

Modern methods of visualization and preprocessing of seismic data for deep learning: a review of python-based approaches

Bekzat Bukhatov^{1,†}, Aizhan Altaibek^{1,2,†} and Marat Nurtas^{1,2*,†}

¹ International Information Technology University, 34/1 Manas St., Almaty, 050000, Kazakhstan

² Institute of Ionosphere, Gardening community IONOSPHERE 117, Almaty, 050020, Kazakhstan

Abstract

In seismic data interpretation, neural networks have significantly advanced tasks such as fault detection and subsurface analysis. However, the quality of input data remains a critical factor in the performance of these models. Seismic data is often noisy, incomplete, or inconsistent, making it necessary to apply robust preprocessing techniques to ensure that neural network or machine learning algorithms can effectively interpret the data. This paper presents an overview of modern preprocessing and visualization techniques tailored for seismic data, with a focus on Python-based implementations. We explore methods such as stratal slicing [1], attribute co-rendering[2], and data interpolation[3], which are crucial for improving both 2D and 3D seismic datasets before they are fed into neural networks. Our focus is not to directly prove the improvement in model performance but to examine how these techniques enhance the overall quality and interpretability of the seismic data. This review aims to provide geophysicists and data scientists with the tools necessary to improve data quality, optimize neural network input, without delving into the actual training of neural networks, but rather focusing on how to better prepare the input data for such tasks.

Keywords

Seismic data, Preprocessing, Neural networks, Deep learning, Fault detection, Seismic visualization, Python, Noise reduction, Stratal slicing, Attribute co-rendering, Data interpolation, 2D seismic data, 3D seismic data

1. Introduction

Seismic data analysis is a fundamental aspect of industries like oil and gas exploration, earthquake hazard mitigation, and geotechnical engineering. One of the critical tasks in seismic interpretation is the detection of faults and other geological structures, which offer valuable insights into subsurface characteristics. Traditionally, these tasks were performed using manual methods or computationally expensive models [4, 5]. While effective, these traditional approaches struggle with the increasing complexity and volume of seismic data [6-8], particularly as modern surveys generate larger datasets with more intricate structures.

In recent years, neural networks have emerged as powerful tools for automating seismic interpretation. Unlike traditional methods, which often require domain expertise and manual feature extraction, neural networks—especially convolutional neural networks (CNNs)—can automatically learn complex patterns from raw seismic data. This makes them highly effective in dealing with large 2D and 3D datasets, where traditional methods might falter due to the complexity and sheer scale of the data. While neural networks offer immense potential, their effectiveness is contingent upon the quality of the input data [9-11].

DTESI 2024: 9th International Conference on Digital Technologies in Education, Science and Industry, October 16–17, 2024, Almaty, Kazakhstan

* Corresponding author.

† These authors contributed equally.

✉ bukhatovbekzat@gmail.com (B. Bukhatov); a.altaibek@iitu.edu.kz (A. Altaibek); m.nurtas@iitu.edu.kz; (M. Nurtas)

ORCID 0009-0007-5684-8288 (B. Bukhatov); 0000-0001-8431-7950 (A. Altaibek); 0000-0003-4351-0185 (M. Nurtas)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Traditional methods for seismic interpretation, such as manual fault picking or applying pre-defined algorithms, often face limitations in handling large-scale datasets and complex geological structures. For example, methods like those described by Wu et al. [10] and Moreno et al. [9] rely on manual interaction or heuristic algorithms to unfault 3D seismic images and assist interpretation. These methods, while useful in specific cases, can struggle with maintaining consistency across larger datasets or detecting subtle features.

In contrast, neural networks, as demonstrated in works like Wu and Hale's automatic fault detection [11], offer a data-driven approach that learns from raw seismic images without manual intervention. By capturing complex spatial relationships and features across 3D datasets, CNNs have shown superior performance in accurately identifying faults, horizons, and other geological structures. This not only reduces the time required for interpretation but also improves accuracy, particularly in regions where traditional methods may miss or misinterpret critical features.

Seismic data, by its nature, is prone to noise and inconsistencies due to variations in acquisition methods and environmental factors. This noise can obscure key geological features, such as faults and horizons, leading to inaccurate results if not properly addressed. Therefore, preprocessing seismic data is essential to ensure that only relevant and clean data is fed into neural network models. Preprocessing includes techniques like noise reduction, data smoothing, and feature enhancement, which can greatly improve the interpretability of the data.

The importance of data quality cannot be overstated, particularly in fields like geophysics, where high accuracy is paramount. Poor-quality data can lead to suboptimal model performance, even with the most advanced neural network architectures. This makes robust preprocessing not just a recommendation but a necessity for successful seismic data analysis.

In this paper, we review modern preprocessing and visualization techniques specifically designed to handle the challenges of seismic data. Focusing on Python-based tools, we will explore methods like stratal slicing, co-rendering of seismic attributes, and data interpolation. These techniques not only prepare the data for neural network models but also enhance human interpretation, offering a more intuitive understanding of complex subsurface structures. By improving data quality and visualization, we aim to facilitate better decision-making processes in seismic data analysis.

2. Problem statement

Seismic data analysis, particularly in the context of fault detection and subsurface feature identification, is challenged by the inherent complexity and variability of the datasets. Both 2D and 3D seismic data can suffer from significant noise, inconsistencies, and artifacts, making it difficult to obtain accurate geological interpretations. While neural networks offer powerful solutions for seismic interpretation, their effectiveness depends heavily on the quality of the input data.

A key issue is the lack of standardized tools and methods that can consistently improve data quality before it is fed into neural network models. Seismic data often requires multiple layers of preprocessing, including noise reduction, attribute co-rendering, and interpolation, to ensure that geological features are clearly visible and interpretable. The challenge lies not only in removing noise but also in preserving the integrity of key features such as faults and horizons, which are critical for accurate interpretations.

