

Imperial College London
Department of Earth Science and Engineering
MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

A Deep Learning Approach to Identify Maximum Stress Distribution Caused by Shockwave in Geo-mechanical Materials

by
Hao Lu

hl1319@imperial.ac.uk
GitHub login: acse-hl1319

Supervisors:
Dr. Robert Bird
Dr. Adriana Paluszny

August 2020

Abstract

Wave propagation patterns in solids have been extensively researched in different areas within the geo-mechanical scope, among which the shockwave has been a concern for many tasks, e.g. mining detonation. Conventionally, the finite element method was used in these studies for its extensibility and accuracy. Recently, successes gained in the deep learning area have attracted geophysical scientists to explore ways of applying neural networks to assist geo-mechanical analysis. In this project, ShockNet, an encoder-decoder neural network, was developed to predict maximum stress distributions in a 2D domain caused by randomly initialised shockwaves. Mean absolute errors (MAE) and mean squared errors (MSE) were used to calculate losses of predicted stress field images. The dataset fed to ShockNet was generated by ANSYS Student Edition (ANSYS-SE). The result shows a satisfying convergence trend of MAE and MSE for ShockNet and its variance, reaching at 0.084 and 0.024 (respectively) after 250 epochs. Stress domains predicted by trained ShockNets can render the input into accurate outputs using only 1/400 of the time using ANSYS-SE. Transferred learning was found capable of decreasing the MAE and MSE to 0.081 and 0.021 (respectively) within 100 epochs. Future work could be focusing on training current ShockNets for more epochs and predicting the numerical value of maximum stress.

Keywords: finite element method, shockwave, deep learning, encoder-decoder.

1. Introduction

Wave propagation in solids has been extensively researched in various engineering areas, e.g. waveform inversion for geophysical modelling (Mosser et al., 2020) and material property characterisation via ultrasonic waves (Lahivaara et al., 2018). Various wave types have been studied over the years: ultrasonic wave was used to determine gasoline volatility characteristics by measuring the wave velocity (Takahashi et al., 2000); acoustic emission wave classification was used to enable comprehensive rail crack monitoring (Li et al., 2019); lamb wave sensing network was used in structural health monitoring for aluminium skin (Sbarufatti et al., 2014).

Wave propagation problems can be described by partial differential equations (PDEs). When such PDEs are solvable, one may expect the analytical result that describes the actual problem. However, solutions to real-world wave propagation problems are likely to be unknown, and therefore numerical methods have been dominant for capturing features of wave propagation problems in solids. The most widely used numerical methods include finite difference method (FDM) and finite element method (FEM). FDM can be quickly developed into high order in space but is limited when addressing unstructured mesh (Nie et al., 2020; Lahivaara et al., 2018). By comparison, FEM can mesh arbitrary domains and return high-quality results if the node resolution is refined (Reddy, 2006). FEM has been used to analyse ultrasonic waves in heterogeneous media in 2D (Freed et al., 2016; Lhuillier et al., 2016; Nakahata et al., 2016) and elastic waves in 3D (Van Pamel et al., 2015 & 2017). Among different types of waves, shock propagation has been focused due to its sophisticated essence of irreversible thermodynamic processes and steeping gradients of velocity and temperature (Salas and Iollo, 1996; Mabssout and Pastor, 2003; Baty et al., 2008).

On the other hand, with the improvement of computational abilities, Neural network (NN) has gained considerable developments recently. NN simulates activities of neurons to “learn” features from large datasets to return concluding results like “instinct” without the knowledge of underlying physical equations in advance (Zhu et al., 2018; Kononenko et al., 2018).

Various NN architectures have been developed. Generative adversarial networks (GANs) had been developed for predicting wave propagating patterns in both homogeneous and heterogeneous domains using a multi-scaled generator (Zhu et al., 2018); however, the generator loss increased significantly after 2×10^5 epochs, and non-existing wavefronts were randomly generated at wrong locations. Sorteberg et al. (2018) applied a recursive NN (RNN) with long short-term memory (LSTM) to predict wavefront contours at different stages, which showed good convergence within the first ten timesteps.

For image generation and classification tasks, convolutional NNs (CNNs) have been extensively used in various areas. Liang et al. (2018) developed an autoencoder to adapt arbitrary aorta shapes in 3D and generate stress distribution. Nie et al. (2019) predicted stress fields for cantilevered structures through an encoder-decoder architecture, where the result showed good convergence and the visually negligible difference between predicted domains

and the ground truths. CNNs were also applied to study wave propagating problems. Lahivaara et al. (2018) used an AlexNet-based CNN to characterise tortuosity and porosity. Li et al. (2019) developed a U-Net-based (Ronneberger et al., 2015) autoencoder to predict fracture locations by training it with 2D-seismic sections. Both studies successfully rendered the initial conditions to comparative predictions. Thus, this project focused on the encoder-decoder architecture to deliver good predicted results.

This project used ANSYS-SE (ANSYS-SE) to simulate the Von Mises stress field when the maximum stress overtime occurred. The domain was set as a rectangle in 2D of $10 \times 20 \text{ mm}^2$ and assigned with six geo-mechanical materials. The lower boundary was fixed, while the upper boundary (UB) was loaded with an instantaneous force in a period of $10^{-9} - 10^{-8} \text{ s}$ to simulate a shockwave. The jump in force was randomly allocated on the UB within the range of 0–2N. Initial conditions images were projected based on boundary conditions, force allocations magnitudes and material densities.

An encoder-decoder NN, ShockNet, was developed in this project. ShockNet extract features from a projected initial condition image (the input) to predict the stress distribution field image (the output). During each training epoch, the output was optimised based on the ground-truth numerical results simulated by ANSYS-SE. ShockNet was based on U-Net (Ronneberger et al., 2015), which enabled multi-scaled feature extractions at different depths. ShockNet also incorporates other state-of-the-art designs of modern CNNs. He et al. (2017) developed residual blocks which enable the NN to deepen without gradients vanishment problems, and therefore further increment learning capabilities. Hu et al. (2018) developed Squeeze and Excitation (SE) blocks which squeeze the feature map into a channel-wise numerical value and excite with the original convolutional block. The training process is governed based on image differences, where the criteria used in this project refer to MAE and MSE.

The following sections will contain: (1) how datasets fed to ShockNets were generated, including the numerical result simulations using ANSYS-SE and initial condition image projections; (2) the structure of ShockNet; (3) the code structure designed for this project; results and discussions about the capability of ShockNet and its variants; and (4) emphasis on further studies.

2. Methodology

ANSYS-SE was used to simulate random shockwaves and generate 30,000 stress fields, where initial conditions have been saved in file names. Initial conditions were then extracted to project input domains for ShockNet. ShockNet was developed for this project based on an encoder-decoder structure. The exact methodology used in this project will be structured as following three sections: dataset generation (2.1), ShockNet architecture development (2.2) and numeric stress value predictions (2.3).

2.1. Dataset Generation

This section describes the general domain setup in ANSYS-SE. 2.1.1 discusses the domain material assignments, mesh size selection and ANSYS-SE automation. Shockwave initialisation based on the Heaviside step function using randomised initial force is introduced in 2.1.2, while how sample inputs were projected from initial conditions is discussed in Initial Condition Projection 2.1.3.

2.1.1. Domain Setup

ANSYS-SE was used to simulate stress fields numerically as ground truths. The target domain was 2D that enabled to generate as large datasets as possible to develop ShockNet's learning ability. A 10x20 mm² rectangular domain was initialised in the XY plane. The mesh size of 1mm was used as it could maintain the capability of capturing sharp wavefront features without taking too long to solve the propagation problem (~20s per sample on AMD Ryzen 3600x). The generation process was fully automated with Python scripts using built-in scripting functionality in ANSYS-SE. The initial condition was saved and reused to project corresponding input images for ShockNet. Domains were assigned with six common geo-mechanical materials: dolomite, sandstones, limestone, granite, basalt and chalks, where densities range from 2000 to 3000 kg/m³ (Table 1). For each sample, the field was homogeneous and isotropic. The lower boundary was fixed during the simulation.

Table 1 Physical features of materials used for shockwave simulations.

