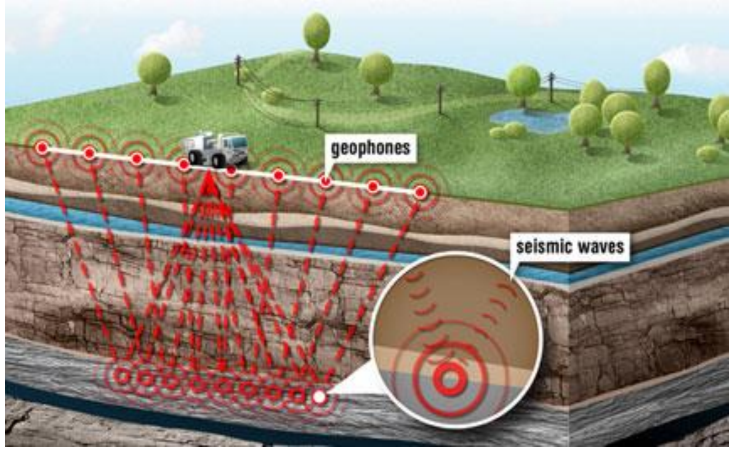
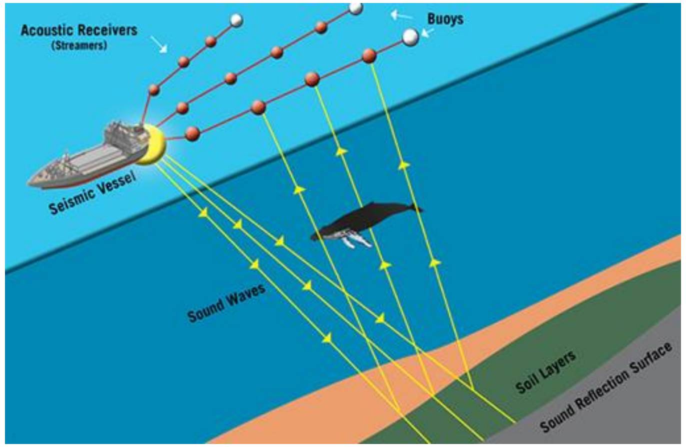
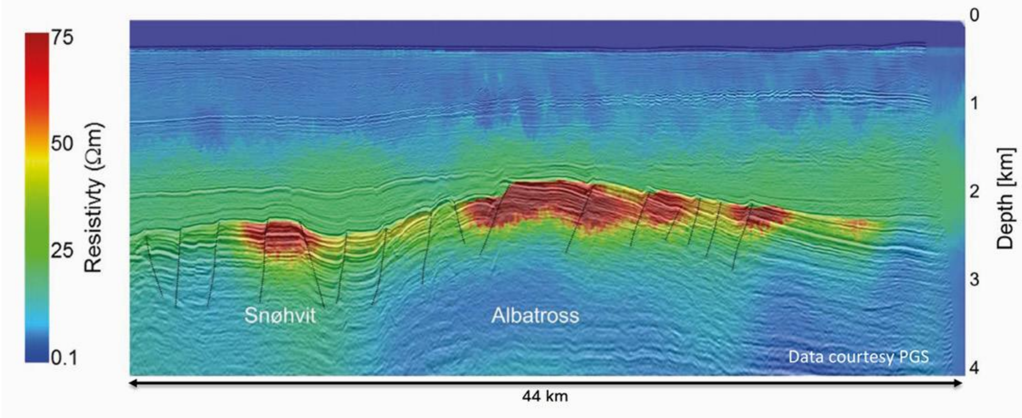
Seismic Facies Classification with Wavelets and Deep Learning

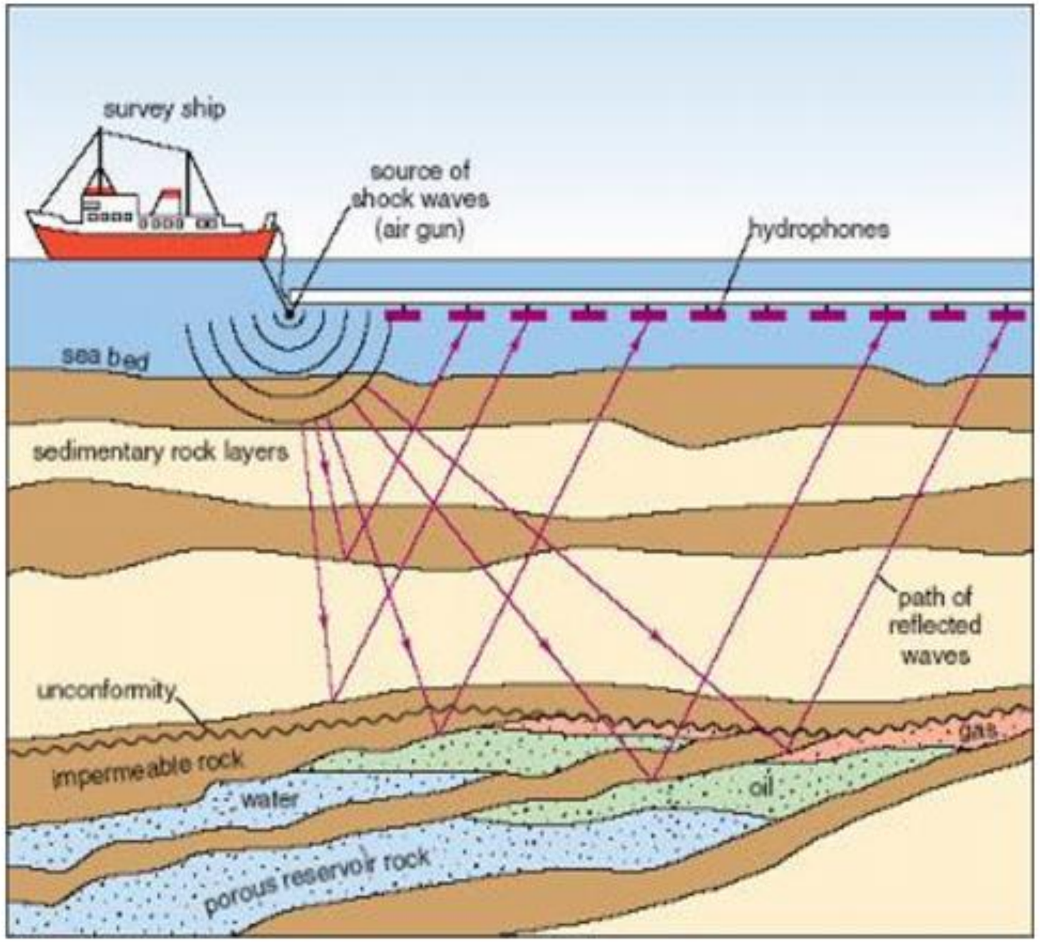






Akhilesh Mishra
Senior Application Engineer
MathWorks, Inc

Seismic data remote sensing



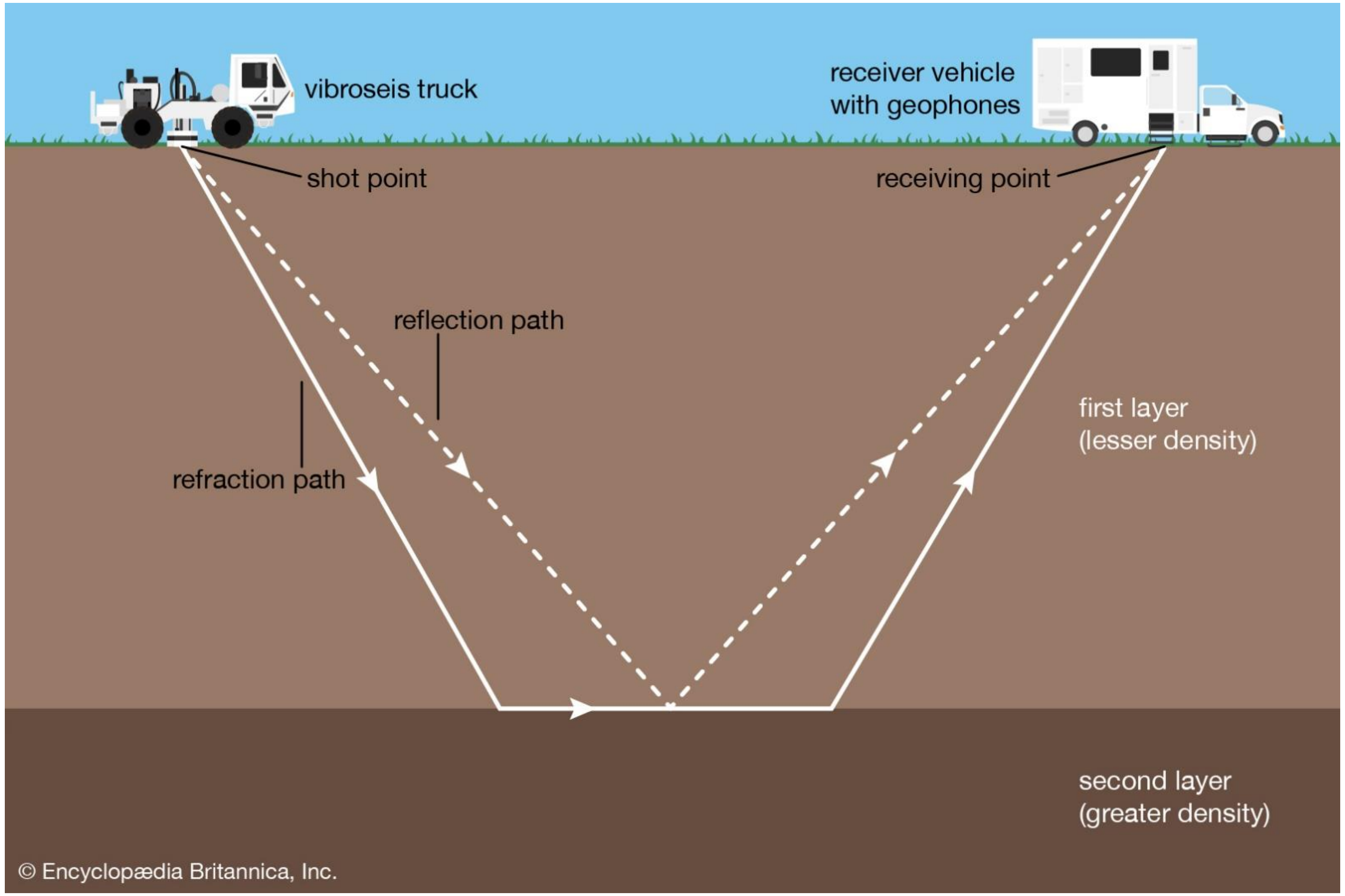


Image source: geologylearn.blogspot.com

Seismic data remote sensing

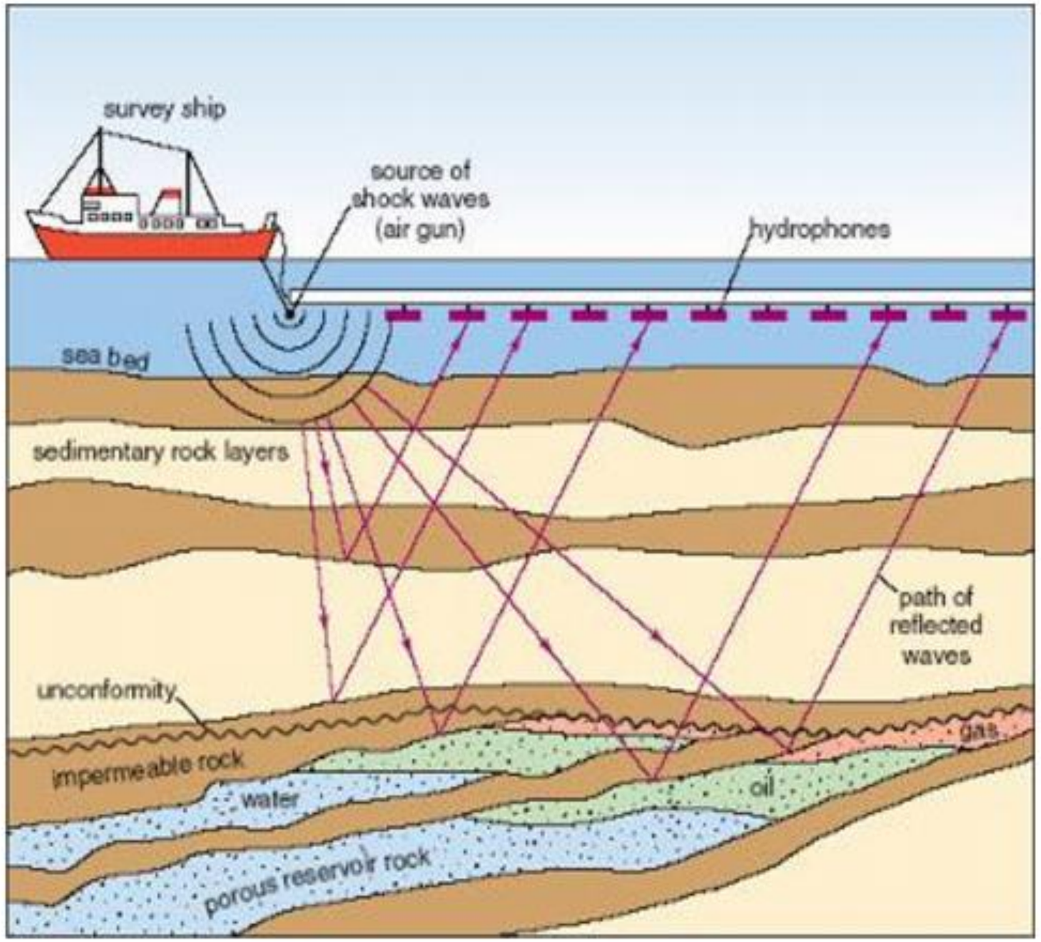


Image source: geologylearn.blogspot.com

- Subsurface reflection proportional to impedance contrast of the layers
- Quantitative interpretation allows determination of reservoir characteristics and reservoir types

Seismic signal processing – quite cumbersome

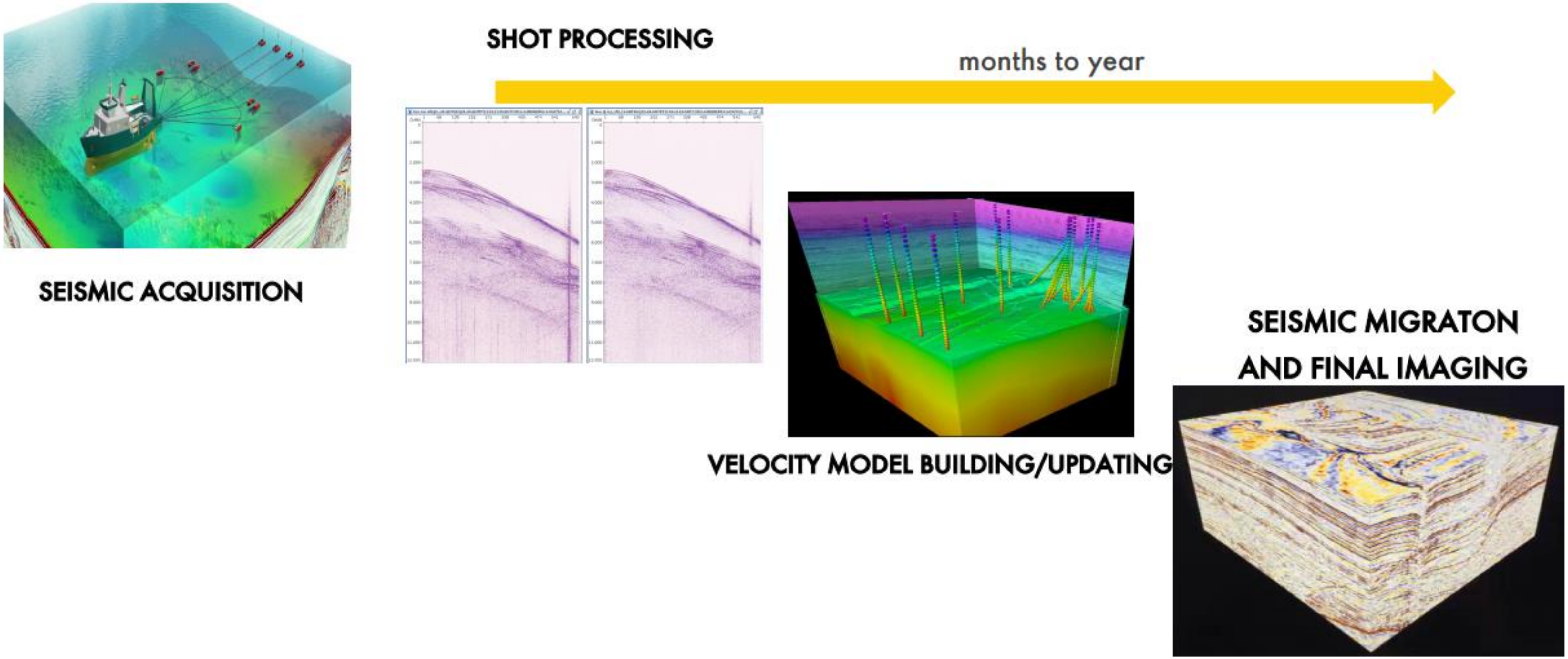


Image source : Shell International B.V.