Another critical problem is the limited ability of existing tools to visualize and explore data before neural network training. Effective visualization methods are essential for geophysicists and data scientists to assess the quality of preprocessing, ensuring that the data is suitable for both human interpretation and neural network tasks. This paper seeks to address these gaps by exploring modern preprocessing techniques and their role in improving the quality and interpretability of seismic datasets.

The main research question is: How can modern preprocessing and visualization techniques enhance the clarity and structure of seismic data, making it more suitable for subsequent analysis and interpretation? By evaluating methods such as noise reduction, stratal slicing, and attribute co-rendering, we aim to provide a clearer understanding of how these techniques can improve both data

quality and the overall seismic analysis workflow. The goal is not to prove improvements in neural network model performance but rather to explore how better data preparation can lead to more robust datasets that are ready for neural network applications or manual interpretation.

3. Methodology

3.1. Data description

3.1.1. Seismic data overview

In this study, we work with both 2D and 3D seismic datasets, which are commonly used for exploring subsurface geological structures, particularly in fault detection and subsurface mapping [12]. Seismic data is collected through surveys that involve generating and recording energy waves reflected from subsurface layers, creating an image of the Earth's subsurface. These images are then used to identify key features like faults, horizons, and fractures.

- 2D seismic data provides cross-sectional views of the subsurface and is often used in early exploration phases. It offers a single vertical slice of data, which is easier to analyze but may miss important details in complex structures.
- 3D seismic data, on the other hand, provides a volumetric representation of the subsurface. It offers a much more detailed and accurate depiction of geological structures, enabling geophysicists to visualize and interpret complex formations. 3D data is essential for detailed fault analysis and exploration of more intricate subsurface features, but it comes with challenges, including increased noise levels and larger data volumes that require more extensive processing.

3.1.2. Seismic data formats

Different datasets are provided in specific formats that cater to large volumes of data and ensure the integrity of the recorded information.

- SEG-Y Format[13]: One of the most common formats for seismic data is SEG-Y (Society of Exploration Geophysicists Y). This format is widely used in the industry and serves as a standard for storing seismic data collected during surveys. SEG-Y files contain a mixture of binary and ASCII data, including the recorded seismic signals (often referred to as traces) and additional metadata, such as location coordinates, acquisition parameters, and recording times.
- .vol Format: The .vol format is another format commonly used for storing 3D seismic volumes. Unlike SEG-Y, which stores individual traces, .vol files represent entire 3D seismic volumes, allowing for faster retrieval and manipulation of data when working with large, complex datasets.
- .npz Format[14]: The .npz format is a data format used for storing large, multi-dimensional arrays, which is supported by the Python library NumPy. This format is especially useful in deep learning workflows where seismic data needs to be efficiently loaded into memory for model training and analysis.

3.1.3. Open seismic data sources

For this study, several open-source seismic datasets were used, allowing for comprehensive analysis and comparison of different preprocessing techniques. Among these are:

- Opunake-3D Dataset[15]: Available at Opunake-3D, this dataset includes detailed 3D seismic images stored in the .vol format. This data is ideal for testing 3D seismic analysis and fault detection methods, particularly in regions with complex subsurface structures.
- FORCE 2020 Machine Learning Competition Dataset[16]: Hosted on the Harvard Dataverse, this dataset includes both 2D and 3D seismic images in the .npy format, specifically designed for machine learning tasks. It provides a diverse set of seismic features, making it a valuable resource for deep learning experiments in fault detection.
- Netherlands F3 Dataset[17]: The F3 dataset is widely used in seismic research and is available through platforms like TerraNubis. It includes 3D seismic data in SEG-Y format, offering a rich dataset for structural and stratigraphic interpretation.
- These datasets were selected for their diversity in format and content, allowing for the testing of various preprocessing techniques.

3.2. Data preprocessing

In seismic data processing, proper preprocessing is crucial for improving data quality, enhancing key features, and preparing the data for further interpretation or machine learning and deep learning tasks. Seismic datasets, particularly those with complex subsurface structures, often contain noise and require interpolation or specialized visualization techniques. The following preprocessing methods were applied to ensure the data is both clean and informative, allowing for clearer visualization and more accurate interpretation.

One of the central techniques used was stratal slicing, which proved invaluable in examining subsurface structures across multiple layers. Traditional vertical cross-sections often obscure important lateral variations, making it difficult to track features such as faults. Stratal slicing addresses this by cutting through seismic volumes horizontally along stratigraphic layers, enabling a more intuitive exploration of how geological formations extend across different parts of the subsurface. This method provided a clear advantage when working with 3D volumes, allowing us to isolate and study specific depositional patterns that would otherwise remain hidden.

In conjunction with stratal slicing, volume flattening [18] was applied to simplify the visualization of complex geological formations. Flattening the volume along a key horizon, this approach removed distortions caused by folding or faulting, offering a clearer view of continuous layers. With the data "flattened," even subtle stratigraphic details became more apparent, aiding in both manual interpretation and deep learning-based feature extraction.

A key addition to our preprocessing toolkit was Crude Spectral Decomposition [19], a method that allows the breakdown of seismic data into its component frequencies. This technique helps in isolating specific frequency ranges that highlight different geological features, making it easier to identify stratigraphic traps and thin beds. Spectral decomposition enriches the dataset by providing more detailed insights into subsurface structures, particularly when combined with other seismic attributes like amplitude.