Material	Sandstones	Dolomite	Limestone	Chalks	Basalt	Granite
Density /kg m ⁻³	2600	2840	2160	2499	3000	2750
Young's Modulus /GPa	34.0	58.6	37.8	16.4	57.0	50.0
Poisson's Ratio	0.234	0.282	0.308	0.078	0.250	0.200

2.1.2. Shockwave Approximation and Randomisation

As a shockwave is initialised by an impact which causes steep kinetic energy difference in a short period, one can expect the discontinuity of a shockwave to be approximated using a Heaviside step function (Villarreal 2006; Baty et al., 2008; Baty and Margolin, 2018):

$$H(E) = \begin{cases} E, & t \in [t_{start}, t_{end}] \\ 0, & \text{others} \end{cases}$$

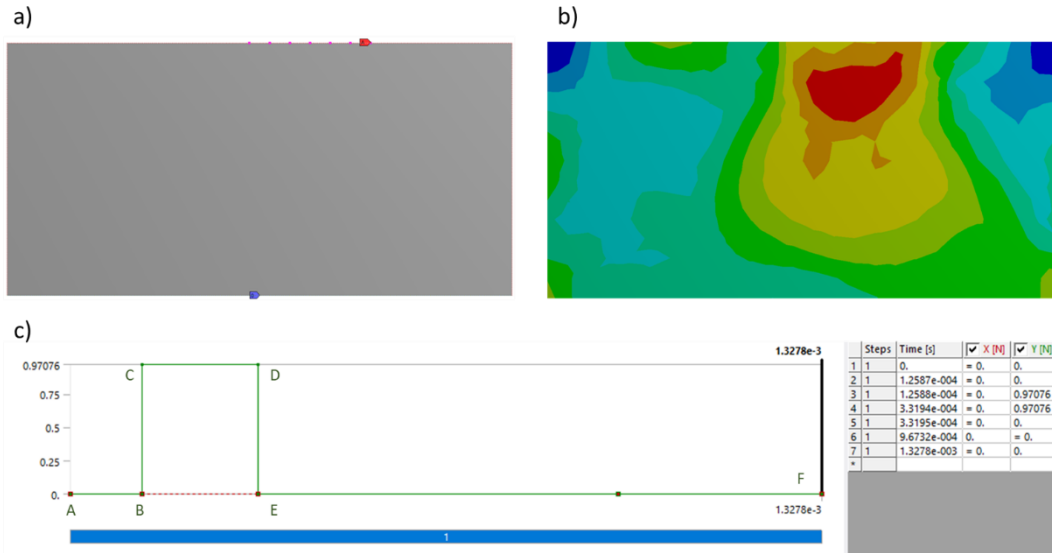


Figure 1: Force initialisation. a) and b) were examples for the initial condition and solved results from ANSYS-SE. The lower part c) refers to discontinuities in force. In Figure 1.c, the instantaneous force is loaded on nodes (or the entire boundary) at B and increased to the randomised magnitude at C where the time difference between B and C is of 10^{-8} - 10^{-9} s, and therefore deliberately created a shockwave at the upper boundary. The force vanishing process is the same as force loading.

where t_{start} and t_{end} define when the force was loaded on and removed from the UB. E stands for the kinetic energy of each node on UB. Random instantaneous forces were applied upon UB to simulate randomised shockwaves. t_{start} and t_{end} were randomly generated during the entire simulation period. The total simulation period was of 10^{-5} s, and the time scale of the force jumping was randomly set between 10^{-9} – 10^{-8} s to better the approximation of shockwaves (Figure 1).

The Explicit Dynamics module from ANSYS-SE Mechanical platform was used in this project which applied a finite element (Lagrange) solver. An improved arbitrary Lagrange-Euler algorithm was developed for the Explicit Dynamics module to address sharp jumps in shock response problems of short time-scales (ANSYS Inc., 2011). The Explicit Dynamics module could apply instantaneous forces to model a shock on the entire UB (boundary force) or selected nodes (nodal force).

The combination of randomised forces on UB and different materials helped to increase the contour shape varieties of the output stress domain. Two UB force randomisation methods were used: the first applied a randomised force on nodes from the entire UB, while the second applied forces on arbitrarily selected nodes on UB, which simulated different shock source lengths (Figure 2). Both randomisations applied a force within the range of 0-2N per node.

Examples for both the boundary force and the nodal force were shown in Figure 2. During the automated ANSYS-SE simulation, predicted stress fields and initial conditions were saved. Saved information included the simulation time, the maximum Von Mises stress over time and the minimum stress recorded at the same stage, and time slots for the force discontinuity. For forces cast on selected nodes on UB, indices of loaded nodes were recorded. Output stress fields were saved as EPS files and reshaped into 48x96 and 64x128.

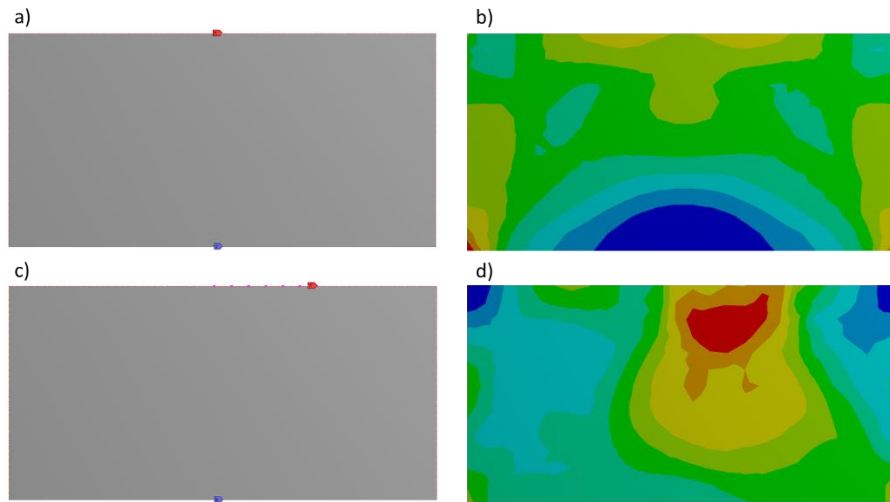


Figure 2: Initial force and supporting conditions (left) and corresponding numerical results (right). For a) forces are cast on the entire upper boundary, where the numerical result showed symmetric stress field in b). For c), the load was added on randomly selected nodes, resulting in asymmetry as shown in c).

2.1.3. Initial Condition Projection

Initial conditions were unpacked and then processed via linear scalings. Figure 3 illustrates examples of the projected initial conditions as inputs for ShockNet. The size (number of pixels in width and height) of the input image was the same as stress field image. The magnitude of shock forces and material densities were scaled into RGB channelled colours to present initial conditions. A scale of 0.085 was set for projecting the density into greyscale pixels, which resulted in 183-255 corresponding to the density range of 2160-3000 kg/m³.

Similarly, a scale of 127 was set to visualise projected forces (darker yellow denotes larger force magnitude). The lower 1/10 of the image was set as red to represent the fixed supporting lower boundary. Other areas were assigned with blue (B: 255) to maintain the 2:1 ratio of width to height and stand out from other scaled sections.

In this project, ShockNet was only implemented to accept one image at a time. Therefore, although Poisson's ratio and Young's modulus were also available initial conditions, they were omitted when projecting the input.

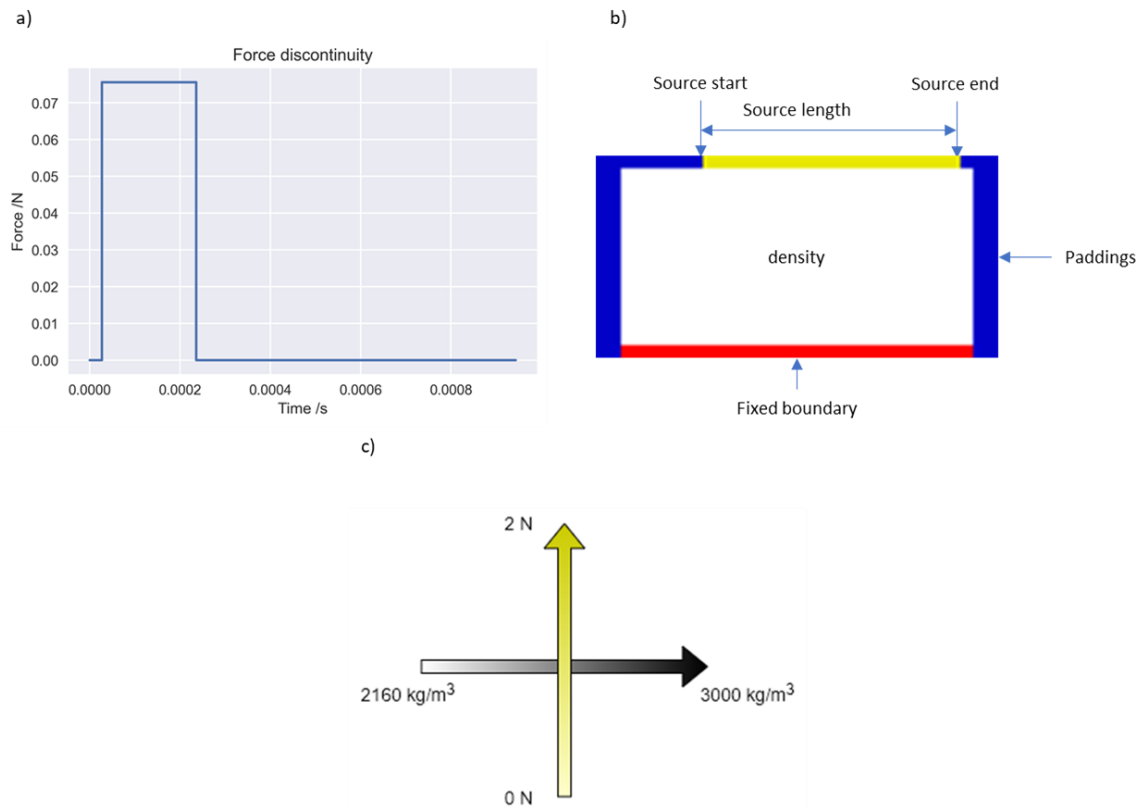


Figure 3: Initial condition projection examples. Figure a) denotes the force discontinuity cast on the source (dark yellow) in figure b). b) illustrates the components in an initial condition image and c) explains how the colour scale related to the physical parameters.

