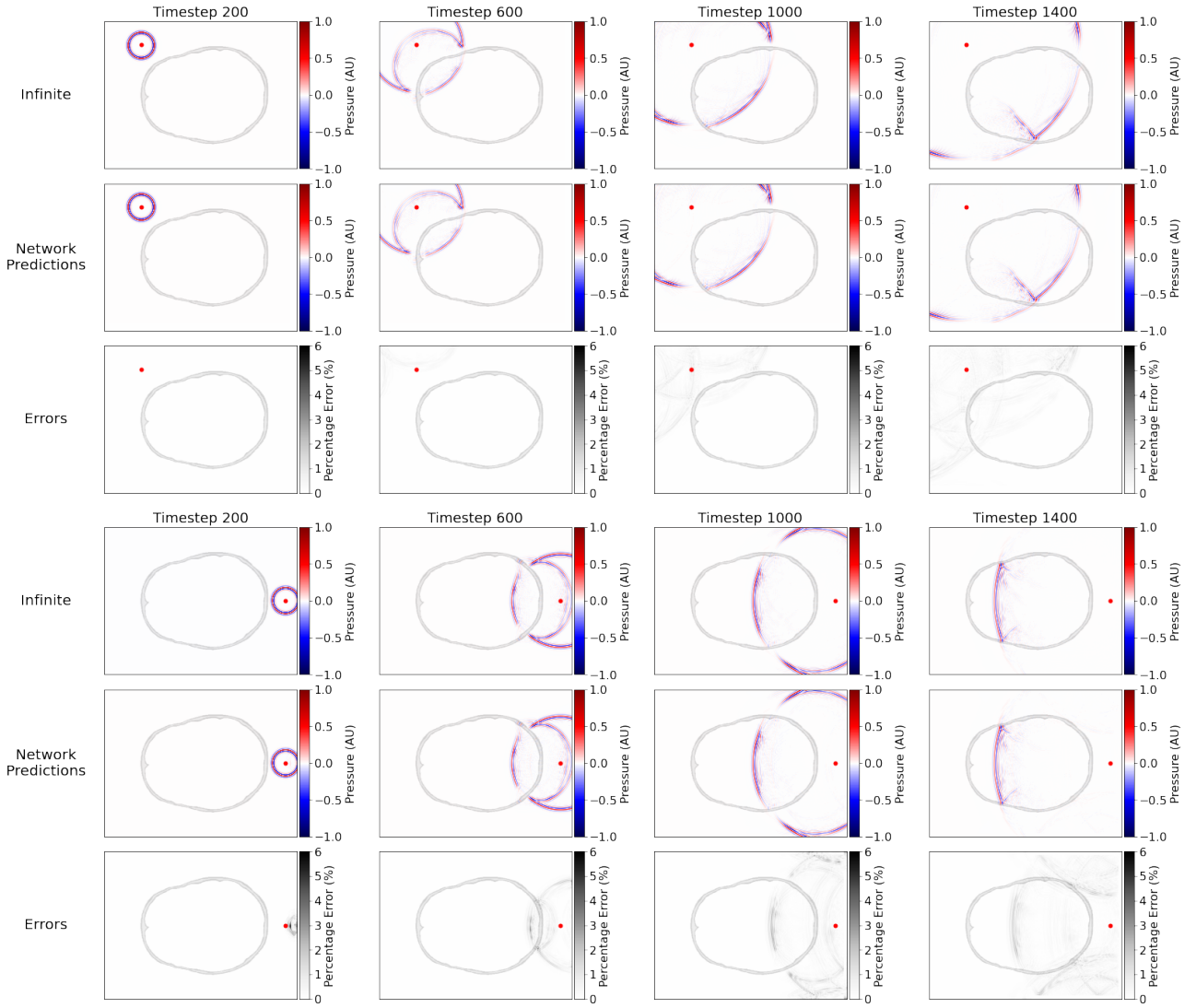


Figure 10: Integration of network predictions with Devito in simulation of a wavefield propagating from a random point source, represented by the red dot. Images labelled Infinite are produced using traditional FD methods. Network Predictions images are produced by substituting in borders predicted by the ConvLSTM network. The Errors of the Network Predictions images, displayed as the percentage difference compared to the Infinite image are relatively low for the duration of the simulation and are not obvious in the Network Predictions images.

ations (flops) are performed.