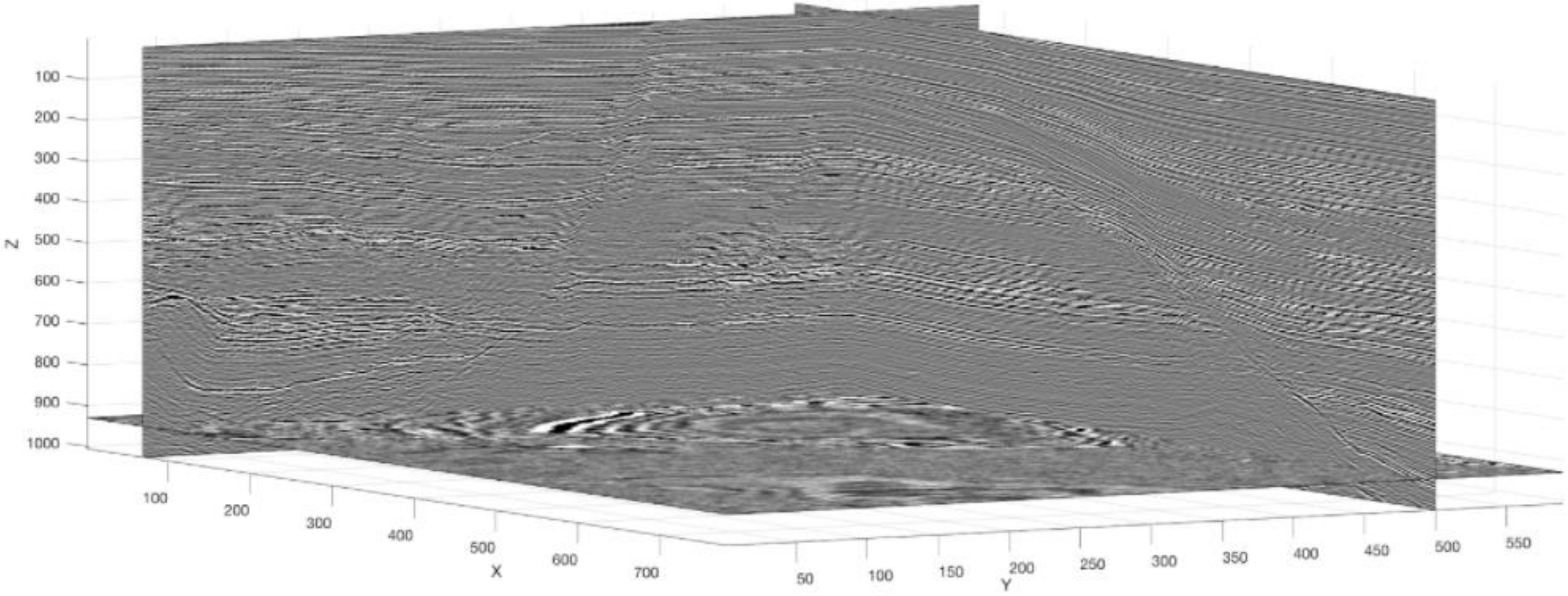
Seismic interpretation

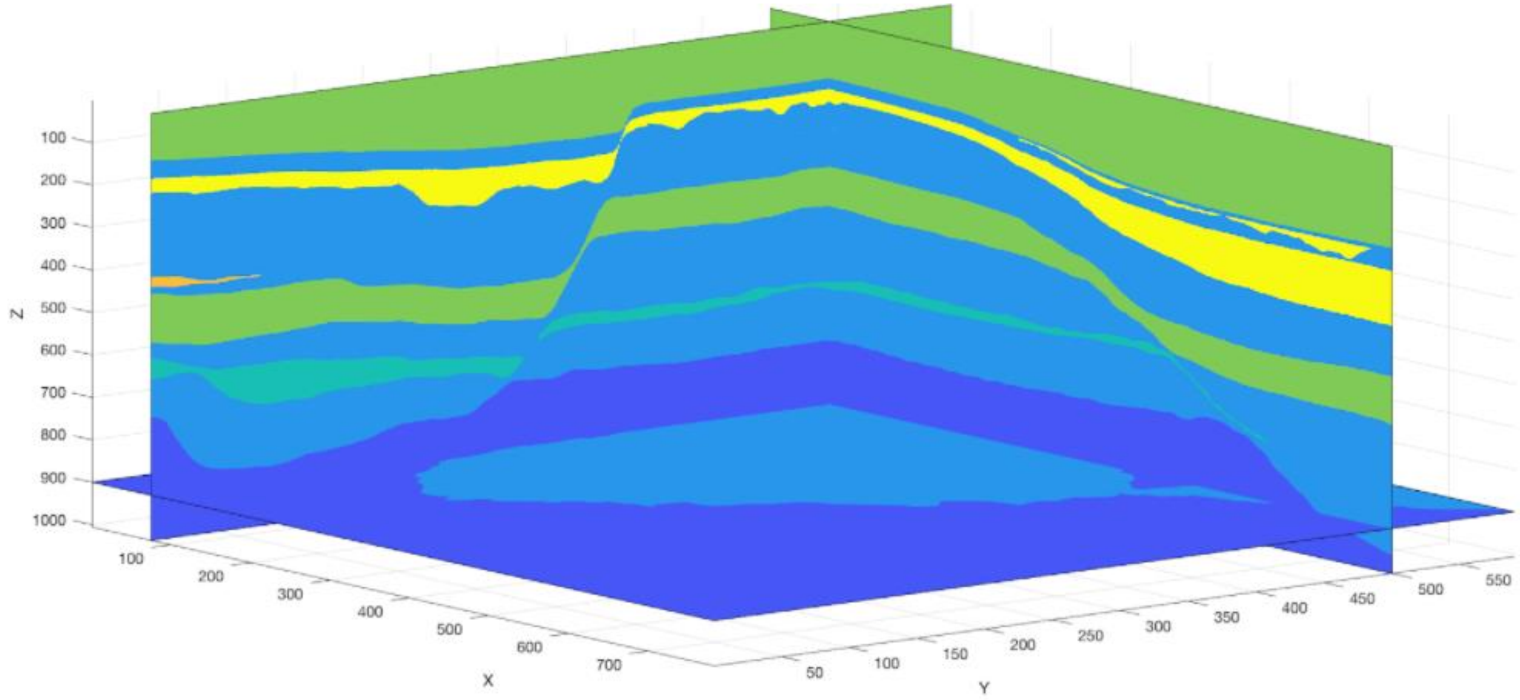
Helps identify subsurface features

Examples of features :

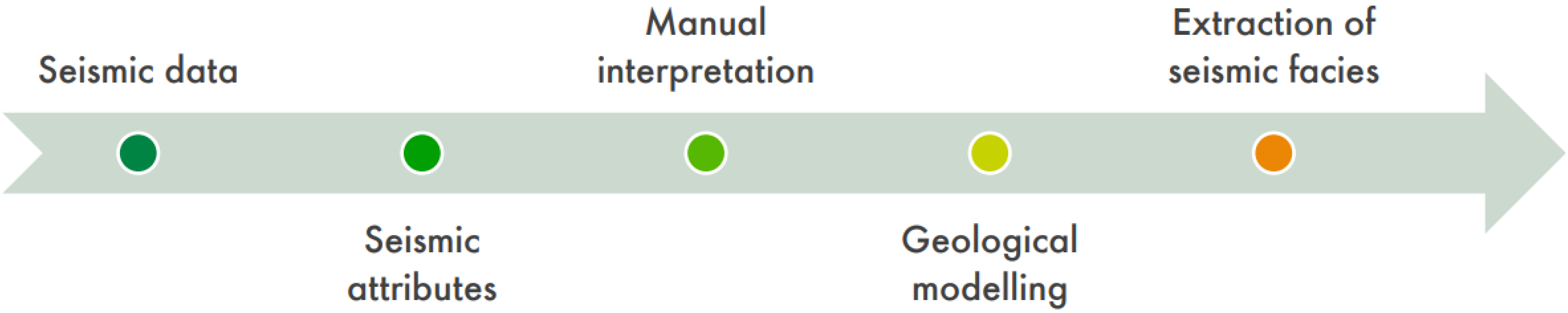
- A mix of sand, silt, and mud deposited in a fan-shaped delta at the mouth of a river (deltaic environment and facies)
- Coarse sandy sediments deposited in a meandering river channel (fluvial environment and facies)
- Extremely fine-grained sediments deposited in a shallow lakebed (lacustrine environment and facies).

Challenges : Time consuming, Reproducibility, and Interpretative





Seismic interpretation – Challenges



Challenges in traditional seismic facies classification		
Time consuming (~3 to 6 months)	Interpretative in nature (Seismic interpretation is a skillset)	Reproducibility (Results can vary based significantly)

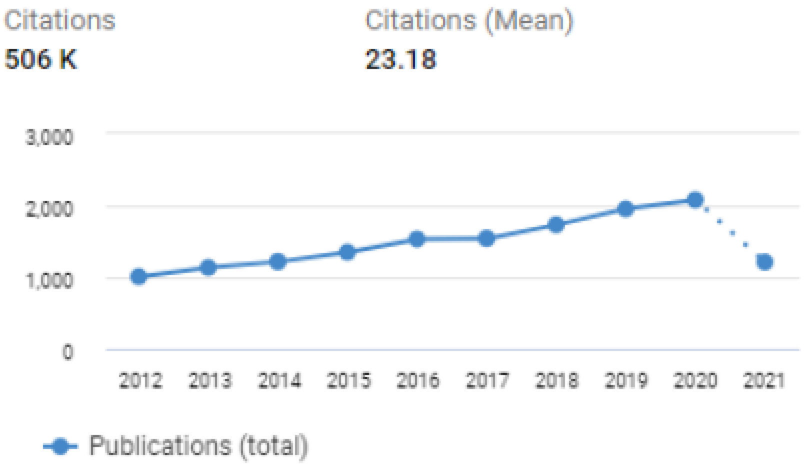
Image source : Shell International B.V.

Artificial Intelligence for seismic interpretation

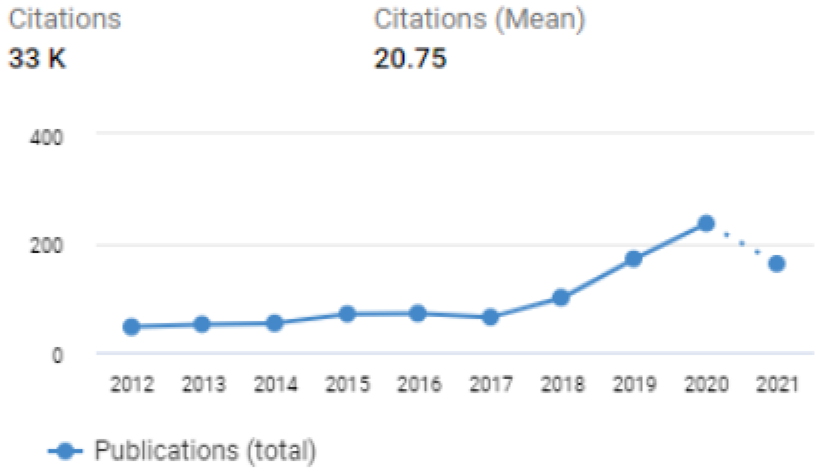
There has been a lot of recent publications on using 2D deep learning models for seismic facies classification

- *Salt classification using deep learning – Waldeland and Solberg (2017)*
- *2D seismic facies classification using state-of-the-art 2D CNN architectures (Dramsch et al., 2018; Zhao, 2018)*