2.2. ShockNet

In recent studies, CNNs have been widely used for image classification and image generation tasks (Bai et al., 2018; Liang et al., 2018; Balu et al., 2019; Nie et al., 2019; Gao et al., 2020). CNNs contain convolutional blocks which apply feature extracting kernels to capture information from the input. The encoder-decoder structure contains CNN blocks to encode features and transpose CNN layers to decode extracted information. Encoder-decoder structures have been extensively applied for image generation and classification tasks. U-Net (Ronneberger et al., 2015) is one of the most successful encoder-decoder structures which have shown practical value in biomechanical (Bai et al., 2018; Liang et al., 2018; Balu et al., 2019) and geo-mechanical areas (Nie et al., 2019; Gao et al., 2020).

2.2.1. Image generation architecture: ShockNet

The original U-Net contains, however, only 12 layers from the input to the bridge block, which is shallow compared to more recent CNN architectures. One of the reasons that the original U-Net cannot be extended to further depth is gradient vanishment: when the NN architecture becomes deep, the small gradient will decrease further under the chain rule through propagations (He et al., 2016). He et al. (2016) introduced the residual network (ResNet), which enables the NN to be as deep as 150 layers by preserving the information from previous blocks. The residual block helps to enable the flow of memory/information from the initial layer to the final layer.

At each ShockNet level (SL), a Squeeze and Excitation (SE) block was added. SE structure was introduced by Hu et al. (2018) which extended the advantage of ResNet. SE blocks enable channel-wise multiplication with a numerical value, which has shown the strength to decrease the top-1 error ranging from 0.61 to 0.85 (depending on the depth) with negligible additional computational cost in classification problems (Hu et al., 2018).

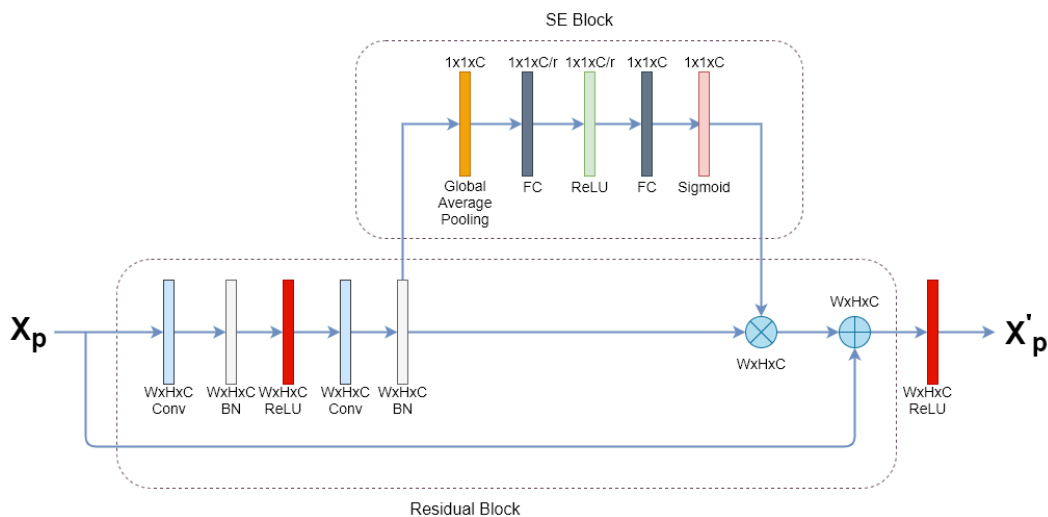


Figure 4: A ShockNet block at one SL. Each residual block contains two convolutional blocks, followed by a batch normalisation for each. A copy of X_p was reserved. The output from the second batch normalisation layer would be (1) sent to the SE block and (2) saved as a copy. Channel-wise scales are output by the SE block and multiply with the saved copy. The multiplied result would then be added with the reserved copy of X_p .

Based on previous successful improvements, ShockNet was developed. Figure 4 presented a single SL. Each SL contains a basic two-layer residual block and an SE block. Before activated by the last ReLU layer, the last batch normalisation layer would be sent to the SE block for channel-wise multiplication.

The overall ShockNet architecture was shown in Figure 5. ShockNet down-sampled the input image to extract initial condition features. A max-pooling layer was added after the SE block to extract the most prominent features from the last level. At the end of the SE block at each depth, the tensor forwards in two directions: one directs to a max-pooling layer for downsampling side, and another is preserved for later concatenation to the upsampling side. Such extracted features are reserved at each SL for concatenation with upsampling blocks, enabling multi-scaled information additions at each SL.

ShockNet2, ShockNet3 and ShockNet4 were implemented corresponding to 2, 3 and 4 SLs, respectively, where this project only experimented the latter two. After a specified number of SLs, a bridge block connects the downsampling part with the upsampling part. The upsampling part contains the same SLs but is connected by transposed convolutional layers.

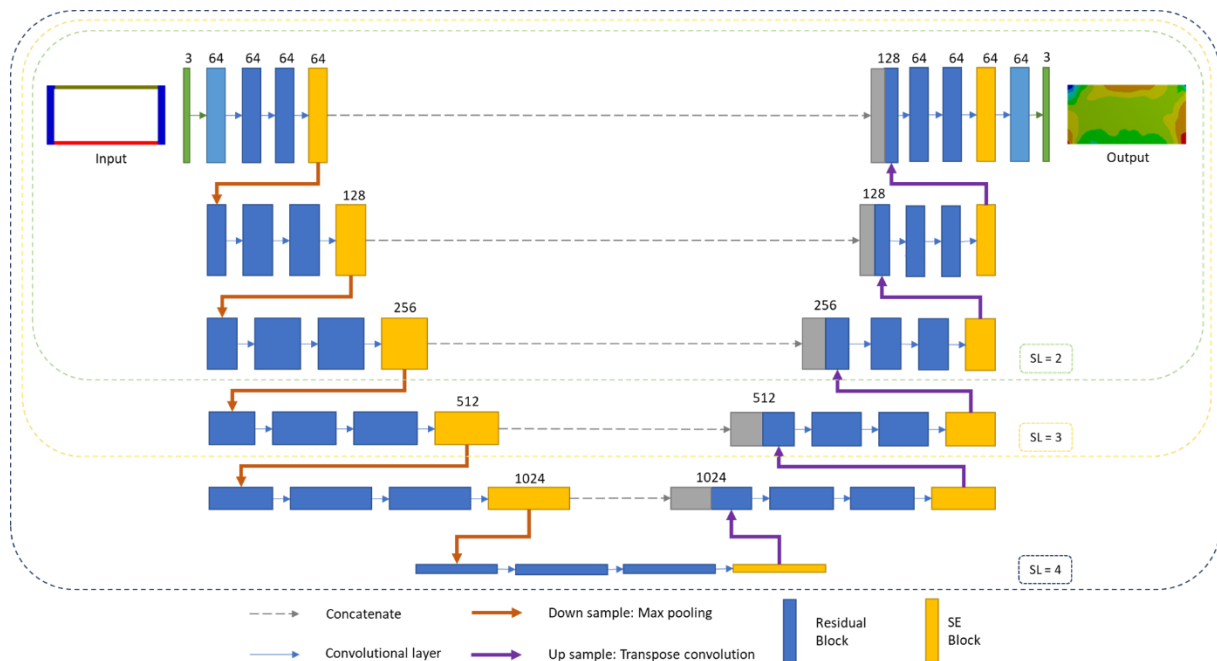


Figure 5: ShockNet architecture. SL numbers were labelled with the defined scope. Red and purple arrows represent max pooling and transpose convolution layers.

2.2.2. ShockNet Training Strategies

This project used MAE and MSE to track the training and validation losses of images:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^3 (y_{i,c} - \hat{y}_{i,c})$$
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \sum_{c=1}^3 (y_{i,c} - \hat{y}_{i,c})^2$$

where $y_{i,c}$ refers to the predicted value of a pixel at each RGB channel whereas $\hat{y}_{i,c}$ refers to the ground truth; n refers to the total number of pixels (8192 and 4608 for 48x96 and 64x128, respectively). ADADELTA was used as the optimiser to automatically changing the learning rate without aggressively decreasing it (Zeiler, 2012). Both ShockNet3 and ShockNet4 were trained to determine how depths would affect the training and validation process.

Data augmentation was frequently used to feed NNs with more diversified data if the dataset size was limited (Antoniou et al., 2017; Shorten et al., 2019; Zhong et al., 2020). However, initial conditions and numerical stress fields in this project were strictly corresponded, suggesting that augmenting only the input would possibly compromise the exact relation between the initialised domain and the ground truth result.

As ShockNet uses ReLU as its primary activation function, Kaiming initialisations were used in this project (He et al., 2016). In this project, Kaiming normal initialisation (KNI) was the primary initialisation method, whereas Kaiming uniform initialisation (KUI) was also applied for comparison.

