Hindawi
International Journal of Energy Research
Volume 2024, Article ID 5576859, 25 pages
https://doi.org/10.1155/2024/5576859

Research Article

Residual Convolutional Neural Network for Lithology Classification: A Case Study of an Iranian Gas Field

Seyed Hamid Reza Mousavi and Seyed Mojtaba Hosseini-Nasab

School of Chemical, Petroleum and Gas Engineering, Iran University of Science and Technology, Tehran, Iran

Correspondence should be addressed to Seyed Mojtaba Hosseini-Nasab; hosseininasab@iust.ac.ir

Received 8 November 2023; Revised 13 March 2024; Accepted 18 March 2024; Published 17 April 2024

Academic Editor: Kathiravan Srinivasan

Gas reservoir development and the estimation of rock properties heavily rely on lithology classification, which can be difficult, time-consuming, and prone to errors. In this study, a novel deep learning-based approach has been developed for the rapid, accurate, and efficient prediction of lithology in a gas field from conventional well-log data. The well-logs, referred to as numerical well-logs (NWLs), are transformed into two-dimensional images through two proposed approaches: shallow images (SIs) and deep images (DIs). In these images, the pixels effectively represent the relationships between different logs. For this purpose, we developed residual convolutional neural networks (ResCNN) named SIs-ResCNN 2D and DIs-ResCNN 2D. The feed data for DIs-ResCNN 2D are images created and referred to as DIs, which are initially formed from a vector in which the order of logs is somehow repeated, ensuring that each pairwise combination occurs only once. This resulted in the incorporation of the connection between the logs within the pixels of the generated images, alongside the integration of unique binary combinations of the logs. We compared the proposed models, including DIs-ResCNN 2D, DIs-ResCNN 2D, and NWLs-ResCNN 1D with baseline methods such as random forest (RF), K-nearest neighbor (KNN), and support vector machine (SVM). Based on the evaluation metrics, DIs-ResCNN 2D outperformed the other proposed and baseline methods on the test dataset. A balanced DIs-ResCNN 2D model achieved 93% accuracy and F1-score of 80% on a test well, highlighting the importance of data balancing during CNN model training.

1. Introduction

Identifying the sedimentary rock that forms a reservoir is vital in the oil and gas industry, since dolomite and carbonate contain more petroleum resources. Lithological analyses can be used to estimate resources based on a variety of petrophysical characteristics. The classification of lithology has traditionally been accomplished by analyzing well-logs, which can be challenging, time-consuming, and difficult [1–3]. Reliable lithology interpretation in mining plays a crucial role in the identification and characterization of geological formations, thereby aiding in exploration and mineral extraction processes. This enables targeted exploration efforts, leading to increased efficiency in recognizing areas with high mineral content. An accurate evaluation of lithology assists in well-informed decision-making for mining operations, ultimately optimizing the extraction of valuable resources [4–6].

The need to develop classification models of lithology in the oil and gas industry is felt for many reasons, especially that (i) correct and quick diagnosis of lithology plays an important role in economic issues due to the high daily cost of the oil and gas industry and reduces costs and time. It also aids petroleum engineers in making better decisions on issues such as well perforation location selection, which leads to maximum and optimal production. (ii) The correct classification of lithology leads to a more accurate determination of reservoir petrophysical properties such as porosity, permeability, and saturation. A correct assessment of these properties is vital because they have a direct impact on reservoir productivity, development, and production [4, 7, 8].

There are three common approaches for classifying facies using artificial intelligence:

(1) Images extracted from geological core plugs containing valuable and rich geological information are utilized. Various studies have employed computer vision and deep learning (DL) techniques to discern significant features and patterns within images. For example, according to Faria et al. [9], a multilayer feed-forward neural network was used to determine lithology by taking core images as input to the network and also for determining model hyperparameters. Bayesian optimization was employed, resulting in an accuracy of 83% on the test data. In other studies [10, 11], machine vision has been used for facies classification. However, due to the high cost of core sampling in the oil and gas industry, collecting images from the entire wellbore column is not economical, which is considered a limitation of this approach

(2) Seismic data, which are measurements of ground vibrations generated by seismic sources and recorded by detectors, serve as features for the classification process. Chaki et al. [12] utilize five attributes to classify different types of rock formations and introduce a probabilistic neural network (PNN) framework. The PNN framework is considered effective due to its ability to handle scenarios and process data quickly. By analyzing well-log data, the researchers categorize lithology into four classes and evaluate their findings using seismic data collected from an Indian hydrocarbon field. The seismic-based classification has also been developed [13] to identify lithofacies, where hidden Markov models (HMMs) are utilized for seismic-based lithofacies classification. HMMs consider conditional probabilities and vertical transitions, yielding accurate predictions for most lithologies, including thin layers. Comparisons with independent methods highlight the spatial correlation advantage of HMMs, while a real case study reveals the need for further research on lateral geological relationships

(3) Conventional well-logs collected after the completion of drilling are often used in the oil and gas industry in various fields to relate these logs to dependent variables. By analyzing well-logs, gamma ray, resistivity, and sonic experts can gain insights into reservoir properties and formation characteristics. He et al.'s [14] work centers on improving reservoir lithology identification using well-logging data. They introduce LSTM-FCN, a hybrid model that outperforms conventional methods, and enhance it with particle swarm optimization. Also, these conventional well reports are used in other fields such as predicting porosity [15], permeability [16, 17], and water saturation [18]. Among the methods discussed, utilizing well-logs proves to be a more practical and cost-effective approach in the oil and gas industry compared to relying on seismic data and core images

Machine learning (ML), one of the subbranches of artificial intelligence and data science, involves the possibility of training without the need for manual programming and the definition of precise rules. In Zhang et al. [19], multiple machine learning models were evaluated to automatically interpret well-log data, employing techniques like cross-validation and Bayesian optimization to fine-tune hyperparameters. The outcomes indicated a preference for ensemble methods, specifically XGBoost and random forest (RF) standing out due to the highest overall accuracy of 0.882 and AUC of 0.947 in the classification of lithologies. In Deng et al. [20], support vector machines (SVM) were improved using synthetic data techniques (SMOTE and Borderline-SMOTE) to tackle imbalanced data, surpassing neural networks in predicting lithology. In Yin et al. [21], a Class-Rebalancing Self-Training (CReST) framework is presented, leveraging well-logging data and limited labels to attain resilient lithology classification. Four high-performing algorithms, namely, Bagging, Extra Trees, RF, and SVM Classifier are selected from 25 options for constructing CReST models. CReST is shown through experimental results to effectively handle challenges associated with label scarcity and class imbalance, leading to a considerable increase in accuracy, especially for categories with limited data samples. Also, other studies have been done including XGBoost [22, 23], SVM [24], and RF [25, 26]. However, ML in complex and difficult problems usually has weaker performance than deep learning. The main reason behind this is the ability of deep learning models to capture complex patterns and dependencies in the data by leveraging the hierarchical structure of multiple layers.

DL has been more successful in extracting complex relationships in various fields, especially when dealing with big data. Imamverdiyev and Sukhostat [27] aim to develop an effective DL model for classifying geological facies in wells. They propose a new 1D-CNN model trained on various optimization algorithms using input data such as photoelectric effect, gamma ray, resistivity logging, and more. Jiang et al. [28] address the challenge of lithology identification in reservoir characterization. They observe three key characteristics in previous studies: the limited consideration of stratigraphic sequence information, the neglect of neighboring formation influence, and the lack of publicly available well-log data for comparison. The experiments demonstrate that the inclusion of geologic constraints significantly improves model performance, with RNN-based networks exhibiting better and more consistent results. Some other DL studies for rock facies classification using conventional well-log data include 1D-CNN [29–31] and RNN [32–34]. These studies have employed one-dimensional layers to learn the model. However, in this study, a unique approach has been taken to process numerical well-logs (NWLs). Instead of analyzing the numerical data directly, these well-logs have been transformed into image data. This transformation involves representing the numerical values as pixels

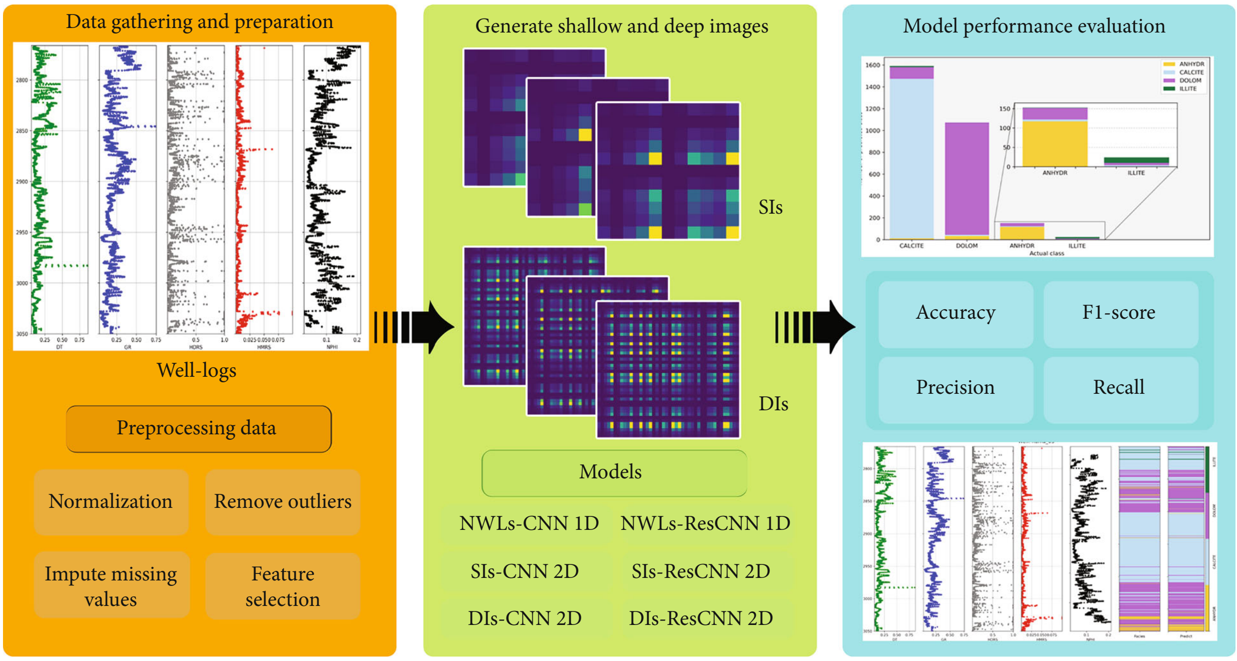


FIGURE 1: The visual process includes important steps, beginning with collecting and organizing the data. Afterward, images are generated from well-logs, and then, the models are trained. Lastly, the performance of these models is assessed.

in an image grid, effectively converting numerical information into visual data.

The core of the workflow design in this research is the CRoss Industry Standard Process for Data Mining (CRISP-DM), which is a methodology utilized for conducting data mining across various industries. This process encompasses six phases including understanding the business background, preparing the data, evaluating performance, and ultimately implementing them [35–37]. The utilization of the CRISP-DM methodology in rock classification brings several benefits. It aids researchers in ensuring the reliability and repeatability of their classification models, allowing for more accurate and consistent results. Additionally, the emphasis on cost-effectiveness and faster project completion enhances the efficiency of rock classification projects, making them more practical and feasible.

The reviewed studies failed to explore the relationships between different features. This study introduces a novel method aimed at incorporating these relationships into image generation for individual samples. Specifically, the approach integrates these feature relationships into the pixels of shallow images (SIs) and deep images (DIs). Two-dimensional images, namely, SIs and DIs, are generated using NWLs. These resultant images serve as inputs for the SIs-ResCNN 2D and DIs-ResCNN 2D models, enhancing the classification accuracy of lithology. This improvement is achieved through the utilization of residual blocks with regular and reduced shortcuts. The proposed methods of generating images offer notable benefits, including faster learning and better performance. The research highlights the significance of employing log-derived images with physical meaning in training CNN models. Additionally, it demonstrates the vital contribution of modified architectures, such as residual architectures to improving the accuracy and overall performance of the model.

2. Study Pipeline and Materials

As shown in Figure 1, this paper can be divided into three general parts: (i) data collection and preparation, (ii) image creation methods (SIs and DIs) and modified convolutional neural network (CNN) architecture, and (iii) classification evaluation. The continuation of this section focuses on data collection and preparation, where the data and its statistical properties are explored in Section 2.1. Additionally, the preprocessing steps applied to the data are described in Section 2.2.

Moving on to the *methodology*, the creation of SIs and DIs is explored in Section 3.1. Section 3.2 covers the residual structure, while Section 3.3 presents the suggested modified CNN architecture. Section 4 of the study addresses classification evaluation methods. The results are presented and analyzed in Section 5. Finally, Section 6 concludes the paper.