Google Scholar Keyword: *Seismic facies classification*



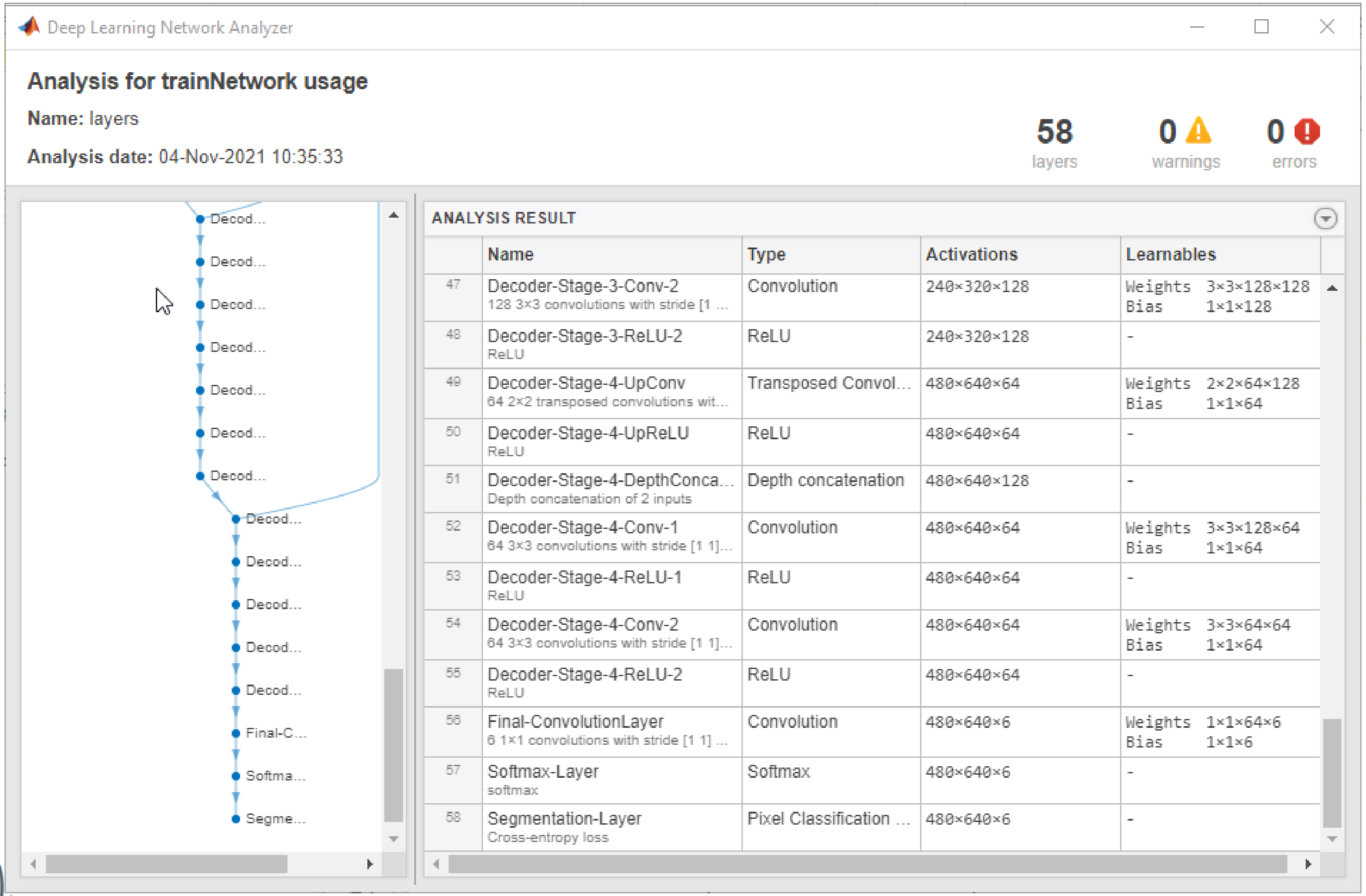
Google Scholar Keyword: *Machine learning seismic facies classification*



Data till June 02, 2021

Most methods include

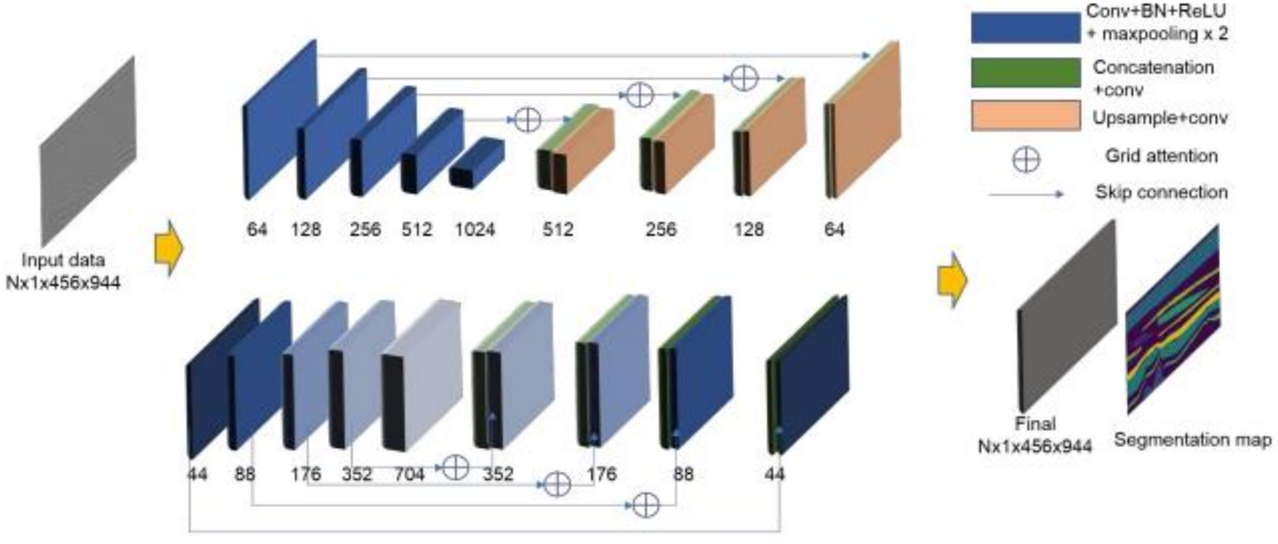
- Semantic segmentation using CNNs
- Use 2D and 3D methods
  - UNet, VGGNet



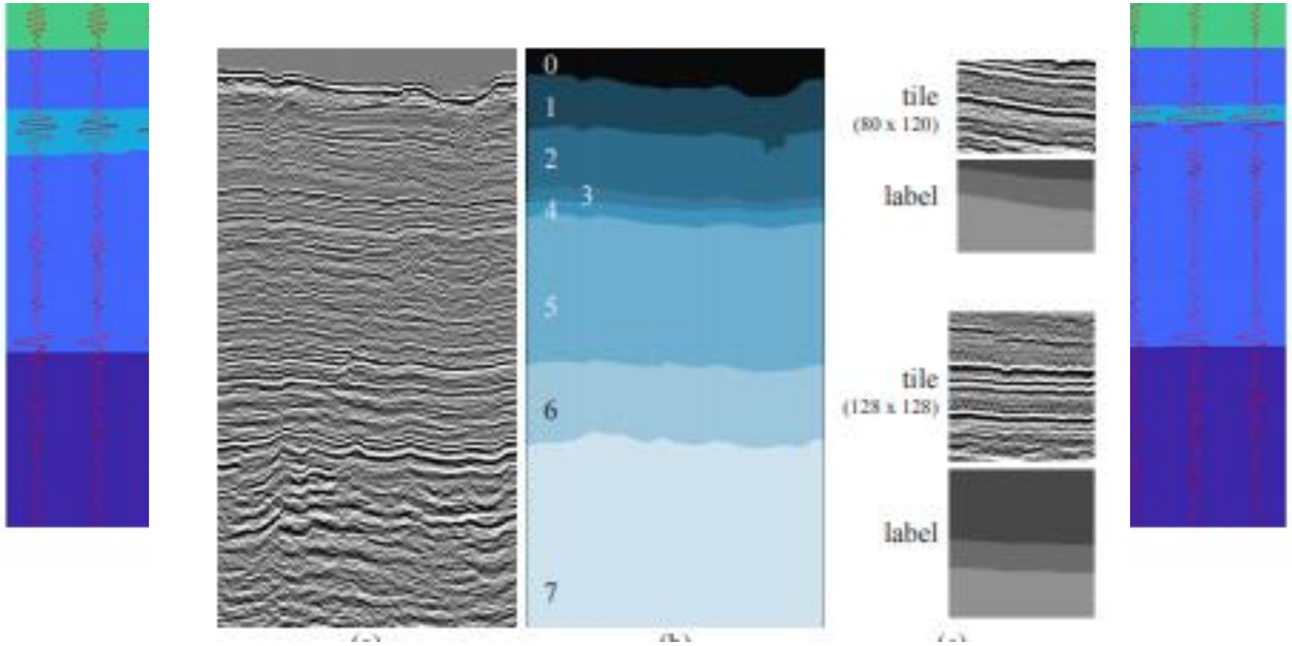
- "3D seismic facies classification using CNN" (Liu et al., 2020).
  - Liu et al., 2020 used VGGNet with 3 convolutional blocks
  - Prediction accuracy for synthetic data validation set = 0.82; F1-score = 0.81
  - Small input patches (32x32x32) and longer compute time

Challenges with Semantic Segmentation

- Accuracy is overall less
- Input image size greatly impacts the prediction results
- Models not data agnostic
- Learned features are all image based, but underlying data is signals



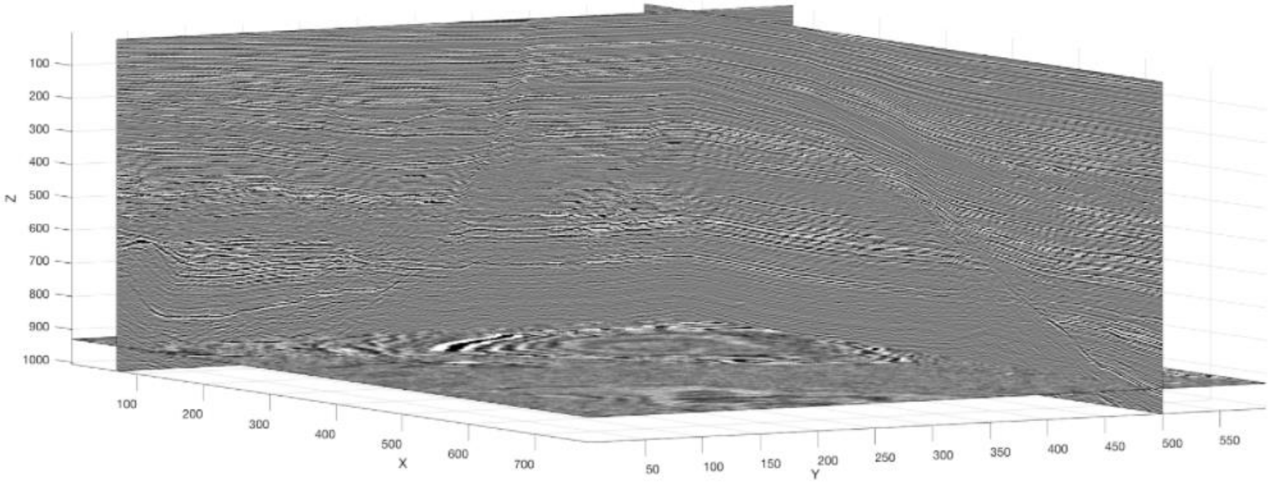
Blue blob in lower brunch indicates dilated convolution + ReLU +Conv + ReLU



Novel approach developed in SEAM AI competition

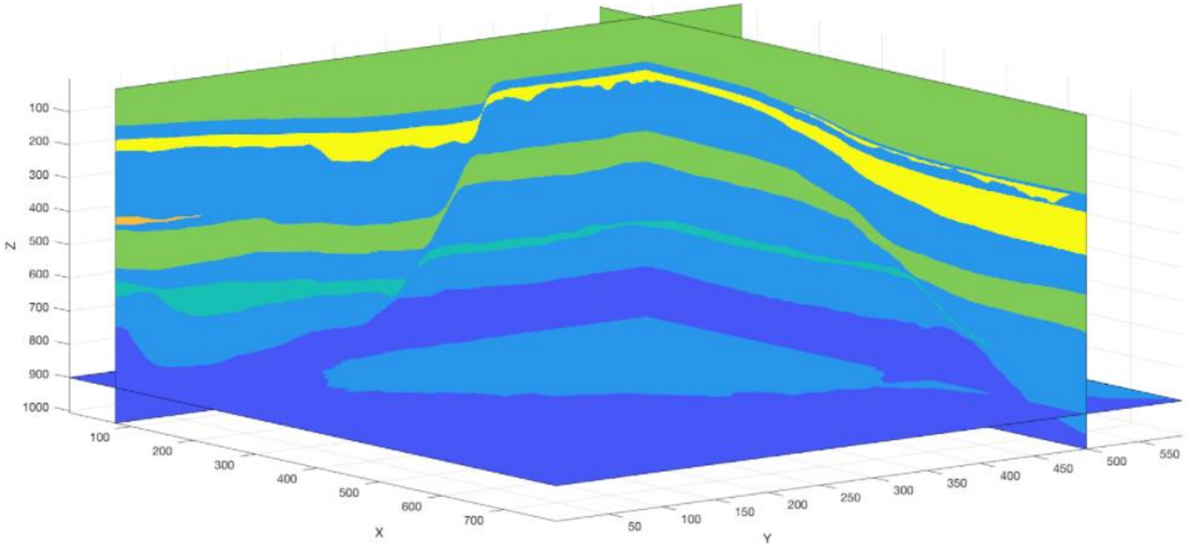
- **Introduction**

SEAM Artificial Intelligence Project presents this data challenge competition in collaboration with AICrowd and Xrathus. This challenge features the Parihaka data set.



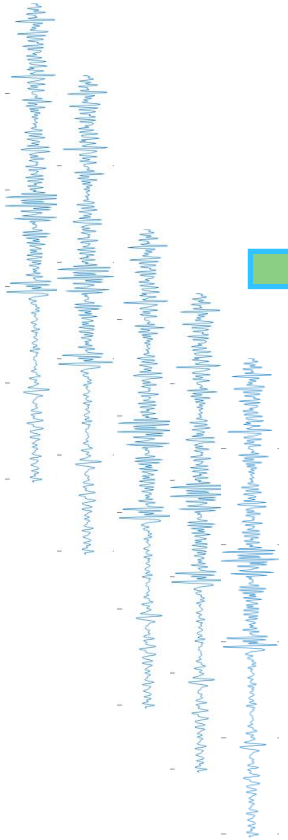
- **Goals**

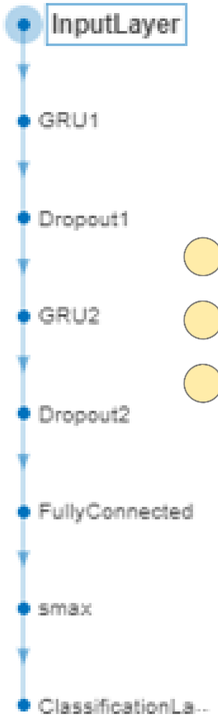
The goal of the SEAM AI Parihaka challenge is to create a machine-learning algorithm which, working from the raw 3D image, can reproduce an expert pixel-by-pixel facies identification.