2.3. Image generation with stress value predictions: TShockNet

The author also developed a two-branched architecture to enable ShockNet not only to generate stress domain images but also to predict numeric values of maximum and minimum stresses (TShockNet). TShockNet would encode the input image first as in ShockNet during down-sampling, but in addition to decoding the low-level features and generate images, TShockNet would also direct to fully connected layers to predict maximum and minimum stress values. As the range of maximum and minimum stress varied substantially ($10^3 - 10^{10}$ Pa), TShockNet predicted the logarithm of the stress:

$$I_{max} = \log_{10} P_{max}$$

$$I_{min} = \log_{10} P_{min}$$

instead of the actual stress value, where P_{max} and P_{min} are maximum and minimum stress, respectively. The L1 loss was used to calculate the summation of maximum and minimum stress losses deviated from the ground truth:

$$L1loss_{stress} = \sum_{i=1}^n \{(I)_{max} - I_{max, true}\} + \{(I)_{min} - I_{min, true}\}$$

where $n = 48 \times 96 = 4608$ for TShockNet.

3. Code Structure

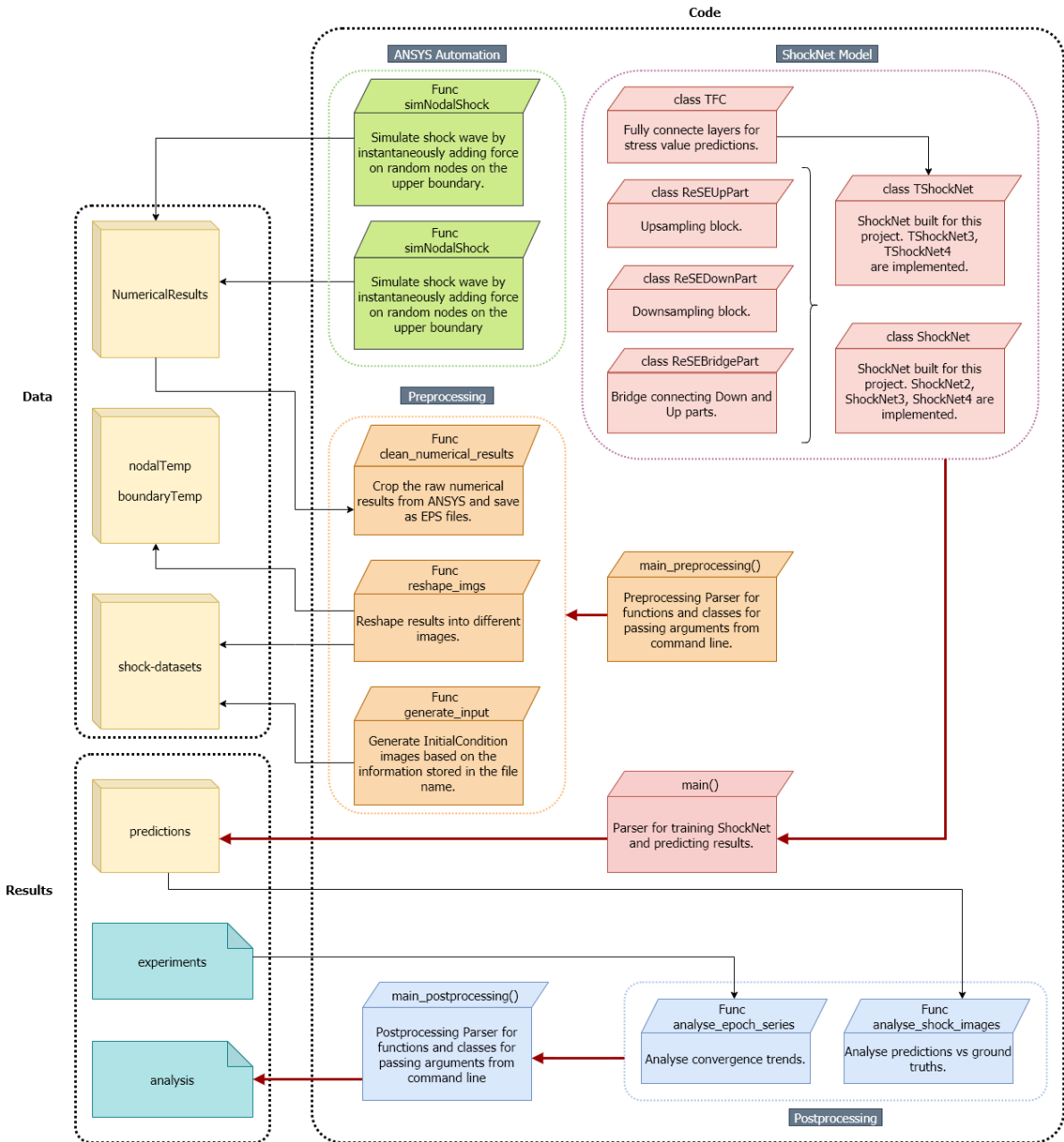


Figure 6: Module structure for codes.

Figure 6 showed the primary code structure, including ANSYS SE automation, ShockNet model designs and pre- and post-processing. ANSYS-SE automation codes enabled fast randomised shockwave simulations, where the numerically calculated stress fields were stored and processed by Preprocessing codes. Preprocessing codes cleaned and reshaped the original output from ANSYS-SE for projecting initial conditions as inputs for ShockNet.

ShockNet codes were composed of basic ShockNet blocks, where TShockNet had a new branch of fully connected layers for numeric stress predictions. ShockNets and TShockNets

GitHub repository: <https://github.com/acse-2019/irp-acse-hl1319>

were parsed in the `main()` function where training and validation losses were recorded in *experiments*, and ShockNet-predicted stress fields were stored in *predictions*.

Postprocessing codes would then pick up loss records per epoch from *experiments* and *predictions* from ShockNets to generate results for analysis and discussions in 4.

4. Results and Discussion

Two types of datasets had been trained in this project: datasets with forces initialised on the entire boundary (BDs) and randomly selected nodes (NDs). The dataset size was 30,000 for each image pixel resolution, which had been divided into 22,000 and 8,000 as training and validation subsets, respectively. ANSYS-SE approximately spent 15–22s on simulating each shockwave case and about 24 hours on running all 30,000 samples (AMD Ryzen 5 3600X CPU). Data were generated by simultaneously running six ANSYS-SE processes on the desktop which consumed over 16GB RAM (ANSYS-SE's Explicit Module does not support parallelised computations for 2D problems within a single workbench).

All codes were implemented based on the open-source framework, Pytorch, and run on Google's Colaboratory, which provided server-level backend GPUs. Training and validation sessions were run on NVIDIA's Tesla P100-PCIE-16GB.

4.3. Training and Validation Losses

This section shows the results of convergence comparisons between ShockNet4 trained on NDs and BDs (4.3.1). As a result, NDs provided more variable stress domains which thus improved the learning capability of ShockNet4. In 4.3.3, comparisons between ShockNet3 and ShockNet4 using KUI and KNI were shown and discussed.

4.3.2. NDs vs BDs

Table 2 shows the mean and standard deviation difference of RGB channels between the two datasets and denotes the validation MAE and MSE of ShockNet4 after trained for 200 epochs. Comparing to BDs, NDs shows more considerable internal variance, which improves ShockNet's learning capability. When trained with BDs, the symmetric instantaneous load lead to higher losses for both MAE and MSE. Figure 7 compares the trend of convergence when trained on BDs and NDs. ShockNet4 trained with BDs converges faster in a more stable pattern but stagnates after only ten epochs. By comparison, when trained with NDs, the ShockNet4 shows the consistent potential to reach lower loss for both MSE and MAE. In this case, in the following experiments, NDs are primarily used for training the ShockNet and predicting stress fields.

Table 2: Input internal variance and validation results after trained for 200 epochs for NDs and BDs of 48x96 resolution.

Dataset	Mean	Std dev	MAE	MSE
NDs	R: 180.15178385	R: 84.79378607	0.0843	0.0230
	G: 204.8688151	G: 61.2799176		
	B: 181.6319987	B: 89.29453818		
BDs	R: 200.43989413	R: 70.55906642	0.0873	0.0317
	G: 229.30037593	G: 11.49446858		
	B: 200.44750417	B: 81.78639965		