2.1. Dataset. The gas field under investigation is located near Asaluyeh, Tabnak, Varavi, and Shanoul in Iran. The estimated recoverable gas reserves in the field amount to 200 billion cubic meters (BCM) [38]. The field's specific location is indicated by marker X in Figure 2. Geological data were obtained from well-logs, core samples, and petrophysical charts. Logging data were recorded at 0.1-meter intervals within the wells. The geological formations within the gas field exhibit distinct layers. The Aghar shale section is characterized by reddish-brown shale interspersed with dolomite layers. The upper part of the Dalan formation consists of brownish limestone and cream-colored dolomite, occasionally accompanied by traces of anhydrite. In the lower segment of the Dalan formation, white limestone predominates with light brown dolomite and small amounts of anhydrite. Transitioning

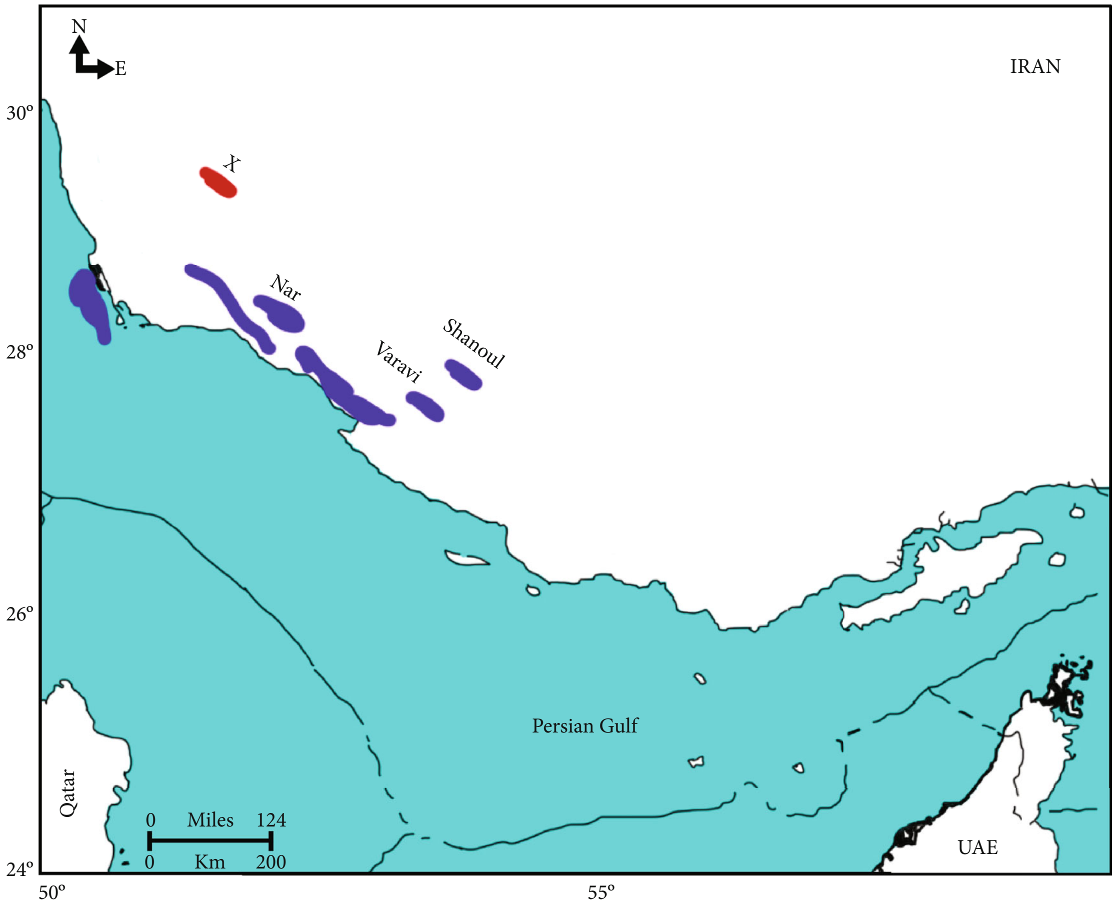


FIGURE 2: The gas field under study is located near Asaluyeh, Tabnak, Varavi, and Shanoul in Iran. This gas field was discovered in August 2000.

to the Nahr section, the composition shifts to light brown to light grayish-brown dolomite and anhydrite. The lithology composition of the different layers of the reservoir has a constant and uniform distribution along the layer and in the extent of the reservoir. For example, the alternation of anhydrite and dolomite layers throughout the Dashtak formation is constant. Only in one well due to a fault or folding has the thickness of the Dashtak formation increased. The lithology of the Kangan formation is also constant throughout the reservoir and consists of limestone.

Evaluatable information was found in 9 of the 14 production wells in this field, resulting in 44,521 recorded depths which include well-logs. Among these 9 wells, one well containing all four types of rock facies was identified as a test well and excluded from the training stage. The data from the remaining 8 wells was selected as training data for learning the model. The training dataset consisted of four rock facies, including ANHYDR (anhydrite), CALCITE (calcite), DOLOM (dolomite), and ILLITE (illite) with 9,643, 15,923, 15,071, and 1,043 instances, respectively. The test dataset consists of all four rock facies, and the test-to-train dataset ratio is six percent.

The collection of data comprises nine specifically chosen well-log characteristics: CGR (capture gamma ray), DFL (deep induction resistivity log), DT (sonic log), GR (gamma ray), HDRS (deep resistivity), HMRS (medium resistivity), NPHI (neutron porosity), PEF (photoelectric effect), and RHOB (bulk density). The calculation of total porosity (PHIT, ϕ_t) and effective porosity (PHIE, ϕ_e) from well-logs was carried out through the utilization of mathematical equations. The formulas for calculating PHIE and PHIT are as follows:

$$\phi_t = \frac{(\rho_{\text{ma}} - \rho_b)}{\left(\rho_{\text{ma}} - \rho_f\right)}, \tag{1}$$

$$\phi_e = \phi_t \times (1 - S_w) = \phi_t \times (1 - V_{\text{sh}}). \tag{2}$$

where ρ_{ma} is matrix density, ρ_b is bulk density, ρ_f is the density of the fluid in the pore space, S_w is water saturation, and V_{sh} is shale volume.

A comprehensive explanation of the well-logging data and calculated porosity (ϕ_t and ϕ_e) is presented in Table 1.

These statistics offer valuable insights into the distribution and range of the data. For example, the mean CGR is 18.14 with a standard deviation of 19.63, indicating a moderate spread of values around the mean. Similarly, other variables exhibit varying levels of dispersion as captured by their respective standard deviations. These statistical measures provide a comprehensive summary of the input well-log data enabling a detailed analysis and interpretation of

Table 1: The description of input well-log data and statistical properties of original data including mean, standard deviation, minimum, 25th/50th/75th percentile, and maximum.

No.	Variable	Mean	std	Min	25%	50%	75%	Max
1	CGR	18.14	19.63	-7.25	6.40	12.45	20.84	150.65
2	DFL	373.56	1055.87	0.00	45.75	115.77	290.60	20000.00
3	DT	58.76	14.93	35.77	50.70	53.61	60.36	214.88
4	GR	31.52	21.12	1.10	18.19	26.16	38.89	169.62
5	HDRS	553.99	777.66	0.00	26.51	104.21	853.05	2000.00
6	HMRS	306.84	646.24	0.26	23.38	64.59	176.22	20000.00
7	NPHI	0.07	0.09	-0.04	0.02	0.04	0.09	1.10
8	PEF	4.34	0.81	0.00	3.80	4.42	4.87	11.34
9	PHIT	0.06	0.06	0.00	0.02	0.03	0.07	0.30
10	PHIE	0.03	0.04	0.00	0.00	0.01	0.05	0.22
11	RHOB	2.72	0.21	0.96	2.66	2.73	2.84	3.23

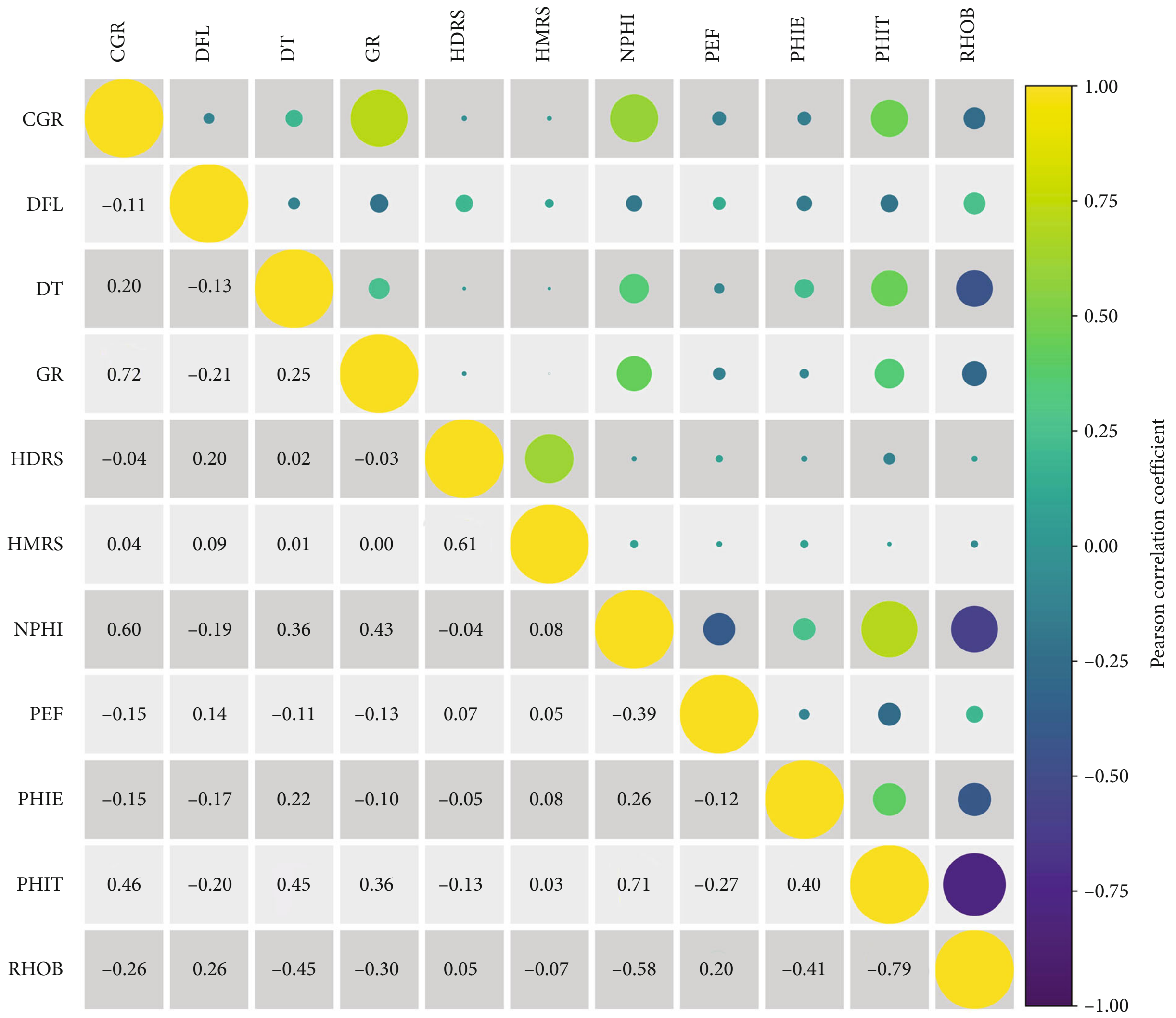


Figure 3: The heat map of the correlation matrix displays Pearson's correlation coefficient values for the examined parameters. The size of the circles and the color yellow to purple show the amount of positive and negative correlation.

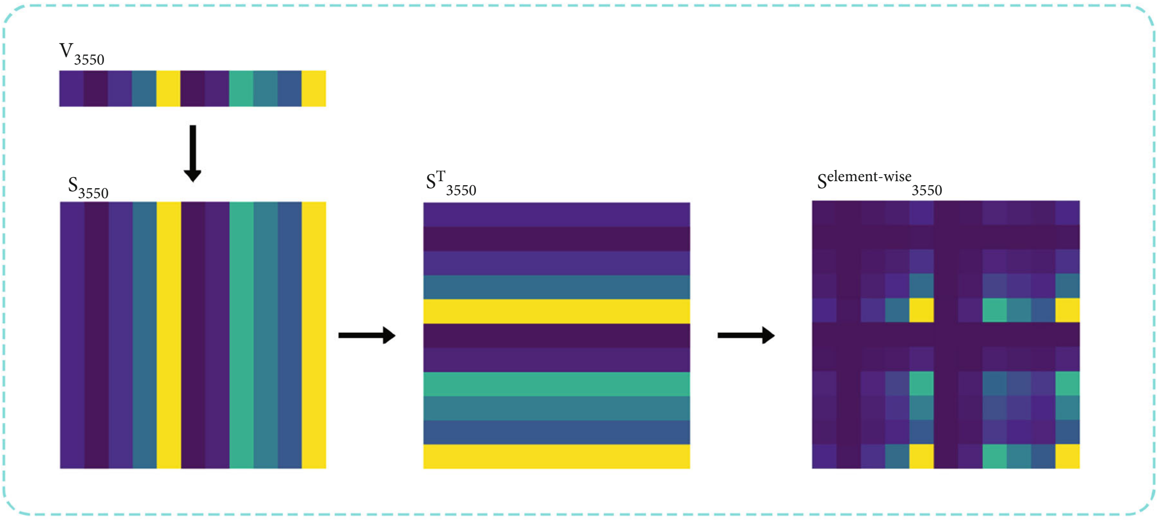


Figure 4: This image illustrates the process of generating shallow images. Initially, the transformation from a vector to a matrix occurs, followed by the creation of shallow images through element-wise multiplication of arrays. This is the shallow image for a depth of 3550 training data where the well-logs have been converted into an image.