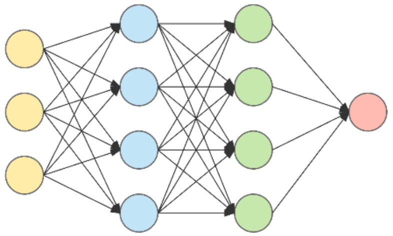


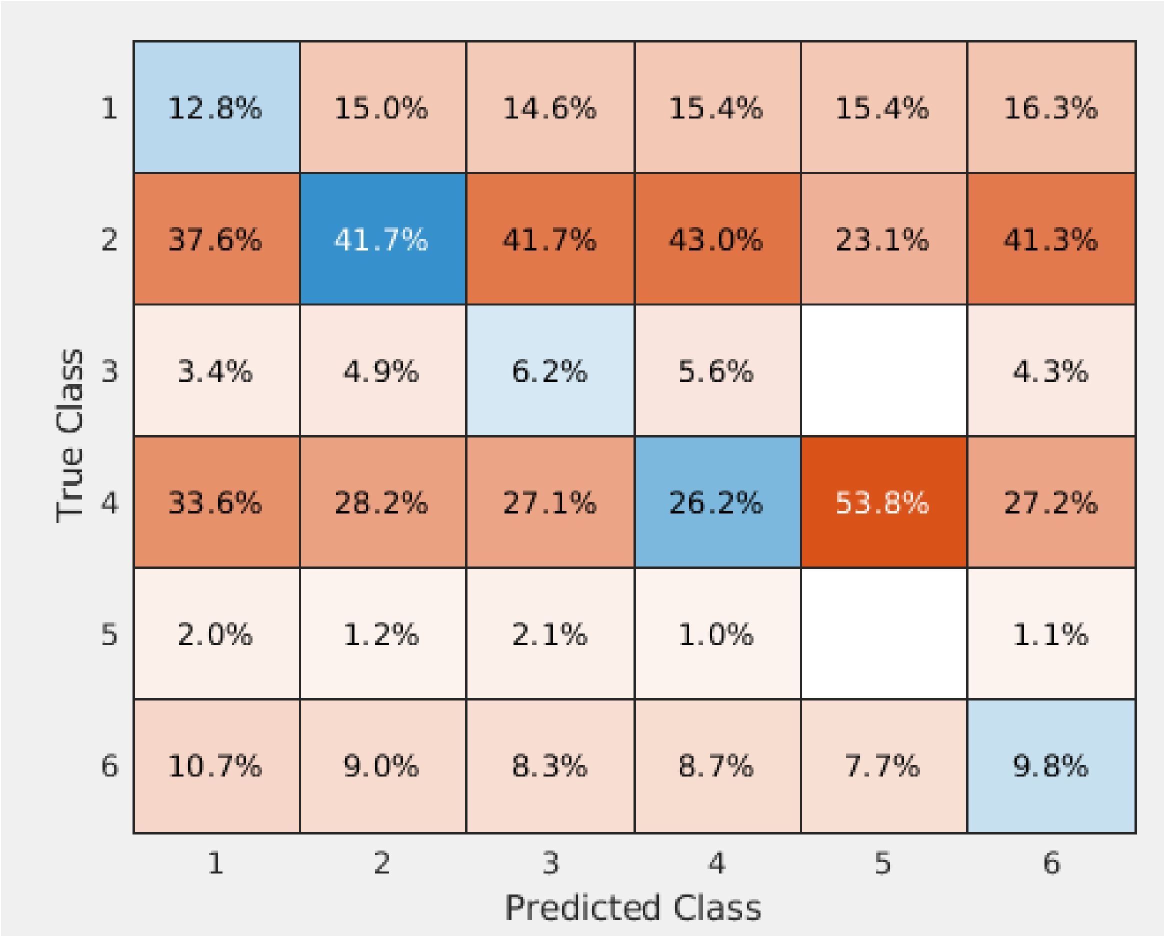
Our solution :
RNN approach with Wavelet pre-processing

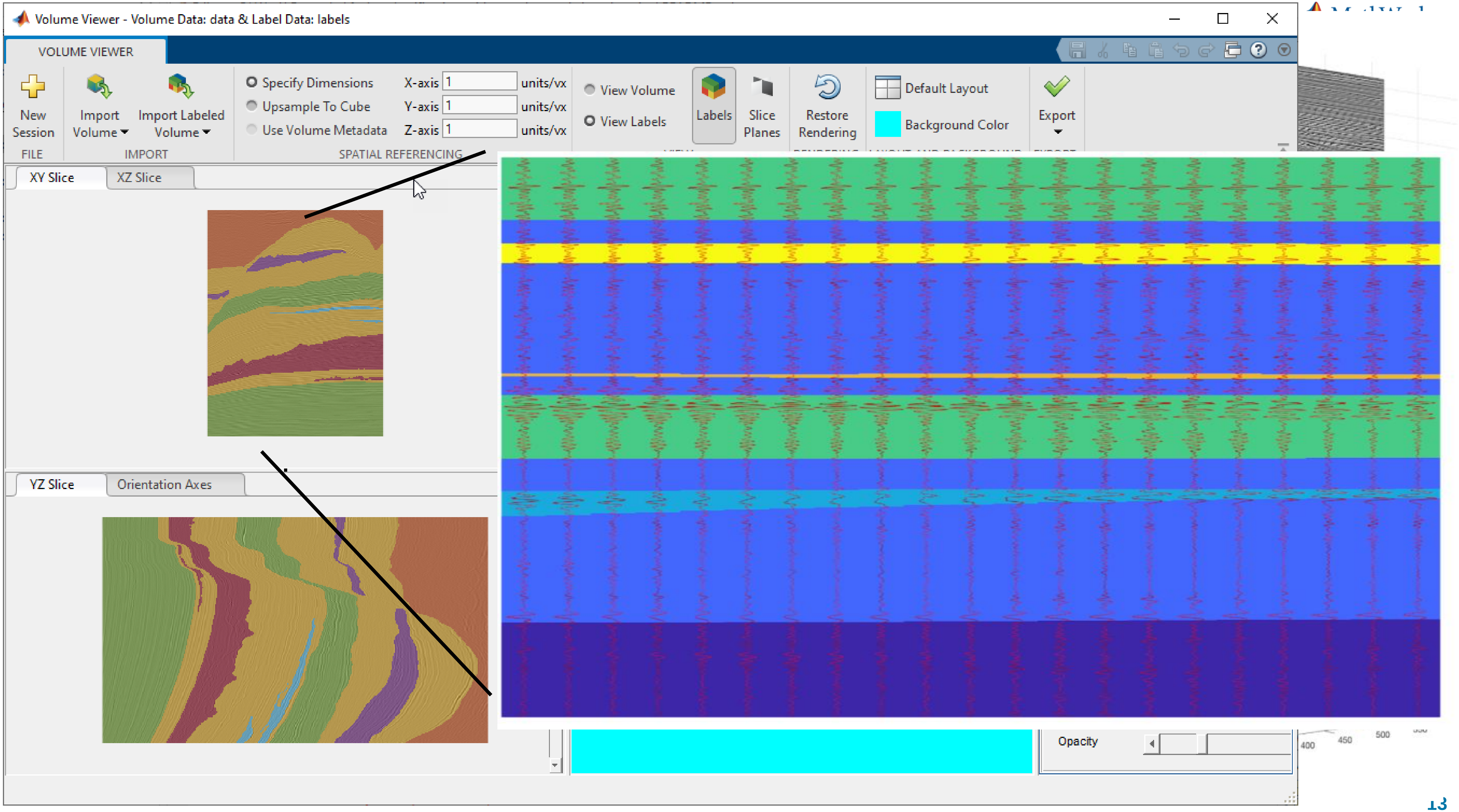
What happens if we train a network with raw data ?





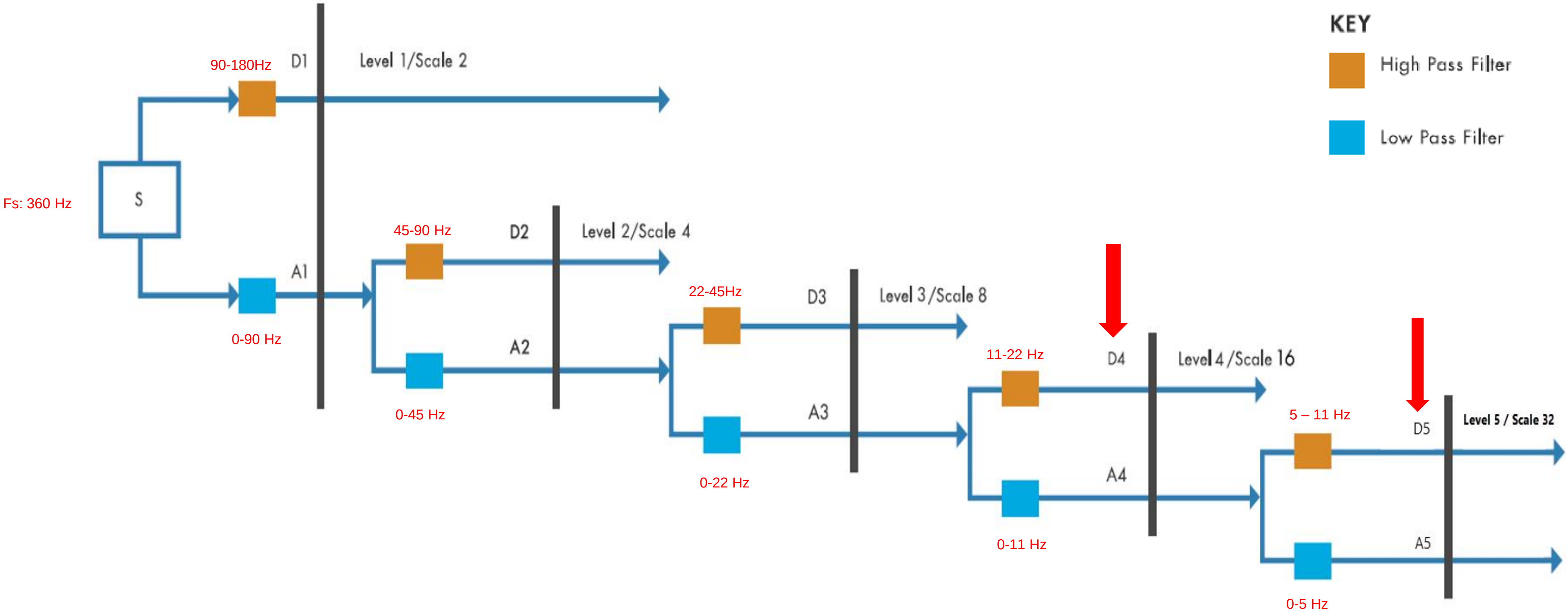





Volume Viewer - Volume Data: data & Label Data: labels
VOLUME VIEWER
New Session
Import Volume
Import Labeled Volume
Specify Dimensions
Upsample To Cube
Use Volume Metadata
X-axis 1 units/vx
Y-axis 1 units/vx
Z-axis 1 units/vx
View Volume
View Labels
Labels
Slice Planes
Restore Rendering
Default Layout
Background Color
Export
FILE
IMPORT
SPATIAL REFERENCING
XY Slice
XZ Slice
YZ Slice
Orientation Axes
Opacity
400
450
500

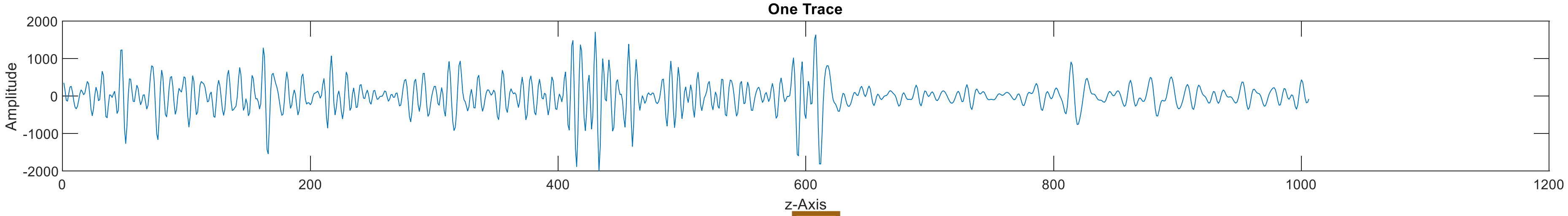
Introduction to Wavelet Multiresolution Analysis

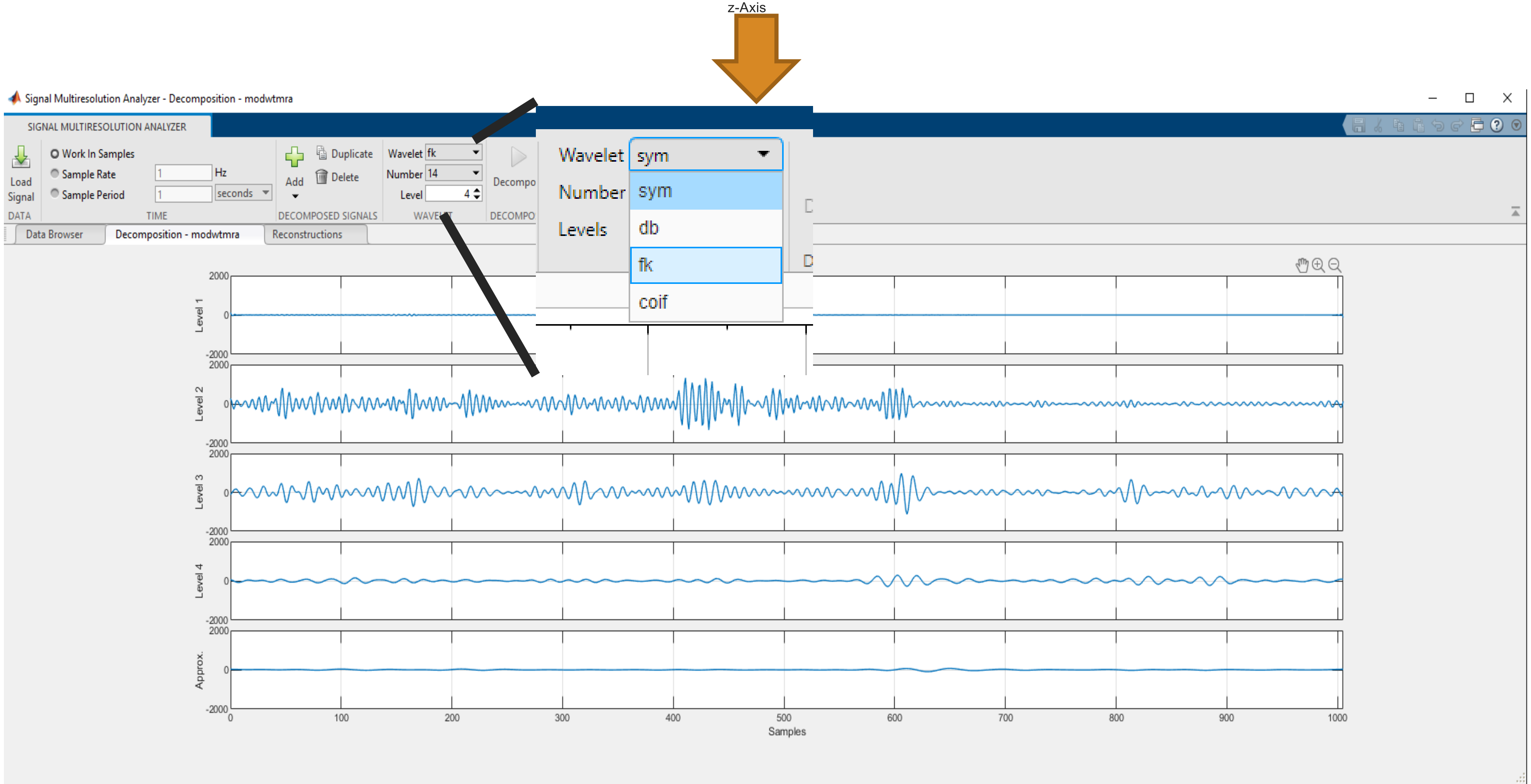
Using DWT (Discrete Wavelet Transform) analyze signals into progressively finer octave bands

