Seismic Inversion Capability on Resource-constrained Edge Devices

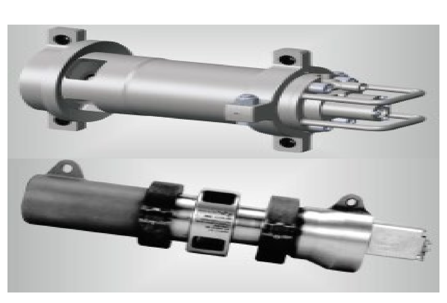
Daniel Manu[1], Petro Mushidi Tshakwanda[1], Lin Youzuo[2], Weiwen Jiang[3], Lei Yang[1]

[1]University of New Mexico, [2]Los Alamos National Laboratory, [3]George Mason University

Introduction, Motivation & Objectives

Introduction

- Seismic FWI presents information about subsurface strata and rock geological properties.
- 2D/3D velocity models is reconstructed from seismic data.
- Seismic sources produce seismic waves.
- Waves are measured using a seismometer.
- Data-driven seismic FWI: Training & Prediction
- Training on GPU, Prediction on Raspberry Pi.

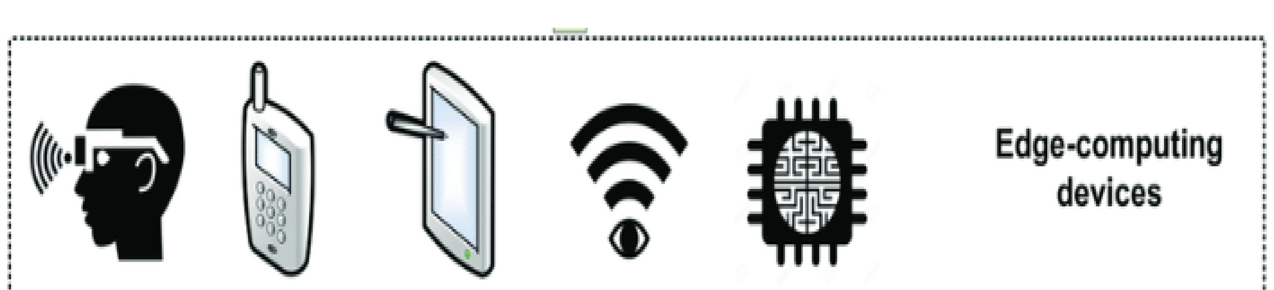


Seismic sources

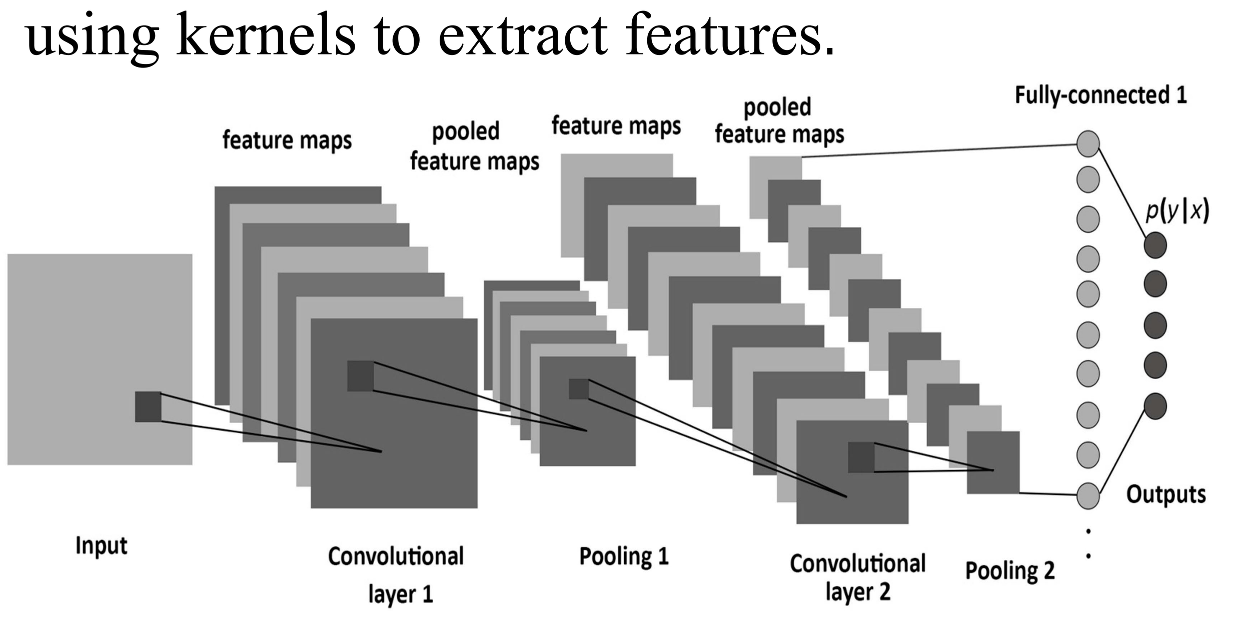
Seismometer

Motivation

- Edge devices can provide real time and on-device computational power for inversion.