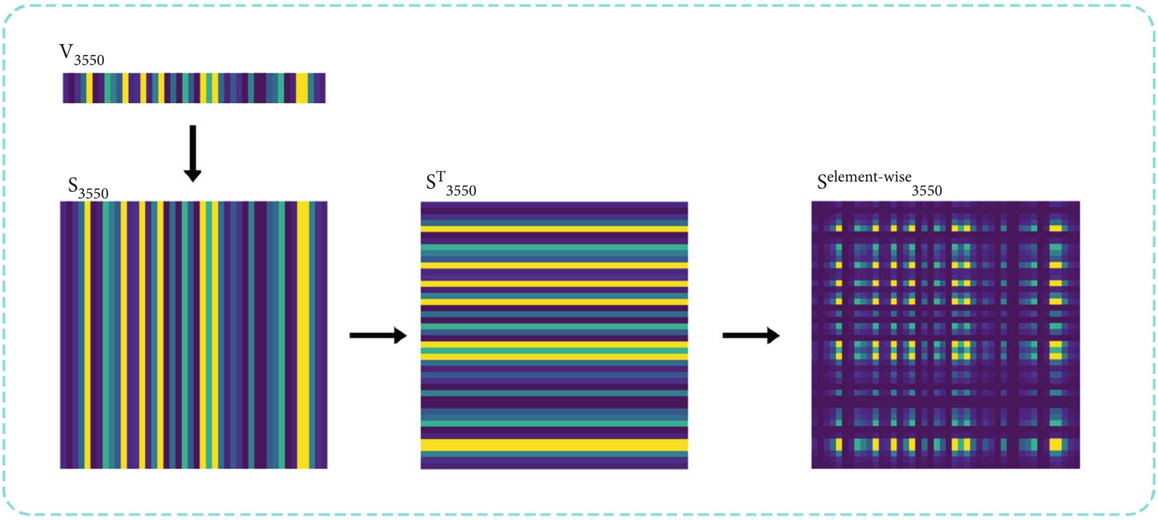


Figure 5: The process of generating deep images encompasses multiple steps. First, a vector is generated utilizing the described algorithm. Subsequently, this vector undergoes conversion into a matrix, and deep images are produced by performing element-wise multiplication of the arrays.

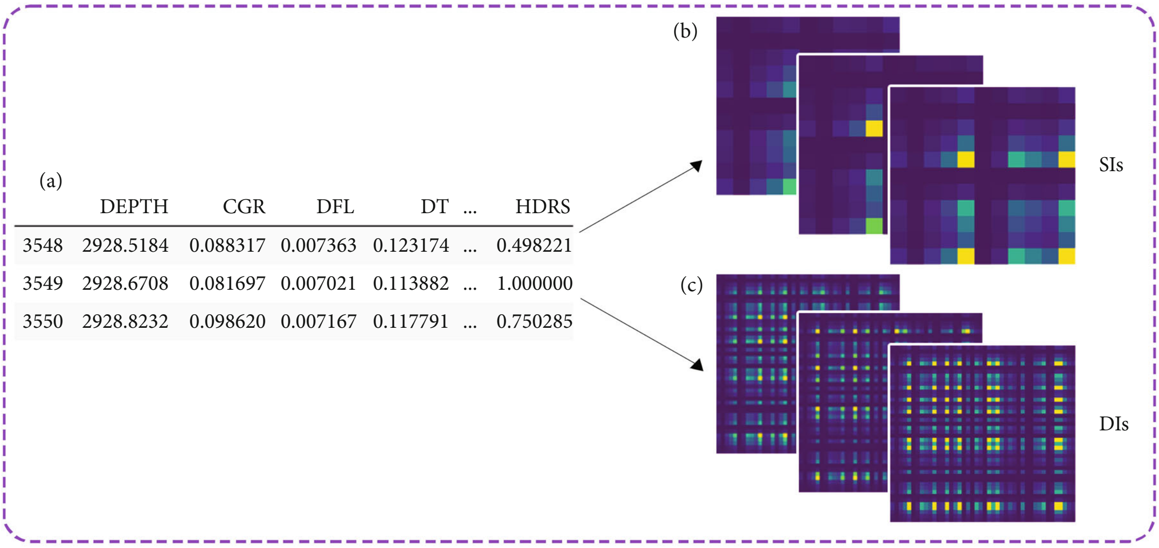
the dataset. Moreover, a visual representation in the form of a heat map correlation plot enhances our understanding of the relationships between different variables. The correlations between the features are shown in Figure 3.

The positive Pearson correlation coefficients indicate strong linear relationships between variables, enhancing the predictive power of measurements. For instance, the positive correlation between GR and CGR (0.72) and NPHI and CGR (0.60) suggests a strong association between these parameters. Conversely, negative correlations, such as RHOB and PHIT (-0.79) and PEF and NPHI (-0.39), signify inverse relationships. These correlations play a crucial role in understanding subsurface conditions and optimizing decision-making in resource exploration.

2.2. Preprocessing. Preprocessing is pivotal for ensuring the quality and reliability of the collected NWLs dataset. A series of steps is performed to prepare the data for analysis; each step is described briefly:

(1) The dataset is carefully examined for missing values. To address this issue, imputation is used, where missing values are replaced with the corresponding feature's value. This preserves the overall structure of the data while minimizing the impact of missing values on the analysis

(2) Normalization techniques are applied to standardize the scales of all features within the dataset, ensuring



	DEPTH	CGR	DFL	DT	...	HDRS
3548	2928.5184	0.088317	0.007363	0.123174	...	0.498221
3549	2928.6708	0.081697	0.007021	0.113882	...	1.000000
3550	2928.8232	0.098620	0.007167	0.117791	...	0.750285

Figure 6: Schematic of creating two-dimensional images in order to use a convolutional neural network. (a) Data shows a table that contains logs recorded for different depths. (b) SIs are created using eleven existing well-logs (each represented as a vector with 11 dimensions). (c) DIs are created using the described algorithm resulting in a vector with 44 dimensions.

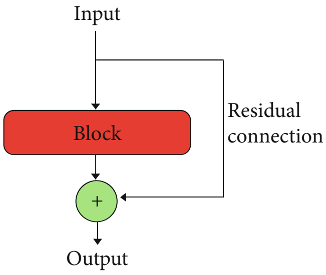


Figure 7: In neural network, a residual connection is a technique that enhances the flow of information during training by combining the original input with the output of a processing block.

equal contribution to the analysis. The chosen method, StandardScaler, adjusts each feature's values to have an average of 0 and a standard deviation of 1. This prevents any single feature from dominating the analysis due to its magnitude

(3) The local outlier factor (LOF) method was selected for data processing due to its higher test accuracy of 87% compared to other methods, such as isolation forest and elliptic envelope based on the K-nearest neighbor (KNN) base model. LOF evaluates the clustering pattern of data points and identifies outliers based on their deviation from this pattern. By eliminating outliers, the analysis is less influenced by extreme values, thereby enhancing the model's reliability. The removal of outliers from the training data resulted in the exclusion of 4168 samples

With the completion of these preprocessing steps, the NWL dataset is now primed for analysis. Missing values have been addressed, the data has been normalized using StandardScaler, and outliers have been identified and removed using LOF. Consequently, the dataset's quality and reliability are assured, facilitating accurate and robust analysis.

3. Methodology

In this section, the proposed method for classifying lithology will be discussed. Our approach relies on two key components: SIs and DIs, as outlined in Section 3.1. The concepts of bottleneck structures are introduced in Section 3.2. Then, in Section 3.3, the core elements of our approach, namely, NWLs-ResCNN 1D, SIs-ResCNN 2D, and DIs-ResCNN 2D, are presented. These components form the foundation of our methodology and enable effective lithology classification.

3.1. Generate Images. Before feeding well-log data into a two-dimensional CNN, it needs to be preprocessed. Well-logs typically consist of NWLs recorded at various depths. To create images, these well-logs need to be converted into a 2D format. This section provides an explanation of how to generate shallow and deep images from well-logs.

3.1.1. Shallow Images (SIs). The process begins by extracting the values of the selected features, which are then used to form a vector V of length 11 (representing the number of selected columns). Next, a matrix S is generated by vertically

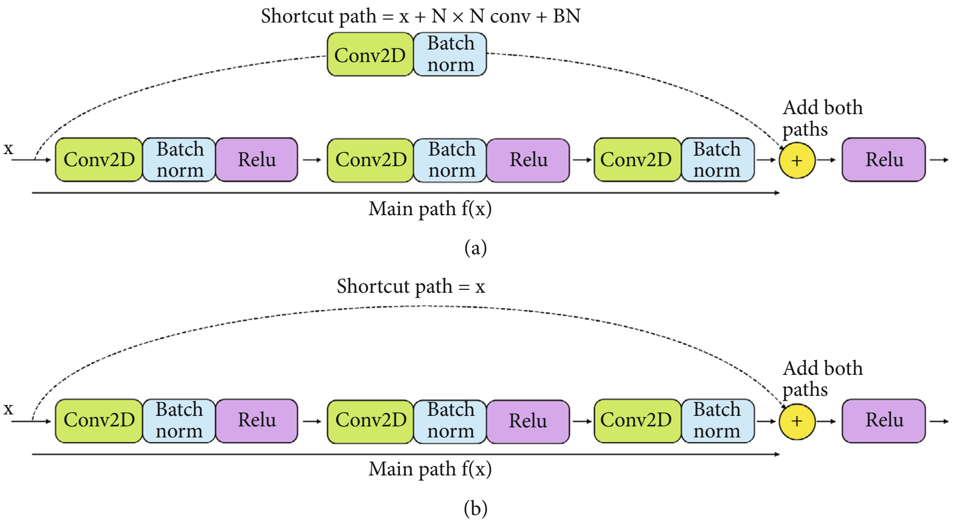


Figure 8: Network with a skip connection. The signal X is transmitted through the shortcut path and then combined with the main path's output, $f(x)$. Afterward, the relu activation function is applied to the sum of main path and shortcut path. (a) Bottleneck residual block with reduce shortcut. In order to decrease the input's dimensionality, a bottleneck layer consisting of an $N \times N$ convolutional layer and batch normalization is introduced to the shortcut path. This configuration is referred to as the reduce shortcut. (b) Bottleneck residual block with regular shortcut. The main pathway's output is combined with the input value via the shortcut before being passed into the relu activation function.

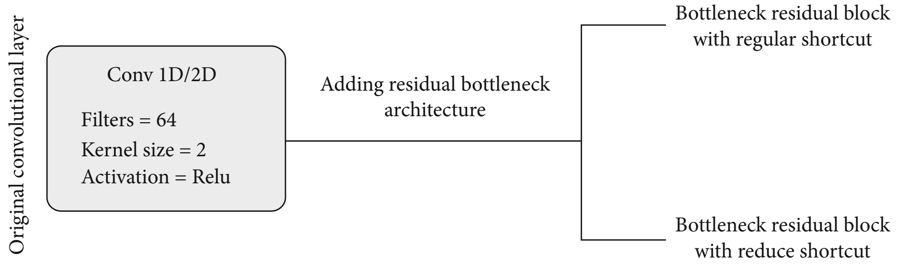

Original convolutional layer
Conv 1D/2D
Filters = 64
Kernel size = 2
Activation = Relu
Adding residual bottleneck architecture
Bottleneck residual block with regular shortcut
Bottleneck residual block with reduce shortcut

Figure 9: To enhance the learning process, reduced and regular shortcuts have been employed instead of the standard convolutional layers. In the original models, the convolutional layer with 64 filters, a kernel size of 2, and relu activation function was replaced with regular and reduce shortcuts. The modified model architectures are illustrated in Section 3.3.

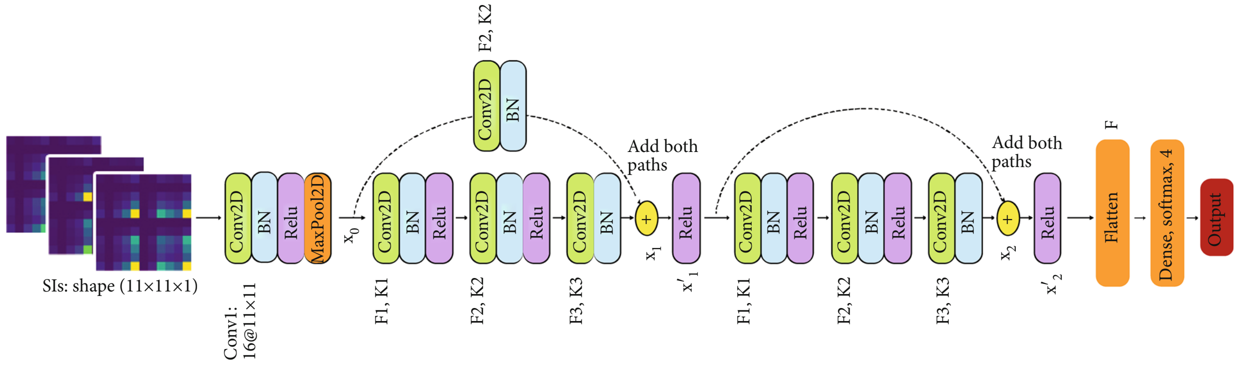


Figure 10: The SIs-ResCNN 2D employs bottleneck residual blocks with a regular shortcut and reduced shortcut, designed for input data with dimensions (11 × 11 × 1). Here, $F1$, $F2$, and $F3$ represent the number of filters, while $K1$, $K2$, and $K3$ denote the kernel size of the convolutional layers. These specific values are determined and set in the hyperparameter tuning section, as detailed in Section 5.1.