Discussion

As shown in Figure 8, the trained ConvLSTM performs reasonably well on the test dataset. The error on this dataset is low, at 2.029×10^{-4} , and predicted borders qualitatively resemble real wavefield borders. A low test error is crucial, as when the network is integrated with Devito, any small errors are propagated through the wavefield and can result in small reflection artefacts, which can affect the entire wavefield over time. Errors in the wavefields are important because they will affect the accuracy of the reconstructed image with FWI.

Predictions of entire borders allow greater appre-

ciation of the ConvLSTM network’s accuracy. Particularly for the bottom border in Figure 9, where the real border contains complex, overlapping ripples, the network’s potential is highlighted. The predicted borders are all qualitatively accurate and almost visually indistinguishable from their real counterparts.

The network is integrated with Devito at each simulation timestep successfully, as shown in Figure 10, with border predictions inserted at the edges of the wavefield, reducing the extension of the model to the width of the stencil, 11 pixels. This results in a computational space for calculating wavefields that is smaller than the traditional methods of calculating propagations to infinity by adding absorbing borders and has potential for significantly reducing computational cost if optimised. Despite requiring significant computational power to train the NN, once trained, only a sin-

gle forward pass through the network is required to predict wavefield borders at each timestep. As a result, the smaller computational space over which FD methods operate when the network is fully integrated results in less memory consumption than traditional methods too, and so less RAM is needed for use in an ultrasound scanner.

It was found that any small errors in wavefield border predictions gave rise to discontinuities that were recognised by Devito, resulting in reflections and small rippled artefacts in the wavefield. These are shown in the Error images in Figure 10. An error arising at the beginning of a simulation could be included in the network’s input image for the next timestep, resulting in a prediction that further produces errors in a positive feedback loop. Over time, these errors can build up, resulting in a wavefield that no longer resembles one with propagations to infinity. It is for this reason that the network must be general enough to predict all parts of a wavefield from any source location, with any skull presented to the model.

Figure 10 demonstrates the trained ConvLSTM’s ability to continually predict wavefield borders without compounding considerable errors over time. The maximum per-pixel error, calculated as the NRMSE between the Infinite and Network Predictions images, remains below 3×10^{-3} at every timestep, and the network is able to handle waves originating from different point source locations. The percentage error never exceeds 6%, further demonstrating the network’s ability.

The wavefield images produced with predicted borders are qualitatively and quantitatively similar to those of wavefields propagating to infinity. If used to reconstruct an ultrasound image with FWI, these solutions to the wave equation would result in accurate ultrasound images. This is important for clinical use, where a clinician must be able to look at and understand an ultrasound image in stroke diagnosis, or other brain imaging scenarios. The wavefields produced do not show errors or artefacts that are significant enough to affect the quality of the image, and so the network can be considered appropriate for predicting wavefield borders.

The techniques employed here offer a computational advantage over other methods in terms of efficiency. The alternatives; using a sponge layer or PML [13, 14, 15], require more computational operations than a forward pass through the network [33, 34], which performs 1.09×10^{11} flops at each timestep. The reduced computational complexity achieved by the coupled FD solver and ConvLSTM network results in a faster solution to the wave equation, and therefore accelerates reconstruction of an ultrasound image with

FWI.

3D brain imaging allows more accurate diagnosis than in 2D, which would be beneficial in stroke diagnosis [9]. For a 3D wavefield simulation, with a depth of 400 pixels (the rough average of the current x- and y-lengths), the computational space for FD operations increases 400 times. In 3D, sponge boundaries of a width of 50 pixels would increase the computational domain by 90%. This almost-doubling of the number of numerical calculations required with FD methods is inefficient. Similarly, a PML in 3D is more complex than in 2D, and requires solving even more PDEs. However, vectorization means that the number of operations performed during each forward pass through the network increases linearly with the number of inputs. Therefore, this technique has the potential for large reductions in computational time when applied to 3D problems. This is significant for stroke diagnosis, where a short delay in receiving treatment results in the destruction of millions of neurons [2].