- Deep convolutional networks are powerful models that can learn representative features using kernels to extract features.



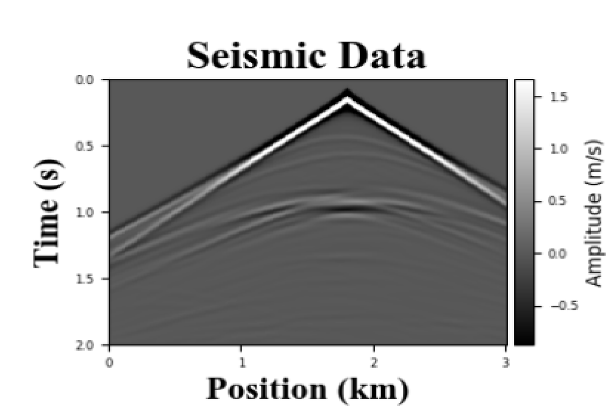
Objectives

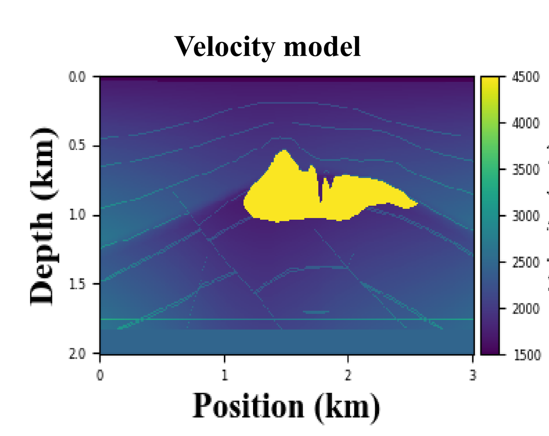
- Design a deep convolutional network (DCN) model based on the UNet architecture.
- Execute the DCN model on the Raspberry Pi to perform inversion with optimum performance.
- Demonstrate robustness of our DCN model to noise by performing inversion on noisy data.
- Design user-friendly interactive graphical user interface (GUI) to automate the inversion process on the Raspberry Pi.

Data and Model Design & DCN Model

Data and Model Design

- 29 sources used to generate seismic data.
- 301 receivers used to record the seismic measurements.
- Each velocity model assumed to have 5–12 layers.
- Velocity values range from 2000 to 4000 m/s.
- Salt body within velocity model has a constant velocity of 4500 m/s.
- Shape of velocity model is $x \times z = 201 \times 301$ grid points.