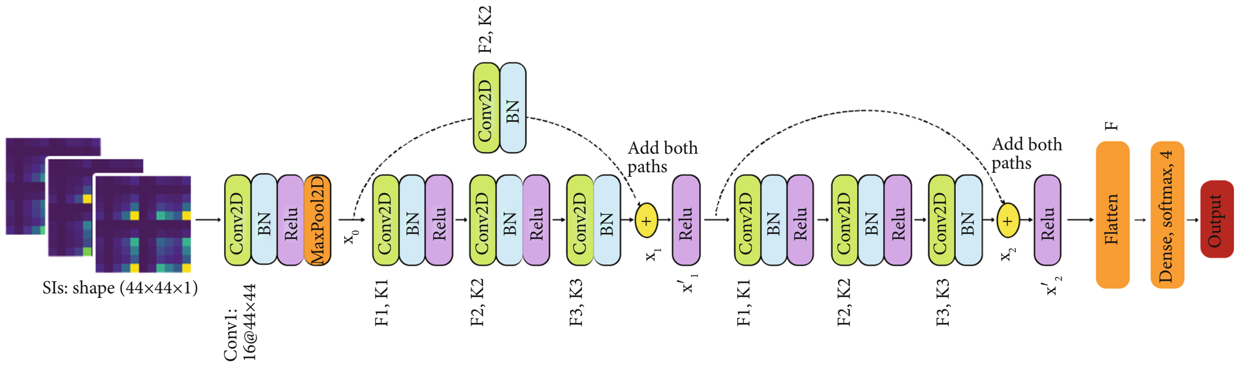


Figure 11: The DIs-ResCNN 2D employs bottleneck residual blocks with a regular shortcut and reduced shortcut, designed for input data with dimensions (44 × 44 × 1). Here, $F1$, $F2$, and $F3$ represent the number of filters, while $K1$, $K2$, and $K3$ denote the kernel size of the convolutional layers. These specific values are determined and set in the hyperparameter tuning section, as detailed in Section 5.1.

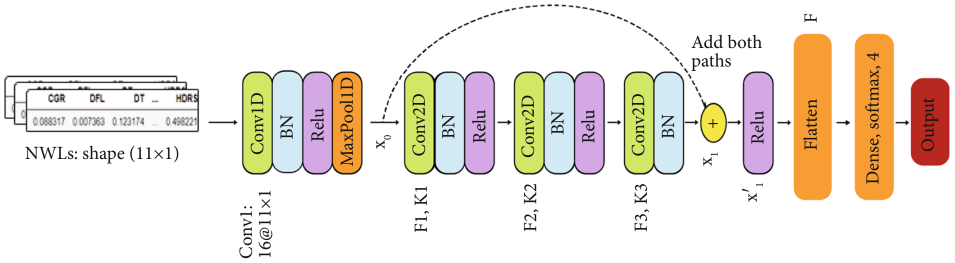


Figure 12: The NWLs-ResCNN 1D employs bottleneck residual blocks with a regular shortcut, designed for input data with dimensions (11 × 1). Here, $F1$, $F2$, and $F3$ represent the number of filters, while $K1$, $K2$, and $K3$ denote the kernel size of the convolutional layers. These specific values are determined and set in the hyperparameter tuning section, as detailed in Section 5.1.

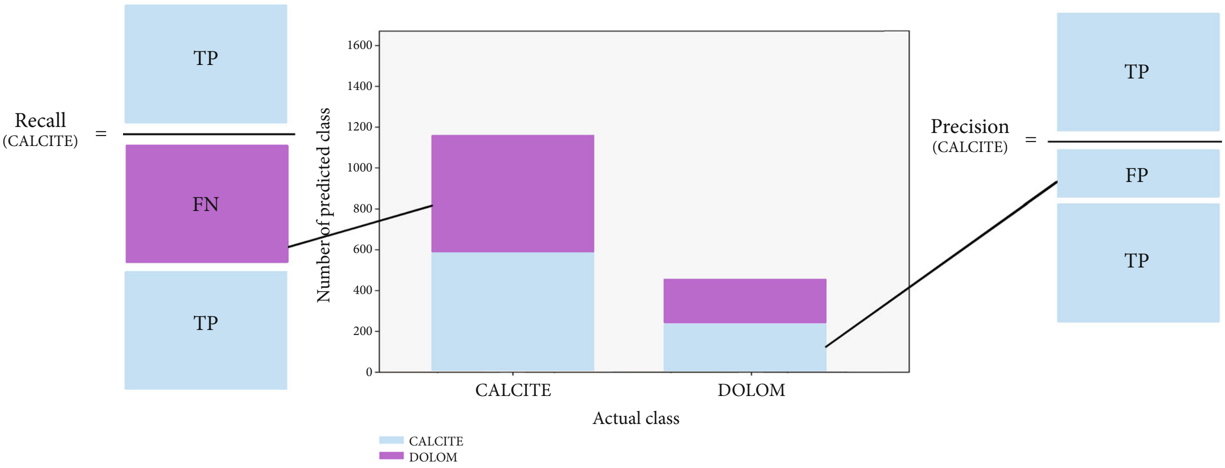


Figure 13: Stacked bar plot showing predicted classes compared to their actual classes. This visualizer provides a way to quickly understand the classifier performance at predicting the right classes. For each class, reducing FN leads to an increase in recall, and reducing FP leads to an increase in precision.

stacking vector V 11 times. As a result, matrix S will have dimensions (N, N), where $N = 11$. This process can be expressed as

$$V_{11\times1} = [a_{\mathrm{CGR}} \quad a_{\mathrm{DFL}} \quad a_{\mathrm{DT}} \quad \cdots \quad a_{\mathrm{PHIT}} \quad a_{\mathrm{RHOB}}], \tag{3}$$

$$S_{11\times11} = \begin{bmatrix} a_{\mathrm{CGR}} & a_{\mathrm{DFL}} & \cdots & a_{\mathrm{RHOB}} \\ a_{\mathrm{CGR}} & a_{\mathrm{DFL}} & \cdots & a_{\mathrm{RHOB}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\mathrm{CGR}} & a_{\mathrm{DFL}} & \cdots & a_{\mathrm{RHOB}} \end{bmatrix}. \tag{4}$$

Table 2: The parameters and search range during Bayesian optimization.

Hyperparameter	Search space
$F1$	[8, 16, 32, 64, 128]
$F2$	[8, 16, 32, 64, 128]
$F3$	[8, 16, 32, 64, 128]
$K1$	[1–4]
$K2$	[1–4]
$K3$	[1–4]
Optimizer	[Adam, SGD, RMSprop]
Learning rate	[0.00001, 0.0001, 0.001, 0.01, 0.1]

A 70-30 ratio was employed for fine-tuning the hyperparameters. These hyperparameters include filter sizes ($F1$, $F2$, and $F3$), kernel sizes ($K1$, $K2$, and $K3$), optimizer choices, and learning rate options.

The final symmetric matrix is obtained by performing element-wise multiplication between the matrix S and its transpose (S^T) resulting in a symmetric matrix with dimensions (N, N). The element-wise multiplication $S_{\text{element-wise}}$ is calculated as

$$S^T_{11\times 11} = \begin{bmatrix} a_{\text{CGR}} & a_{\text{CGR}} & \cdots & a_{\text{CGR}} \\ a_{\text{DFL}} & a_{\text{DFL}} & \cdots & a_{\text{DFL}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\text{RHOB}} & a_{\text{RHOB}} & \cdots & a_{\text{RHOB}} \end{bmatrix}, \tag{5}$$

$$S_{\text{element-wise}}(i, j) = S(i, j) \times S^T(i, j), \tag{6}$$

where $S(i, j)$ and $S^T(i, j)$ represent the elements at row i and column j in matrices $S_{11\times 11}$ and $S^T_{11\times 11}$, respectively. The resulting matrix $S_{\text{element-wise}}$ will also be of size 11×11.

This multiplication ensures that the interaction between feature i and feature j is the same as between feature j and feature i, resulting in a matrix that is symmetrical. This method allows us to represent relationships among the selected features in matrices which can provide valuable insights into geological and petrophysical measurements for analysis. Figure 4 illustrates the process of creating the shallow image (SI) at a depth of 3550.

3.1.2. Deep Images (DIs).

The algorithm provided by Jiang and Yin [39] was employed to generate a series of numbers. It starts with a value of 1 and checks each pair of consecutive numbers. The two numbers have not been paired before the second number is added to the sequence. This process continues until the first and second numbers become the same. Throughout each iteration, the algorithm maintains a list to keep track of all the generated numbers.

The algorithm confines the sequence within the range of 1 to 11, each number denoting a distinct well-log. If the second number in a pair exceeds this range, it loops back to 1. Moreover, the algorithm avoids including pairs of numbers that have already been incorporated in the sequence. By iterating through these steps, the algorithm generates a unique sequence of numbers characterized by alternating increments and specific skips. The resulting sequence is as follows: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 3, 5, 7, 9, 11, 2, 4, 6, 8, 10, 2, 5, 8, 11, 4, 7, 10, 3, 6, 9, 2, 6, 10, 4, 8, 2, 7, 11, 5, 9, 3, and 7. The generated sequence adheres to a pattern of alternating ascending numbers, punctuated by specific skips before repeating. In the context of the well-log data, the sequence is represented as a column vector as illustrated in

$$V_{44\times 1} = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \end{bmatrix}. \tag{7}$$

Here, $V_{44\times 1}$ is a column vector with 44 rows and 1 column, composed of coefficients a_1 through a_{11}.

Equations (8)–(10) define square matrices $S_{44\times 44}$, $S^T_{44\times 44}$, and $S_{\text{element-wise}_{44\times 44}}$, respectively. These matrices involve various combinations of coefficients a_1 through a_{11}, both element-wise and transposed. The structures of these matrices are described as follows:

$$S_{44\times 44} = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \\ a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \\ a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \\ \vdots & \vdots & \ddots & \vdots & & \\ a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \\ a_1 & a_2 & a_3 & \cdots & a_3 & a_7 \end{bmatrix}, \tag{8}$$

$$S^T_{44\times 44} = \begin{bmatrix} a_1 & a_1 & a_1 & \cdots & a_1 & a_1 \\ a_2 & a_2 & a_2 & \cdots & a_2 & a_2 \\ a_3 & a_3 & a_3 & \cdots & a_3 & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_3 & a_3 & a_3 & \cdots & a_3 & a_3 \\ a_7 & a_7 & a_7 & \cdots & a_7 & a_7 \end{bmatrix}, \tag{9}$$

$$S_{\text{element-wise}_{44\times 44}} = \begin{bmatrix} a_1 \cdot a_1 & a_2 \cdot a_1 & a_3 \cdot a_1 & \cdots & a_3 \cdot a_1 & a_7 \cdot a_1 \\ a_1 \cdot a_2 & a_2 \cdot a_2 & a_3 \cdot a_2 & \cdots & a_3 \cdot a_2 & a_7 \cdot a_2 \\ a_1 \cdot a_3 & a_2 \cdot a_3 & a_3 \cdot a_3 & \cdots & a_3 \cdot a_3 & a_7 \cdot a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 \cdot a_3 & a_2 \cdot a_3 & a_3 \cdot a_3 & \cdots & a_3 \cdot a_3 & a_7 \cdot a_3 \\ a_1 \cdot a_7 & a_2 \cdot a_7 & a_3 \cdot a_7 & \cdots & a_3 \cdot a_7 & a_7 \cdot a_7 \end{bmatrix}. \tag{10}$$

The presented content combines mathematical equations representing matrix operations with an algorithm that dynamically generates a sequence of indices based on specific rules. This integration serves as an illustrative example of theoretical concepts being translated into practical computational implementation. The element-wise equations show matrix structures involving coefficients, while the algorithm demonstrates a systematic approach to index

Table 3: Bayesian optimized values for hyperparameters of NWLs-ResCNN 1D, SIs-ResCNN 2D, and DIs-ResCNN 2D.

Model	$F1$	$F2$	$F3$	$K1$	$K2$	$K3$	Optimizer	Learning rate	Max trials
NWLs-ResCNN 1D	16	32	16	1	1	1	SGD	0.0001	30
SIs-ResCNN 2D	16	16	16	1	2	1	Adam	0.001	30
DIs-ResCNN 2D	32	64	32	1	2	1	SGD	0.0001	30

Table 4: Showing the precision, recall, and $F1$-score for both the original and proposed models.