To handle incomplete or irregular data, we employed various interpolation techniques. While nearest-neighbor interpolation [20] preserved sharp boundaries in regions with missing data, more refined methods such as bilinear interpolation and cubic interpolation were used to create smoother transitions in areas where the data was less complex. Bilinear interpolation proved useful for filling gaps in 2D data, whereas cubic interpolation was more effective for creating smooth, continuous surfaces in 3D seismic volumes.

Filtering and denoising were applied to mitigate the noise inherent in seismic data, ensuring that critical features like faults and horizons remained clear. This process involved several techniques, including:

- Denoising through methods like CLAHE [21] (Contrast Limited Adaptive Histogram Equalization), which enhanced contrast in regions with poor signal-to-noise ratios, making features easier to detect.

- Gaussian smoothing [22], which was used to reduce high-frequency noise while maintaining the overall structural integrity of the data.
- Outlier removal [23] to eliminate anomalous data points that could distort the seismic image.
- Spatial transformations to correct for any geometrical distortions introduced during data acquisition.

These filtering and noise-reduction techniques worked together to enhance the quality of seismic data, ensuring that deep learning models and human interpreters had access to clean, high-fidelity data.

Finally, co-rendering of multiple seismic attributes was applied to enhance the interpretability of the data. This technique involved overlaying various attributes such as amplitude and spectral decomposition to create a composite view that emphasized subtle geological features. By combining these attributes, we were able to detect features like fault zones and stratigraphic traps with greater clarity, ensuring that no critical details were overlooked during the interpretation process.

Together, these preprocessing techniques form a comprehensive approach to seismic data preparation, ensuring that the datasets are not only clean and continuous but also rich in geological detail. These steps are critical for both manual interpretation and for providing high-quality input to deep learning models.

3.3. Tools and libraries

Handling and visualizing seismic data at this scale requires a combination of specialized tools and libraries, each playing a distinct role in the preprocessing and analysis pipeline. The integration of industry-standard tools with Python-based solutions allowed for a flexible yet robust workflow, particularly suited for the challenges of seismic data.

Geoprobe, for example, served as a cornerstone for interpreting seismic volumes. This tool allowed for efficient horizon and fault picking, automating many of the tasks that would otherwise require manual intervention. The ability to quickly extract key structural information from the seismic volumes significantly accelerated the interpretation process, laying the groundwork for more in-depth analysis.

For 3D visualization, Mayavi [24] played an essential role. With its ability to render large 3D volumes interactively, Mayavi allowed us to explore seismic data in real time, adjusting parameters on the fly to better understand complex subsurface features. Its strength lies in the detailed, dynamic visualizations it produces, offering geophysicists the ability to inspect subsurface structures layer by layer and identify anomalies or points of interest with precision.

Complementing Mayavi was PyVista [25], a more Pythonic interface built on top of the VTK framework. Where Mayavi excelled in real-time visualization, PyVista provided a smoother experience for creating high-quality, static visualizations, perfect for documenting the effects of different preprocessing techniques. PyVista's integration with Python allowed seamless transitions between data manipulation and visualization, making it an indispensable part of our workflow.

When it came to handling the data itself, particularly in SEG-Y format, Obspy [26] provided the necessary tools for reading, writing, and manipulating seismic traces. This library was pivotal in extracting the necessary metadata from seismic files, preparing the data for further processing steps like volume flattening or stratal slicing. Its broad compatibility with seismic data formats made it a go-to tool for data preprocessing.

By using this suite of tools and libraries, we were able to streamline the preprocessing pipeline, enabling efficient handling of large seismic datasets. The integration of these tools into a cohesive Python-based workflow ensured flexibility and scalability, essential for both seismic interpretation and deep learning applications.

3.4. Research

3.4.1. Metrics for evaluation

In this study, several key metrics are used to assess the quality and effectiveness of our preprocessing techniques. These metrics provide a quantitative evaluation of how well each method contributes to improving the seismic data before it is fed into neural networks. The following metrics were chosen for their ability to measure both the information content and the structural integrity of the data.

Correlation: This metric measures the relationship between the original seismic data and the processed data. A lower correlation after processing can indicate that new, meaningful information has been introduced, particularly after methods like spectral decomposition. This suggests that the processed data contains additional, independent features not present in the original dataset.

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}, \quad (1)$$

where, $\text{cov}(X,Y)$ is the covariance between variables X and Y . σ_X and σ_Y are the standard deviations of X and Y , respectively.

Information Entropy: Entropy is a measure of uncertainty or information content in the data. Higher entropy values indicate that the processed data contains more distinguishable information, which can be useful for further interpretation or deep learning tasks. An increase in entropy after preprocessing suggests that new features or patterns have been extracted from the data.

$$H(X) = - \sum_{i=1}^n P(x_i) \log_b P(x_i), \quad (2)$$

where, $P(x_i)$ is the probability of occurrence of event x_i . \log_b is the logarithm to the base b (commonly base 2).

PSNR (Peak Signal-to-Noise Ratio): PSNR quantifies the similarity between the original and processed data by measuring the ratio between the maximum possible signal and the noise introduced during processing. Higher PSNR values indicate that the processed data closely resembles the original, meaning that while noise has been reduced, the integrity of the original signal has been preserved.

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right), \quad (3)$$

where, MAX_I is the maximum possible pixel value of the image. MSE is the mean squared error between the original and compressed image.

SSIM (Structural Similarity Index): SSIM evaluates the structural similarity between the original and processed data. It compares features such as luminance, contrast, and structure to determine how much the key features of the data have been preserved. A high SSIM score suggests that important geological features remain intact after preprocessing, even after noise reduction or smoothing techniques have been applied.

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (4)$$

where, μ_x and μ_y are the mean values of x and y . σ_x^2 and σ_y^2 are the variances of x and y . σ_{xy} is the covariance between x and y . C_1 and C_2 are constants to stabilize the division.