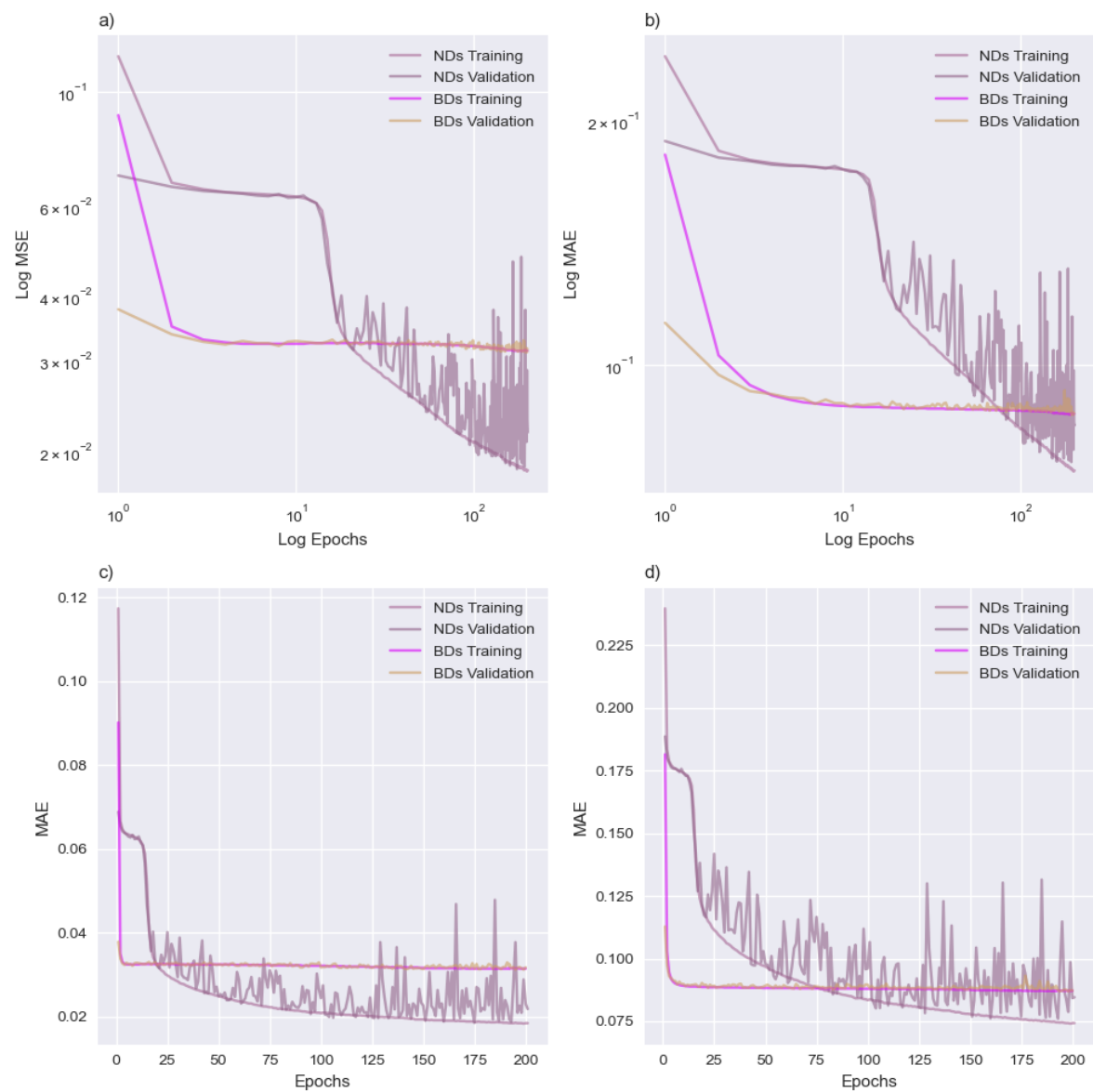


Figure 7: Comparison of ShockNet4 between training with BDs and training with NDs. Figure a) and b) refer to log-log plots.

4.3.3. Comparisons between ShockNet and its variant

Figure 8 shows the performance of ShockNet for different SL numbers and initialisation methods when trained with NDs. ShockNet4 was initialised with both KUI and KNI methods (ShockNet4-uni and ShockNet4-norm, respectively), and ShockNet3 with KNI (ShockNet3-norm) was also added for parallel comparison.

All three curves in Figure 8 show good convergence trends where the optimisers are working to reach the global minimum. All experiments share a gradually more considerable oscillation pattern of validation losses, where ShockNet3-norm shows the most stable convergence. MAE appears to be more sensitive to the training process, showing a steeper decrease with epochs in Figure 8.

The platform-like pattern showed by all curves is a typical period when the optimiser is temporarily trapped within the local minimum. ShockNet3-norm escapes from the platform-like feature faster than the ShockNet4 variants (10 epochs for ShockNet3-norm comparing to 15 and 20 epochs for ShockNet4-norm and ShockNet4-uni respectively).

ShockNet4-norm presents the lowest MAE and MSE within 200 epochs, where ShockNet4-uni shows similar training losses yet unstable validation errors. It appears that the KUI method will cause the losses to fluctuate substantially. The increase starts at 125th epoch, where the validation MSE reaches 0.10 and MAE at 0.19. The amplitude continues to soar to 0.16 and 0.23 for MSE and MAE, respectively. One reason for this is due to the improper parameter initialisation via KUI that caused the ADADELTA optimiser to be stuck within a local minimum which was also seen by Nie et al. (2019).

ShockNet3-norm, on the other hand, presents the most considerable losses, ending at 0.024 (MSE) and 0.085 (MAE) after 200 epochs, whereas ShockNet4-norm reaches at 0.019 (MSE) and 0.075 (MAE). The curve shape is similar to it of ShockNet4-norm, though, which has been continuously decreasing stably until 200 epochs.

Results suggest that (1) KNI is superior to KUI for ShockNet initialisations, (2) both ShockNet3-norm and ShockNet4-norm can

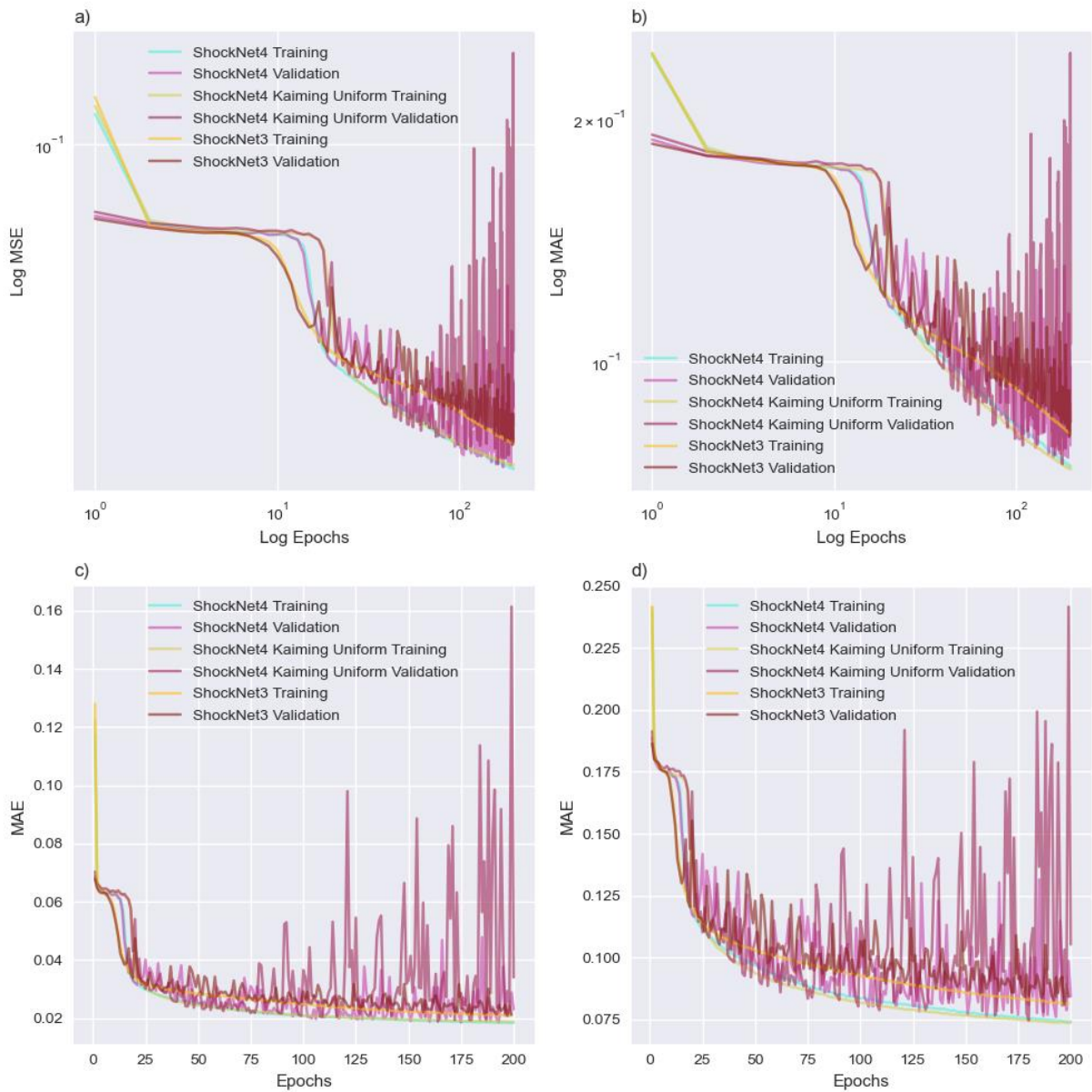


Figure 8: Convergence comparisons. Each graph contains training and validation MSE and MAE of experimenting with (1) ShockNet4 using KNI, (2) ShockNet4 using KUI, and (3) ShockNet3 using KNI. Figure a) and b) refer to log-log plots.