Method	Model	Train			Test		
		Precision	Recall	$F1$-score	Precision	Recall	$F1$-score
Original	NWls-CNN 1D	84	93	87	62	74	65
Proposed	NWls-ResCNN 1D	81	92	84	64	82	67
Original	SIs-CNN 2D	77	85	80	62	63	62
Proposed	SIs-ResCNN 2D	83	90	86	65	73	68
Original	DIs-CNN 2D	86	94	89	71	86	74
Proposed	DIs-ResCNN 2D	93	98	95	78	81	79

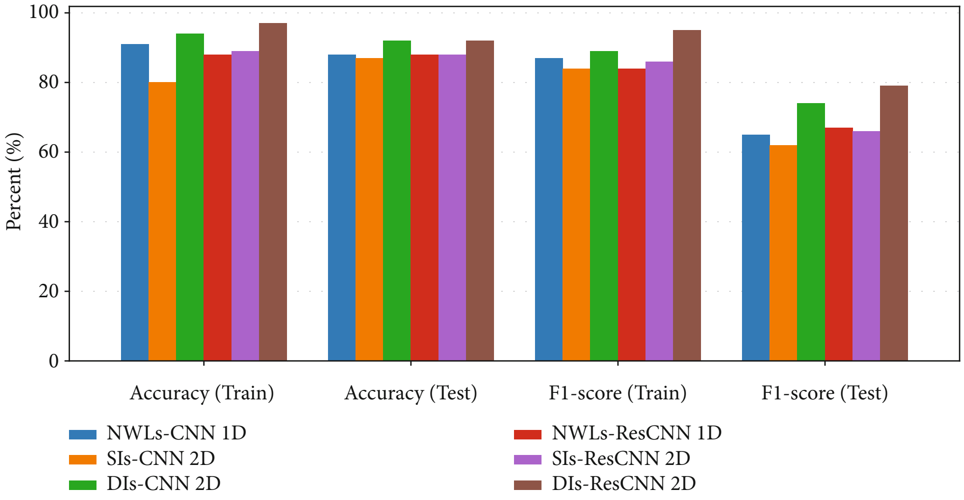


Figure 14: Display bar charts for accuracy and $F1$-score for both original and proposed models on the training and testing data. It is evident that the DI method has led to an improvement in performance compared to the other methods.

selection. Together, these elements provide a comprehensive understanding of both the mathematical and computational aspects presented. Figure 5 illustrates the procedure for generating the deep image (DI) at a depth of 3550.

Both SIs and DIs for all depths were created by combining the previously mentioned methods. In Figure 6(a), a data table containing logs recorded at various depths is shown. SIs, depicted in Figure 6(b), are generated using eleven well-logs, resulting in an 11-by-11-dimensional image. Additionally, DIs, as shown in Figure 6(c), were constructed using the described algorithm, resulting in a 44-by-44-dimensional image.

3.2. Residual Bottleneck. The layered structure of CNNs enables them to automatically recognize features and patterns in images, resulting in more effective learning, for example, in image classification [40, 41], transfer learning [42], object detection [43], segmentation [44–46], and a variety of other visual applications. In the fields of surveillance systems, as well as facial recognition and medical imaging, CNNs have demonstrated their impact. MRI and CT scans can be analyzed to identify and diagnose diseases using these images [47, 48].

He et al. [49] introduced a technique in which gradients can be directly backpropagated to earlier layers. This method involves using skip connections, which enable the flow of information from early layers to later ones, creating an alternative path for the gradient to traverse. An additional advantage of skip connections is that they enable the model to learn an identity function by directly passing the input, ensuring that a layer performs as effectively as the preceding layer and then combines both signals, namely, the skip connection and the main path, as shown in Figure 7. The direct path ensures a straightforward flow of information to the output. On the other hand, the second path passes through the block to enable

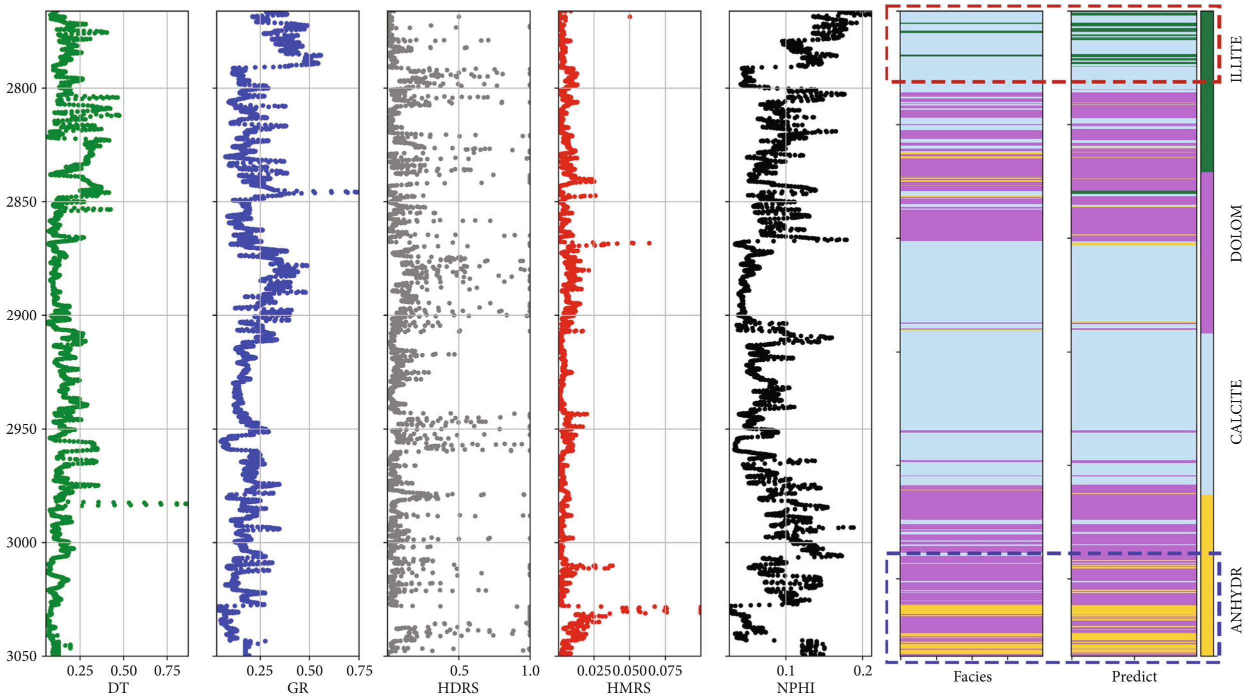


FIGURE 15: Displaying the DT, RG, HDRS, HMRS, and NPHI well-log test well, real facies predicted by the NWLs-ResCNN 1D are found, respectively. The sections marked with dashed lines highlight where this model makes the most significant errors in classifying ANHYDR and ILLITE facies.

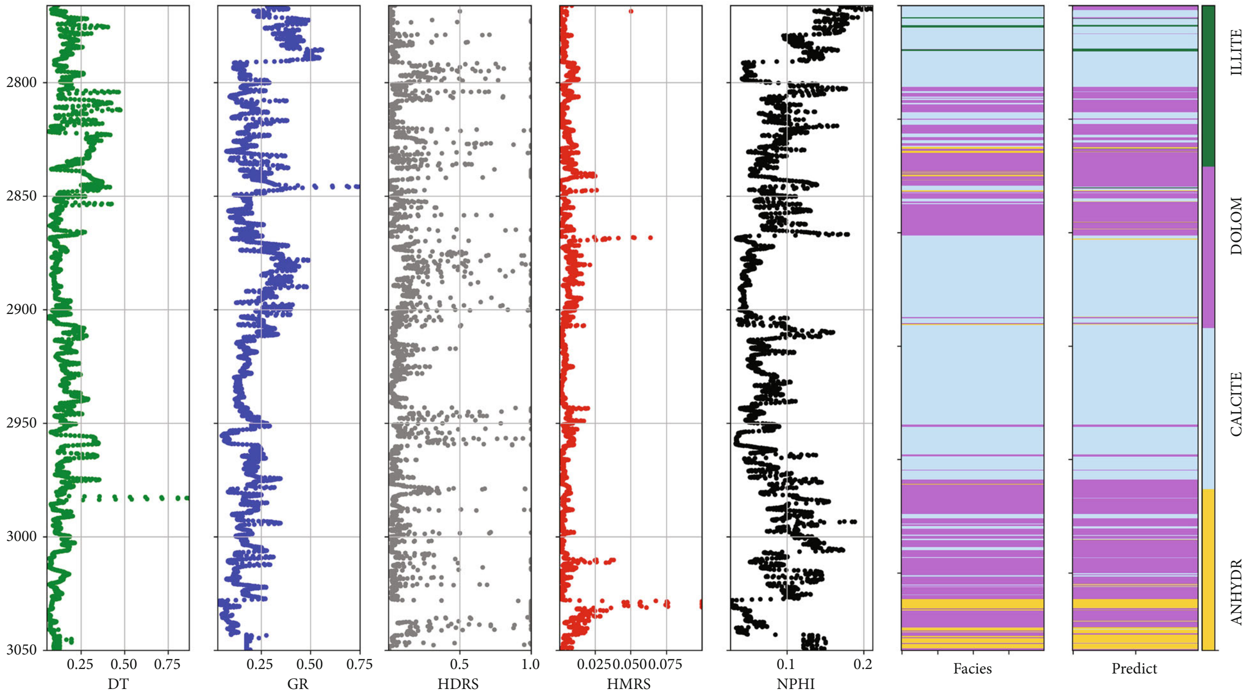
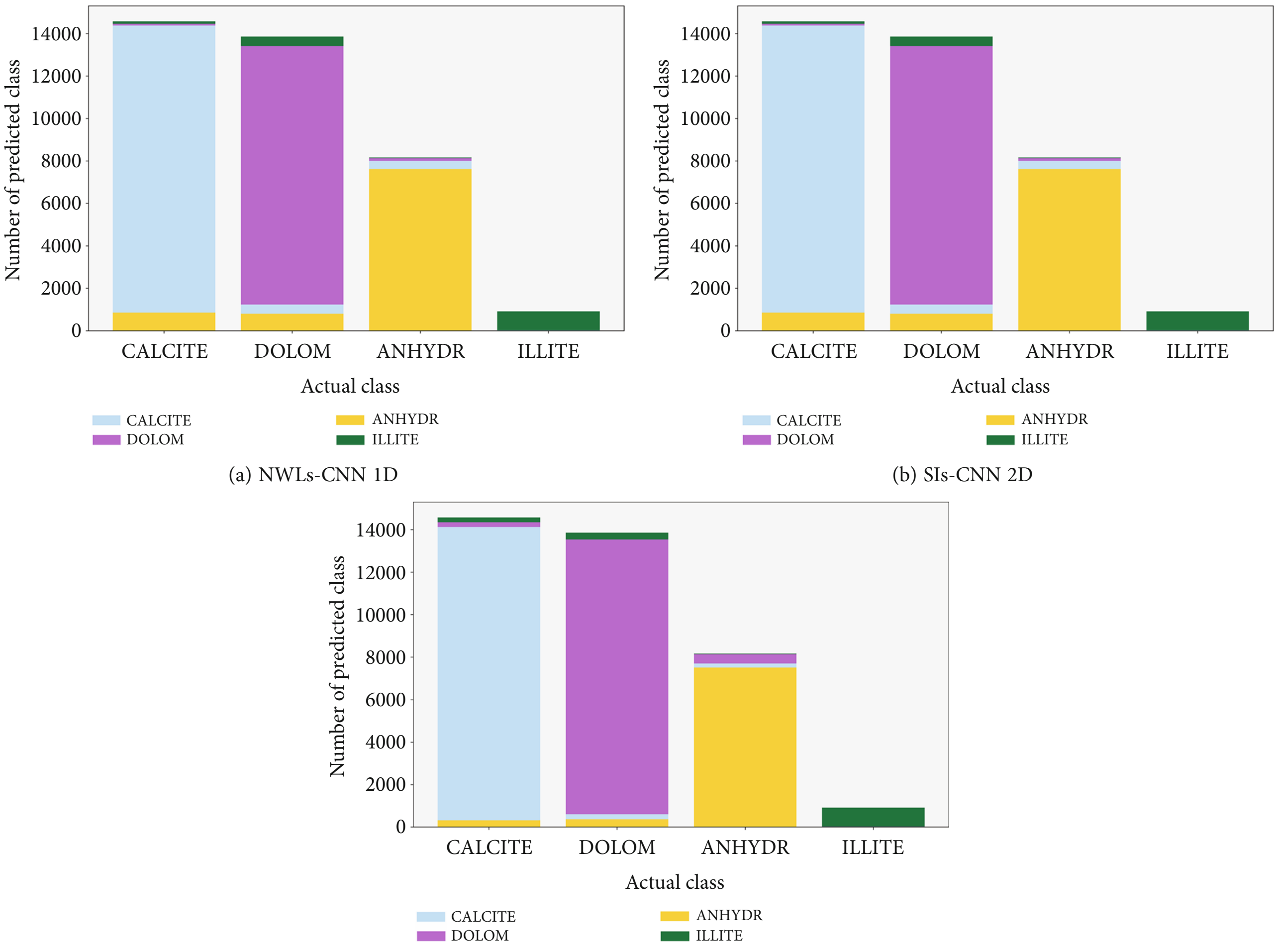


FIGURE 16: Displaying the DT, RG, HDRS, HMRS, and NPHI well-log test well, real facies predicted by the DIs-ResCNN 2D are found, respectively. It is evident that in most of the classifications for these facies, the DIs-ResCNN 2D model has performed well.

the extraction and integration of additional features. This enhances the model's ability to capture complex patterns and improves overall performance [50].