3.4.2. Spectral decomposition

Spectral decomposition is a powerful technique used to break down seismic signals into their frequency components. This method allows for a more detailed analysis of subsurface features, revealing subtle geological structures that may not be visible in the original amplitude data. By decomposing the signal into its constituent frequencies, we can isolate specific frequency bands that highlight different geological characteristics, such as thin beds or stratigraphic traps.

Amplitude and Coherence Extraction: We begin by extracting amplitude and coherence attributes from the seismic data. These attributes form the basis for spectral decomposition, as they provide insight into the seismic signal's strength and continuity across different layers.

Combining Spectral Components: After extracting the spectral components, we combine them with the amplitude and coherence data to form a more comprehensive dataset. This allows us to create new features that enhance the interpretability of the data, providing additional insights into subsurface structures.

Correlation and Entropy Evaluation: Once the spectral decomposition is complete, we evaluate the processed data using correlation and information entropy metrics. A lower correlation between the original and decomposed data indicates that new features have been introduced. An increase in entropy suggests that the decomposed data contains more distinguishable information, enhancing its value for subsequent analysis or deep learning models.

3.4.3. Interpolation

Interpolation is critical for filling in gaps in seismic data, ensuring continuity and completeness across the dataset. Seismic surveys often produce incomplete data due to technical limitations or environmental factors, and interpolation helps to mitigate these issues by reconstructing missing values.

Bilinear Interpolation: This method is used for simple regions of the seismic dataset, where a smooth transition between data points is sufficient. Bilinear interpolation calculates the value of a missing data point as the weighted average of its neighboring points, resulting in a seamless integration of the interpolated data into the original dataset.

Cubic Interpolation: For more complex regions, cubic interpolation is applied. This method provides smoother transitions between data points, making it ideal for areas with intricate geological structures. Cubic interpolation ensures that the reconstructed data points are more accurate, preserving the continuity of the geological features.

PSNR and SSIM Evaluation: After interpolation, we evaluate the data using PSNR and SSIM. High PSNR values indicate that the interpolated data closely matches the original dataset, while high SSIM values suggest that the structural integrity of the geological features has been preserved. These metrics ensure that the interpolation methods have effectively reconstructed the missing data without introducing significant distortions.

3.4.4. Filtering and noise reduction

To further enhance the quality of the seismic data, a combination of filtering and noise reduction techniques is applied. These methods are essential for removing high-frequency noise, outliers, and other artifacts that can obscure critical geological features.

Gaussian Smoothing: This technique is used to reduce high-frequency noise while maintaining the overall structure of the seismic reflections. Gaussian smoothing applies a weighted average to the data, effectively blurring out noise while preserving the key features of the subsurface.

CLAHE (Contrast Limited Adaptive Histogram Equalization): CLAHE is applied to enhance the contrast of the seismic data, particularly in regions with low signal-to-noise ratios. By improving contrast, CLAHE makes subtle features more visible, aiding in the interpretation of geological structures.

Outlier Removal: Anomalous data points, which could distort the interpretation of the seismic data, are removed to ensure that the dataset remains clean and interpretable. Outlier removal is particularly important in regions where data acquisition may have been less accurate.

Spatial Transformations: Geometrical distortions in the data, introduced during acquisition, are corrected through spatial transformations. These transformations align the data with the expected subsurface geometry, ensuring that the features are accurately represented.

PSNR and SSIM Evaluation: Following the application of these noise reduction and filtering techniques, the data is evaluated using PSNR and SSIM to ensure that the integrity of the original seismic signal has been preserved. High PSNR and SSIM scores confirm that the noise has been effectively reduced without compromising the structural features of the data.

This approach ensures that each preprocessing step contributes to the overall goal of improving data quality and interpretability. By combining spectral decomposition, interpolation, and noise reduction techniques, we can significantly enhance the seismic dataset, making it more suitable for deep learning models and geological interpretation.

4. Results and discussion

In our study, all experiments were conducted using real seismic data, ensuring the practical validation of the proposed methods. Specifically, we utilized both 2D and 3D seismic datasets. The Opunake-3D Dataset [15], stored in the .vol format, provided high-resolution 3D seismic images ideal for testing fault detection and subsurface analysis techniques, particularly in regions with complex geological structures. Additionally, the dataset includes detailed subsurface information, allowing for thorough evaluation of our preprocessing and interpretation methods on real-world data.

4.1. Spectral decomposition

In the spectral decomposition, we applied Gaussian filters with different levels (`np.linspace(1, 9, 4)` and `np.linspace(1, 15, 6)`). The correlation between color channels (R, G, B) and the average correlation were moderate, while entropy slightly increased with finer decomposition.

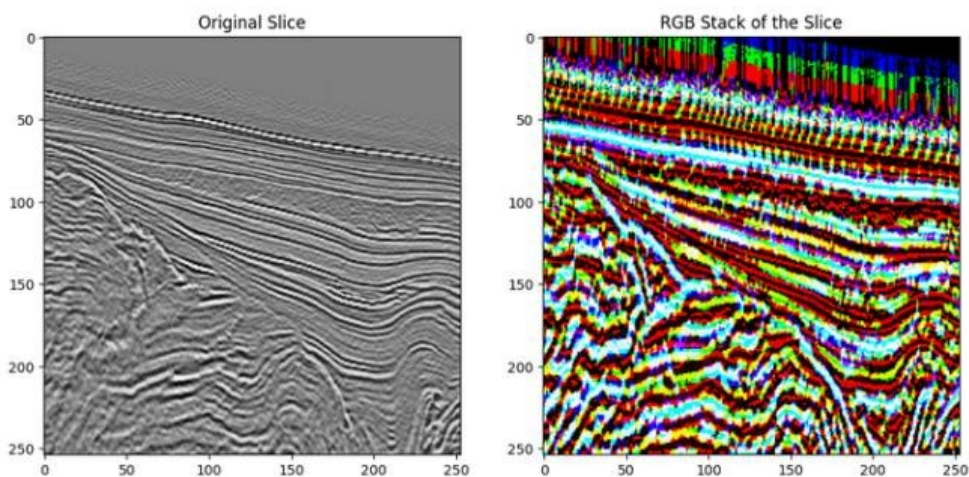


Figure 1: Original Slice and RGB Stack of the Slice after set level `linspace(1, 9, 4)`. This is the RGB spectral stack of the same seismic slice. The RGB colors represent different spectral components, highlighting additional geological features or details that are not visible in the original grayscale slice.