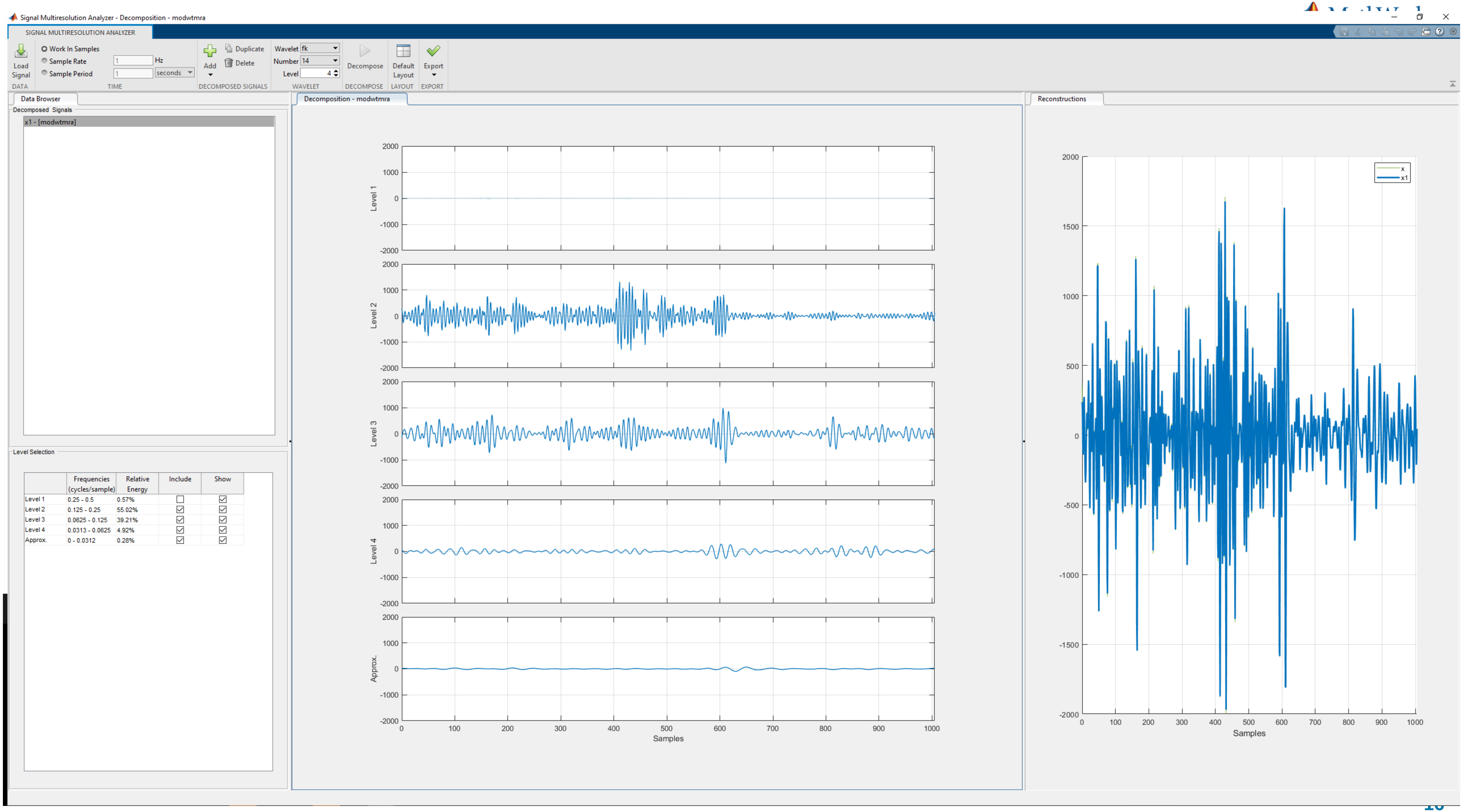


Wavelet MultiResolution Analysis

Wavelet → fk14
Levels → 4



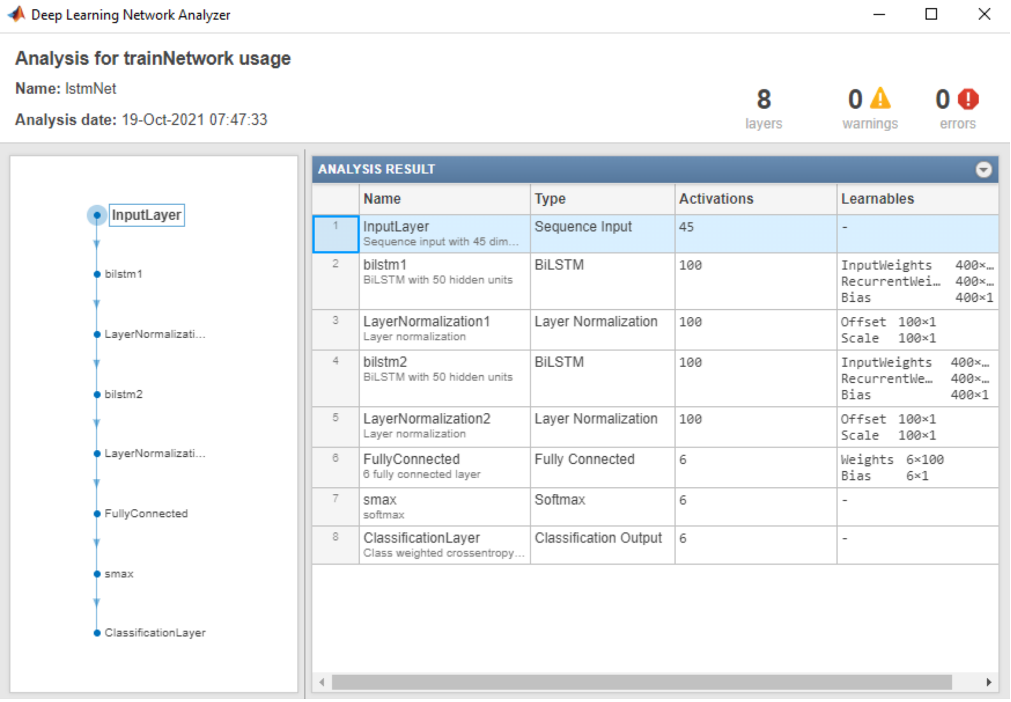



Signal Multiresolution Analyzer - Decomposition - modwtmra
SIGNAL MULTIRESOLUTION ANALYZER
Load Signal
DATA
Work In Samples
Sample Rate
Sample Period
Hz
seconds
TIME
Add
Duplicate
Delete
DECOMPOSED SIGNALS
Wavelet fk
Number 14
Level 4
WAVELET
Decompose
DECOMPOSE
Default Layout
LAYOUT
Export
EXPORT
Data Browser
Decomposed Signals
x1 - [modwtmra]
Level Selection
Frequencies (cycles/sample)
Relative Energy
Include
Show
Level 1 0.25 - 0.5 0.57%
Level 2 0.125 - 0.25 55.02%
Level 3 0.0625 - 0.125 39.21%
Level 4 0.0313 - 0.0625 4.92%
Approx. 0 - 0.0312 0.28%
Decomposition - modwtmra
Level 1
Level 2
Level 3
Level 4
Approx.
Samples
Reconstructions
x
x1
Samples

Recurrent Neural networks

- Started with LSTMs, moved to GRUs instead

- Started with 1 trace at a time, changed it to 3x3 trace to capture spatial correlation

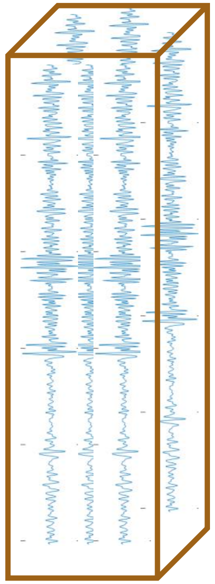


	Name	Type	Activations	Learnables
1	InputLayer Sequence input with 45 dim...	Sequence Input	45	-
2	bilstm1 BiLSTM with 50 hidden units	BiLSTM	100	InputWeights 400×... RecurrentWei... 400×... Bias 400×1
3	LayerNormalization1 Layer normalization	Layer Normalization	100	Offset 100×1 Scale 100×1
4	bilstm2 BiLSTM with 50 hidden units	BiLSTM	100	InputWeights 400×... RecurrentWe... 400×... Bias 400×1
5	LayerNormalization2 Layer normalization	Layer Normalization	100	Offset 100×1 Scale 100×1
6	FullyConnected 6 fully connected layer	Fully Connected	6	Weights 6×100 Bias 6×1
7	smax softmax	Softmax	6	-
8	ClassificationLayer Class weighted crossentropy...	Classification Output	6	-

3

3

1006



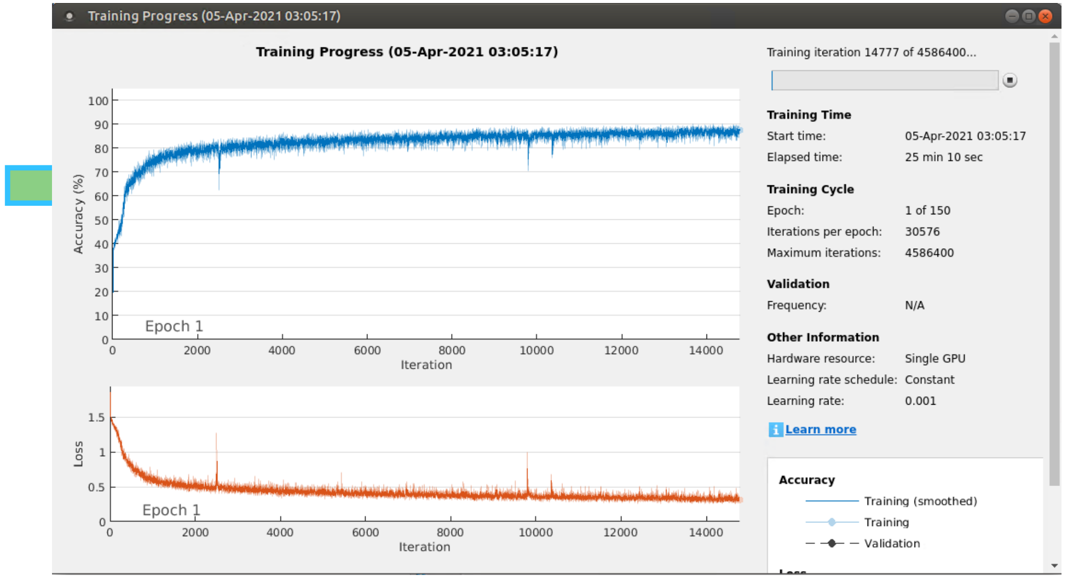
Wavelet MRA

Entire training data →

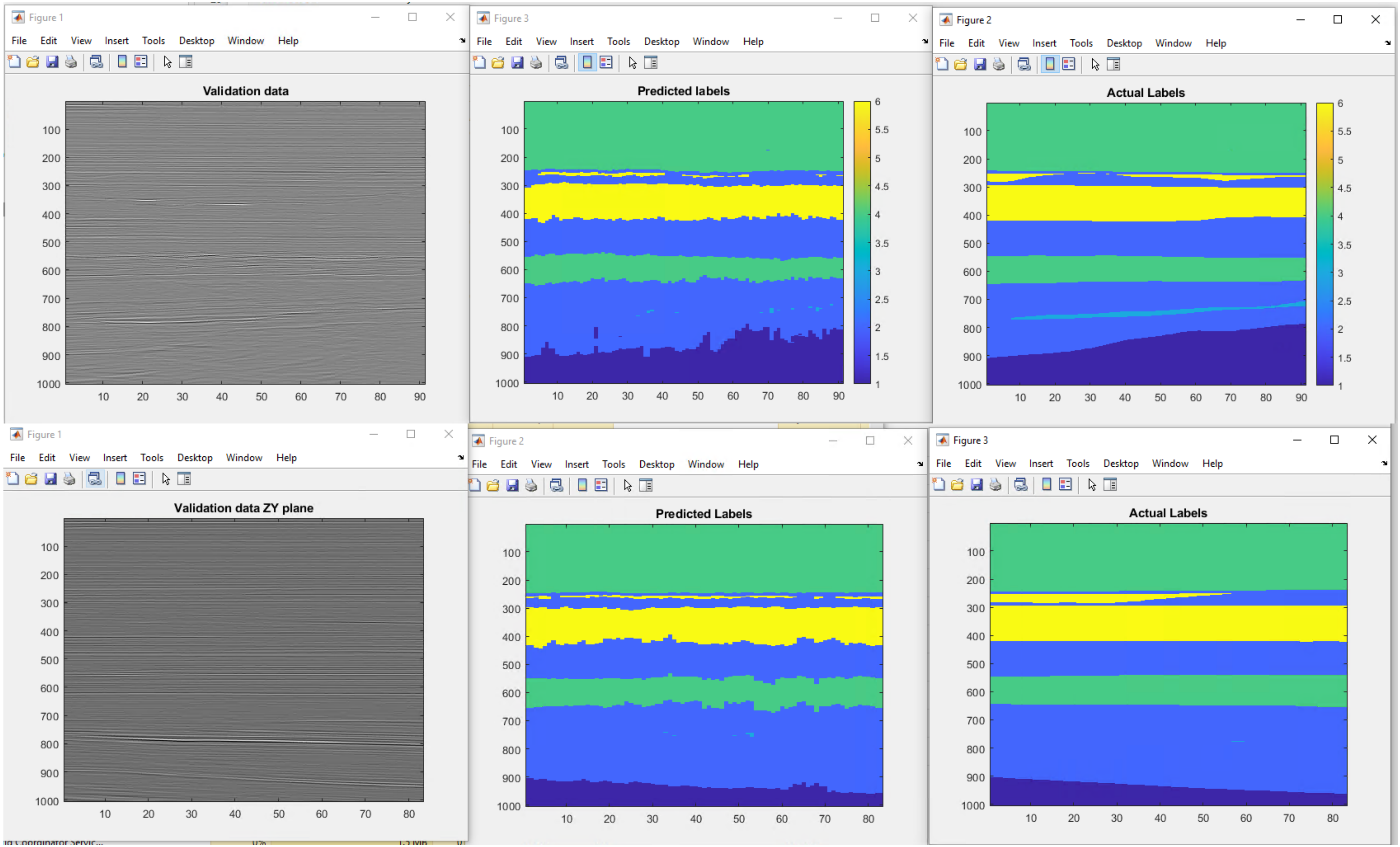
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
(1006 x 3 x 3 x 5)
…
…
…
(1006 x 3 x 3 x 5)