The network created and implemented here can also be considered an improvement on Graef’s CNN for predicting wavefield borders [16]. While Graef’s CNN is unable to predict wavefield borders accurately, as shown in Figure 2, with a perceptual test error of 8.331×10^{-1} [16], the ConvLSTM network produces more realistic borders, shown in Figure 9 with a significantly lower test error of 2.241×10^{-4} . Furthermore, the integration of the ConvLSTM network with Devito shows its ability to continually produce accurate wavefield border predictions, a feat not achieved by Graef.

This project serves as proof of concept for the use of machine learning in the acceleration of brain ultrasound imaging. Faster ultrasound image reconstruction will enable faster diagnosis of stroke, enabling treatment to be applied sooner and increasing chances of survival [3]. Although the solution implemented here has not been completely integrated with FWI, an accurate solution to the wave equation, provided by the coupled FD solver and NN, would produce desirable results once combined. This is because FWI simply seeks to find the differences between a model and the true results provided by the FD solver and network. Therefore, the NN can be assumed to have decreased the computational time required for FWI, confirming the suitability of ultrasound use for fast brain imaging with this method.

The potential for acceleration of 3D ultrasound brain imaging has been demonstrated in 2D. For this to be applied to 3D brain imaging, where the computational time advantage will be even more apparent, a similar ConvLSTM network architecture can be

implemented. 2D convolution of ConvLSTM transitions can be replaced by 3D convolution, and 4D $[9 \times 23 \times 23 \times 23]$ tensors can serve as inputs to the network, used to predict a $[1 \times 1 \times 1]$ border slice, rotated to the correct 3D axis. This is a more complex task for the network to learn, and a larger dataset and longer training time than utilised here will likely be required to achieve high accuracy. A 3D solution must be fully explored to confirm its viability, but the successful implementation of a machine learning solution in 2D here strongly suggests that application to 3D is feasible.

Conclusion

This project has demonstrated the use of a ConvLSTM network to predict the borders of a 2D wavefield in computational space accurately. The methods introduced reduce the time taken for FD methods to calculate wave propagations by reducing the number of operations necessary. Machine learning has been shown to have potential for the acceleration of 3D ultrasound brain imaging by expanding on the techniques used here in the third dimension. This has important applications in faster stroke diagnosis, where the sooner diagnosis is achieved, the sooner treatment can be applied, minimising the risk to a patient [3]. The integration of a NN with a FD solver must be optimised to fully take advantage of the reduced computation required to solve the PDEs for FWI with this method, and the time advantages of this will be most obvious in 3D.

Acknowledgements

I am grateful Dr Mengxing Tang and Mr Carlos Cueto Mondéjar for their supervision whilst undertaking this project. Mr Cueto Mondéjar's endless support and guidance over the last year has been instrumental to its success.