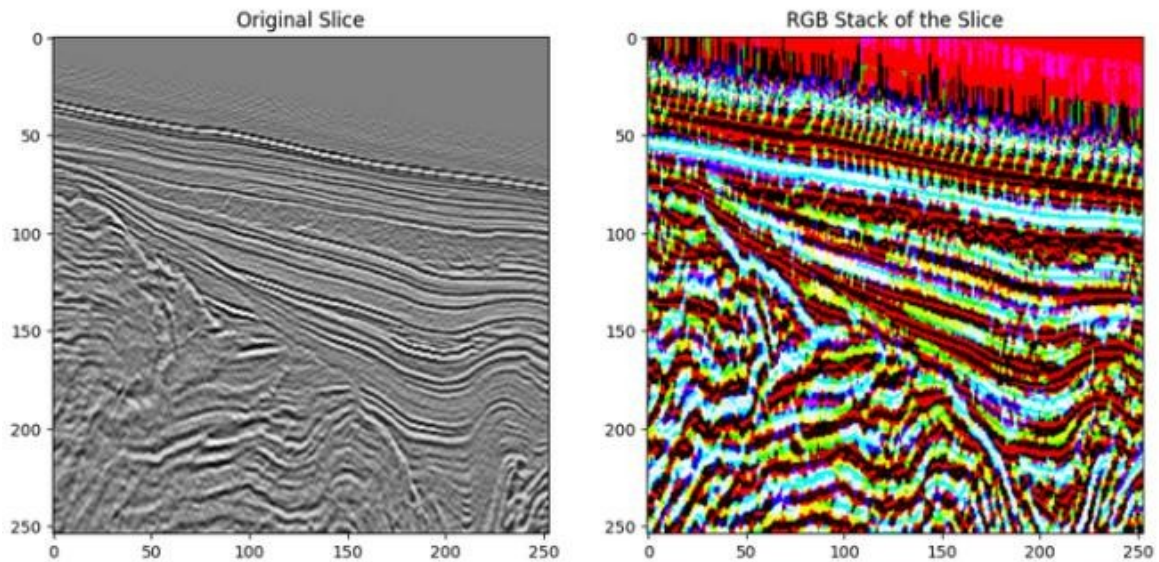


Figure 2: Original Slice and RGB Stack of the Slice after set level linspace(1, 15, 6). Another RGB spectral stack, emphasizing different spectral components from the original slice, with more vibrant colors (including red and blue) showing subtle variations in the data.

Table 1

Spectral Decomposition results

Attempt	Avg Correlation	R Correlation	G Correlation	B Correlation	Original Entropy	RGB Entropy
linspace (1, 9, 4)	0.2990	0.5746	0.2114	0.1110	11.0114	11.3138
linspace (1, 15, 6)	0.2943	0.5719	0.2012	0.1097	11.0114	11.3550

Correlation: The moderate correlation values (avg_corr = 0.2990 and avg_corr = 0.2943) indicate that spectral decomposition captures useful but noisy features.

Entropy: A slight increase in entropy (from 11.01 to 11.35) reflects added complexity in the data, but this complexity may also introduce noise.

Impact on Neural Networks: Moderate correlation and increased entropy suggest that while spectral decomposition introduces more detail, it may also make it harder for a neural network to distinguish meaningful patterns. Neural networks may benefit from this decomposition if the right balance between signal and noise is maintained, but too much complexity could overwhelm the model, making it harder to learn key features.

4.2. Co-rendered Seismic Amplitude and Coherence

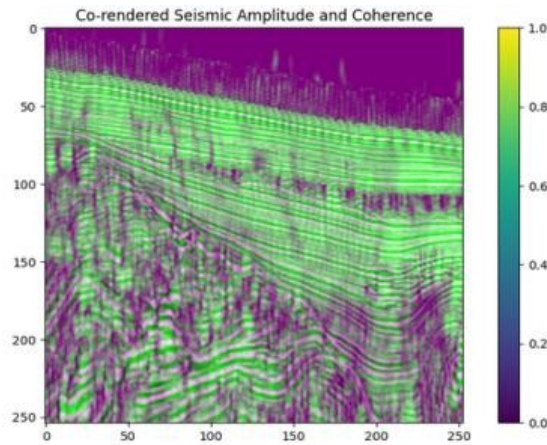


Figure 3: This image visualizes the co-rendering of seismic amplitude and coherence data. Different colors (yellow, green, purple, etc.) are used to emphasize areas where amplitude and coherence correlate, helping identify geological structures or features.

Table 2

Co-rendered Seismic Amplitude and Coherence results

Correlation	Seismic Entropy	Coherence Entropy
-0.0041	11.0002	10.8595

The combination of seismic amplitude and coherence showed a very low correlation ($\text{corr} = -0.0041$), indicating minimal linear relationship between the two. However, the entropy values for both datasets were close, suggesting that they contain comparable levels of complexity.