4.4. Predictions vs Ground Truths

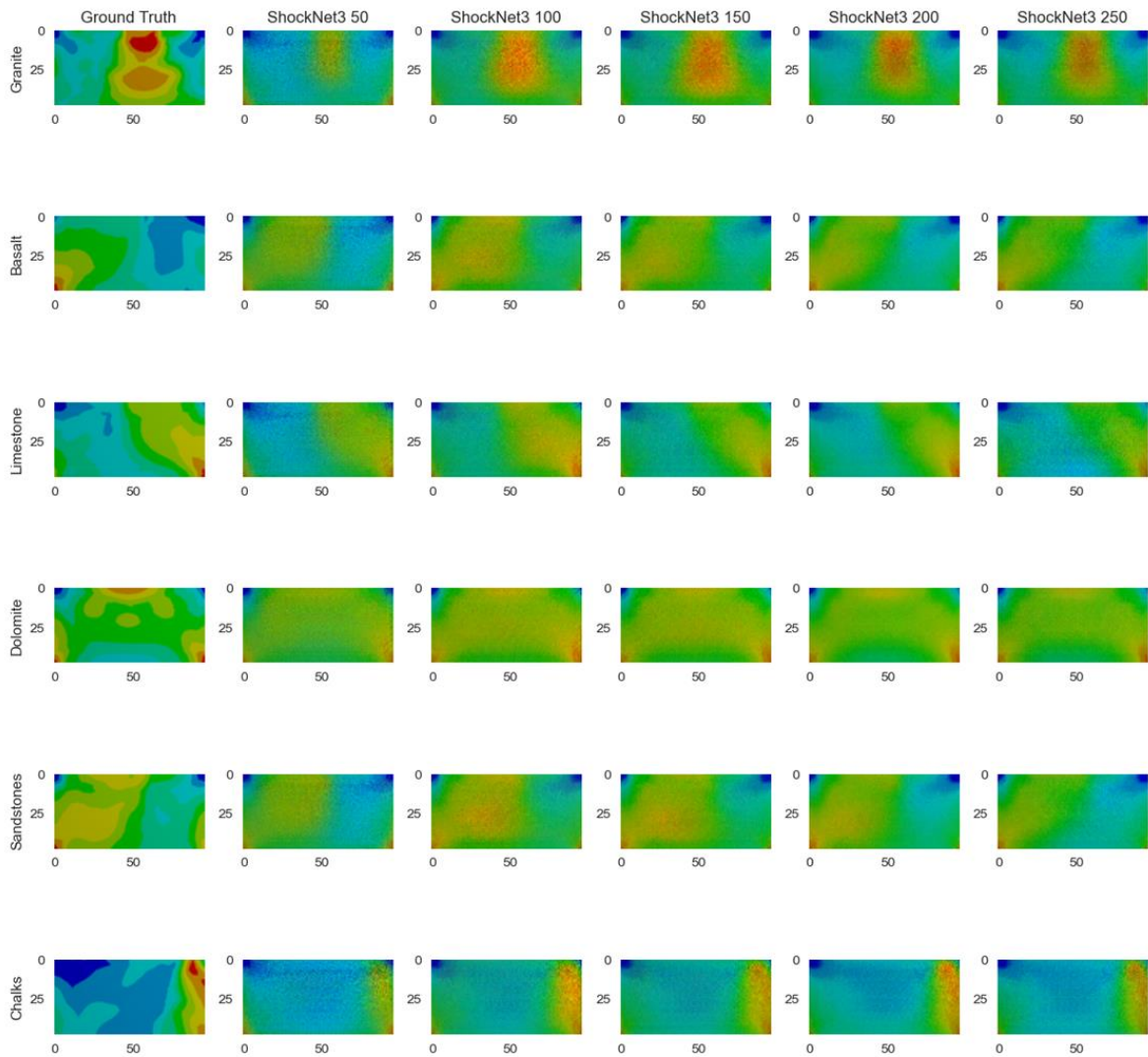
ShockNet3 and ShockNet4 trained with different epochs were saved and reused to predict stress fields (Figure 9). Samples were randomly extracted from the prediction subset for each material initialised in the dataset. It took an average of 0.051s and 0.063s to generate a stress field prediction for ShockNet3 and ShockNet4 (on NVIDIA's Tesla P100-PCIE-16GB), respectively, comparing to the average time consumption of 20s for ANSYS-SE.

Both ShockNet3 and ShockNet4 have been generating progressively more precise outputs when trained with more epochs. Contours are indistinguishable at 50th and 100th epoch yet become generally sharper after 250 epochs (Figure 9). The area of maximum stress is becoming more appropriately proportioned with the contours refined every 50 epochs. Noise features are dominant at 50th epoch but substantially suppressed at 250th epoch. Comparing to ShockNet3, ShockNet4 shows more promising ability to render more details and reduce noises in the output stress domain, which, however, costs about 26.3% more time to train with NDs per epoch than ShockNet3 (~19 minutes for ShockNet3 and ~24 minutes for ShockNet4).

Figure 10 shows six basalt results generated from ShockNet4 after trained for 250 epochs. Among six samples, Basalt 1 shows minimum MSE (0.0112) and MAE (0.0630) whereas Basalt 5 shows maximum MSE (0.0347) and MAE (0.1109). MSE and MAE increase when the maximum stress area (i.e. red area) become more substantial. All samples indicate that ShockNet4 can identify the maximum and minimum stress areas at corners. The comparison between Basalt 1 and Basalt 4 suggests the effect of the internal variance of the ground truth image: the large minimum (blue) stress area in Basalt 1 is sharply differentiated from the medium stress (green) area whereas, in Basalt 4, the area of minimum stress is much smaller which reduces ShockNet4's ability to refine the stress boundary.

GitHub repository: <https://github.com/acse-2019/irp-acse-hl1319>

a)



b)

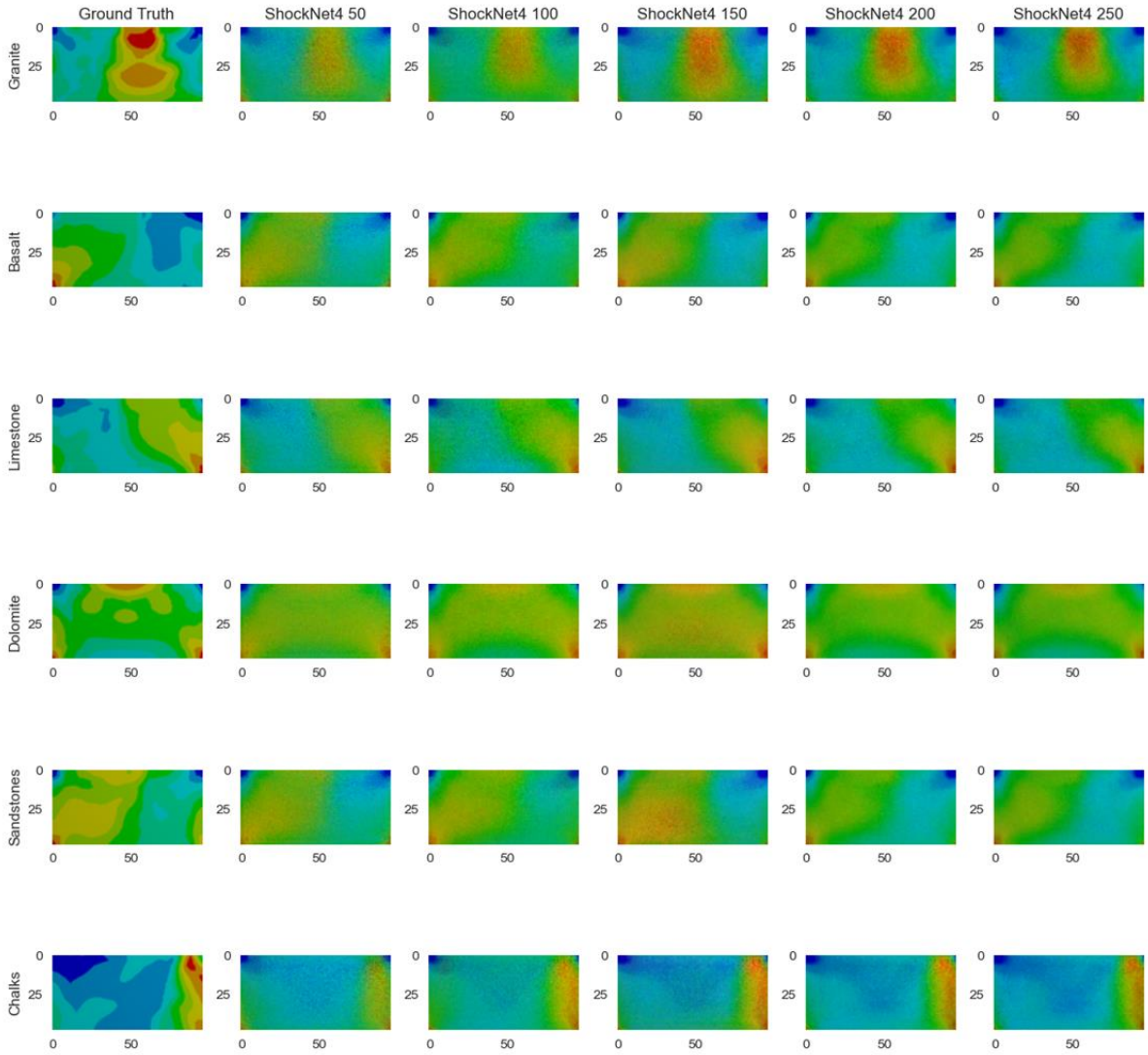


Figure 9: Visual comparisons between predicted maximum stress field and the ground truth result (first column). 3-SL (a) and 4-SL (b) ShockNets were used to generate the predictions. Rows represent different material samples, whereas the last five columns refer to the different number of epochs of which 3-SL and ShockNet4s had been trained.

