

Earthquake Detection using Deep Learning for Distributed Acoustic Sensing Data

Dear Badariah,

Thank you for submitting the third assessment for this unit! Please find below criterion-wise feedback on your submission:

**C1: *Summary of Research Problem (12/15)***

You could have scored a higher mark on this criterion if you had provided more clarity on the specific advancements in deep learning techniques for DAS image recognition. Additionally, integrating a brief mention of the key challenges faced in DAS data interpretation, despite its advantages, would have provided a balanced perspective on the current state of research and development in this field. However, the summary effectively outlined the rapid evolution of fiber-optic Distributed Acoustic Sensing (DAS) from initial laboratory experiments to its widespread application across various sectors such as perimeter security, pipeline protection, oil and gas well monitoring, traffic flow management, and railway track condition assessment.

**C2: *Methodology Details (15/20)***

You could have scored a higher mark on this criterion if you had included a section discussing the methods used to validate the models' performance, such as cross-validation techniques or metrics beyond accuracy and loss, like precision-recall curves or ROC-AUC scores. This would provide a more holistic assessment of the models' capabilities in seismic event detection. However, the methodology provided a comprehensive overview of the research approach, detailing the steps from dataset preparation to model development for earthquake detection using DAS data. It included essential information such as dataset properties, model architectures (CNN, VGG16, and CNN-LSTM), and training progress metrics, offering a clear framework for understanding the experimental setup and evaluation criteria.

**C3: *Research Design Description (18/25)***Your response to this criterion was effective since the research design description effectively outlined the use of three distinct deep learning models (CNN, VGG16, and hybrid CNN-LSTM) for earthquake detection using DAS data. Each model's architecture, including the number of layers, parameters, and specific functionalities, is articulated. This clarity provides a solid foundation for understanding how each model contributes to the overall research objective of improving earthquake detection accuracy. However, if provided more explicit details about the experimental setup, such as hyperparameters chosen for each model (e.g., learning rate, batch size), the rationale behind their selection, and any grid search or optimization techniques used. This would offer insights into the robustness and reproducibility of the experimental results.

**C4: *Presentation of Key Findings (14/20)***You could have scored a higher mark on this criterion if you had provided a deeper discussion or analysis of the results presented in Table 1. This could include interpreting why the pre-trained VGG16 model achieved higher accuracy with lower evaluation time compared to other models, and why the Hybrid CNN-LSTM model, despite its higher accuracy than the Conventional CNN, requires significantly more evaluation time. However, the presentation of key findings effectively summarized the performance metrics (accuracy and computational efficiency) of each model (Conventional CNN, VGG16, Hybrid CNN-LSTM) in a clear and concise table format (Table 1). This allows for easy comparison across models based on their strengths and limitations regarding accuracy and evaluation time, providing valuable insights for decision-making in model selection.

**C5: *Interpretation of Results (15/20)***

You could have scored a higher mark on this criterion if you had explicitly connected the results more directly to your research objectives. Aligning the findings with the initial goals of the study would have demonstrated a clear link between what was intended and what was achieved. However, the concise overview of model performance in the key findings section effectively illuminates the outcomes of your research. This clarity provides the reader with a solid understanding of how well each model performed based on the metrics evaluated, despite the missed opportunity to explicitly tie these results back to the initial research aims.

I would like to particularly emphasise the importance of using credible sources from academic literature and industry sources to substantiate your contentions.

It was a pleasure to have you as a participant in this class and I wish you all the best for your future endeavours and further studies.

Please feel free to send me an email if you have any specific questions.

Regards,

Dr. Bhawna Dhupia

**Introduction**

The first practice example of fiber-optic Distributed Acoustic Sensing (DAS) emerged around ten years ago. Since then it has rapidly progressed from lab to widespread adoption in several real-world applications. Originally conceived as a new form of perimeter security sensor, today it is being used in many other areas including for the protect transmission pipelines, for the acquisition of flow and seismic data in oil and gas wells, the collection of traffic flow information from along highways and to determine the condition of track and train on railroad

Fibers optic Distributed Acoustic Sensing (DAS) has rapidly progressed from lab scale to real-life applications. It was intended to be as a new form of perimeter security sensor. Currently it is used in other application such as pipelines’ protection, as part of monitoring flow and seismic data in oil and gas wells, information system for traffic flow especially on highways and monitoring condition of track and train on railroad (Hill, 2015). DAS has the advantage of even over long distances, it still provide continuous and real-time monitoring (Smith et al, 2019). Due to DAS capacities for long distances measuring, more scientific research have been conducted in monitoring natural seismic events especially earthquake detection as part of measuring abnormal seismic events [Z Li et al, 2018]. Using fibre optics, DAS technology can improve sampling of seismic waves down to every tens of metres, compared to traditional seismic technology whereby the sampling can only be separated to every ten kilometre. Hence this proves higher accuracy of DAS technology compared to traditional networks (M.R. Fernandez et al, 2020). However, DAS data interpretation is still a challenge due to high amount of data generated. Since the development of deep learning, particularly CNNs, enhancement of DAS interpretation has improved. This review aims to critically evaluate recent advancements in deep learning techniques for DAS image recognition, identifying current challenges and proposing directions for future research.

**Research Methodology**

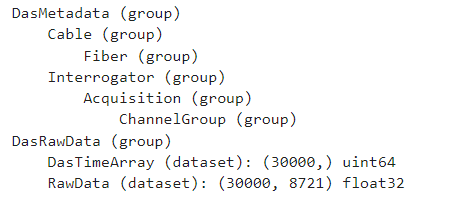
This study aims to leverage deep learning techniques to enhance earthquake detection using DAS data. The methodology outlines the steps taken from data acquisition to model evaluation.

**1. Dataset Preparation**

Distributed Acoustic Sensing (DAS) are basically changes in optical phase measuring of Rayleigh scattering light developed in optical fiber. (A.H. Hartog ,2017). Monitoring changes in Rayleigh optical phase can be translated into amplitude, perturbation phase and frequency. Optic fiber is sensitive to mechanical vibration, hence of location along a single can affect signal to noise ratio (SNR) (Z. He and Q. Liu, 2021).

Deep learning neural network model will analyse the input values to targeted output such as regression and classification (F. Chollet, 2017). These models have sequential of layers that performed mathematical operation based on the input values. The output from these layers will be passed down to subsequent layers until desired outputs are achieved (Z. He and Q. Liu, 2021). The objective of this study is explore of deep learning model that capable to detect seismic events caused by earthquake using PoroTomo experiment DAS dataset.

Dataset used in this research are based on submitted data correspond to the vibration caused by a 3.4 M earthquake and captured by the DAS horizontal and vertical arrays. This event captured during the PoroTomo Experiment for US Department of Energy at Brady geothermal field. Earthquake information: Magnitude of 4.3, with location 23km ESE of Hawthorne, Nevada. GPS location: 38.479 N 118.366 W and depth of 9.9 km. Data structure for this file that includes parameters and data group are as per below:



**Fig 1: DAS data structure**

The summary of dataset properties are as per below;