The combination of the skip connection and convolutional layers is called a residual block. A residual module consists of two branches: a shortcut path and a main path.



(a) NWLs-CNN 1D

(b) SIs-CNN 2D

(c) DIs-CNN 2D

Figure 17: Error plot of training data for original models: NWLs-CNN 1D, SIs-CNN 2D, and DIs-CNN 2D. Each bar is segmented to represent the proportion of predictions assigned to each class.

The main path consists of three convolutional layers with relu activations. Batch normalization was added to each convolutional layer to reduce overfitting and accelerate training. The main path architecture looks like this: $[\mathrm{CONV} \Rightarrow \mathrm{BN} \Rightarrow \mathrm{relu}] \times 3$. A bottleneck layer ($N \times N$ convolutional layer $+$ batch normalization) was added to the shortcut path as depicted in Figure 8(a) which is called the reduce shortcut. On the other hand, if the short path does not include the bottleneck layer, the output in add both path will be $f + x$ according to Figure 8(b) which is called the regular shortcut.

Modifications were applied to the original CNN structure by incorporating a residual connection (Figure 9) to enhance its efficiency in addressing the specific problem of lithology classification.

3.3. Proposed Residual Convolutional Neural Networks. The proposed models, illustrated in Figures 10–12, have been enhanced through the integration of residual structures. In SIs-ResCNN 2D, the X_0 signal is passed along the shortcut path and then added to the main path, $f(X_0)$. Then, the relu activation is applied to $f(X_0) + X_0 + K2 \times K2\mathrm{CONV} + \mathrm{BN}$ to generate the output signal: $\mathrm{relu}(f(X_0) + X_0 + K2 \times K2\ \mathrm{CONV} + \mathrm{BN})$ (The resulting output is denoted as X'_1). The output in the second add both path becomes $\mathrm{relu}(f(X'_1) + X'_1)$, as illustrated in the architecture shown with X'_2.

The DIs-ResCNN model receives DIs as input, as shown in Figure 11. The features of these images after applying reduce shortcut and the relu activation function can be represented as

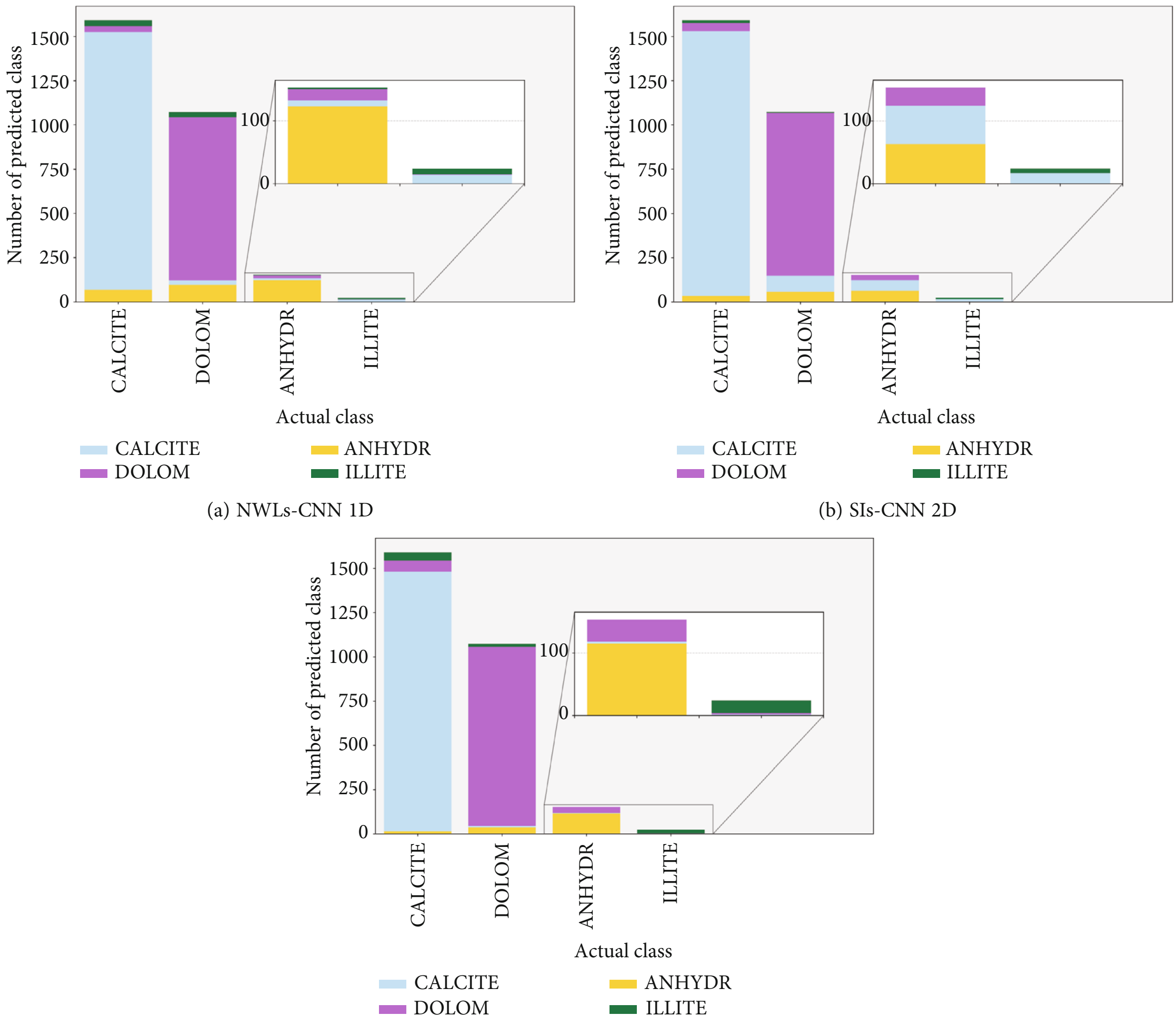
$$\text{DIs-ResCNN}\quad 2D : X_1' = \mathrm{relu}(f(X_0) + X_0 + K2 \times K2\quad \mathrm{CONV} + \mathrm{BN}). \tag{11}$$

Finally, by applying the regular shortcut, the features are flattened according to

$$\text{DIs-ResCNN}\quad 2D : F = \mathrm{relu}\left(f\left(X_1'\right) + X_1'\right). \tag{12}$$

The NWLs-ResCNN 1D model receives NWL data as input, as shown in Figure 12. The features after applying regular shortcut and the relu activation function can be represented as

$$\text{NWLs-ResCNN}\quad 1D : F = \mathrm{relu}(f(X_0) + X_0). \tag{13}$$



(a) NWLs-CNN 1D

(b) SIs-CNN 2D

(c) DIs-CNN 2D

Figure 18: Error plot of testing data for original models: NWLs-CNN 1D, SIs-CNN 2D, and DIs-CNN 2D. Each bar is segmented to represent the proportion of predictions assigned to each class.

These flattened features denoted as F are then fed into a dense layer. This layer consists of four neurons with the softmax activation function, which outputs probabilistic values. These values represent the probabilities of each class for each sample. To determine the most probable class for each sample, the Argmax function is applied based on these features (Argmax is a mathematical operation that identifies the argument or input value that results in the maximum value from a given target function).

In these proposed architectures, focus was placed on two hyperparameters in the model's architecture: the number of filters ($F1$, $F2$, and $F3$) and the kernel size ($K1$, $K2$, and $K3$). These hyperparameters were tuned in Section 5.1 to obtain their respective values.

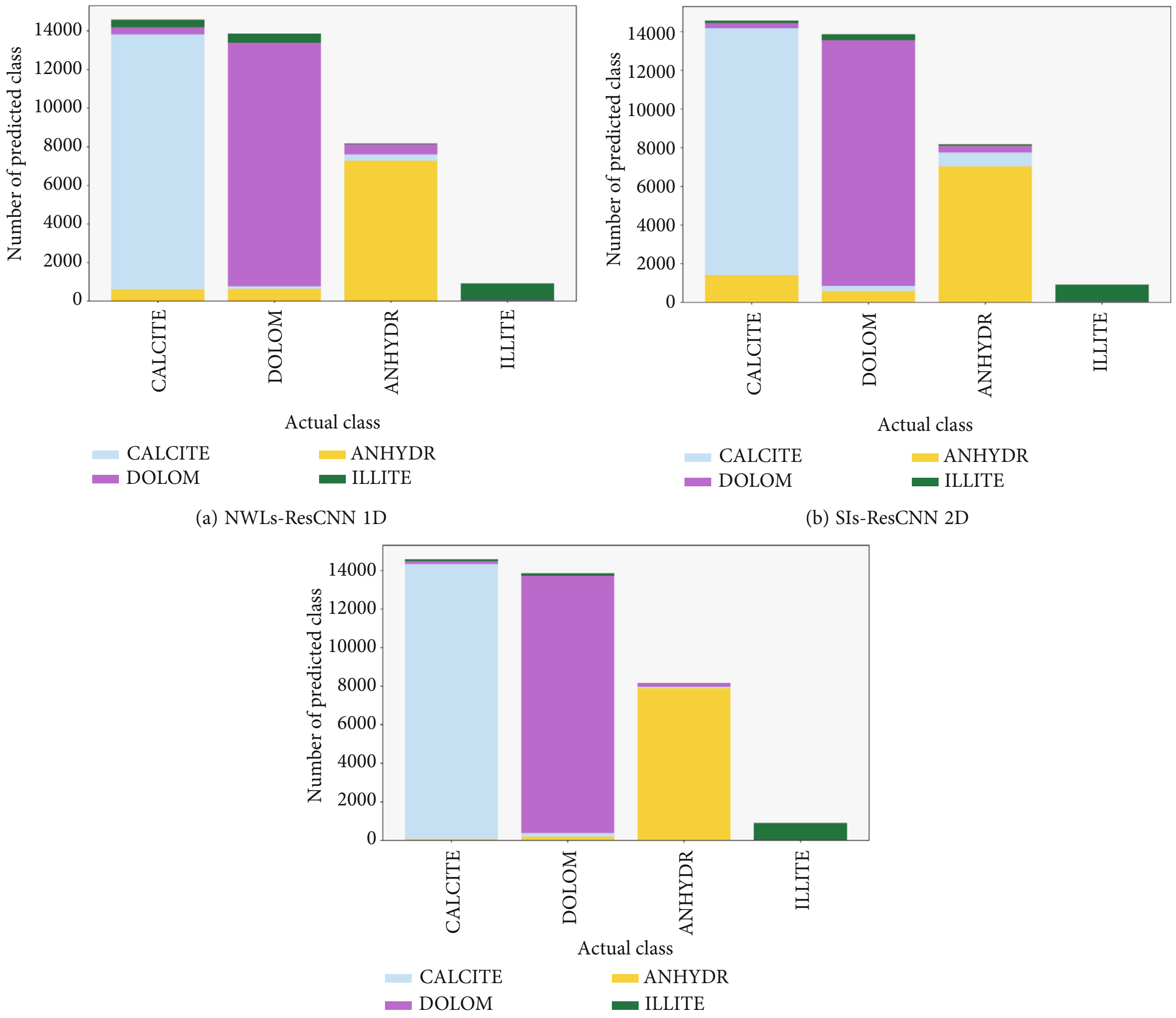
4. Model Evaluation

4.1. Metrics. Precision and recall are two metrics used to evaluate the performance of a classifier for individual classes. Precision refers to the probability that a sample truly belongs to a particular class given the classification result, while recall is the probability that a sample is correctly classified for a given class. The $F1$-score is a combined measure of accuracy and precision, providing a single measure of the relevance of classifier results [51]. The metrics for evaluating a model's performance are precision and recall, which are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{15}$$

The abbreviations TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. Here are the equations for calculating the $F1$ -score and accuracy:



(a) NWLs-ResCNN 1D

(b) SIs-ResCNN 2D

(c) DIs-ResCNN 2D

FIGURE 19: Error plot of training data for proposed models: NWLs-ResCNN 1D, SIs-ResCNN 2D, and DIs-ResCNN 2D. Each bar is segmented to represent the proportion of predictions assigned to each class.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (16)$$