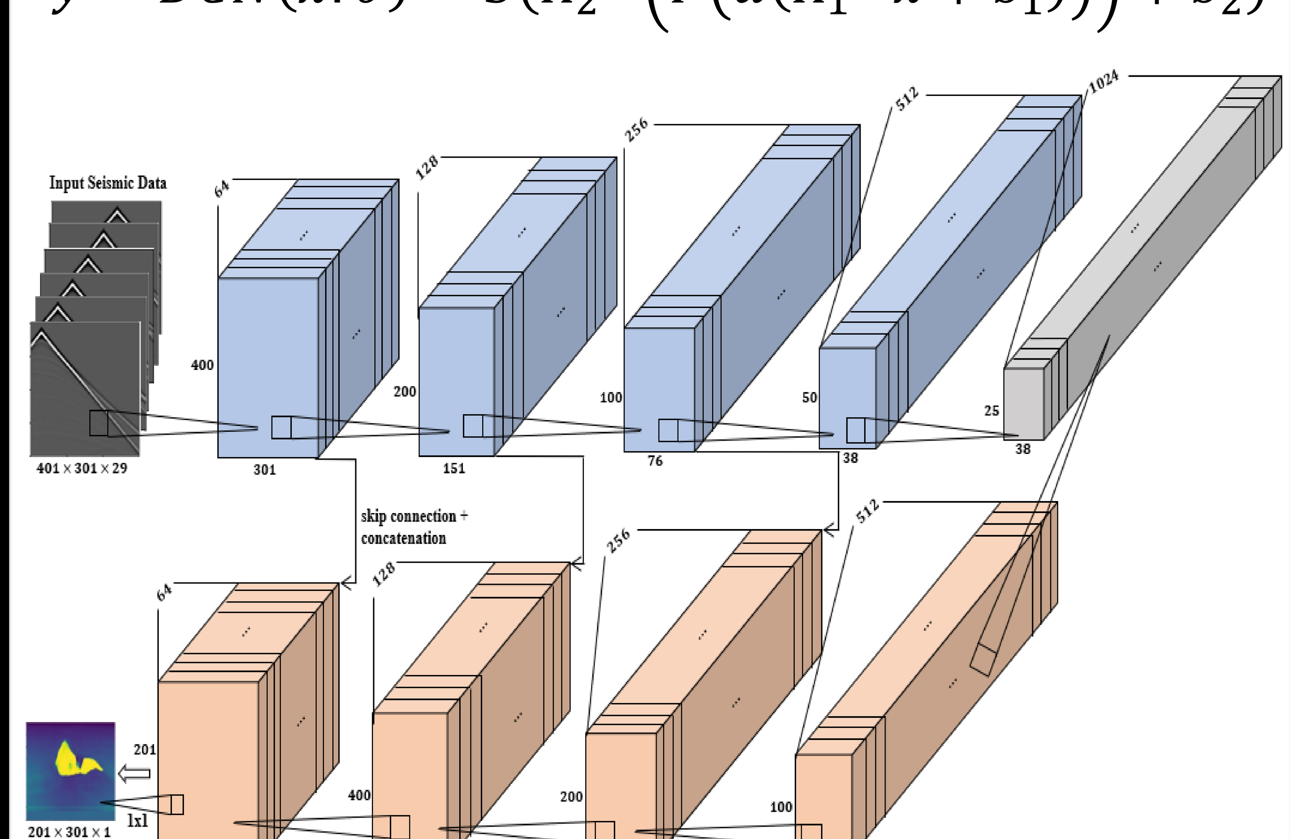




DCN Model

- DCN model implemented using convolution, max pooling and deconvolution layers.
- Convolution layer uses a fixed kernel size of 3×3.
- Channel dimensions of 64, 128, 256, 512 and 1024.
- Convolution layer followed by max pooling and then deconvolution layer.
- Softmax function used to obtain the predicted label.

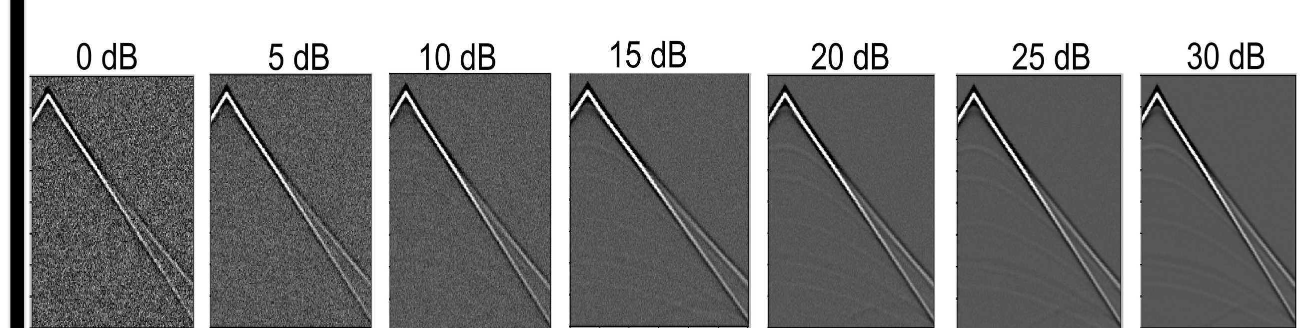
$$y = DCN(x:\theta) = S(K_2 \cdot (P(\alpha(K_1 \cdot x + b_1))) + b_2)$$