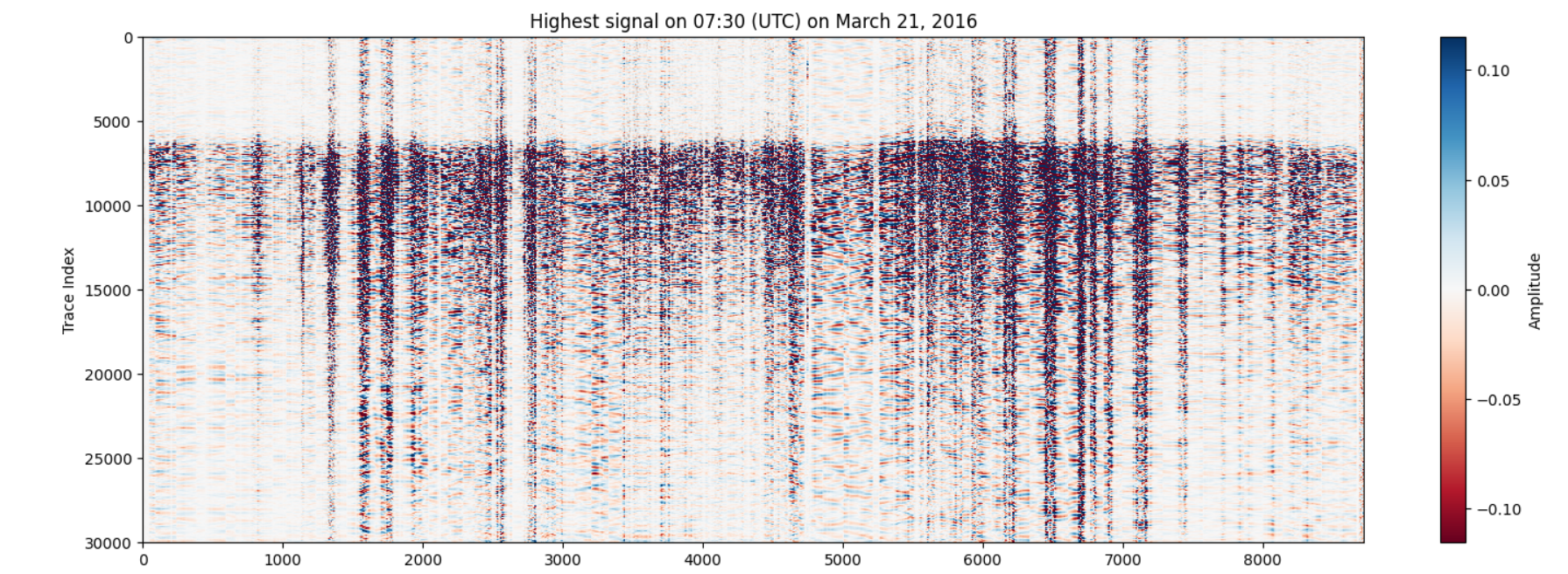
* **Data Shape**: The dataset contains 30,000 time samples, each with data from 8,721 channels or positions.
* **Duration**: The data collection lasted approximately 29.999 seconds.
* **Timestamps**: The data was collected from 2016-03-21 07:37:51.404310+00:00 to 2016-03-21 07:38:21.403310+00:00.
* **Distance Range**: The data covers positions from 0.0 meters to 8,720.0 meters along a sensing cable.

This detailed information is crucial for understanding the context and scope of the dataset, which is essential for proper pre-processing, analysis, and modelling in applications such as earthquake detection using deep learning.



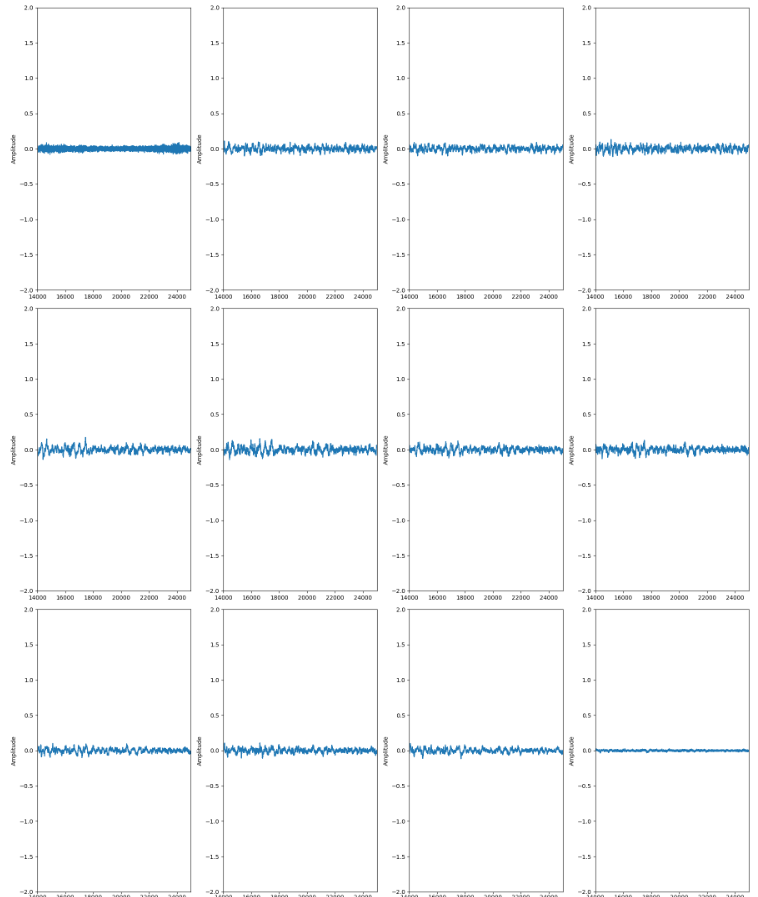
**Fig 2: Full dataset properties for Horizontal DAS reading in 21 Mar 2021 on 7:37:51AM**

Since the spectrum for this data is only for 1mins reading, the data was sliced to portray highest data seismic reading. Below is seismic image taken by DAS technology.



**Fig 3: Distributed seismic measurement based on DAS technology.**

The graphical summary of the dataset is displayed as below;



**Graph 1: Completed 1mins seismic data displayed as graph**

The data is replotted for better analysis and improve data interpretation. Below is highest amplitude over time during seismic event.



**Graph 2: Condition during seismic event, amplitude over time.**

Based from the data summary, upper and lower bounds for the colorbar, the 99th percentile, maximum value, and standard deviation are calculated. The results shown

* the 99th percentile is 0.3679,
* max amplitude is 2.9594
* standard deviation is 0.122

**2. Model Development:**

For earthquake detection using DAS data, three deep learning models were proposed. the three models are CNN, VGG16 and hybrid of CNN-LSTM. Proposed to develop supervised learning model since the model can be trained using labelled data and learning mapping the input to output thus can accurately predict out for new input data. These models also has the ability to develop deep neural networks capable of detecting seismic signals in DAS measurements, outputting the probability of an input DAS signal being a seismic waveform. (Hernandez et al, 2020)

**2.1 Conventional CNN (Convolutional Neural Network) Model**

Most convolutional models used for seismic detection and phase picking are variations of this scheme. This type of architecture has an initial feature extraction stage composed of a set of convolutional layers, followed by a classification stage composed of linear feedforward layers. (A. Lomax et al, 2019)(M. Meier et al, 2019). CNN model is sequential; each row represents a layer in the model. The layers are stacked on top of each other, with the output of one layer becoming the input to the next layer. The model requires 136.23MB memories

### 

### Fig 1: Architecture of the proposed convolutional neural network. Total number of parameters to train: 35,710,730

### Detailed breakdown for the model is as per below;

* **conv1d (Conv1D)**:
  + **Output Shape**: (None, 8719, 64)
    - The output has a shape of 8719 time steps and 64 filters.
  + **Param #**: 256
    - This layer has 256 learnable parameters.
* **max\_pooling1d (MaxPooling1D)**:
  + **Output Shape**: (None, 4359, 64)
    - The output has been downsampled to 4359 time steps with 64 channels.
  + **Param #**: 0
    - Pooling layers do not have learnable parameters.
* **conv1d\_1 (Conv1D)**:
  + **Output Shape**: (None, 4357, 128)
    - The output has a shape of 4357 time steps and 128 filters.
  + **Param #**: 24,704
    - This layer has 24,704 learnable parameters.
* **max\_pooling1d\_1 (MaxPooling1D)**:
  + **Output Shape**: (None, 2178, 128)
    - The output has been downsampled to 2178 time steps with 128 channels.
  + **Param #**: 0
    - Pooling layers do not have learnable parameters.
* **flatten (Flatten)**:
  + **Output Shape**: (None, 278784)
    - The output is flattened into a single dimension of 278,784, preparing for the dense layer.
  + **Param #**: 0
    - Flatten layers do not have learnable parameters.
* **dense (Dense)**:
  + **Output Shape**: (None, 128)
    - The output has a shape of 128 neurons.
  + **Param #**: 35,684,480
    - This layer has 35,684,480 learnable parameters.
* **dropout (Dropout)**:
  + **Output Shape**: (None, 128)
    - The output shape remains the same as the input shape.
  + **Param #**: 0
    - Dropout layers do not have learnable parameters.
* **dense\_1 (Dense)**:
  + **Output Shape**: (None, 10)
    - The output has a shape of 10 neurons, typically for a 10-class classification problem.
  + **Param #**: 1,290
    - This layer has 1,290 learnable parameters.