References

- [1] Abolfazl Avan, Hadi Digaleh, Mario Di Napoli, Saverio Stranges, Reza Behrouz, Golnaz Shojaeianbabaei, Amin Amiri, Reza Tabrizi, Naghme Makhber, J. David Spence, and Mahmoud Reza Azarpazhooh. Socioeconomic status and stroke incidence, prevalence, mortality, and worldwide burden: An ecological analysis from the Global Burden of Disease Study 2017. *BMC Medicine*, 17(1), 10 2019.
- [2] Jeffrey L. Saver. Time is brain - Quantified, 1 2006.
- [3] Tapuwa D. Musuka, Stephen B. Wilton, Mouhieddin Traboulsi, and Michael D. Hill. Diagnosis and management of acute ischemic stroke: Speed is critical, 9 2015.
- [4] Duncan S. Maru, Ryan Schwarz, Andrews Jason, Sanjay Basu, Aditya Sharma, and Christopher Moore. Turning a blind eye: The mobilization of radiology services in resource-poor regions. *Globalization and Health*, 6, 10 2010.
- [5] Godwin Inalegwu Ogbale, Adekunle Olakunle Adeyomoye, Augustina Badu-Pepurah, Yaw Mensah, and Donald Amasike Nzeh. Survey of magnetic resonance imaging availability in West Africa. (3), 2018.
- [6] Lluís Guasch, Oscar Calderón Agudo, Meng Xing Tang, Parashkev Nachev, and Michael Warner. Full-waveform inversion imaging of the human brain. *npj Digital Medicine*, 3(1), 12 2020.
- [7] Oscar Calderon Agudo, Lluís Guasch, Peter Huthwaite, and Michael Warner. 3D imaging of the breast using full-waveform inversion. Technical report, 2017.
- [8] Jacqueline N. Welch, Jeremy A. Johnson, Michael R. Bax, Rana Badr, Sam So, Thomas Krummel, and Ramin Shahidi. Real-time free-hand 3D ultrasound system for clinical applications. In *Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures*, volume 4319, pages 724–730. SPIE, 5 2001.
- [9] Qinghua Huang and Zhaozheng Zeng. A Review on Real-Time 3D Ultrasound Imaging Technology, 2017.
- [10] Michael Warner, Andrew Ratcliffe, Tenice Nangoo, Joanna Morgan, Adrian Umpleby, Nikhil Shah, Vetle Vinje, Ivan Štekl, Lluís Guasch, Caroline Win, Graham Conroy, and Alexandre Bertrand. Anisotropic 3D full-waveform inversion. *Geophysics*, 78(2), 2013.
- [11] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6), 2009.
- [12] Qingkai Kong, Timmy Siau, and Alexandre M. Bayen. *Python Programming and Numerical Methods*. Elsevier, 2020.

- [13] Yingjie Gao, Hanjie Song, Jinhai Zhang, and Zhenxing Yao. Comparison of artificial absorbing boundaries for acoustic wave equation modelling. 2015.
- [14] J. Sochacki, R. Kubichek, J. George, W. R. Fletcher, and S. Smithson. Absorbing boundary conditions and surface waves. *Geophysics*, 52(1):60–71, 2 1987.
- [15] Jean Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, 10 1994.
- [16] Cosima Graef. UROP: Accelerating 3D Ultrasound brain imaging with machine learning. Technical report, 2020.
- [17] C.E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, jan 1949.
- [18] Mathias Louboutin, Michael Lange, Fabio Luporini, Navjot Kukreja, Philipp A. Witte, Felix J. Herrmann, Paulius Velesko, and Gerard J. Gorman. Devito (v3.1.0): An embedded domain-specific language for finite differences and geophysical exploration. *Geoscientific Model Development*, 12(3):1165–1187, 3 2019.
- [19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [20] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, Wang-Chun Woo, and Hong Kong Observatory. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. Technical report, 2015.
- [21] Y LeCun and Y Bengio. Convolutional Networks for Images, Speech, and Time-Series SpeechBrain View project Unsupervised Learning of Speech Representations View project. Technical report, 1997.
- [22] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology, 8 2018.
- [23] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. 5 2015.
- [24] Sepp Hochreiter. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. Technical report, 1997.
- [25] Alex Graves and Jürgen Schmidhuber. Frame-wise Phoneme Classification with Bidirectional LSTM Networks. Technical report, 2005.
- [26] Alex Sherstinsky. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, 404, 8 2018.
- [27] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. Technical report, 2016.
- [28] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 9 2014.
- [29] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [31] TESLA P100 PCIe GPU ACCELERATOR. Technical report, 2016.
- [32] Ekaba Bisong. Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pages 59–64. Apress, 2019.
- [33] Barbara Kaltenbacher, Manfred Kaltenbacher, and Imbo Sim. A modified and stable version of a perfectly matched layer technique for the 3-d second order wave equation in time domain with an application to aeroacoustics. *Journal of Computational Physics*, 235:407–422, 2 2013.

- [34] Gang Yao, Nuno V. Da Silva, and Di Wu. An effective absorbing layer for the boundary condition in acoustic seismic wave simulation. *Journal of Geophysics and Engineering*, 15(2):495–511, 2 2018.