782 x590

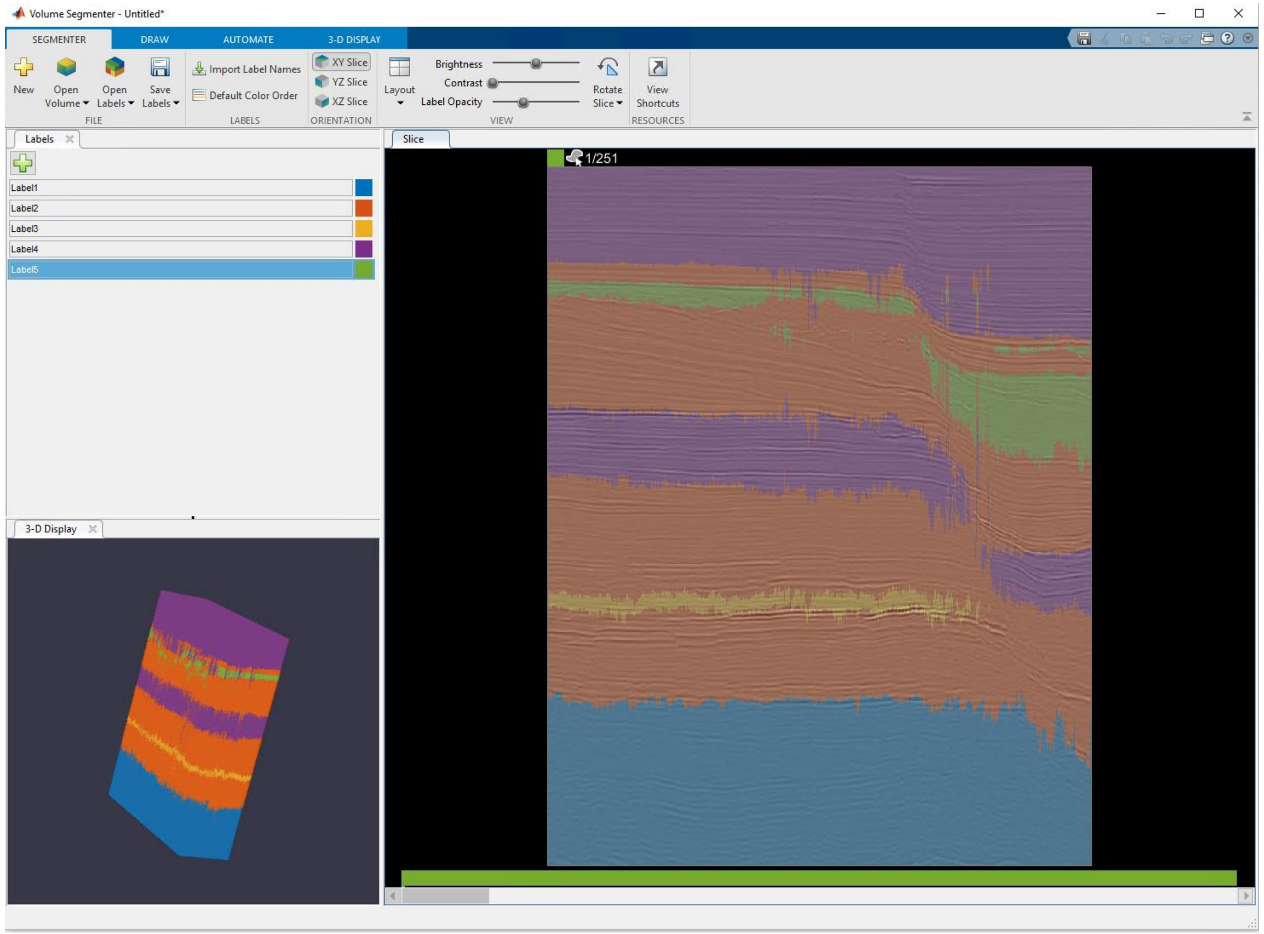
~200 GB data



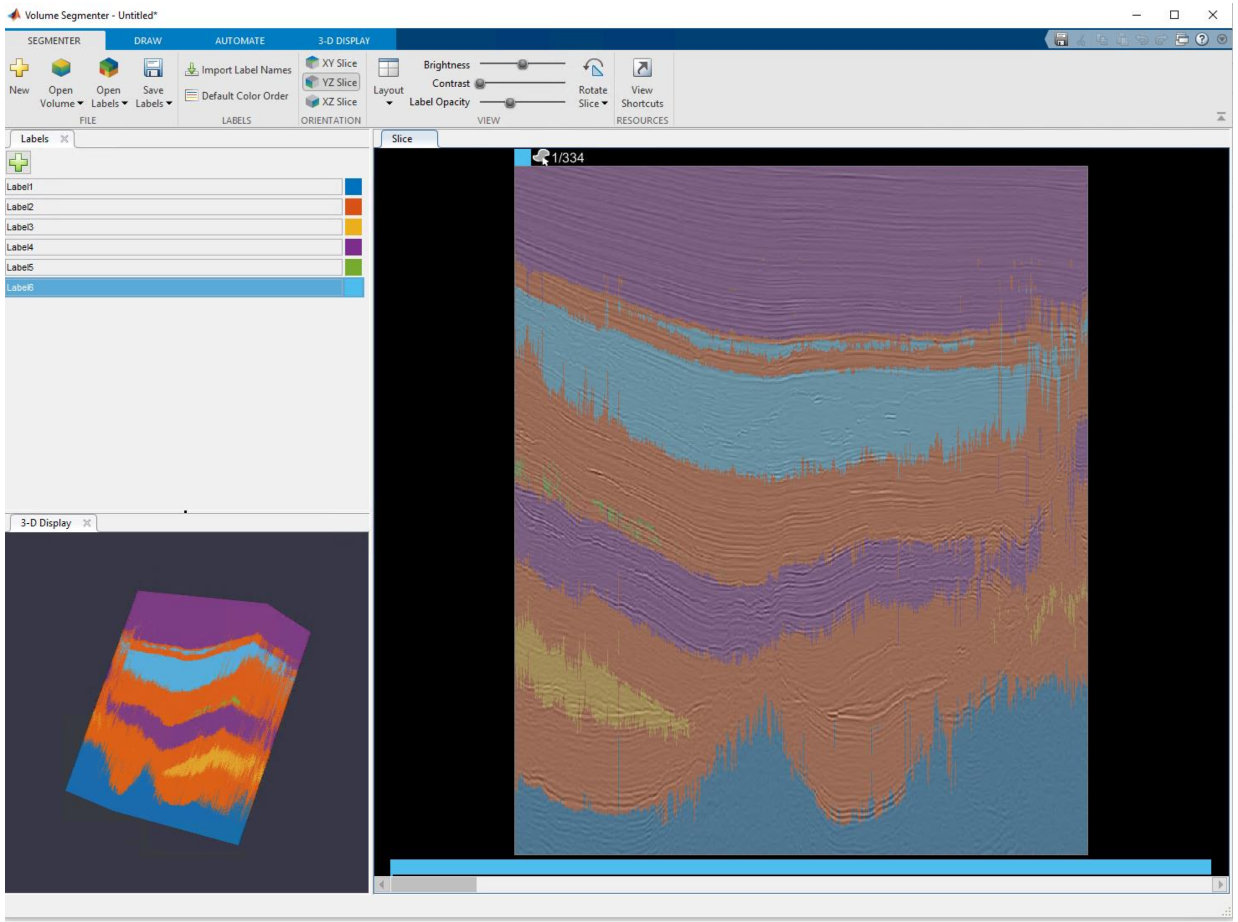
RNN Results on Validation Data



RNN Results on Test Data



Data _test_1 predicted labels



Data _test_2 predicted labels

FINAL SOLUTION

Results:

- Accuracy: how much accuracy did you achieve?
  - Overall 93% on Validation data set

- Performance Numbers: With NVIDIA Volta GPU
  - ~3 Hours

- Prediction time using GPU :
  - RNN : 2-3 mins for ~1000 traces

Volume Viewer - Volume Data: data & Label Data: labels

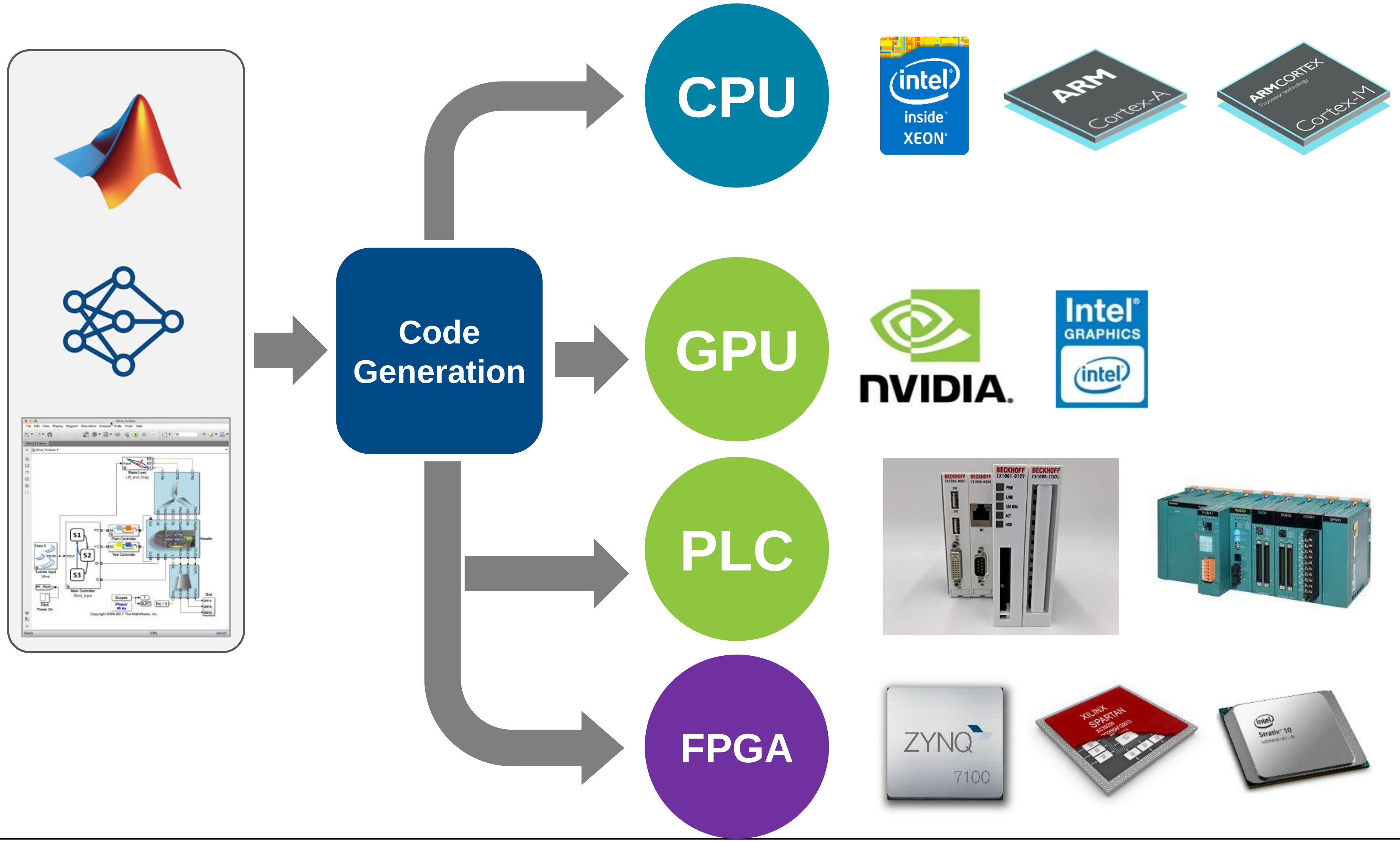
VOLUME VIEWER

Facies Challenge Leaderboard – Test 1

Test 1	F1-Equal	%-Equal	**F1-Weighted**	%-Weighted	F1-Interface	%-Interface
MathWorks	0.796	0.770	**0.763**	0.700	0.697	0.676
	0.656	0.817	**0.395**	0.826	0.602	0.743
	0.607	0.755	**0.358**	0.760	0.520	0.670
	0.587	0.723	**0.335**	0.728	0.466	0.601
	0.481	0.708	**0.294**	0.924	0.417	0.606
	0.480	0.633	**0.192**	0.633	0.407	0.558

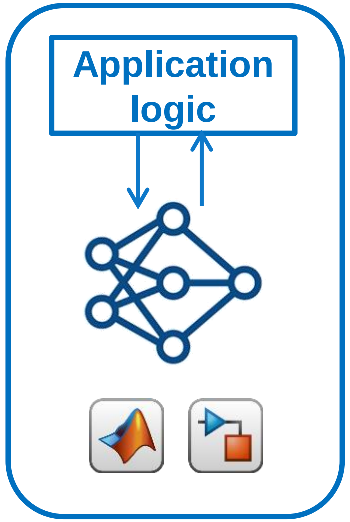
Opacity

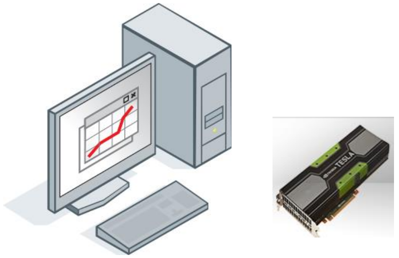
Deploy to Any Processor with Best-in-class Performance



All models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop

Multi-Platform Deployment





Desktop

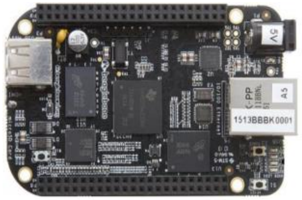
Data Center

FPGA

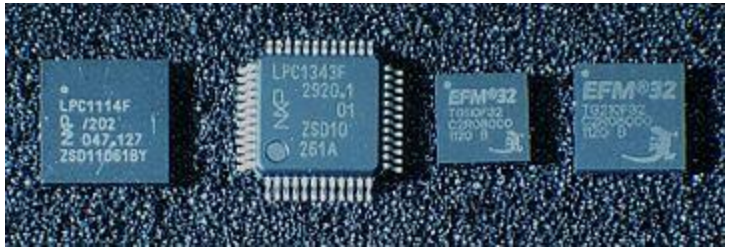
NVIDIA Jetson

Raspberry pi

Mobile



Beaglebone

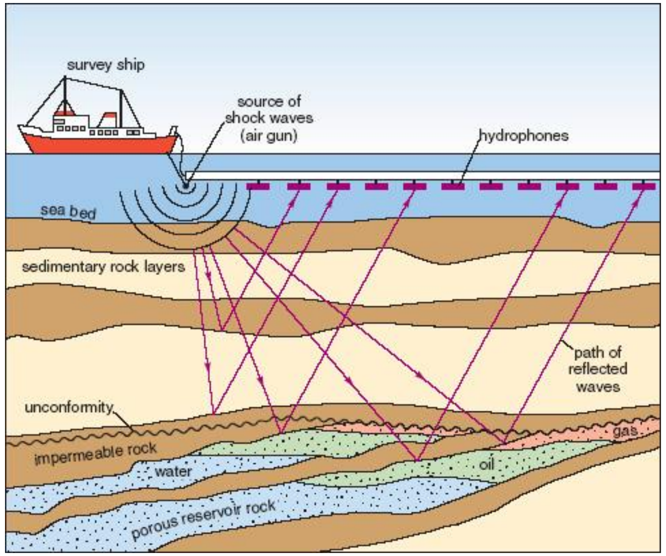


ARM Microcontrollers

. . .