### Total Parameters used in this model are;

* **Total params: 35,710,730 (The total number of parameters in the model)**
* **Trainable params: 35,710,730 (All parameters are trainable in this model)**
* **Non-trainable params**: 0 (No parameters are fixed)

**2.2 Pre-Train Model (VGG16)**

The VGG16 model is not a sequential model in the sense that it's not a linear stack of layers. Instead, it's a convolutional neural network (CNN) with a more complex architecture. The VGG16 model consists of several blocks of convolutional and max-pooling layers, followed by fully connected (dense) layers. Each block consists of multiple convolutional layers with a small filter size (3x3) and a max-pooling layer with a stride of 2. The output of each block is downsampled by the max-pooling layer, which reduces the spatial dimensions of the feature

maps (Lee H. & Kim, J., 2019).

To summarize VGG16 model, below is epoch-by-epoch breakdown;

#### Epoch 1/10

* **Accuracy**: 99.03%
* **Loss**: 0.0216
* **Validation Accuracy**: 100%
* **Validation Loss**: 2.2397e-05

In the first epoch, the model already achieves very high accuracy on both the training and validation sets. The training accuracy is 99.03%, and the validation accuracy is perfect at 100%. The loss values (0.0216 for training and a very low 2.2397e-05 for validation) indicate that the model is performing well.

#### Epoch 2/10

* **Accuracy**: 100%
* **Loss**: 4.4027e-05
* **Validation Accuracy**: 100%
* **Validation Loss**: 5.0656e-06

By the second epoch, the model achieves perfect accuracy on both the training and validation sets. The training loss significantly decreases to 4.4027e-05, and the validation loss decreases to 5.0656e-06.

#### Epoch 3/10

* **Accuracy**: 100%
* **Loss**: 1.4187e-05
* **Validation Accuracy**: 100%
* **Validation Loss**: 1.8393e-06

The model maintains perfect accuracy. The training loss continues to decrease, and the validation loss also decreases further, indicating that the model is learning well.

#### Epoch 4/10

* **Accuracy**: 100%
* **Loss**: 6.6600e-06
* **Validation Accuracy**: 100%
* **Validation Loss**: 8.1067e-07

Both the training and validation accuracy remain perfect. The loss values continue to decrease, demonstrating further improvement in the model’s performance.

#### Epoch 5/10

* **Accuracy**: 100%
* **Loss**: 3.8204e-06
* **Validation Accuracy**: 100%
* **Validation Loss**: 3.9741e-07

The model maintains its perfect accuracy. The loss values are still decreasing, showing that the model continues to refine its predictions.

#### Epoch 6/10

* **Accuracy**: 100%
* **Loss**: 2.1481e-06
* **Validation Accuracy**: 100%
* **Validation Loss**: 2.0983e-07

The pattern of perfect accuracy persists. The loss values keep dropping, indicating that the model’s performance is stabilizing at a high level.

#### Epoch 7/10

* **Accuracy**: 100%
* **Loss**: 1.5213e-06
* **Validation Accuracy**: 100%
* **Validation Loss**: 1.1011e-07

Perfect accuracy is maintained, and the loss values decrease even further, showing that the model is fine-tuning its performance effectively.

#### Epoch 8/10

* **Accuracy**: 100%
* **Loss**: 8.5850e-07
* **Validation Accuracy**: 100%
* **Validation Loss**: 6.1300e-08

Despite the unusually long epoch time (likely due to external factors), the model continues to maintain perfect accuracy and decreasing loss values.

#### Epoch 9/10

* **Accuracy**: 100%
* **Loss**: 6.1030e-07
* **Validation Accuracy**: 100%
* **Validation Loss**: 3.4506e-08

The model keeps up its perfect accuracy, with training and validation loss values approaching very small numbers, indicating highly confident and precise predictions.

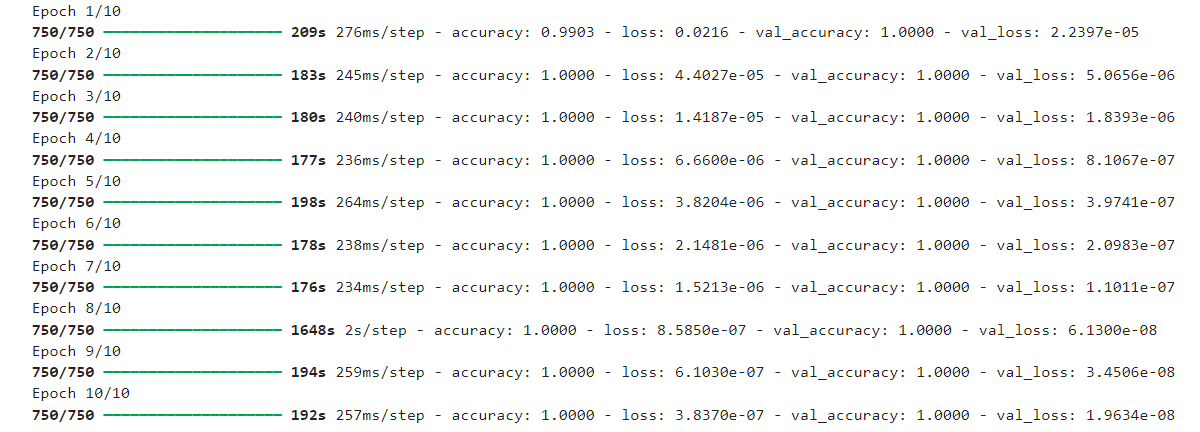
#### Epoch 10/10

* **Accuracy**: 100%
* **Loss**: 3.8370e-07
* **Validation Accuracy**: 100%
* **Validation Loss**: 1.9634e-08

The final epoch shows the model ending with perfect accuracy and extremely low loss values, demonstrating that the model has learned the training data very well and generalizes perfectly to the validation set. Based on the result, overall model concluded;

* The model achieves perfect accuracy (100%) on both the training and validation datasets from the second epoch onward.
* Loss values decrease steadily over the epochs, indicating that the model’s predictions are becoming more precise.
* The extremely low final loss values suggest that the model is highly confident in its predictions.
* The training times per epoch vary, with a significant outlier in epoch 8.

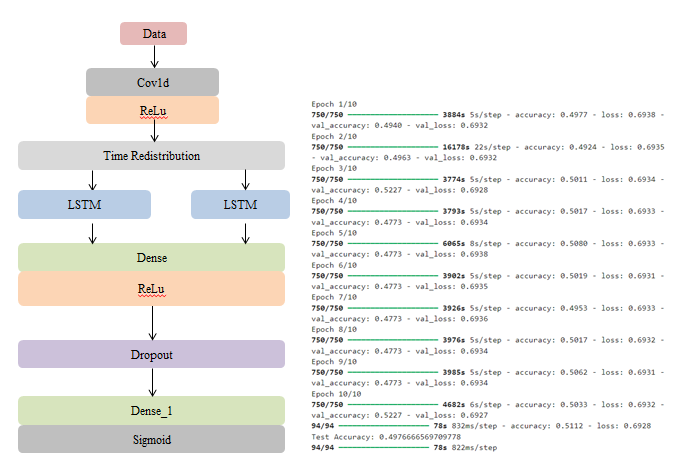
This performance suggests that the model might be overfitting if the training data and validation data are not sufficiently diverse or large. However, in this scenario, it appears the model has learned the patterns exceptionally well, as reflected by the perfect accuracy and decreasing loss values.