Impact on Neural Networks: The low correlation suggests that seismic amplitude and coherence provide complementary information, which could enhance the model's ability to learn diverse features. However, the lack of linear correlation may require the neural network to learn more complex, non-linear patterns between the datasets. The close entropy values imply that both datasets contribute a similar amount of information, which could help the model generalize better.

4.3. Interpolated seismic amplitude and coherence (bilinear and cubic)

4.3.1. Bilinear interpolation

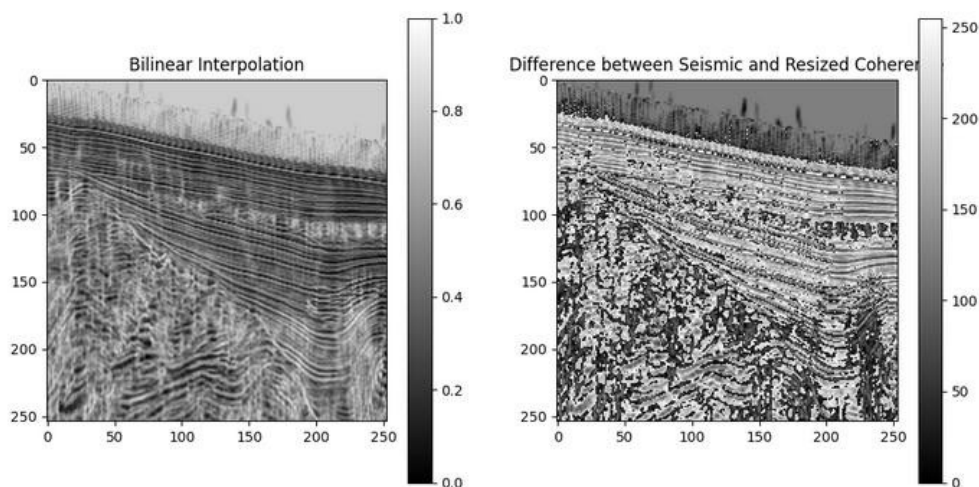


Figure 4: Cubic Interpolation and Difference between Seismic and Resized Coherence this visualizes the difference between the seismic data and resized coherence after cubic interpolation, with the grayscale showing regions of greater mismatch or error.

Table 3

Bilinear Interpolation (Seismic Amplitude and Coherence) results

PSNR	SSIM	Correlation	Seismic Entropy	Coherence Entropy
8.7239	0.0246	-0.0049	10.9792	10.8354

These values indicate that bilinear interpolation introduces considerable noise and poorly preserves the structure of the data.

Effect on Neural Networks: A low PSNR and SSIM suggest that bilinear interpolation could distort critical patterns, making it difficult for a neural network to extract meaningful features. The model may struggle to learn from this data, as the interpolation distorts the spatial relationships that are important for effective learning.

4.3.2. Cubic interpolation

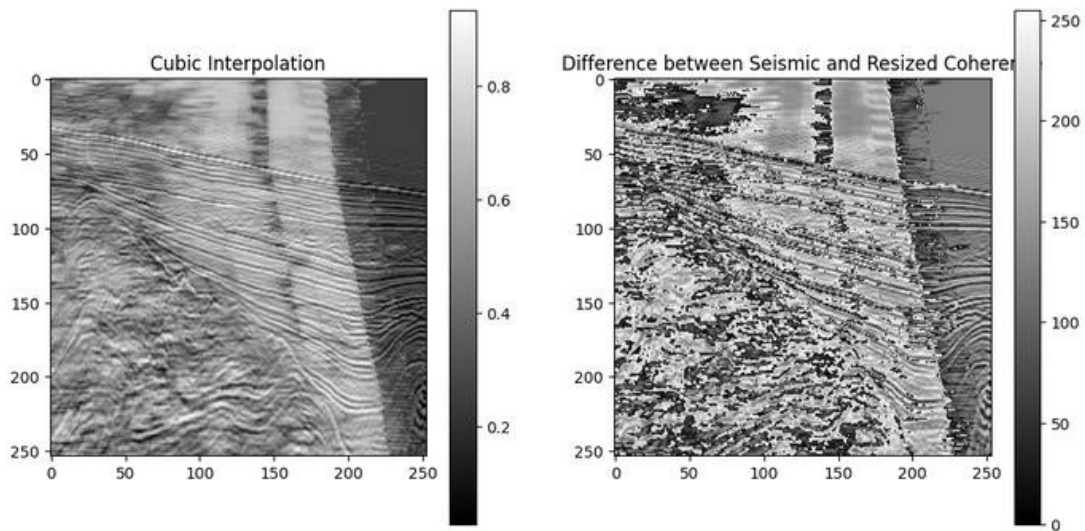


Figure 5: Bilinear Interpolation and Difference between Seismic and Resized Coherence this visualizes the difference between the seismic data and resized coherence after bilinear interpolation, with the grayscale showing regions of greater mismatch or error.

Table 4

Cubic Interpolation (Seismic Amplitude and Coherence) results

PSNR	SSIM	Correlation	Seismic Entropy	Coherence Entropy
8.6663	0.0379	0.0033	10.9962	10.8551

Cubic interpolation marginally improves structural similarity over bilinear interpolation, but the overall noise remains high.

Effect on Neural Networks: Cubic interpolation is slightly better at preserving structure, but the network may still face challenges learning from this data due to the introduced noise. Although it retains more information than bilinear interpolation, the distortion could negatively impact model performance, especially for tasks requiring fine detail recognition.