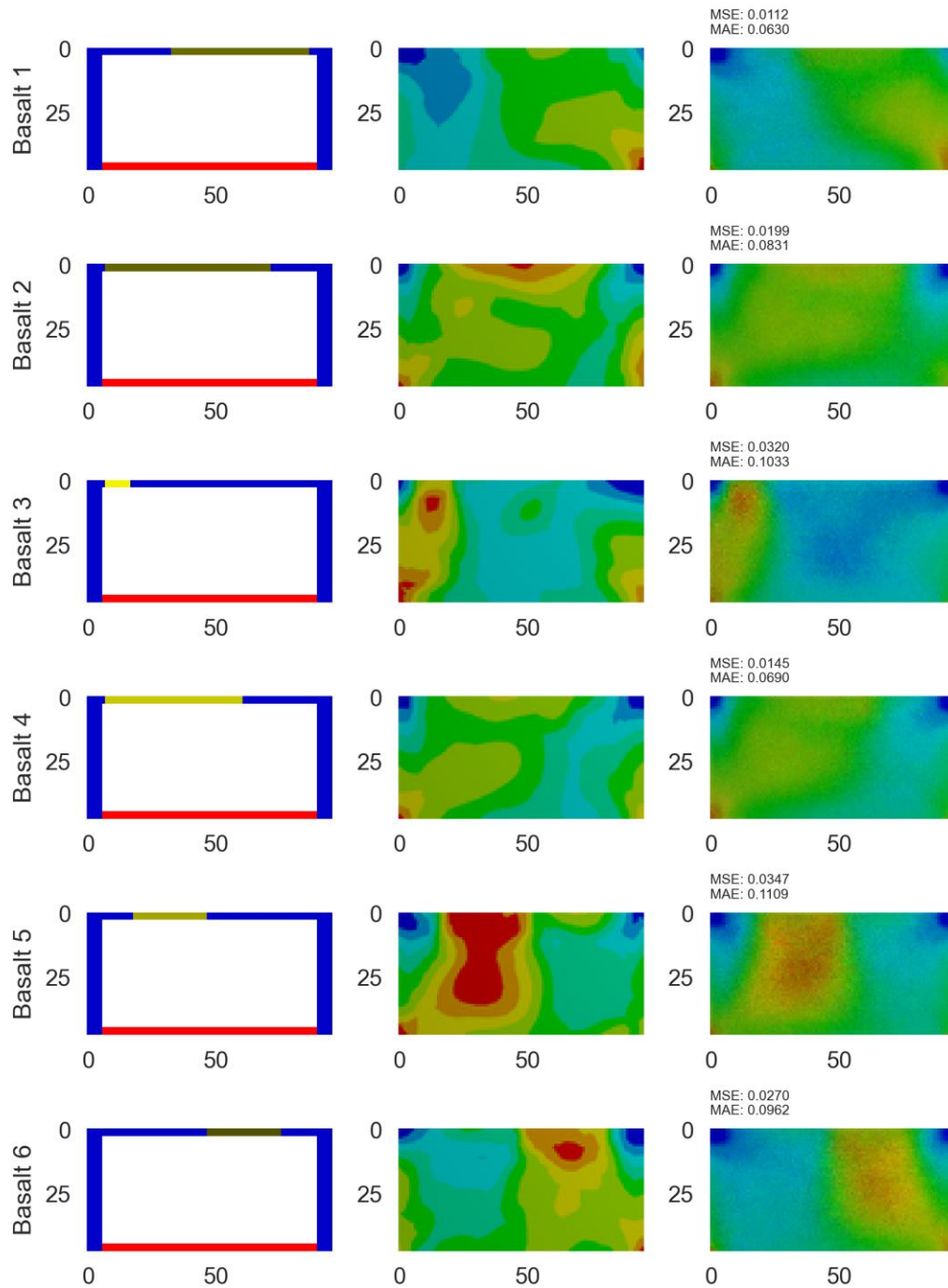


Figure 10: Results generated from ShockNet4-norm where corresponding initial conditions and ground truths are presented. MSE and MAE are calculated. Basalt was selected as the sample material.

4.5. Transferred Learning Capabilities

Transferred learning shows that the ability of ShockNet to generalise what has been learnt to a new dataset of different resolution. Experiments on the performance of transferred learning are shown in Figure 11. Unlike ShockNet4, transferred 4-SL ShockNet (ShockNet4-transferred) does not display the second time drop in losses. The oscillation pattern of validation losses shows a more stable convergence trend towards training losses. Although only trained with less than 100 epochs, ShockNet4-transferred reaches approximately the same training and validation losses as ShockNet4 trained for 250 epochs. This shows the promising capability of ShockNet4 to be applied on high-resolution images when it has only been trained with coarse datasets.

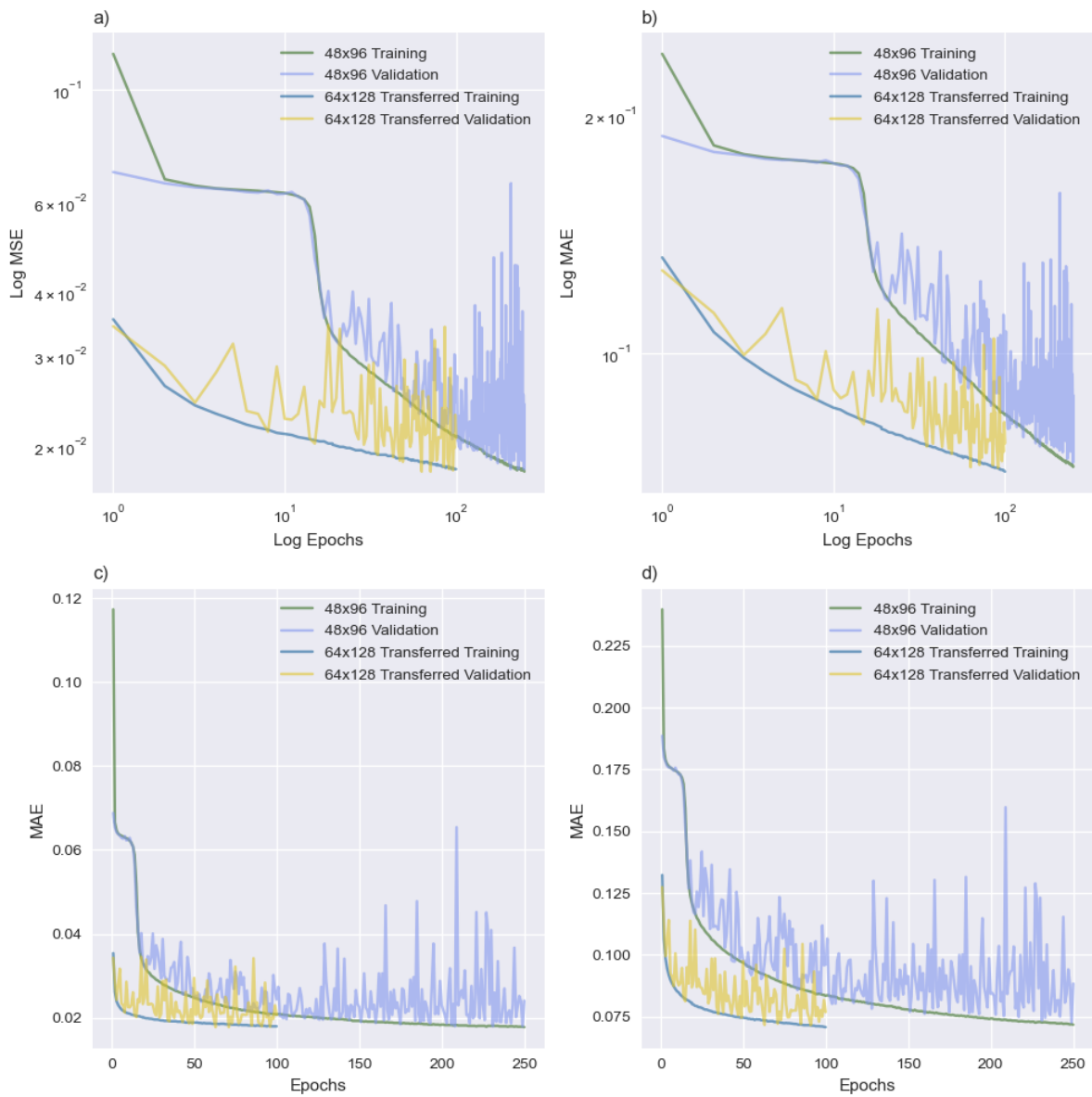


Figure 11: Comparison between ordinary learning using 48x96 dataset and transferred learning using 64x128 dataset. Parameters were inherited from the ShockNet4 after trained for 250 epochs with 48x96 dataset.