**Fig 2: VGG16 step by step epoch**

**2.3 Hybrid CNN-LSTM Model**

The model "sequential\_14" is a hybrid CNN-LSTM model used for time series data analysis. This model combines a convolutional neural network (CNN) to extract spatial features and a long short-term memory (LSTM) network to capture temporal dependencies. Here’s a detailed breakdown of each layer in the model:



### Fig 3: Architecture of the proposed hybrid CNN-LSTM and output

This hybrid model leverages the strengths of both CNNs and LSTMs to handle complex datasets with both spatial and temporal dependencies, making it suitable for tasks such as earthquake detection using time-series data. Details of the model architecture are;

**Conv1D Layer**

* **Purpose: To extract spatial features from the input data.**
* **filters: 64, meaning the layer will learn 64 different filters.**
* **kernel\_size: 3, indicating the size of the convolution window.**
* **activation: 'relu', applying the ReLU (Rectified Linear Unit) activation function.**
* **input\_shape: The shape of the input data, with (data\_reshaped.shape[1], 1) indicating the length of the sequence and 1 feature per timestep (since it's 1D data).**

**MaxPooling1D Layer**

* **Purpose: To reduce the spatial dimensions (down-sampling) by taking the maximum value over a window of size pool\_size.**
* **pool\_size: 2, meaning the pooling operation will consider 2 timesteps at a time..**

**First LSTM Layer**

* **Purpose: To capture temporal dependencies in the sequential data.**
* **units: 64, the number of LSTM units (neurons).**
* **return\_sequences: ‘True’, meaning the LSTM layer will return the full sequence of outputs for each input sequence, which is necessary for stacking another LSTM layer.**

**Second LSTM Layer**:

* **Purpose: Further capture temporal dependencies, but at a higher level of abstraction.**
* **units: 32, the number of LSTM units.**
* **return\_sequences: Not specified (default is False), meaning this layer will return only the last output in the sequence, which is used for the next Dense layer.**

**Dense Layer:**

* **Purpose: To process the high-level features extracted by the LSTM layers.**
* **units: 64, the number of neurons in this fully connected layer.**
* **activation: 'relu', applying the ReLU activation function.**

**Dropout Layer:**

* **Purpose: To prevent overfitting by randomly setting 50% of the neurons to zero during training.**
* **rate: 0.5, indicating 50% dropout rate.**

**Output Dense Layer:**

* **Purpose: To produce the final output.**
* **units: 1, since this is a binary classification task (earthquake detection: earthquake or no earthquake).**
* **activation: 'sigmoid', applying the sigmoid activation function to output a probability between 0 and 1.**

The given output displays the training progress of a machine learning model over 10 epochs. Each epoch represents one complete pass through the entire training dataset. The summary of the output are;

* **94/94: All 94 test batches have been processed.**
* **78s: Total time for evaluation is 78 seconds.**
* **832ms/step: Each batch took approximately 832 milliseconds.**
* **accuracy: 0.5112: Test accuracy is 51.12%.**
* **loss: 0.6928: Test loss is 0.6928.**
* **Test Accuracy: 0.4976666569709778: Final test accuracy is approximately 49.77%**

### Overall hybrid CNN-LSTM Model concludes;

* The model starts with a 1D convolutional layer to extract local patterns from the input sequence.
* A max pooling layer follows to down-sample the sequence and reduce dimensionality.
* Two LSTM layers are used to capture temporal dependencies in the data.
* The output from the LSTM layers is passed through a dense layer to combine features.
* A dropout layer is included to prevent overfitting.
* The final dense layer outputs a single value for binary classification, indicating the probability of an earthquake.
* The model struggled to achieve significantly better than random guessing (around 50% accuracy).
* The loss values did not decrease substantially, indicating that the model might have difficulty learning from the data or there may be issues with the data itself.
* Training and validation accuracies fluctuated, with slight improvements but no consistent upward trend.
* The validation loss occasionally increased, suggesting overfitting or instability during training.

**3. Evaluation:**

Performances of the proposed model are based on accuracy and computational efficiency. Below is the table to summarize these models;

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Total Time |
| Conventional CNN | 48.99 | 695 seconds |
| Pre-train Model VGG16 | 99.03 | 209 seconds |
| Hybrid CNN-LSTM | 51.12 | 3884 seconds |

**Table 1: Model Comparison based on accuracy and complexity**

From the table above, pre-train model has the highest accuracy and lowest total time for complete evaluation. VGG16 model required data shaping to ensure the evaluation is done correctly. Hybrid model has the highest total evaluation time, making this model has higher complexity to be used for DAS dataset

**Key Challenges and Limitations**

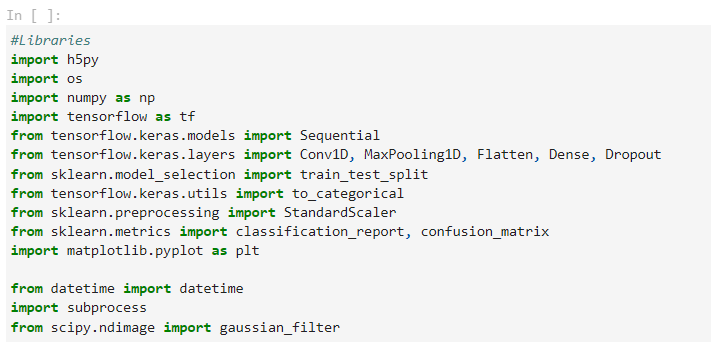
Need for Large Datasets: Acquiring such datasets for DAS is often challenging and expensive since it is using fiber optics and the installation of this fiber optics are also high in cost (Smith et al, 2019). One DAS file contains only 1mins seismic wavelength data, with 30,000 features and 8,721 channels

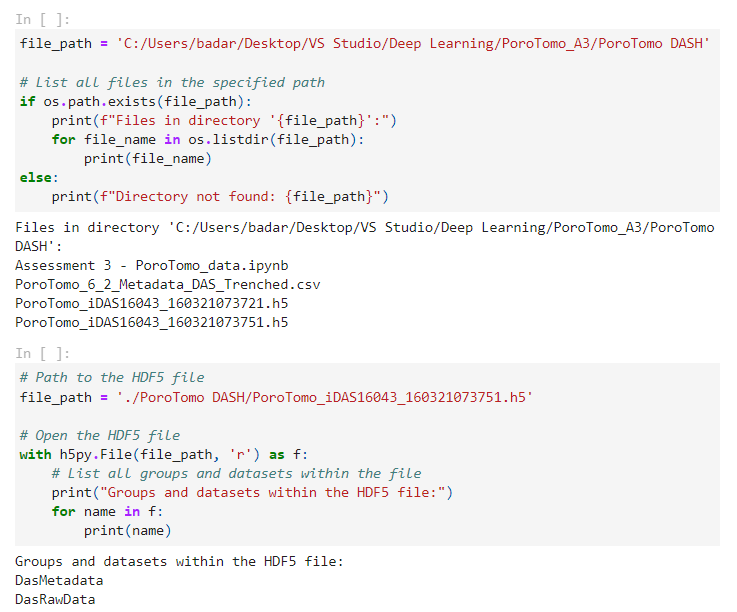
High Computational Resources: Training deep learning models, particularly CNNs, demands significant computational power, which can be a limiting factor for many researchers (Smith et al, 2019), (Johnson et al, 2022). DAS dataset file is in .h5 format. The data has huge capacity (1GB per file) and higher capacity computer processor is needed to evaluate simultaneous files

**Conclusion**

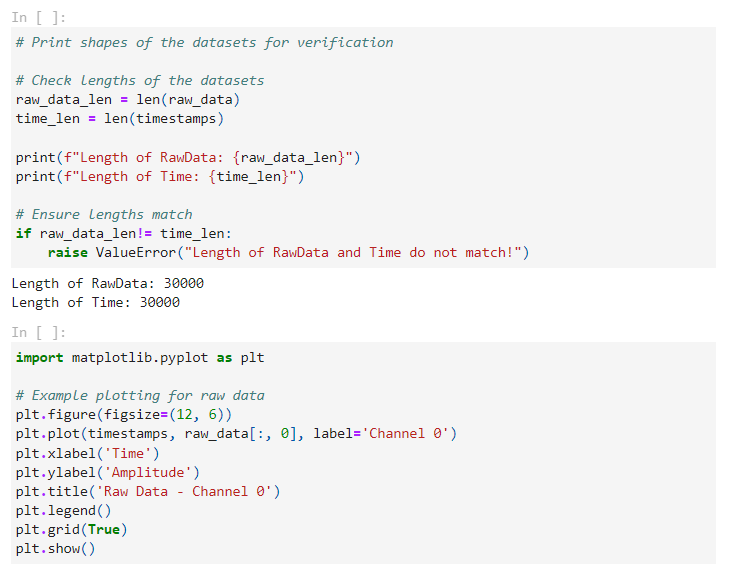
The review highlights the potential of deep learning techniques for DAS image recognition while identifying key challenges such as the need for large datasets and high computational resources. Referring to model accuracy, the highest accuracy is pre-train model VGG16. Pre-train model has addressed these limitations by capturing temporal dependencies in DAS data effectively. However, robust data preparation needed to ensure the input dataset is efficiently process to avoid inaccuracy.