4.4. Denoising and Filtering

Table 5
Denoising with Median Filtering results

Attempt	Filter Size	PSNR	SSIM
1	3	19.13	0.7096
2	5	14.81	0.2579
3	7	14.82	0.1837
4	9	14.88	0.1644
5	11	14.87	0.1564

Table 6
Gaussian Smoothing results

Attempt	Sigma	PSNR	SSIM
1	1	18.28	0.6346
2	2	15.66	0.2876
3	3	15.19	0.1988
4	4	15.05	0.1750
5	5	15.01	0.1677

Table 7
Outlier Removal results

Attempt	Percentiles	PSNR	SSIM
1	1-99	44.80	0.9998
2	2-98	42.24	0.9995
3	3-97	39.81	0.9992
4	4-96	38.09	0.9988
5	5-95	36.48	0.9981

Table 8
Spatial Transformations (Rotation) results

Attempt	Rotation Angle	PSNR	SSIM
1	15°	10.71	0.0848
2	30°	9.94	0.0656
3	45°	9.70	0.0468
4	60°	9.92	0.0426
5	75°	10.69	0.0387

Table 9

Denoising with Median Filtering results

Attempt	Clip Limit	Grid Size	PSNR	SSIM
1	2.0	(8, 8)	19.32	0.8832
2	3.0	(8, 8)	18.03	0.8474
3	4.0	(8, 8)	17.47	0.8318
4	2.0	(16, 16)	18.95	0.8755
5	3.0	(16, 16)	18.39	0.8470

Among the different methods, CLAHE stands out as the most balanced for enhancing image quality, with a PSNR of 19.32 and SSIM of 0.8832 at Clip 2.0, Grid (8, 8). It effectively enhances contrast while preserving structure, making it ideal for neural network training, as it makes features more distinguishable without introducing artifacts. Outlier removal produces the highest PSNR (44.80) and SSIM (0.9998), significantly improving signal clarity by removing extreme values. This method ensures that the neural network learns from clean, high-quality data, enhancing its ability to generalize. For denoising, a small median filter (size 3) performs well, removing noise while retaining important details (PSNR = 19.13, SSIM = 0.7096). Larger filters, however, lead to over-smoothing, reducing the network's ability to capture key features. Gaussian smoothing with a small sigma also helps reduce noise without losing too many details, but higher sigma values risk blurring important patterns.

CLAHE proved especially effective in improving contrast and detail visibility, allowing for better feature recognition by the model. However, excessive filtering or contrast enhancement could lead to over-smoothing or artifacts, reducing model performance.

In summary, CLAHE offers the best balance for feature enhancement, while outlier removal is ideal for improving signal clarity, making both methods excellent for neural network preprocessing.

4.5. Summary of results and limitations

The preprocessing techniques we applied, such as spectral decomposition, co-rendering, and denoising, significantly improved the quality of the input data, making it more suitable for neural network training. For example, spectral decomposition results showed moderate correlation values (average 0.2990 and 0.2943), indicating useful but noisy features, while entropy increased slightly, from 11.01 to 11.35, reflecting added complexity. Co-rendering of seismic amplitude and coherence also revealed complementary information, with a very low correlation (-0.0041) but similar entropy levels (around 11), which could enhance the neural network's ability to learn diverse features.

One of the main benefits of these methods is the enhancement of important geological features, making it easier for neural networks to identify patterns and detect anomalies. Techniques like spectral decomposition and co-rendering allowed us to extract complementary information from the data, improving the richness of the input. Denoising and filtering steps, such as median filtering with a PSNR of 19.13 and SSIM of 0.7096, further reduced noise, ensuring that the network learns from cleaner, more consistent data.

However, these improvements come with some challenges. The increased complexity of the data, especially after adding new features through spectral decomposition, means that training neural networks will likely take longer. More computational resources are needed, and the network might require a more sophisticated architecture to handle the complexity. There is also the risk that with too much added detail, the model could struggle to generalize and may overfit to the noise or irrelevant features.

In summary, while our preprocessing methods greatly enhance data quality and can improve neural network performance, they also introduce challenges like longer training times and increased complexity, which require careful consideration when applying these techniques in practice.

5. Conclusion

This study has demonstrated the effectiveness of various preprocessing techniques applied to seismic data in improving data quality and preparing it for neural network-based analysis. By utilizing methods such as spectral decomposition, interpolation, denoising, and filtering, we were able to enhance the clarity and structure of the data while preserving critical geological features.

Spectral decomposition successfully added new features by breaking down the seismic signal into its frequency components, although it also introduced some noise. The balance between added complexity and potential noise is crucial, as it provides more detail but requires careful management to avoid overwhelming neural networks.

Interpolation methods, including bilinear and cubic interpolation, provided continuity in areas with missing data, although cubic interpolation was slightly better at preserving structural integrity. Nonetheless, noise introduced during interpolation remains a challenge that could impact neural network performance, particularly in detailed tasks.

Denoising and filtering techniques such as CLAHE and median filtering played a critical role in improving the quality of the data by reducing noise and enhancing contrast. These methods preserved key features necessary for both human interpretation and deep learning tasks, ensuring that seismic data retains its structural integrity throughout the preprocessing pipeline.

Finally, spatial transformations and outlier removal helped further refine the data, removing artifacts and extreme values that could have negatively impacted the learning process. These techniques not only improved data quality but also ensured that neural networks receive clean, well-structured input.

In conclusion, the preprocessing techniques discussed in this paper are integral to enhancing seismic data, making it more interpretable and suitable for deep learning applications. These methods offer a robust foundation for improving the quality of seismic datasets and facilitating more accurate subsurface geological interpretations.