Noise Addition, Results & GUI Design

Noise Addition

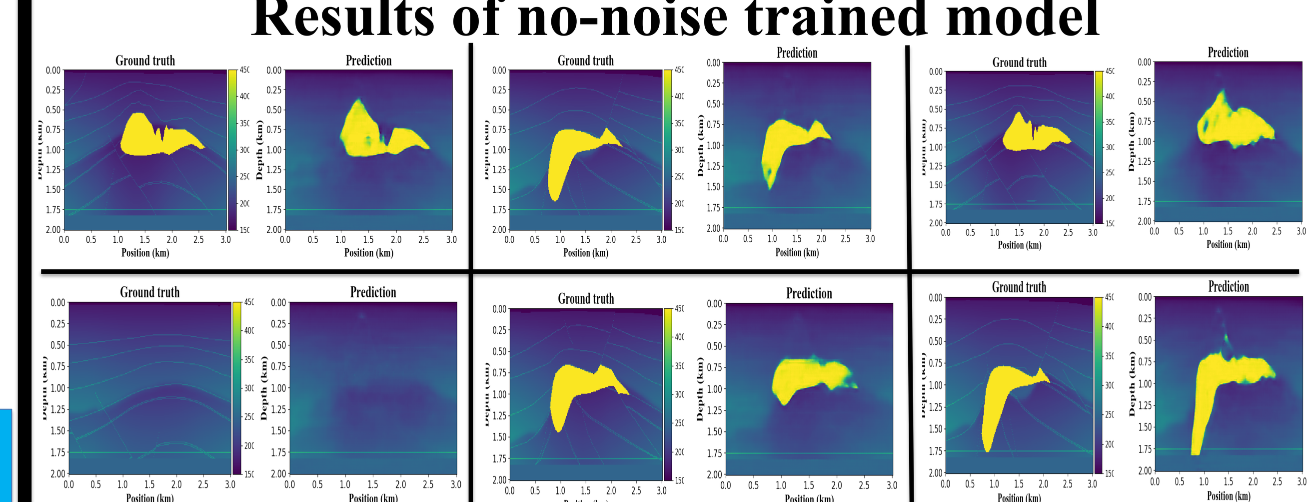
- No-noise model training and Noise-aware testing – 120 training data
- Noise-aware model training and testing – 960 training data (clean + 0, 5, 10, 15, 20, 25, 30 dB)



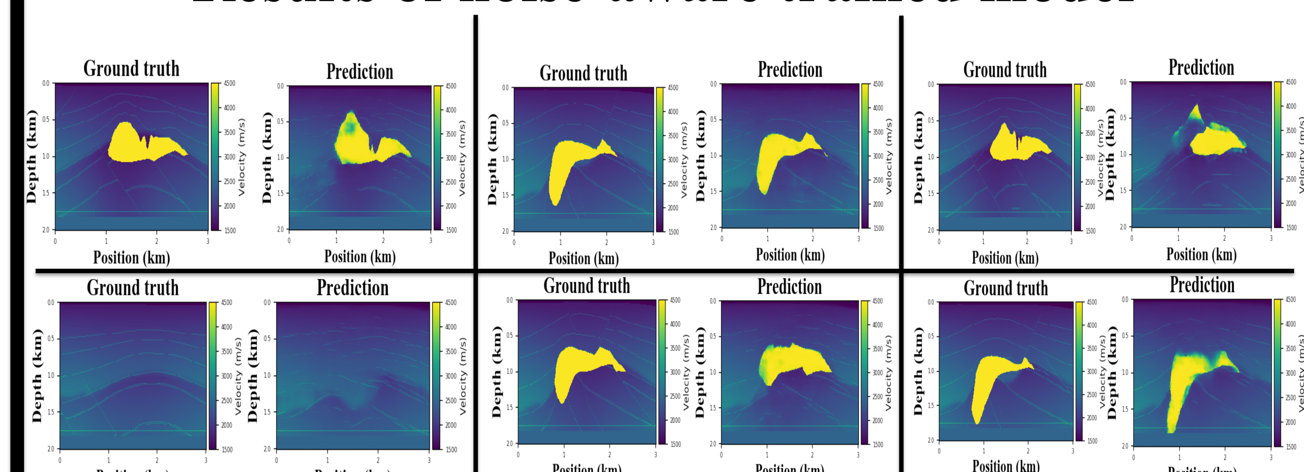
Results

- Prediction time on Raspberry Pi is 25 seconds.
- Inference performance in terms of PSNR & SSIM