Function for deployment

```
function outLabels = RNNClassificationTestingCodegen(data)

% Convert data to single and initialize dataMRA
% data = single(data);
dataMRA = zeros([size(data),5], 'single');
outLabels = categorical(randi(6,size(data)));

% X and Y dimensions
dim1 = size(data,2);
dim2 = size(data,3);

% Extract MRA from each trace
for ii = 1 : dim1
    for jj = 1:dim2
        dataMRA(:,ii,jj,:) = modwt(data(:,ii,jj), 'fk14',4)';
    end
end

% Load the saved deep learning network
filename = 'netLstm.mat';
persistent mynet;

if isempty(mynet)
    mynet = coder.loadDeepLearningNetwork(filename);
end

% Reshape the data to the network input requirements
for ii = 2: dim1-1
    for jj = 2:dim2-1
        tempData = permute((dataMRA(:,ii-1:ii+1, jj-1:jj+1, :)), [1 4 2 3]);
        outLabels(:,ii,jj)= mynet.classify(reshape(tempData, [1006 45])') ;

    end
end

% Extend the labels to full area
outLabels(:,1,:) = outLabels(:,2,:);
outLabels(:,dim1,:) = outLabels(:,dim1-1,:);
outLabels(:,:,1)= outLabels(:,:,2);
outLabels(:,:,dim2)= outLabels(:,:,dim2-1);

end
```

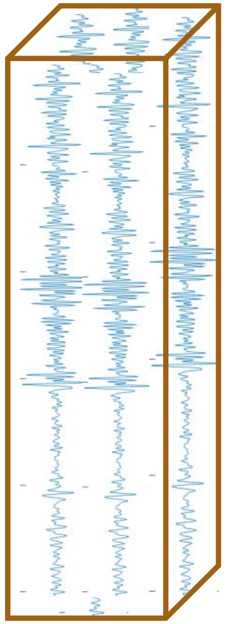


Field sensors

15

15

1006



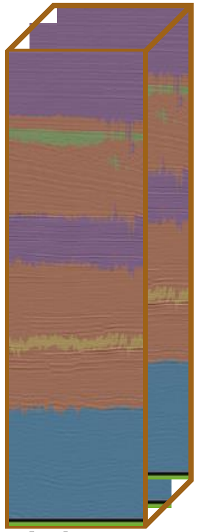
Streaming data

Edge Device

15

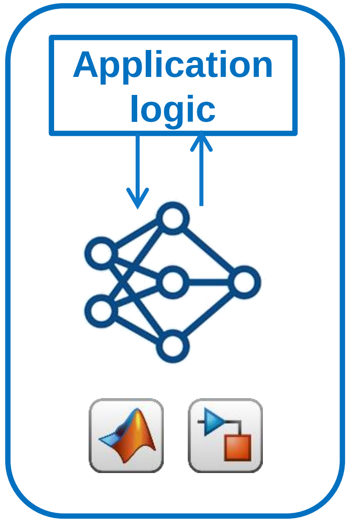
15

1006



Processed output : Save to disk/server

Edge GPU Deployment



GPU Coder

MATLAB Coder

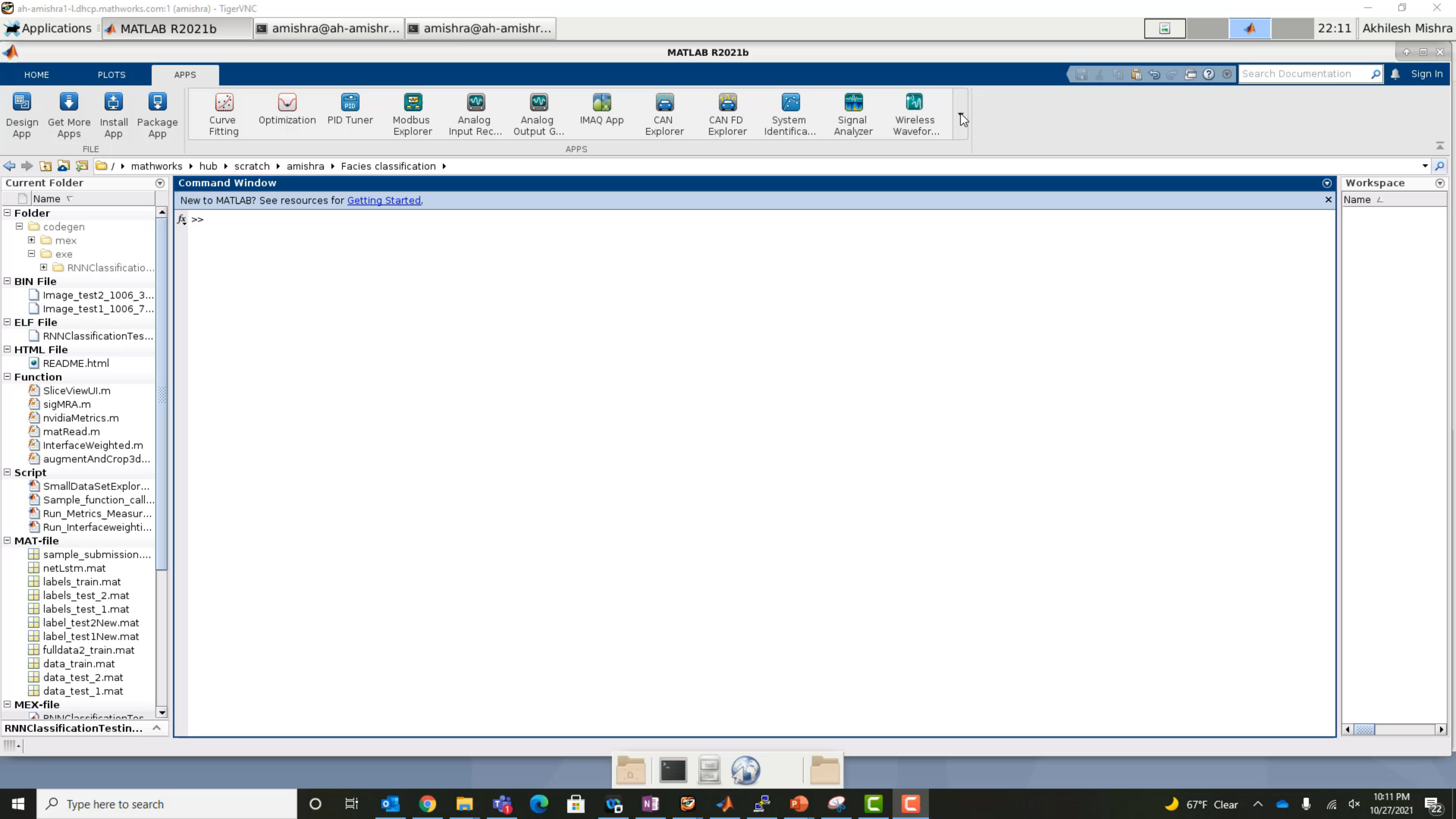
Target Libraries

NVIDIA TensorRT & cuDNN
cuFFT
cuBLAS

Intel MKL-DNN Library

ARM Compute Library
CMSIS
…

CUDA Code with GPU Coder


ah-amishra1-l.dhcp.mathworks.com:1 (amishra) - TigerVNC
Applications
MATLAB R2021b
amishra@ah-amishr...
amishra@ah-amishr...
22:11
Akhilesh Mishra
MATLAB R2021b
HOME
PLOTS
APPS
Search Documentation
Sign In
Design App
Get More Apps
Install App
Package App
FILE
Curve Fitting
Optimization
PID Tuner
Modbus Explorer
Analog Input Rec...
Analog Output G...
IMAQ App
CAN Explorer
CAN FD Explorer
System Identifica...
Signal Analyzer
Wireless Wavefor...
APPS
/ mathworks hub scratch amishra Facies classification
Current Folder
Name
Folder
codegen
mex
exe
RNNClassificatio...
BIN File
Image_test2_1006_3...
Image_test1_1006_7...
ELF File
RNNClassificationTes...
HTML File
README.html
Function
SliceViewUI.m
sigMRA.m
nvidiaMetrics.m
matRead.m
InterfaceWeighted.m
augmentAndCrop3d...
Script
SmallDataSetExplor...
Sample_function_call...
Run_Metrics_Measur...
Run_Interfaceweighti...
MAT-file
sample_submission....
netLstm.mat
labels_train.mat
labels_test_2.mat
labels_test_1.mat
label_test2New.mat
label_test1New.mat
fulldata2_train.mat
data_train.mat
data_test_2.mat
data_test_1.mat
MEX-file
RNNClassificationTestin...
Command Window
New to MATLAB? See resources for Getting Started.
>>
Workspace
Name
Type here to search
67°F Clear
10:11 PM
10/27/2021

Compare the speedups

Load the data

```
inp = IMAGE(:,1:15, 1:15);
inp = single(inp);
```

C code execution time

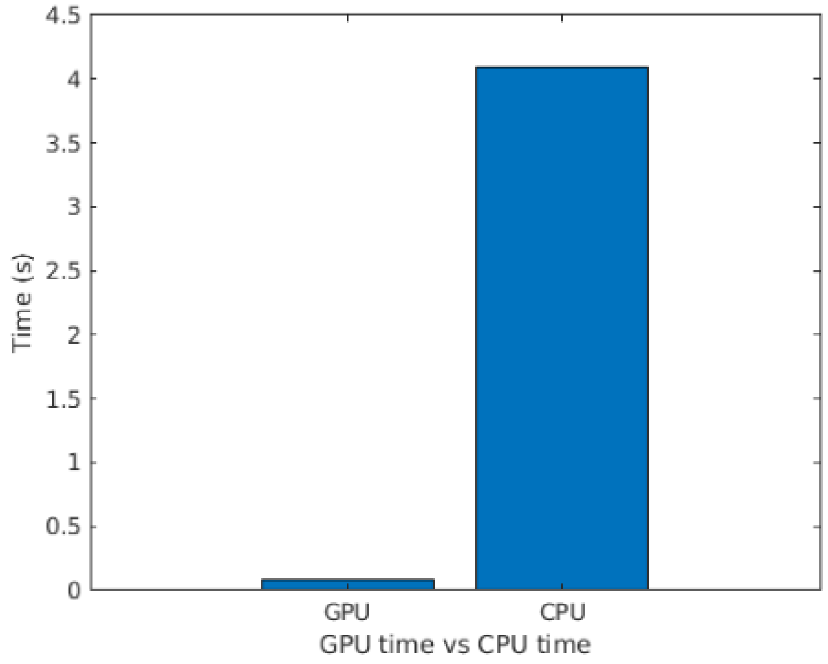
```
tic
outLabels = RNNClassificationTestingCodegen_mex(inp);
cpuTime = toc;
```

GPU code execution time

```
tic
outLabels = RNNClassificationTestingCodegen_mex(inp);
gpuTime = toc;
```

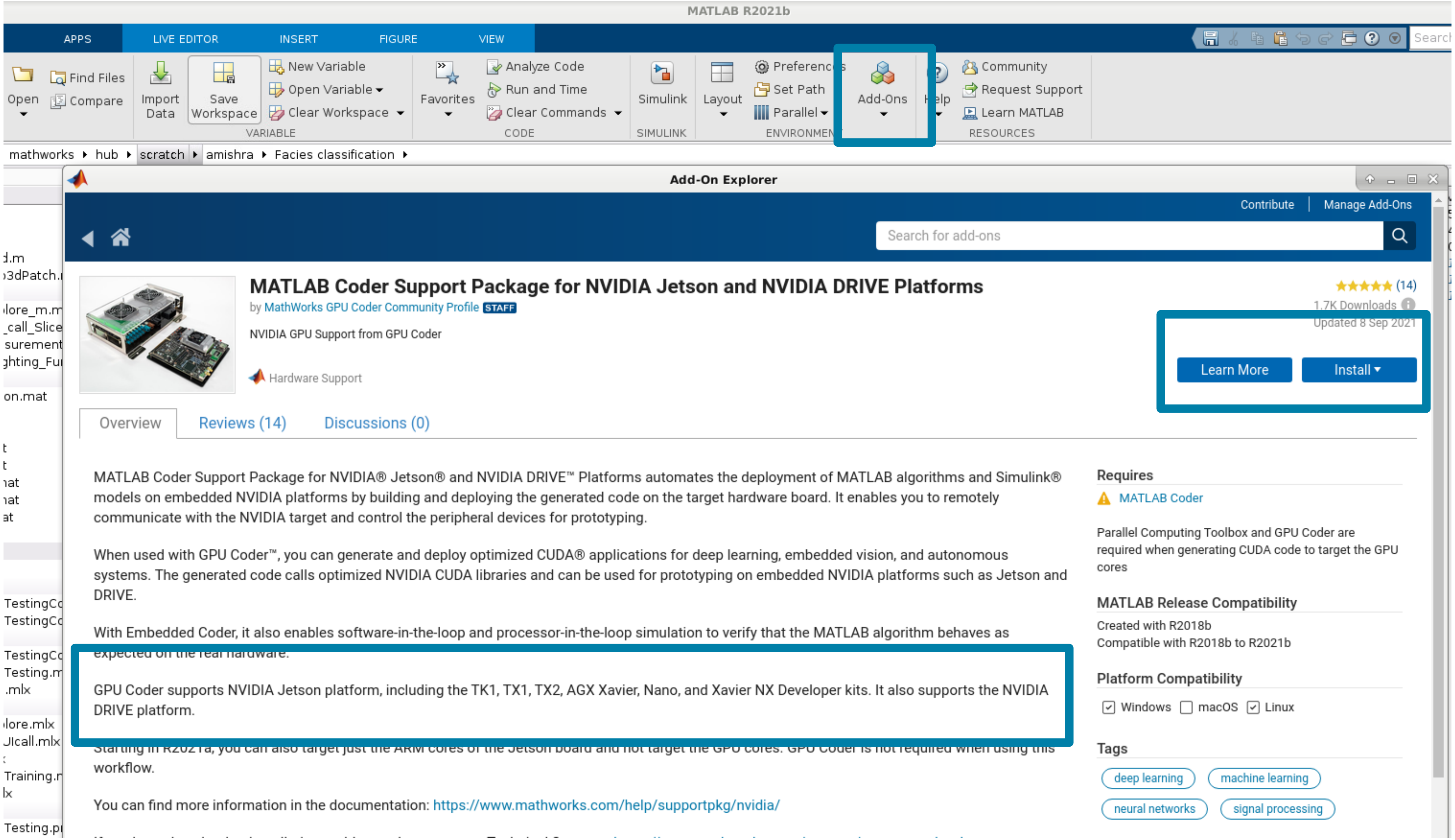
Plot the results

```
bar([gpuTime, cpuTime])
xlabel('GPU time vs CPU time')
xticklabels({'GPU', 'CPU'})
ylabel('Time (s)')
```