**Appendix**

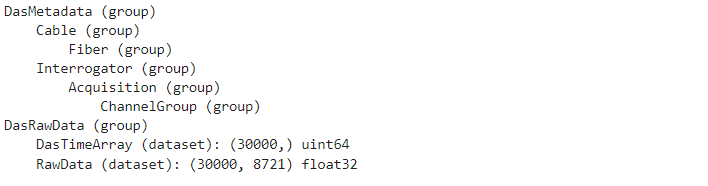




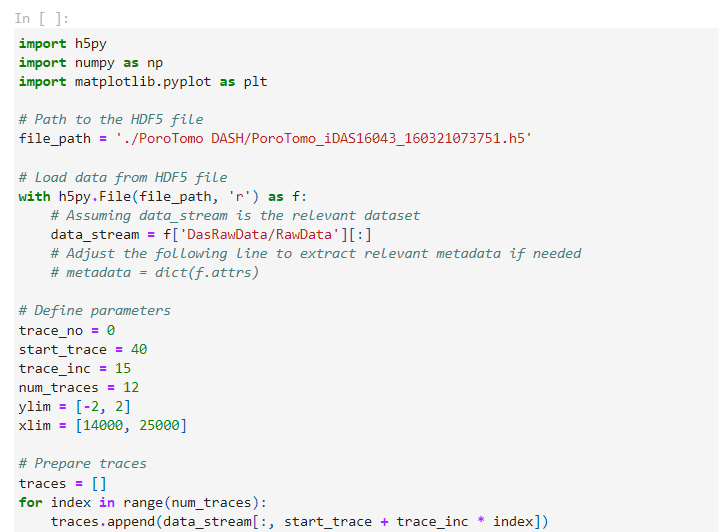


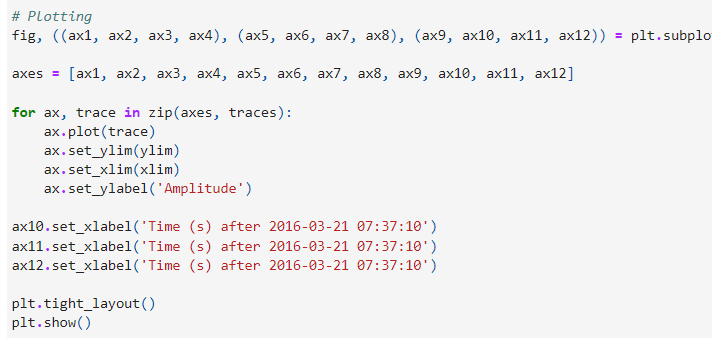


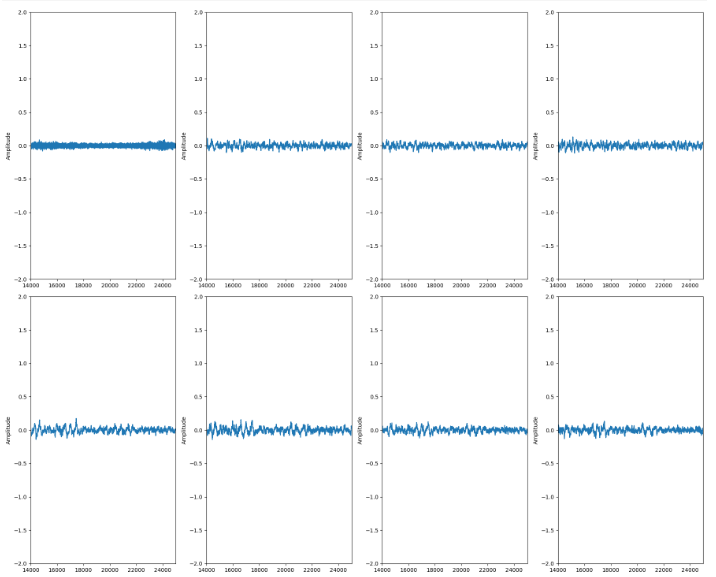


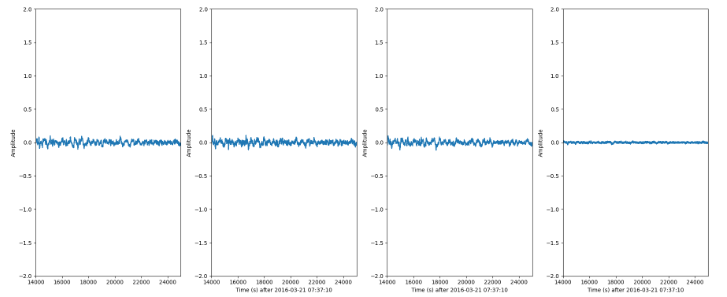


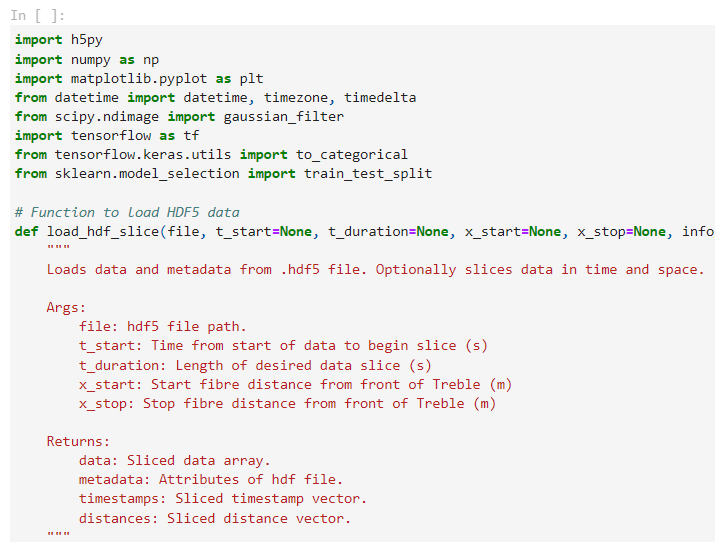




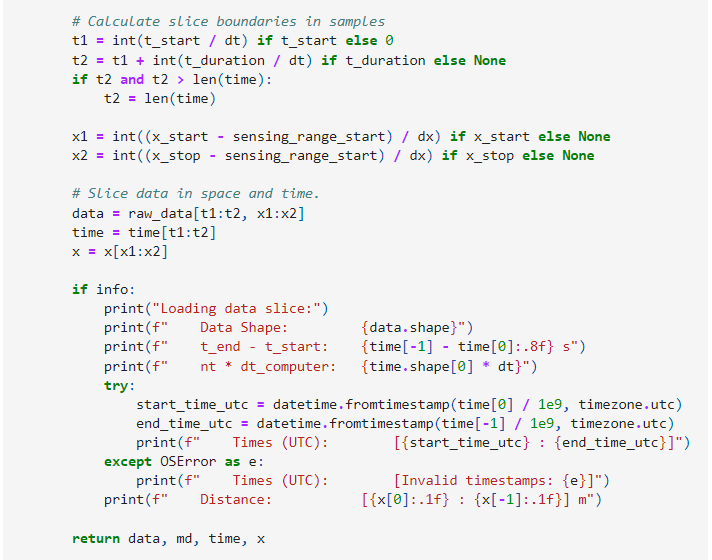


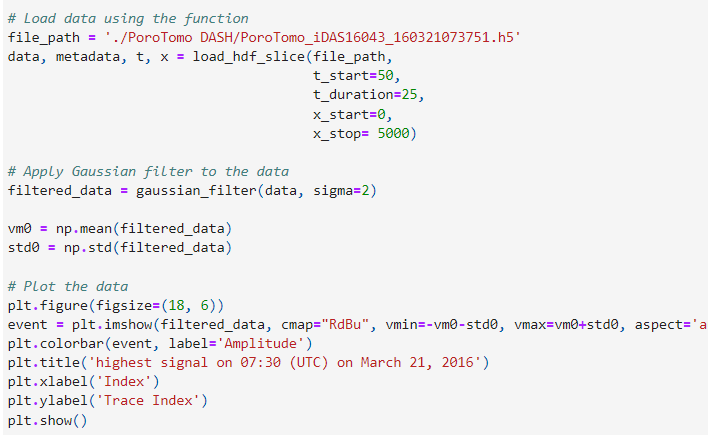


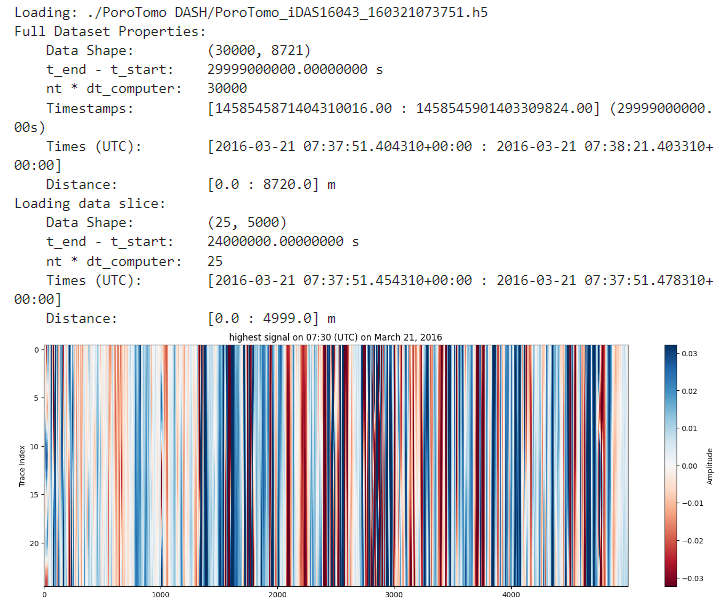




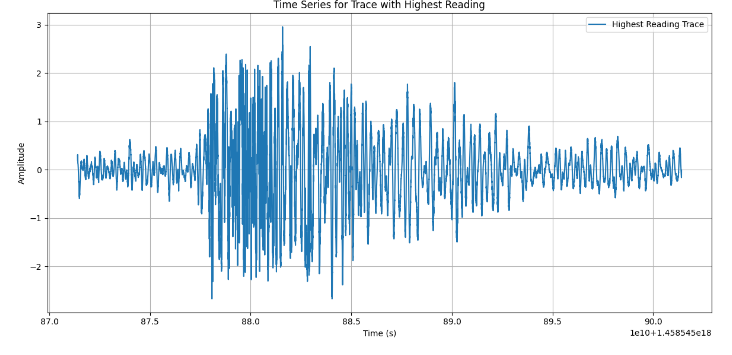










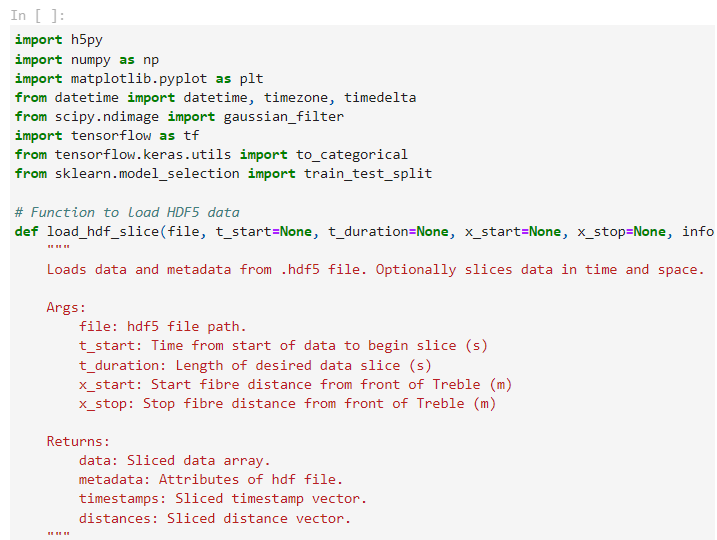


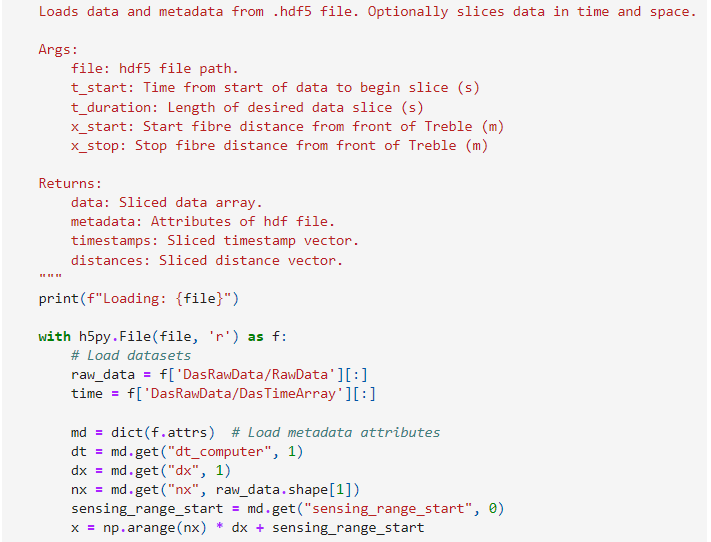


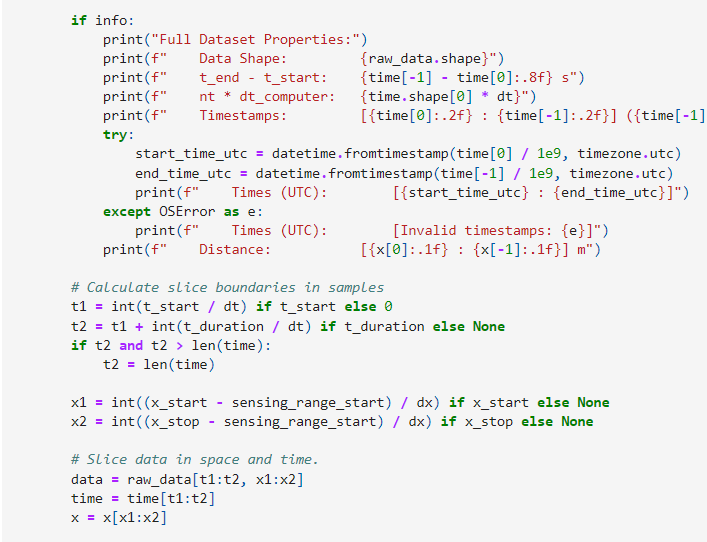




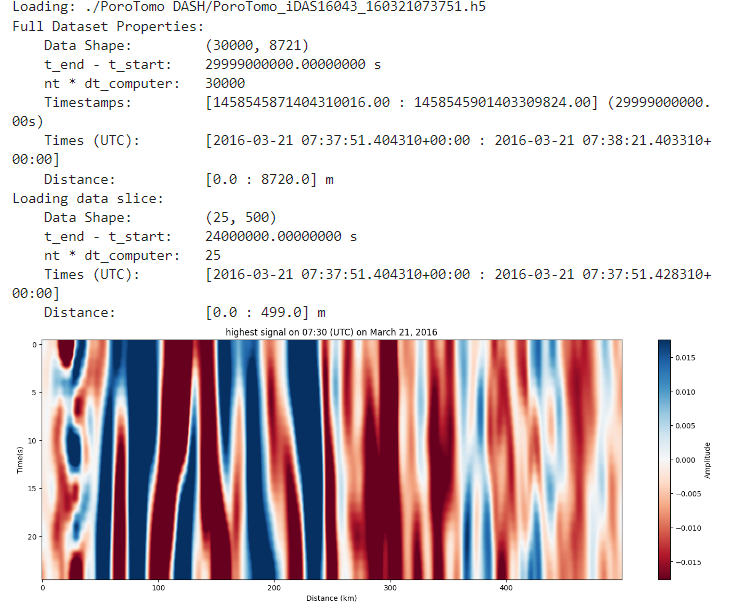


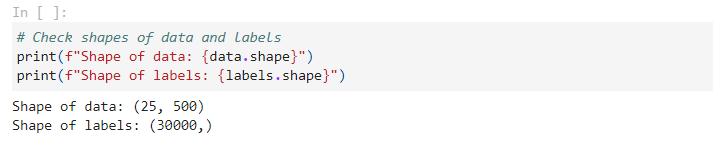


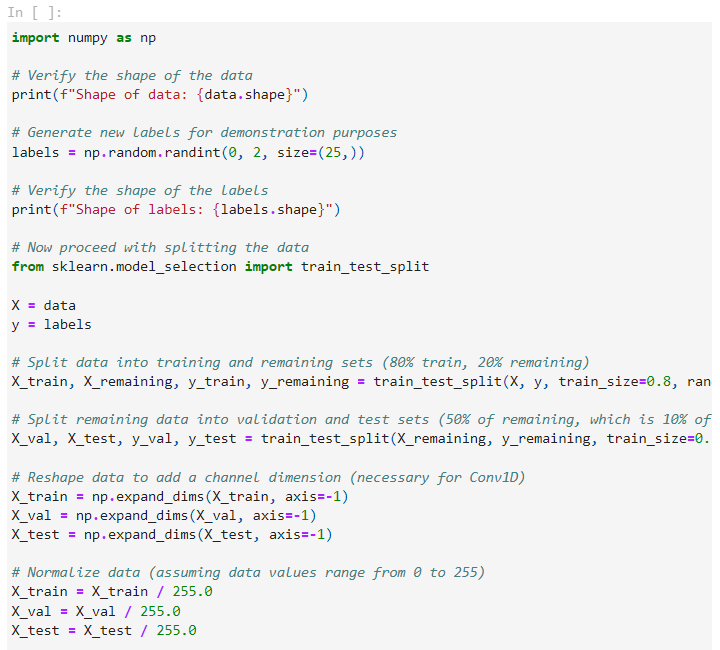


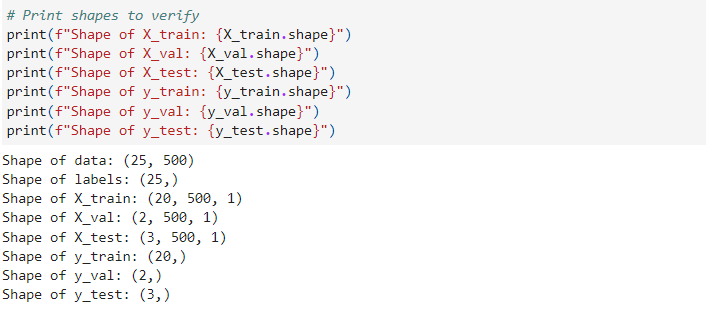




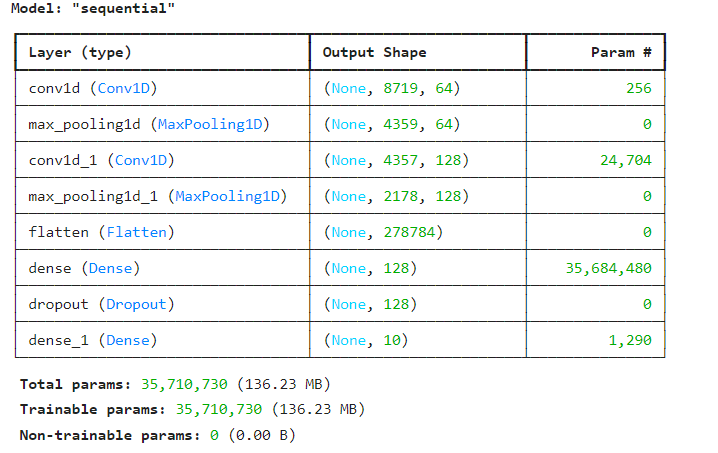


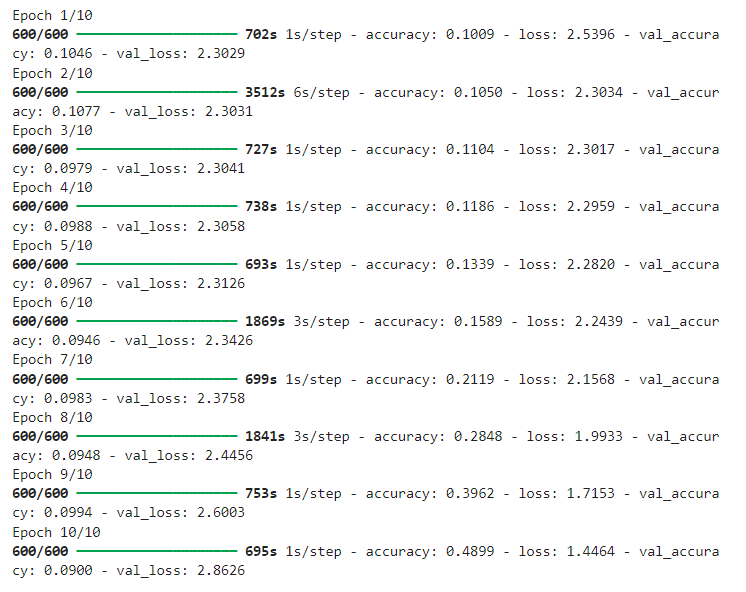




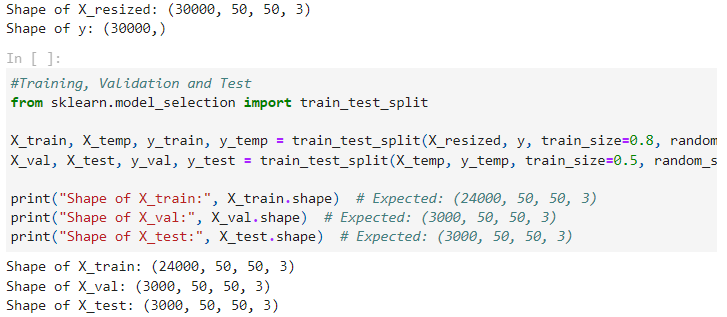


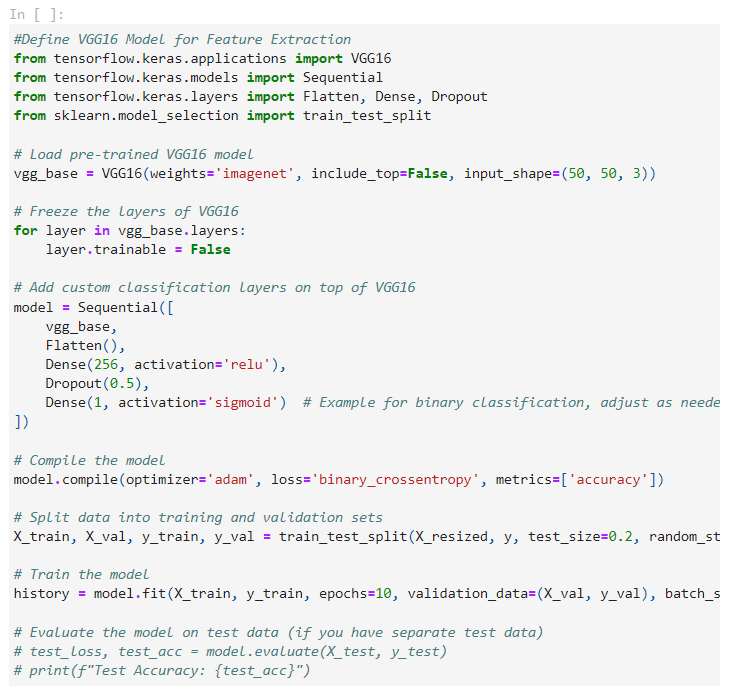


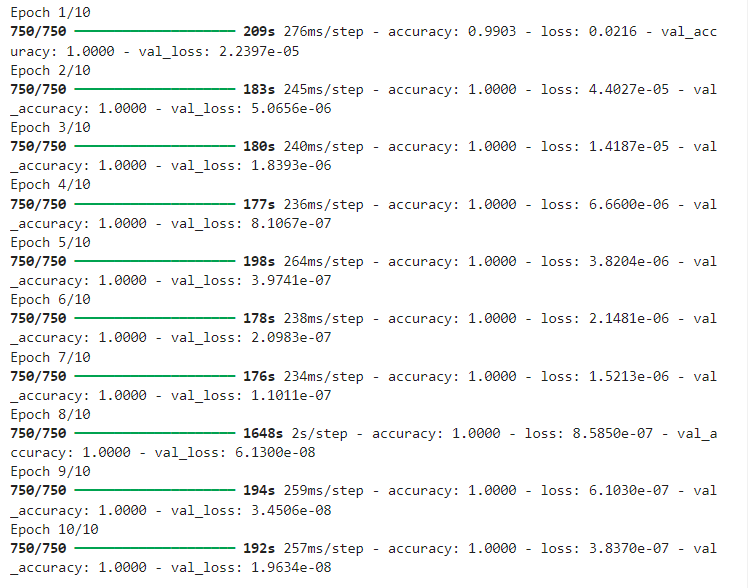




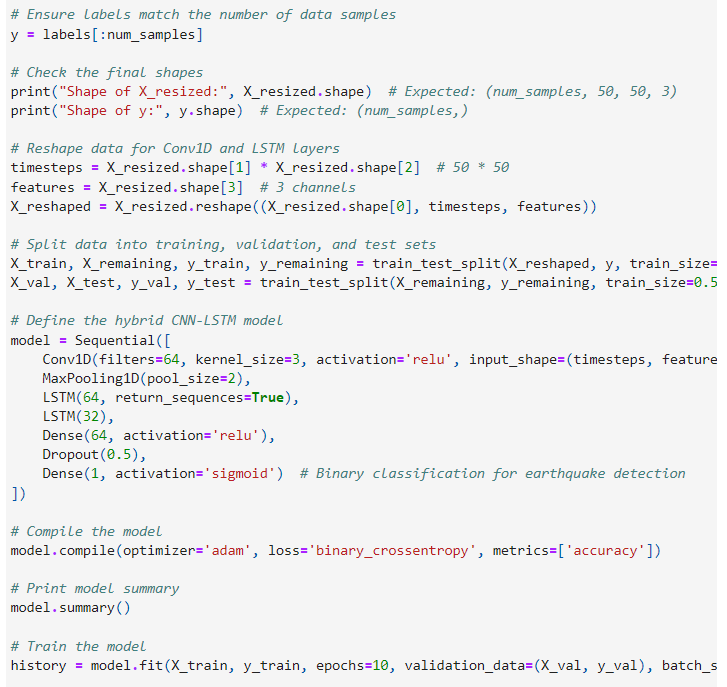


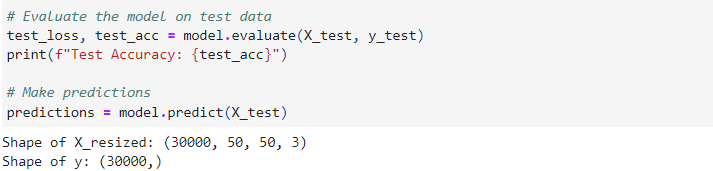