4.6. Maximum stress predictions

TShockNet was developed to enable ShockNet to learn from both the geometric distribution of stresses and the numeric value of maximum and minimum stresses. However, TShockNet took ~4 hours to finish a training & validating cycle on NVIDIA's Tesla P100-PCIE-16GB, and therefore only 22 epochs had been trained.

Figure 12 shows the TShockNet experiment results. Training loss for L1stress shows a bounce-back pattern. Such a pattern is due to the small batch sizes used for training TShockNet. When the batch becomes too small, the learning ability of a deep NN will suffer due to non-monotonically improving the loss function. On the other hand, the validation losses have been continuously decreasing overtime for all three criteria.

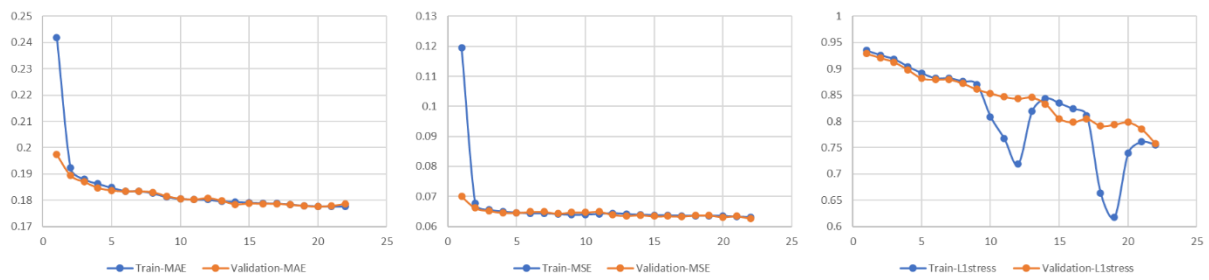


Figure 12: TShockNet training results. L1stress is the L1 loss of the numerical stress exponent.

5. Conclusions and Future Work

In this project, ShockNet, an encoder-decoder-based image generation architecture, had been built to explore its potential to replace numerical shock stress predictions. By training with NDs and BDs, ShockNets show abilities to learn features from the datasets overtime and produce comparable results to highlight the maximum stress area.

A summary of results is as follow:

1. NDs vs BDs: NDs could improve the learning capability of ShockNet due to more considerable internal variance of the datasets.
2. KNI vs KUI: ShockNet architecture, KNI worked as the best initialisation method.
3. ShockNet depth effects: The effect of depth of ShockNets was studied by comparing ShockNet3 and ShockNet4. ShockNet showed better prediction qualities and lower validation losses. By comparison, ShockNet3 showed faster and more stable convergence.
4. Transferred learning: Transferred learning showed promising potentials of ShockNet to be generalised and applied on datasets of various resolutions. Stress prediction architecture was implemented, but the training time was limited.

Additionally, ShockNets were able to predict results that depicted the general contour of the stress field, which was a less time-consuming way comparing to ANSYS-SE. The resolution of predicted images became clearer when trained with more epochs. Noises had been substantially suppressed but still existed. One reason for this was the insufficient training epochs. From previous successful studies, about 1,000-20,000 epochs were trained to achieve relatively low losses and visually satisfying qualities (Zhu et al., 2018; Nie et al., 2019), about 10^1 - 10^2 times larger than what had been done in this project.

Though ShockNet has shown its primary ability and value to be further studied as an alternative way to analyse shockwave-induced stress distribution, further efforts are required to improve the performance of it. Future work could focus on the following directions: further training on ShockNet3 and ShockNet4 to explore the actual stop point of the training process (where the validation loss soars); continuing the transferred learning process for higher resolutions to examine the generalisation capabilities; more experiments on TShockNet to explore the accuracy and performance of it to predict actual stress range.

Bibliography

1. ANSYS Inc., 2011. Explicit Dynamics. Available at: <https://www.ansys.com/-/media/ansys/corporate/resourcelibrary/brochure/ansys-explicit-dynamics-brochure-140.pdf>.
2. Antoniou, A., Storkey, A. and Edwards, H., 2017. Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340.
3. Bai, W., 2018. Automated cardiovascular magnetic resonance image analysis with fully convolutional networks, 1–12. <http://doi.org/10.1186/s12968-018-0471-x>.
4. Balu, A., Nallagonda, S., Xu, F., Krishnamurthy, A., Hsu, M.-C., & Sarkar, S., 2019. A Deep Learning Framework for Design and Analysis of Surgical Bioprosthetic Heart Valves. Scientific Reports, 1–12. <http://doi.org/10.1038/s41598-019-54707-9>.
5. Baty, R.S. and Margolin, L.G., 2018. Modern infinitesimals and the entropy jump across an inviscid shockwave. International Journal of Aeroacoustics, 17(4-5), pp.502-520.
6. Baty, R.S., Tucker, D.H. and Stanescu, D., 2008. Nonstandard jump functions for radially symmetric shockwaves (No. LA-14380). Los Alamos National Laboratory (LANL), Los Alamos, NM.
7. Freed S, Blackshire J, Na J., 2016. Ultrasound finite element simulation sensitivity to anisotropic titanium microstructures. In Proc. of the Review of Progress in Quantitative NDE, Minneapolis, MN, 25–31 July 2015. Melville, NY: AIP Publishing.
8. Gao, W., Lu, X., Peng, Y. and Wu, L., 2020. A Deep Learning Approach Replacing the Finite Difference Method for In Situ Stress Prediction. IEEE Access, 8, pp.44063-44074.
9. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
10. Hu, J., Shen, L. and Sun, G., 2018. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).
11. Kononenko, O. and Kononenko, I., 2018. Machine learning and finite element method for physical systems modeling. arXiv preprint arXiv:1801.07337.
12. Lähivaara, T., Kärkkäinen, L., Huttunen, J.M. and Hesthaven, J.S., 2018. Deep convolutional neural networks for estimating porous material parameters with ultrasound tomography. The Journal of the Acoustical Society of America, 143(2), pp.1148-1158.
13. Lhuillier P-E et al., 2016. Overview of the finite elements modeling of the ultrasonic propagation in complex media at EDF R&D. In Proc. 19th World Conf. on Non-Destructive Testing, Munich, Germany, 13–17 June 2016.
14. Liang, L., Liu, M., Martin, C. and Sun, W., 2018. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. Journal of The Royal Society Interface, 15(138), p.20170844.
15. Mabssout, M. and Pastor, M., 2003. A Taylor–Galerkin algorithm for shockwave propagation and strain localisation failure of viscoplastic continua. Computer Methods in Applied Mechanics and Engineering, 192(7-8), pp.955-971.

16. Mosser, L., Dubrule, O. and Blunt, M.J., 2020. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, 52(1), pp.53-79.
17. Nakahata K, Mizokami N., 2016. Image-based finite element simulation of ultrasonic wave in polycrystalline metal using phase-field modeling. In *Proc. 19th World Conf. on Non-Destructive Testing*, Munich, Germany, 13–17 June 2016.
18. Nie, Z., Jiang, H. and Kara, L.B., 2020. Stress field prediction in cantilevered structures using convolutional neural networks. *Journal of Computing and Information Science in Engineering*, 20(1). (2019).
19. Reddy, J. N. (2006). *An Introduction to the Finite Element Method* (Third ed.). McGraw-Hill. ISBN 9780071267618.
20. Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
21. Salas, M.D. and Iollo, A., 1996. Entropy jump across an inviscid shockwave. *Theoretical and computational fluid dynamics*, 8(5), pp.365-375.
22. Sbarufatti, C., Manson, G., & Worden, K., 2014. A numerically enhanced machine learning approach to damage diagnosis using a Lamb wave sensing network. *Journal of Sound and Vibration*, 333(19), 4499–4525. <http://doi.org/10.1016/j.jsv.2014.04.059>.
23. Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), p.60.
24. Sorteberg, W.E., Garasto, S., Pouplin, A.S., Cantwell, C.D. and Bharath, A.A., 2018. Approximating the solution to wave propagation using deep neural networks. *arXiv preprint arXiv:1812.01609*.
25. Takahashi, T., Kondo, T., Saitou, H., Okazaki, T. and Ishikiriya, M., Toyota Motor Corp, 2000. Method and apparatus for determining gasoline characteristics by using ultrasonic wave. U.S. Patent 6,032,516.
26. Van Pamel A, Brett CR, Huthwaite P, Lowe MJS., 2015. Finite element modelling of elastic wave scattering within a polycrystalline material in two and three dimensions. *J. Acoust. Soc. Am.* 138, 2326–2336. (doi:10.1121/1.4931445).
27. Van Pamel A, Sha G, Rokhlin SI, Lowe MJS., 2017. Finite-element modelling of elastic wave propagation and scattering within heterogeneous media. *Proc. R. Soc. A*.
28. Villarreal, F., 2006. Heaviside generalised functions and shockwaves for a Burger kind equation. *Integral Transforms and Special Functions*, 17(2-3), pp.213-219.
29. Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
30. Zhong, Z., Zheng, L., Kang, G., Li, S. and Yang, Y., 2020. Random Erasing Data Augmentation. In *AAAI* (pp. 13001-13008).
31. Zhu, W., Sheng, Y. and Sun, Y., 2017. Wave-dynamics simulation using deep neural networks.