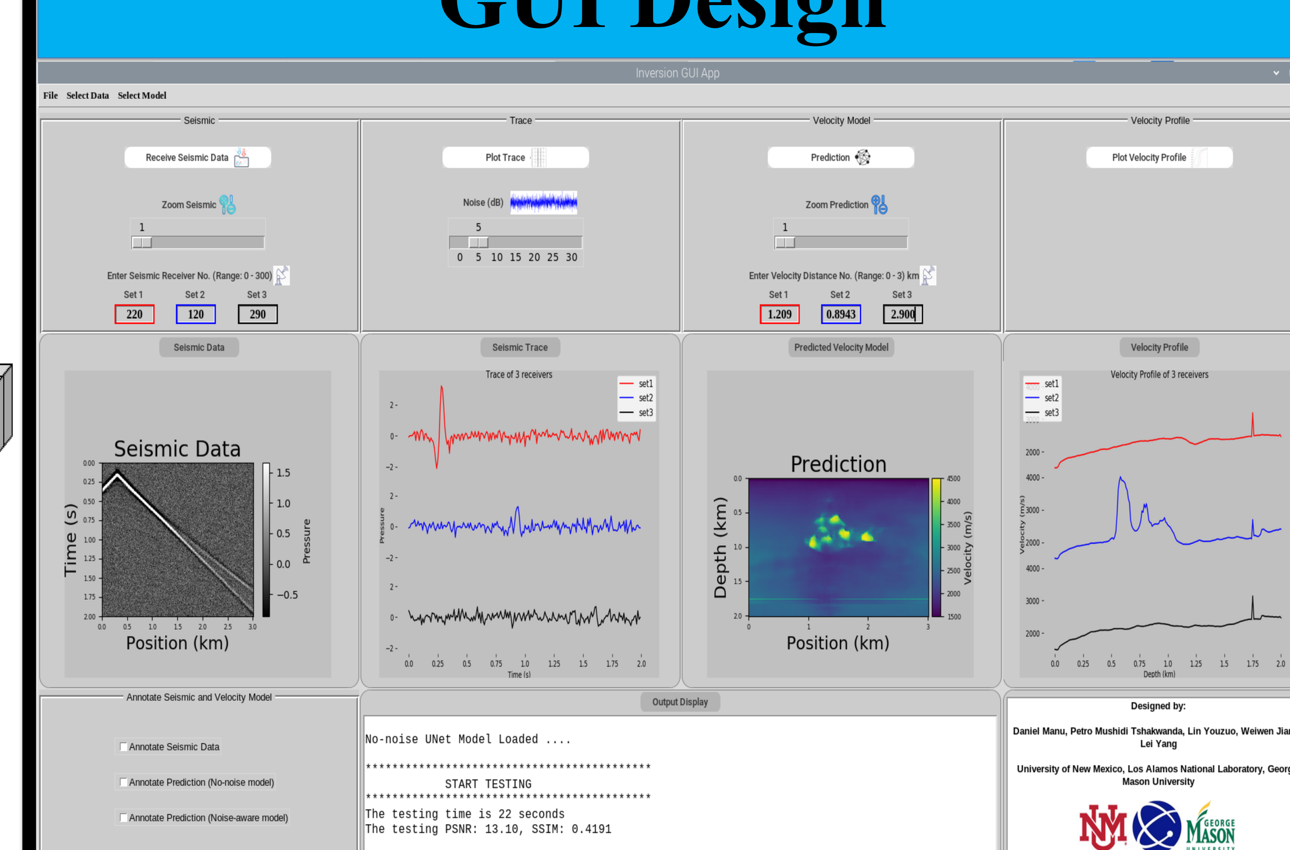
Results of no-noise trained model



Results of noise-aware trained model



GUI Design



Conclusion

- Seismic inversion successfully performed on rasp pi.
- Our models show robustness in presence of noise.