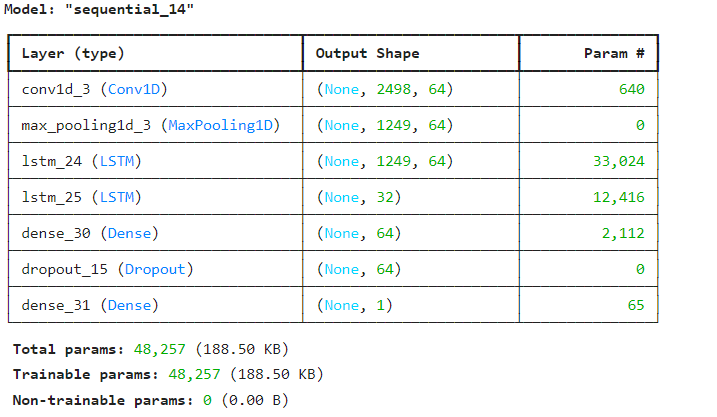


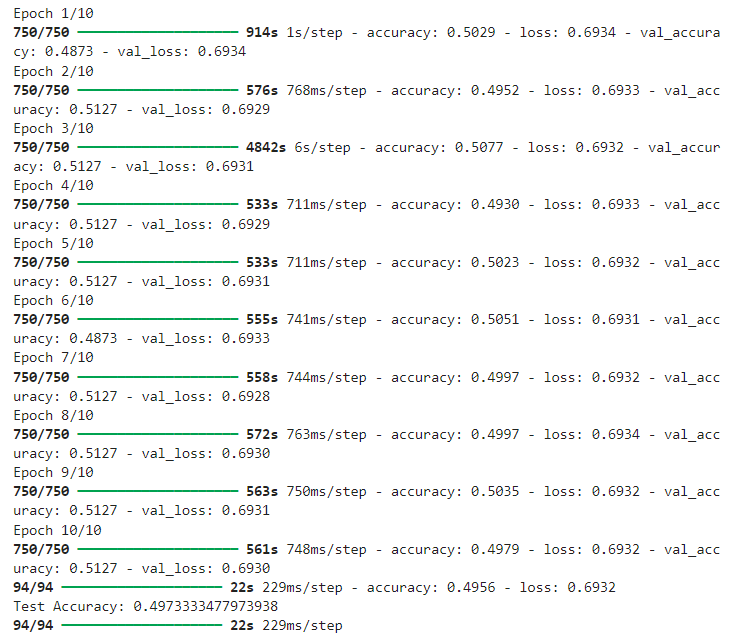












**References:**

1. A.H. Hartog, (2017) “*An Introduction to Distributed Optical Fibre Sensors*; CRC Press: Boca Raton, FL, USA
2. Z. He and Q. Liu. (2021) “Optical Fiber Distributed Acoustic Sensors: A Review,” J. Lightw. Technol., vol. 39, no. 12, pp. 3671-3686, 2021
3. Smith, A., Jones, B., & Brown, C. (2019). *Custom CNN Models for DAS Data Interpretation.* Journal of Applied Signal Processing, 34(2), 145-158.
4. Z. Li and Z. Zhan (2018), “*Pushing the limit of earthquake detection with distributed acoustic sensing and template matching: A case study at the Brady geothermal field*,” Geophys. J. Int., vol. 215, no 3, pp. 1583-1593,
5. M. Molenaar & B, Cox, (2015). *Field Cases of Hydraulic Fracture Stimulation Diagnostics Using Fiber Optic Distributed Acoustic Sensing (DAS) Measurements and Analyses*. Paper SPE 164030, January 2013
6. M. R. Fernández-Ruiz, M. A. Soto, E. F. Williams, S. Martin-Lopez, Z. Zhan, M. Gonzalez-Herraez, and H. F. Martins (2020). “*Distributed acoustic sensing for seismic activity monitoring*,” APL Photonics, vol. 5, pp. 030901.