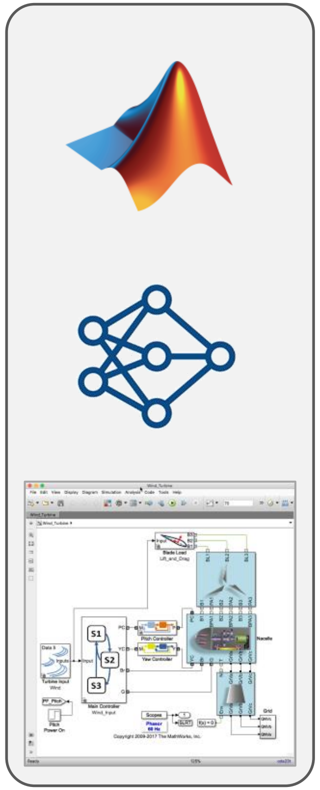


50x Speed Up

Step 2 : Deploy on NVIDIA Jetson Target



Recap: Deploy to Any Processor with Best-in-class Performance



Code Generation

CPU — **MATLAB Coder + Embedded Coder**

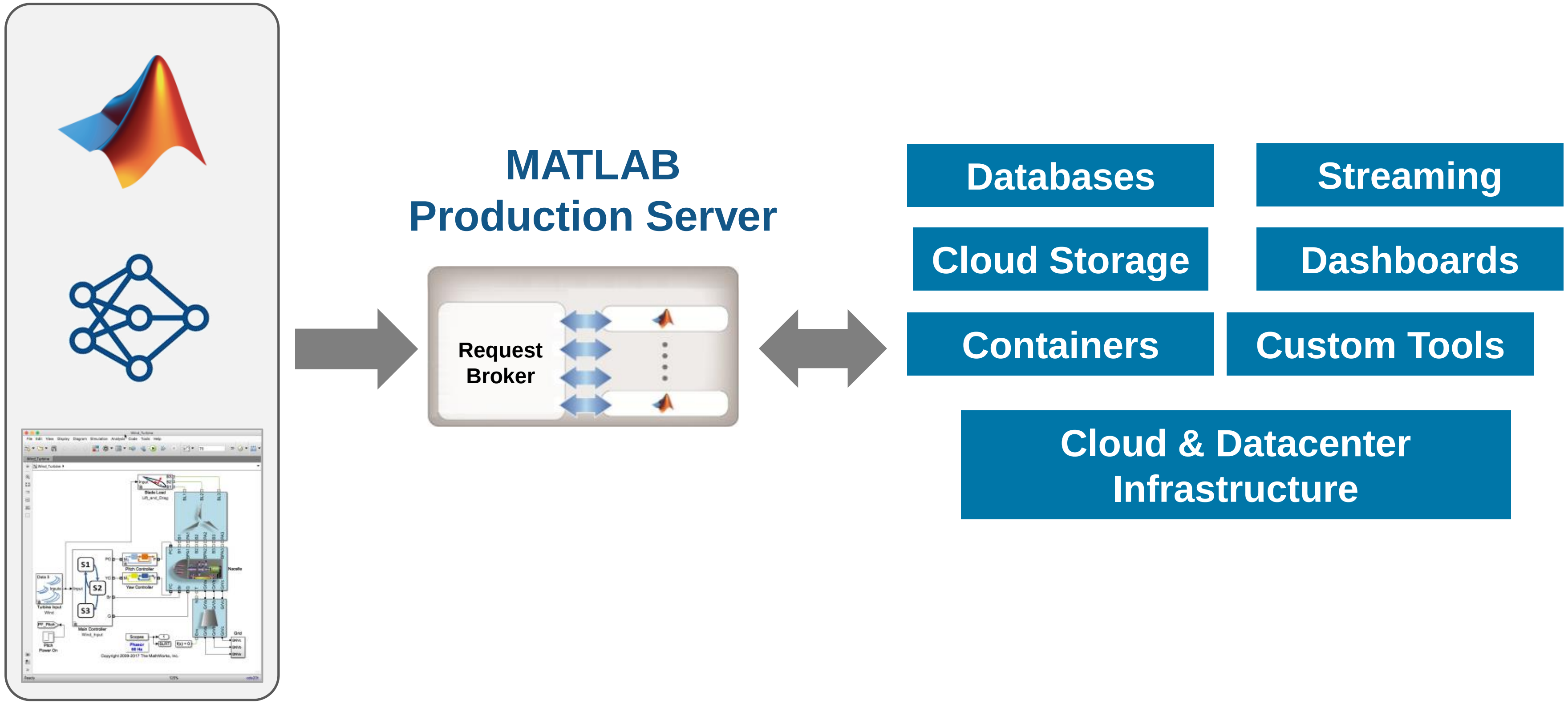
GPU — **GPU Coder + Embedded Coder**

PLC — **PLC Coder**

FPGA — **HDL Coder**
HDL Verifier
Fixed Point Designer
Deep Learning HDL Toolbox

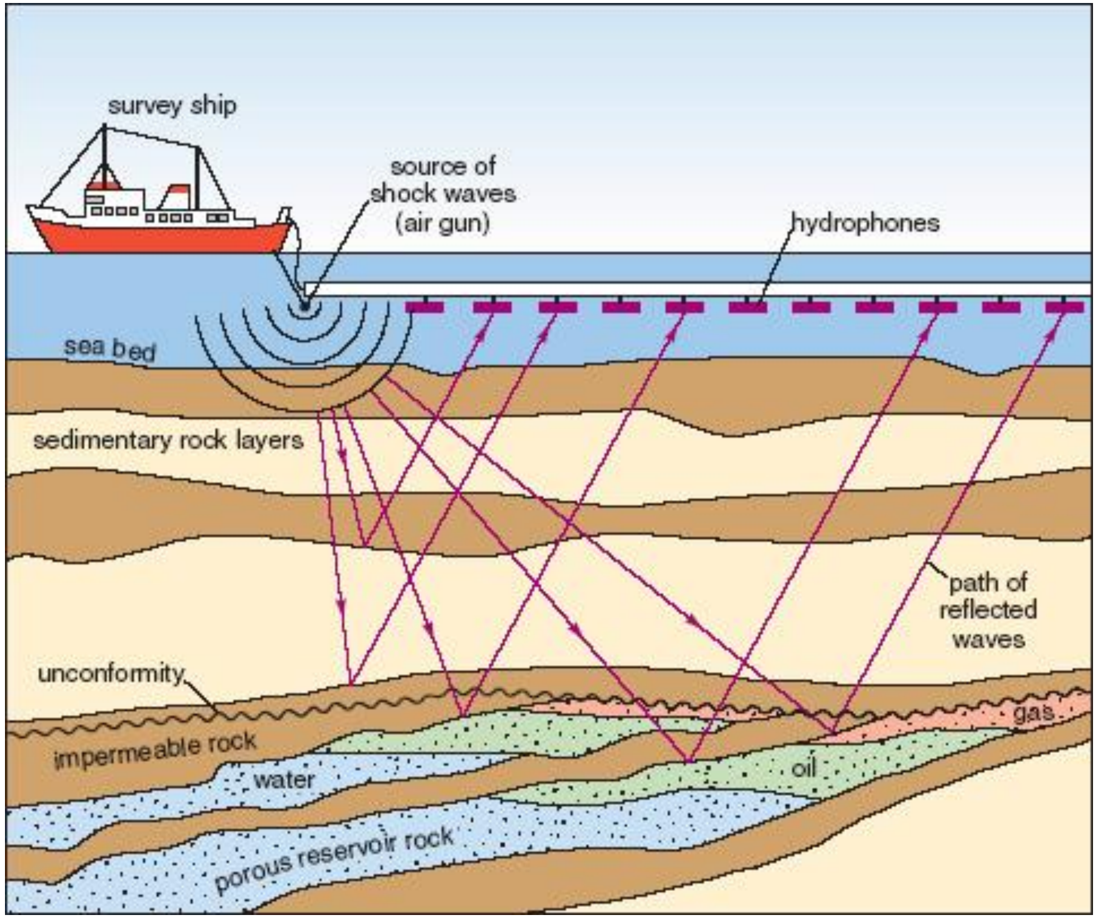
All models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop

Additionally : Deploy to Enterprise IT Infrastructure



Recap

- Building complex algorithms with low code / no code approach
- Easy Iterating signal processing + AI with MATLAB
- Handling big data and scaling compute intensive algorithms – AWS, NGC
- Automated Edge computing deployment



Access to full code and article :

https://blogs.mathworks.com/deep-learning/2021/08/03/mathworks-wins-geoscience-ai-gpu-hackathon/

< Jumpstart your DCASE Challenge 2021... Auto-Categorization of Content using Deep... >

MathWorks Wins Geoscience AI GPU Hackathon

Posted by Johanna Pingel, August 3, 2021 172 views (last 30 days) | 7 Likes | 0 comment

The following post is from Akhilesh Mishra, Mil Shastri and Samvith V. Rao from MathWorks here to talk about their participation and in a Geoscience hackathon. Akhilesh and Mil are Applications Engineers and Samvith is the Industry Marketing Manager supporting the Oil and Gas industry.

Background

SEAM (SEG Advanced Modeling Corp.) is a petroleum geoscience industry body that fosters collaborations among industry, government, and academia to address major Geological challenges. Their latest event was a hackathon (SEAM AI Applied Geoscience GPU Hackathon) that sought to explore the use of AI to improve both qualitative and quantitative interpretation of geophysical images of Earth's interior, and speed up the applications using NVIDIA GPUs.

A total of 7 teams participated from all over the world, including commercial companies (Chevron, Total, Petrobras) and a mix of industry and university students. Each team was assigned a mentor who is an expert geoscientist working for a top oil and gas company.

The Challenge

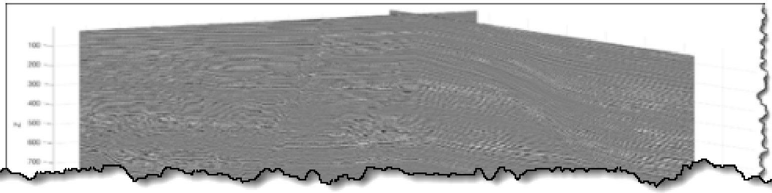
Geologic interpretation of survey data—especially raw seismic images of Earth's interior is an important step for the oil and gas industry. Seismic images delineate volumes of rock inside the Earth with different physical and geologic characteristics summarized by the term **"facies"**. An important step in interpretation of the images—which guide exploration, drilling, production, and abandonment of underground reservoirs—is identification and classification of all distinct geologic facies in a seismic image, often called *seismic facies identification or classification*.

This process is still done largely geologists assisted by specialists in geophysical data collection, processing, imaging, and display. Successful interpreters are experts in identifying features such as channels, mass transport complexes, and collapse features.

The problem statement of the hackathon was to **train an algorithm to recognize distinct geologic facies in seismic images automatically**, producing an interpretation that could pass for that of an expert geologist, or be used as a starting point to speed up human interpretation.

The Data

We were given the following data set drawn from the Parihaka region off the coast of New Zealand. This data is open to the public and has been labeled by a Chevron geoscientist. The figure below shows a rendering of two vertical slices and one horizontal slice through the 3D seismic image used in this challenge. Standard cartesian coordinate system is used to plot the image, with X and Y measuring the horizontal positions near the earth's surface and the Z measuring depth below the earth.



MathWorks Engineering Support

Training

Guided Evaluations

Onsite Workshops

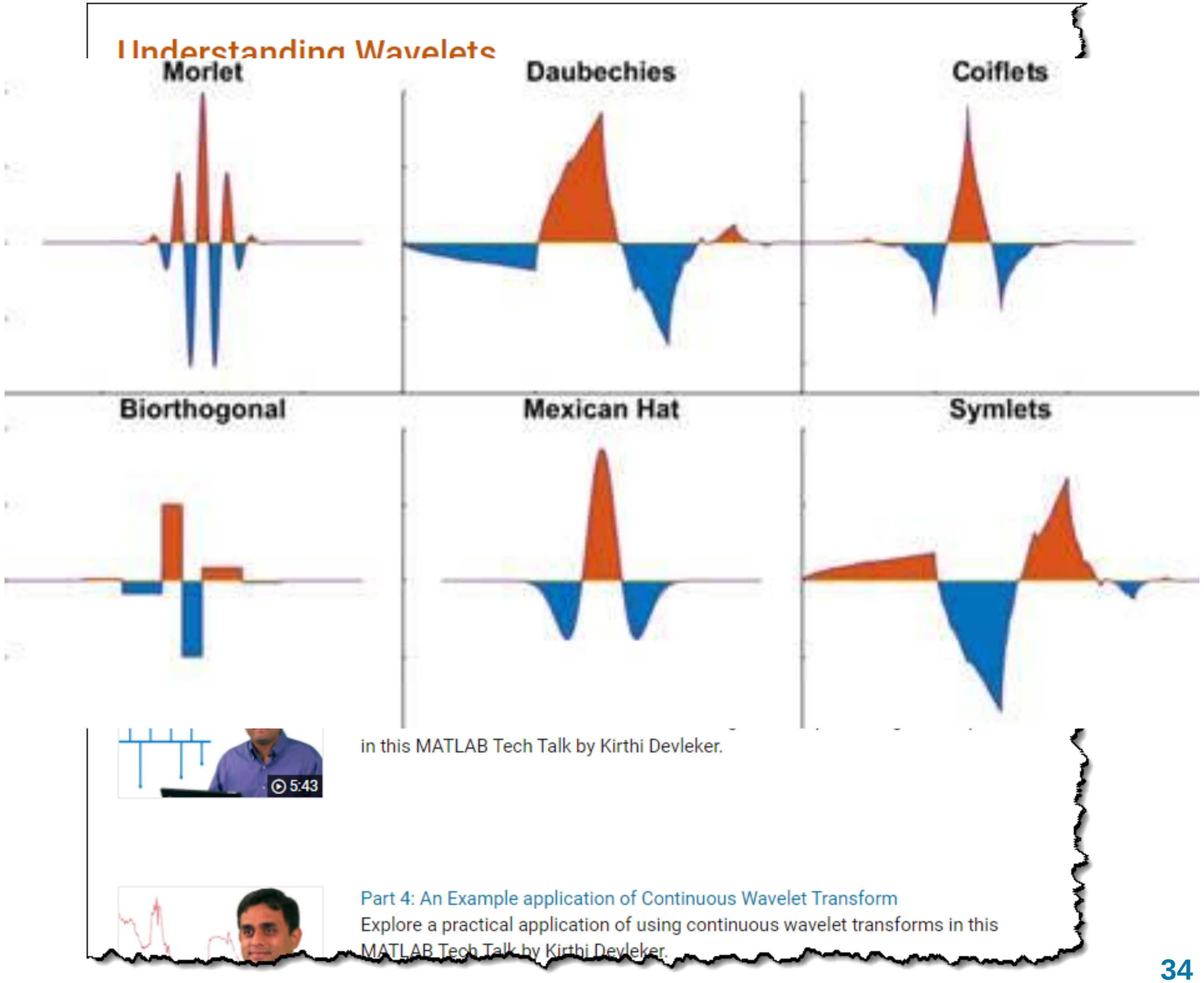
Consulting

Technical Support

Further Learning & Teaching

- Wavelets Analysis with MATLAB
  (7 hr Instructor led training)
  - Continuous Wavelet Analysis
    - Time frequency analysis
    - Wavelet coherence
    - Wavelet synchro-squeezing
    - Time-localized filtering
  - Discrete Wavelet Analysis
    - Multiresolution analysis
    - Denoising with wavelets
    - Wavelet packet transform
  - Wavelets for AI
    - Wavelet scattering networks
    - Wavelet for feature extraction

Wavelets tech-talk series



Further Learning & Teaching

- Deep Learning Onramp
  - 2 hr online tutorial
- Deep Learning Workshop
  - 3 hr hands on session
  - Contact us to schedule
- Deep Learning Training
  - 16 hr in depth course
  - Online or Instructor Lead
- Teaching Deep Learning with MATLAB
  - Curriculum support

Thank you !

Linkedin:
www.linkedin.com/in/akhilesh-mishra-mathworks

Email :
amishra@mathworks.com