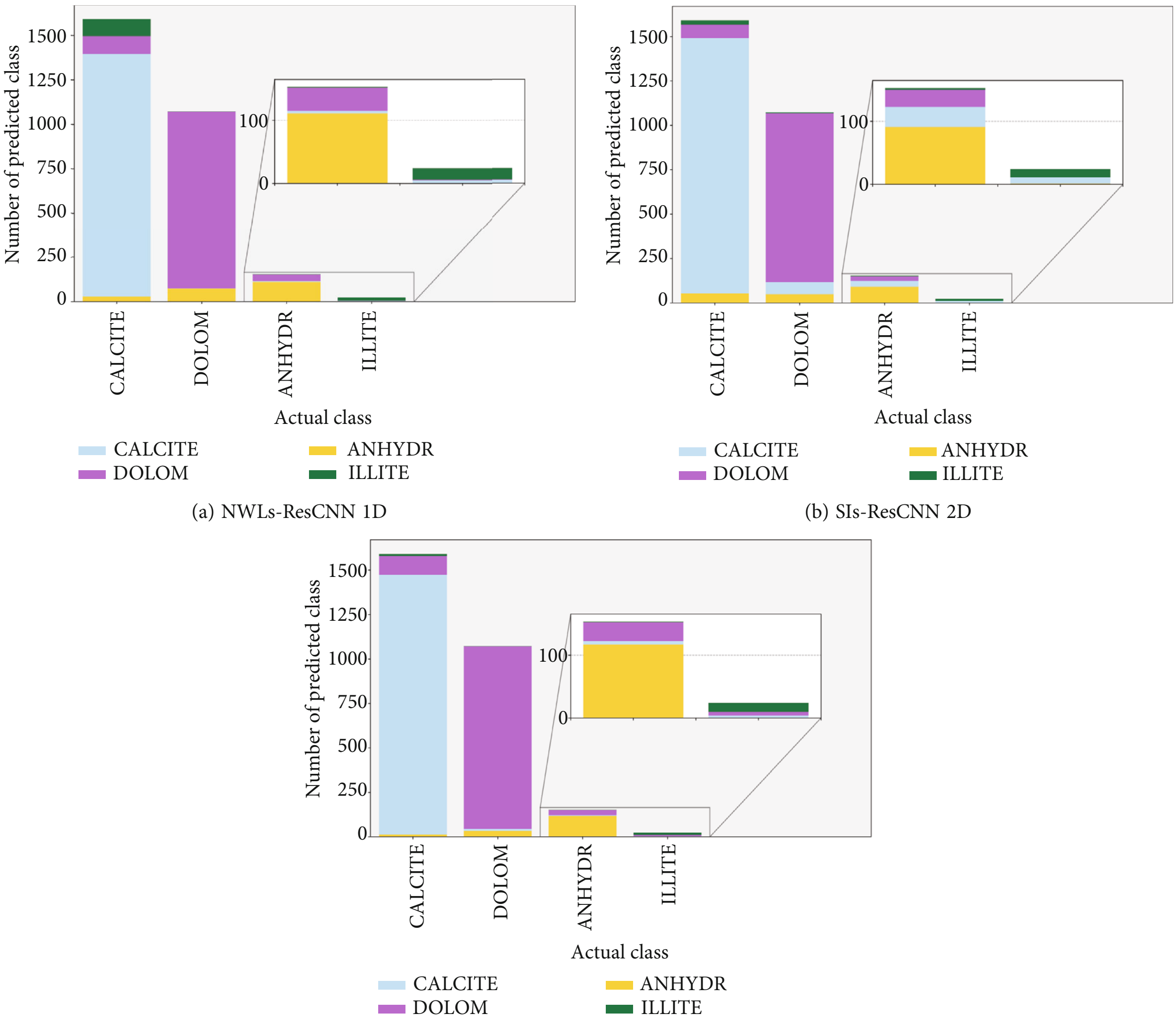
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (17)$$

4.2. Class Prediction Error Plot. Figure 13 visual representation uses a bar graph to show the support for each class in a classification model. Each bar is divided to indicate the percentage of predictions assigned to each class including negatives and false positives (similar to a confusion matrix). This plot assists us in comprehending both the strengths and weaknesses of models as well as specific challenges present in the dataset. Additionally, the class prediction error plot provides a way to assess how accurately the classifier predicts relevant classes. It is like a version of a confusion matrix that clearly highlights prediction errors. The error plot is particularly useful in classification problems allowing us to have a better understanding of the classification errors made by the model. As depicted in Figure 13, changes in recall and precision for faces CALCITE and DOLOM can be easily tracked. Additionally, the overall model performance can be evaluated in comparison to other models.

5. Results and Discussions

In this section, hyperparameters are initially considered, and their tuning process is explained. Subsequently, the results of the original and proposed models are compared. Following this, the best model is compared with baseline models such as support vector machine (SVM), K-nearest neighbor (KNN), and decision tree (DT). Additionally, the stability of the model and the impact of data balancing are investigated. Finally, the assessment of feature importance on NWL data is conducted.

5.1. Hyperparameter Tuning. An objective of hyperparameter tuning is to maximize the validation set's evaluation



(a) NWLs-ResCNN 1D

(b) SIs-ResCNN 2D

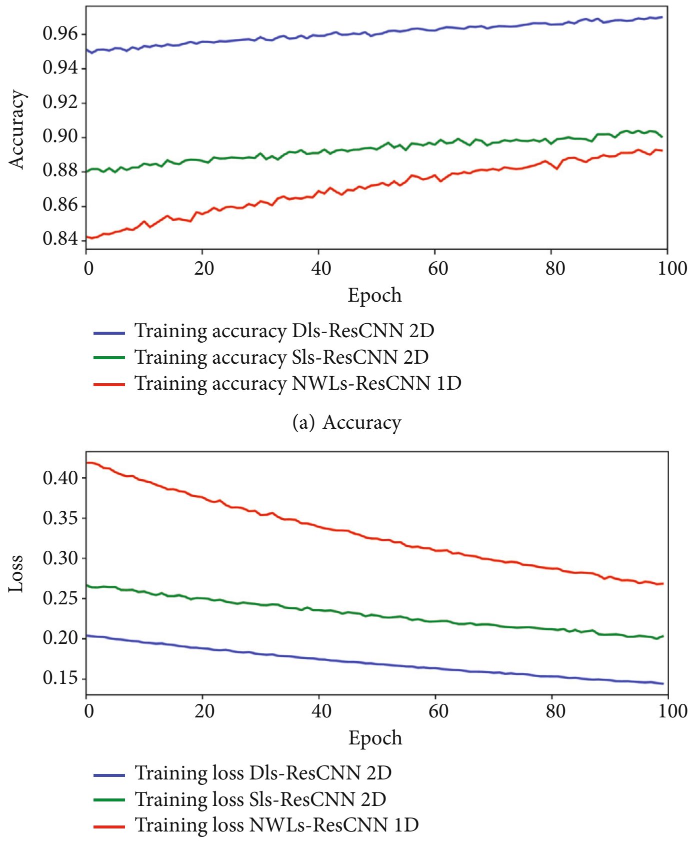
(c) DIs-ResCNN 2D

Figure 20: Error plot of testing data for proposed models: NWLs-ResCNN 1D, SIs-ResCNN 2D, and DIs-ResCNN 2D. Each bar is segmented to represent the proportion of predictions assigned to each class.

score without overfitting [52]. The optimal selection of filters (F) and kernel size (K) significantly influences deep learning model performance. Filters extract features by capturing intricate details which in turn requires greater computation. Kernel size affects feature extraction; larger kernels encompass more context but require more computation. A careful balance is crucial because excessive filters/kernels can lead to overfitting, while insufficient filters can result in underfitting. For the three proposed models, the hyperparameters were tuned by dividing the training data into a 70-30 ratio to determine these parameters. Subsequently, the model was trained on the entire dataset and evaluated using the test data that had been excluded. The hyperparameters and their corresponding tuned values for each model are detailed in Table 2 and Table 3, respectively.

Bayesian optimization is an important technique that differs from grid search or random search methods because it adapts the search process based on previous trials. This assists the tuner in making choices about which hyperparameters to test next, focusing on configurations that seem promising for better performance. By doing this, it significantly cuts down on the number of trials required to find the hyperparameters, which saves both time and computational resources. Bayesian optimization is employed to improve the efficiency of hyperparameter tuning. By setting max trials to 30, it means that Keras Tuner will explore and evaluate 30 different combinations of hyperparameters to find the best configuration (Table 3). The tuner will perform multiple training runs, adjusting various hyperparameters like the learning rate and the number of filters for each trial. Then, it will compare the performance of these models based on a specified evaluation metric (e.g., validation accuracy).

5.2. Overall Performance. Table 4 provides a comparative overview of model performance using SIs and DIs in the context of well-log data analysis. SIs involve a process of extracting specific features, creating a vector and generating a corresponding matrix. On the other hand, DIs ensure a



(a) Accuracy

(b) Loss

Figure 21: The training loss and accuracy of three proposed models, namely, DIs-ResCNN 2D, SIs-ResCNN 2D, and NWLs-ResCNN 1D. Loss and accuracy have been examined for the proposed models over 100 epochs. The plot on (a) displays the model's accuracy, while the plot on (b) illustrates the model's loss.

sequence of numbers between 1 and 11 with unique patterns formed by alternating and skipping steps.

The different models are evaluated based on precision, recall, and $F1$-score metrics for both training and testing datasets. The baseline models, NWls-CNN 1D, SIs-CNN 2D, and DIs-CNN 2D, represent the original approaches while the proposed models, NWls-ResCNN 1D, SIs-ResCNN 2D, and DIs-ResCNN 2D, incorporate refinements and enhancements. Comparing the original and proposed models, it is clear that the residual convolutional neural network (ResCNN) architecture significantly improves the performance across all methods. Notably, the DIs-ResCNN 2D model achieves remarkable precision, recall, and $F1$-score values on both training and testing datasets, indicating its effectiveness in handling the complexities of the well-log data. These findings highlight the importance of utilizing structured and DIs in combination with advanced deep learning architectures such as ResCNN to enhance the accuracy and reliability of well-log data analysis. The distinct patterns generated by DIs contribute to the improved performance of the models demonstrating the potential of this approach in the field of geological data processing and interpretation. The proposed DIs-ResCNN 2D model had the highest metrics among all the models on both the training and test sets. The training set had a precision of 93%, recall of 98%, and $F1$-score of 95%. The test set also showed strong results with a precision of 78%, recall of 81%, and $F1$-score of 79%.

Table 5: The precision, recall, and $F1$-score metrics were calculated for five baseline classifiers: support vector machines (SVM), K-nearest neighbor (KNN), decision tree (DT), random forest (RF), and extra trees.

Model	Train			Test		
	Precision	Recall	$F1$-score	Precision	Recall	$F1$-score
SVM	95	92	94	65	68	67
KNN	97	96	96	75	64	63
DT	98	97	97	62	67	63
RF	99	97	98	72	68	64
Extra trees	89	80	83	64	64	63

Figure 14 illustrates bar charts comparing the original models with the proposed models in terms of accuracy and $F1$-score evaluation metrics on both training and testing datasets. The utilization of the DIs technique has clearly resulted in improved performance for the DIs-CNN 2D and DIs-ResCNN 2D models in comparison to the other

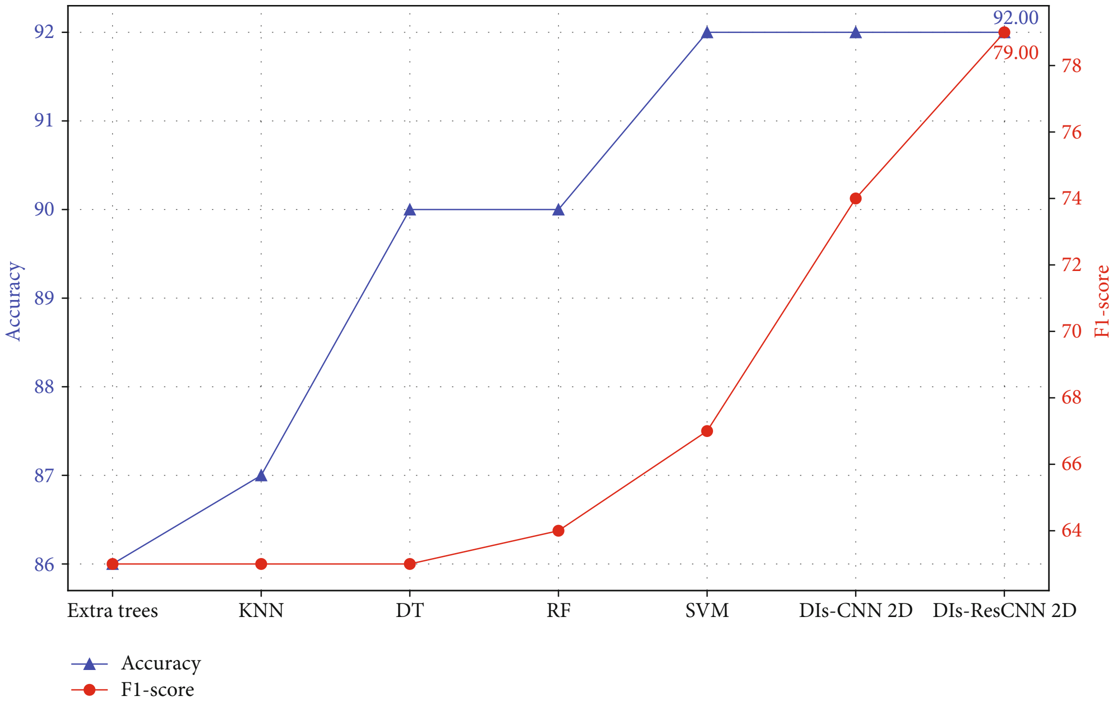


Figure 22: The accuracy and $F1$-score plots for the baseline models compared to the DIs-CNN 2D and DIs-ResCNN 2D models show that both proposed models outperformed the baseline models. The DIs-ResCNN 2D model achieved an accuracy of 92% and an $F1$-score of 79%.

models. This highlights the effectiveness of generating images using the proposed DI method.