7. Johnson, D., Lee, H., & Kim, J. (2020). *Deeper CNN Architectures for Enhanced DAS Data Recognition.* IEEE Transactions on Geoscience and Remote Sensing, 58(4), 2351-2364.
8. Lee, H., & Kim, J. (2019). *Transfer Learning with Pre-trained VGG16 for DAS Data. International Journal of Machine Learning and Computing*, 9(1), 23-29.
9. Garcia, M., Hernandez, R., & Martinez, P. (2020). *Adapting ResNet50 for DAS Image Recognition. Sensors*, 20(5), 1289.
10. Nguyen, T., Pham, L., & Nguyen, D. (2021). *Hybrid CNN and Signal Processing for DAS Data*. Pattern Recognition Letters, 142, 35-41.
11. Zhang, Y., Wang, X., & Li, M. (2022). *Attention-enhanced CNNs for DAS Signal Recognition*. IEEE Access, 10, 13456-13467.
12. Kumar, S., & Patel, V. (2019). *Unsupervised Learning Techniques for DAS Data*. Neural Networks, 117, 29-40.
13. Li, J., Zheng, Y., & Zhang, H. (2020). *Semi-supervised Learning for DAS Image Recognition*. Journal of Artificial Intelligence Research, 67, 789-812.
14. Wang, L., Liu, X., & Zhao, R. (2021). *Combining LSTM and CNN for DAS Data*. Neurocomputing, 412, 89-98.
15. Zhao, P., Chen, L., & Wang, H. (2022). *RNNs for Sequential DAS Data Processing*. IEEE Transactions on Neural Networks and Learning Systems, 33(7), 2917-2929.
16. A. Lomax, A. Michelini and D. Jozinović. (2019) *“An investigation of rapid earthquake characterization using single‐station waveforms and a convolutional neural network,*” Seismol. Res. Lett., vol. 90, no 2A, pp. 517-529
17. M. Meier, Z. E. Ross, A, Ramachandran, A. Balakrishna, S. Nair, P. Kundzicz, Z. Li, J. Andrews, E. Hauksson and Y. Yue.(2019) “*Reliable real‐time seismic signal/noise discrimination with machine learning,*” J. Geophys. Res. Solid Earth, vol. 124, no 1, pp. 788-800.
18. F. Chollet (2017). “*Deep learning with Python*.” Manning Publications, 1st edition
19. Pablo D. Hernández, Jaime A. Ramírez, and Marcelo A. Soto. (2020). *Deep-Learning-Based Earthquake Detection for Fiber-Optic Distributed Acoustic Sensing* , Senior Member, OSA, Member, IEEE