Acknowledgements

The work was performed according to the grant funding of the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan, Grant number: AP23489938.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] Zeng, H. (2010). Stratal slicing: Benefits and challenges. *The Leading Edge*, 29(9), 1040-1047. <https://doi.org/10.1190/1.3485764>.
- [2] Chopra, S., & Marfurt, K. (2011). Volume co-rendering of seismic attributes — A great aid to seismic interpretation. *SEG Technical Program Expanded Abstracts 2011*, 30(1), 1150-1154. <https://doi.org/10.1190/1.3627406>.
- [3] Chen, Y. A. (2010). An improved digital image interpolation algorithm. *2010 Second International Conference on Multimedia and Information Technology*, 183-186. <https://doi.org/10.1109/MMIT.2010.141>.
- [4] Meirmanov A.M., Nurtas M. (2016). Mathematical models of seismic in composite media: elastic and poroelastic components. *Electronic Journal of Differential Equations*. 184, 1-22. EID: 2-s2.0-84978512564, ISBN: 10726691.
- [5] Meirmanov A.M., Mukhambethzanov, S., Nurtas M. (2016)_Seismic in composite media: elastic and poroelastic components. *Siberian Electronic Mathematical Reports*. 13. 75 - 88. DOI: 10.17377/semi.2016.13.006.

- [6] Nurtas M., Baishemirov Z. (2020). Numerical simulation of the acoustic wave equation by the method of reverse time migration. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(5), 8223-8227. <https://doi.org/10.30534/ijatcse/2020/189952020>.
- [7] Nurtas M, Ydyrys A, Altaibek A. (2020). Using of Machine Learning algorithm and Spectral method for simulation of Nonlinear Wave Equation. *ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020*, 43, 1-6. <https://doi.org/10.1145/3410352.3410778>.
- [8] Nurtas M, Baishemirov Zh, Ydyrys A, Altaibek A. (2020). 2-D Finite Element method using "eScript" for acoustic wave propagation. *ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020*, 40, 1-7. <https://doi.org/10.1145/3410352.3410774>.
- [9] Moreno, M., Santos, R., Mozart, R., Santos, W., & Cerqueira, R. (2018). Assisting seismic image interpretations with Hyperknowledge. In *2018 First International Conference on Artificial Intelligence for Industries (AI4I)* (pp. 48-51). IEEE. <https://doi.org/10.1109/AI4I.2018.8665691>.
- [10] Wu, X., Luo, S., & Hale, D. (2016). Moving faults while unfaulting 3D seismic images. *Geophysics*, 81(2), MA1-MA17. <https://doi.org/10.1190/geo2015-0381.1>
- [11] Wu, X., & Hale, D. (2016). Automatically interpreting all faults, unconformities, and horizons from 3D seismic images. *Interpretation*, 4(2), T260-T277. <https://doi.org/10.1190/INT-2015-0160.1>.
- [12] Yilmaz, Ö. (2001). *Seismic data analysis: Processing, inversion, and interpretation of seismic data* (Vol. 1). Society of Exploration Geophysicists. <https://doi.org/10.1190/1.9781560801580>.
- [13] Aziz, I. A., Mazelan, N. A., Samiha, N., & Mehat, M. (2008). 3-D seismic visualization using SEG-Y data format. In *2008 International Symposium on Information Technology* (pp. 1-7). IEEE. <https://doi.org/10.1109/ITSIM.2008.4631705>.
- [14] NumPy Developers. (n.d.). *numpy.lib.format — NumPy v1.22.dev0 Manual*. NumPy. Retrieved September 8, 2024, from <https://numpy.org/devdocs/reference/generated/numpy.lib.format.html>.
- [15] Society of Exploration Geophysicists. (n.d.). *Opunake-3D Dataset*. SEG Wiki. Retrieved from <https://wiki.seg.org/wiki/Opunake-3D>.
- [16] FORCE. (2020). *FORCE 2020 Machine Learning Competition Dataset*. Harvard Dataverse. <https://doi.org/10.7910/DVN/2020>.
- [17] dGB Earth Sciences. (n.d.). *Netherlands Offshore F3 Block*. Open Seismic Repository. Retrieved from <https://opendtect.org/osr/>.
- [18] Wu, X., Li, Y., & Sawasdee, P. (2022). Toward accurate seismic flattening: Methods and applications. *Geophysics*, 87(5), 1SO-V558. <https://doi.org/10.1190/geo2021-0662.1>.
- [19] Mahadik, R., Singh, G., & Routray, A. (2022). Multispectral coherence analysis for better fault visualization in seismic data. *IEEE Geoscience and Remote Sensing Letters*, 19, 1-5, Art no. 5000905. <https://doi.org/10.1109/LGRS.2021.3076213>.
- [20] Shengda, C. (2012). The comparison and analysis of several magnification of image magnification. *Jinhua Polytechnic Journal*, 12(3), 66-70.
- [21] Shome, S., & Mandal, B. (2019). Multidimensional Contrast Limited Adaptive Histogram Equalization. *arXiv*. <https://arxiv.org/abs/1906.11355>.
- [22] Zhao, T., & Huang, J. (2009). Structure-oriented Gaussian filter for seismic detail preserving smoothing. *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 601-604. <https://doi.org/10.1109/ICIP.2009.5654>.
- [23] Zhao, T., Zhang, G., & Chen, Y. (2018). An efficient outlier removal method for scattered point cloud data. *PLOS ONE*, 13(7), e0201280. <https://doi.org/10.1371/journal.pone.0201280>.
- [24] Ramachandran, P., & Varoquaux, G. (2011). Mayavi: 3D Visualization of Scientific Data. *Computing in Science & Engineering*, 13(2), 40-51. <https://doi.org/10.1109/MCSE.2011.35>.
- [25] Sullivan, C., & Kaszynski, A. (2019). PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). *Journal of Open Source Software*, 4(37), 1450. <https://doi.org/10.21105/joss.01450>.
- [26] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4). Springer. https://doi.org/10.1007/978-3-642-00296-0_5.