In Figures 15 and 16, for NWLs-ResCNN 1D and DIs-ResCNN 2D models, the visual display of their quality and accuracy in determining the facies for the well that was not included in the training (test well) can be seen. As is clear, especially considering the parts marked with a dashed line, using the presented approach to convert NWLs to DIs and using the generated images as input to CNN have resulted in a significant improvement in the determination of ANHYDR and ILLITE facies.

Figures 17 and 18 show the class prediction error plots for the original models, and Figures 19 and 20 show the proposed models. As mentioned earlier, these plots provide better insights into the model's performance compared to the confusion matrix. They also offer a simpler analysis, particularly when dealing with multiple classification problems. In error plot visualizations, the horizontal axis represents the actual lithological facies. The accompanying legend enables a clear distinction between accurately identified and misclassified facies by the model. These insights are imperative for refining the model and enhancing its predictive capabilities, thereby ensuring more accurate facies predictions in complex geological scenarios.

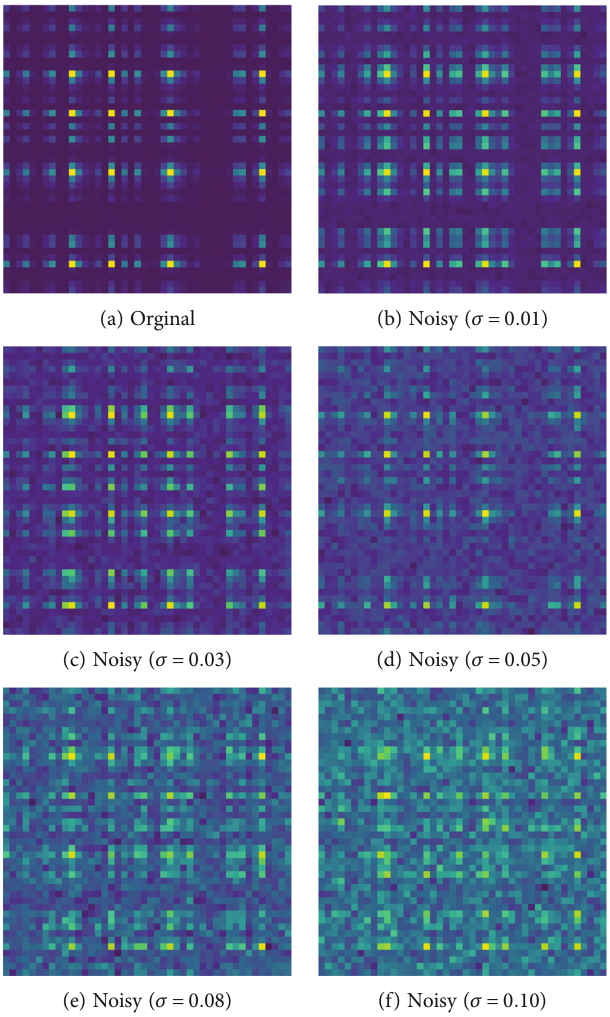
In Figures 17 and 18, the DIs-CNN 2D model has a higher recall in predicting illite than the other two models, namely, NWLs-CNN 1D and SIs-CNN 2D. On the other hand, it can be seen that this model exhibits higher precision and recall in determining the dolomite facies. By comparing Figures 18(c) and 20(c), the most obvious change is the trade-off between precision and recall, which, according to Table 4, has improved the $F1$-score value. The rest of the models are comparable with the same process in terms of precision, recall, $F1$-score, and accuracy.

In Figure 21, the presented models, namely, DIs-ResCNN 2D, SIs-ResCNN 2D, and NWLs-ResCNN 1D, depict the values of loss and accuracy on the training data during 100 epochs. As is evident, accuracy values exhibit a gradual increase, while loss values decrease for all three models. The model trained with DIs demonstrates the effectiveness of DIs in enhancing the performance of the proposed model. Additionally, the learning process of the DIs-ResCNN 2D and SIs-ResCNN 2D occurs more rapidly in the initial epochs indicating an enhancement in the models' ability to recognize patterns.

5.3. Comparison to Baseline Models. In this study, a method for generating SIs and DIs, as well as residual convolutional neural networks, was introduced. The performance of the proposed model was evaluated by comparing it to established baseline models in this section (Table 5). The primary objective was to assess the effectiveness and superiority of our approach in addressing the problem under investigation.

Baseline models serve as applied algorithms that act as reference points when evaluating approaches. By comparing our model to these baseline models, the aim is to demonstrate its ability to surpass or at least match the existing state of techniques. The analysis results highlight the strengths of our proposed model and identify any possible limitations and areas, for further improvement. The DIs-CNN 2D model has demonstrated improvements over the baseline models in terms of test accuracy as well as $F1$-score (Figure 22).

The DIs-ResCNN 2D outperformed other models and achieved a train accuracy of 97% and a test accuracy of


(a) Orginal

(b) Noisy ($\sigma = 0.01$)

(c) Noisy ($\sigma = 0.03$)

(d) Noisy ($\sigma = 0.05$)

(e) Noisy ($\sigma = 0.08$)

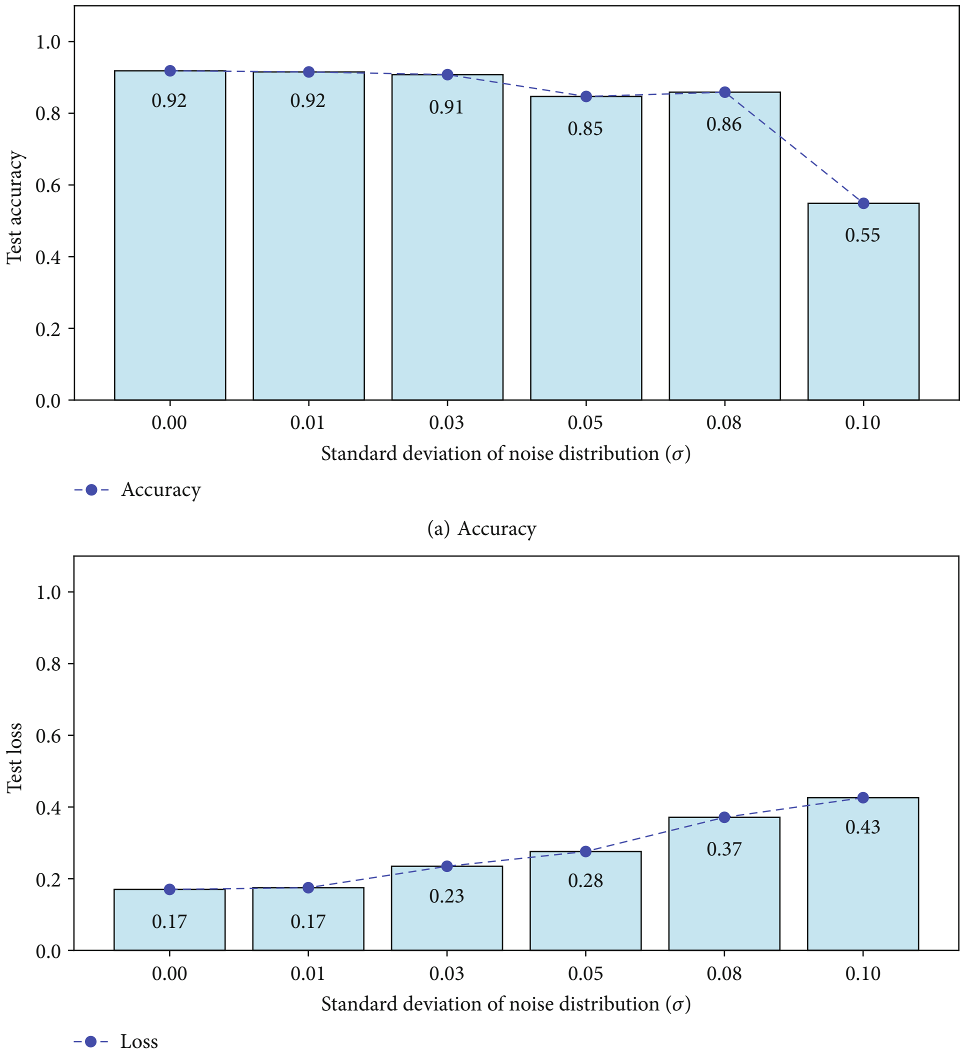
(f) Noisy ($\sigma = 0.10$)

Figure 23: The effect of Gaussian noise on DIs has been investigated. (a) The first image is without any noise, while (b–f) the rest of the images depict different levels of noise being applied.

92%. Additionally, it obtained a test $F1$-score of 79% indicating its proficiency in handling both precision and recall. The incorporation of connections (residual block), in the architecture, likely contributed to efficient training and improved generalization. Both DIs-CNN 2D and DIs-ResCNN 2D models exhibit better performance compared to other proposed and baseline models. The application of images obtained from data and the incorporation of residual connections have shown to be successful techniques for this specific lithology classification assignment. Using DIs and residual CNNs provides improved accuracy and $F1$-score, which are measures for ensuring accurate predictions and managing imbalanced categories.

5.4. Model Robustness. Incorporating the Gaussian noise technique into the model design enhances the robustness and performance of the system. This technique serves as a regularization method by adding controlled noise to the input data during training real-world variations and enhancing our model to handle unexpected inputs. By integrating Gaussian noise within the layers of our network using Keras [53], we introduce an element of controlled randomness that encourages the model to learn resilient and adaptable features. This approach prevents overfitting and also equips our model with the ability to make accurate predictions on unseen data ultimately resulting in improved overall performance and reliability. Figure 23 illustrates the impact of Gaussian noise on the DI at different levels. In order to evaluate the robustness of the proposed approach, Gaussian noise was deliberately introduced to the original data with standard deviations ranging from 0 to 0.10. This variation in standard deviations represents diverse intensities of noise, facilitating a comprehensive assessment of the model's performance across varying degrees of noise interference.

Controlled randomness is encouraged by incorporating Gaussian noise into the input, promoting the learning of features, preventing overfitting, and enabling predictions on new noisy datasets. Increased model resilience in challenging scenarios has been observed through validation. Although there is a decrease in accuracy with levels of noise, the model still performs well overall indicating improved generalization (Figure 24).



(a) Accuracy

(b) Loss

Figure 24: Loss and accuracy have been examined for the DIs-ResCNN 2D model under various levels of Gaussian noise applied to the DIs. The plot on (a) displays the model's accuracy across different noise levels, while the plot on (b) illustrates the model's loss across various noise levels.

5.5. Effect of Data Balancing. Data balancing is a crucial technique that involves redistributing data in a dataset to achieve a balanced representation across various classes. This is especially important when dealing with imbalanced data [54]. In supervised learning, it is crucial to train a model using balanced data to ensure that the model is equally informed about all classes [55].

The investigated data is unbalanced; in other words, the number of facies is unequal. Thus, another aspect to be considered is balancing the data. SMOTE proves to be a technique for dealing with imbalanced datasets when it comes to machine learning tasks. Particularly, it works when one class has more samples compared to the others. The number of samples for the minority classes was increased by utilizing SMOTE to balance the dataset. This led to an improvement in $F1$-score for classes as shown in Table 6, before using SMOTE, and in Table 7, after using SMOTE.

Table 6: Classification report before using SMOTE.

	Precision	Recall	$F1$-score	Support
0	0.72	0.76	0.74	153
1	0.99	0.92	0.95	1591
2	0.88	0.96	0.92	1073
3	0.52	0.58	0.55	24
Accuracy			0.92	2841
Macro avg	0.78	0.81	0.79	2841
Weighted avg	0.93	0.92	0.92	2841

A classification report provides precision, recall, $F1$-score, and support values for each class in a classification model's evaluation. Classes zero, one, two, and three correspond to the ANHYDR, CALCITE, DOLOM, and ILLITE facies, respectively.

Table 7: Classification report after using SMOTE.

	Precision	Recall	$F1$-score	Support
0	0.68	0.78	0.73	153
1	0.99	0.93	0.96	1591
2	0.91	0.96	0.93	1073
3	0.44	0.83	0.58	24
Accuracy			0.93	2841
Macro avg	0.76	0.88	0.80	2841
Weighted avg	0.94	0.93	0.93	2841

A classification report provides precision, recall, $F1$-score, and support values for each class in a classification model's evaluation. Classes zero, one, two, and three correspond to the ANHYDR, CALCITE, DOLOM, and ILLITE facies, respectively.

Table 8: Classification report after using attention mechanism.

	Precision	Recall	$F1$-score	Support
0	0.71	0.79	0.75	153
1	0.98	0.92	0.95	1591
2	0.89	0.95	0.92	1073
3	0.54	0.79	0.64	24
Accuracy			0.92	2841
Macro avg	0.78	0.86	0.82	2841
Weighted avg	0.93	0.92	0.93	2841

A classification report provides precision, recall, $F1$-score, and support values for each class in a classification model's evaluation. Classes zero, one, two, and three correspond to the ANHYDR, CALCITE, DOLOM, and ILLITE facies, respectively.

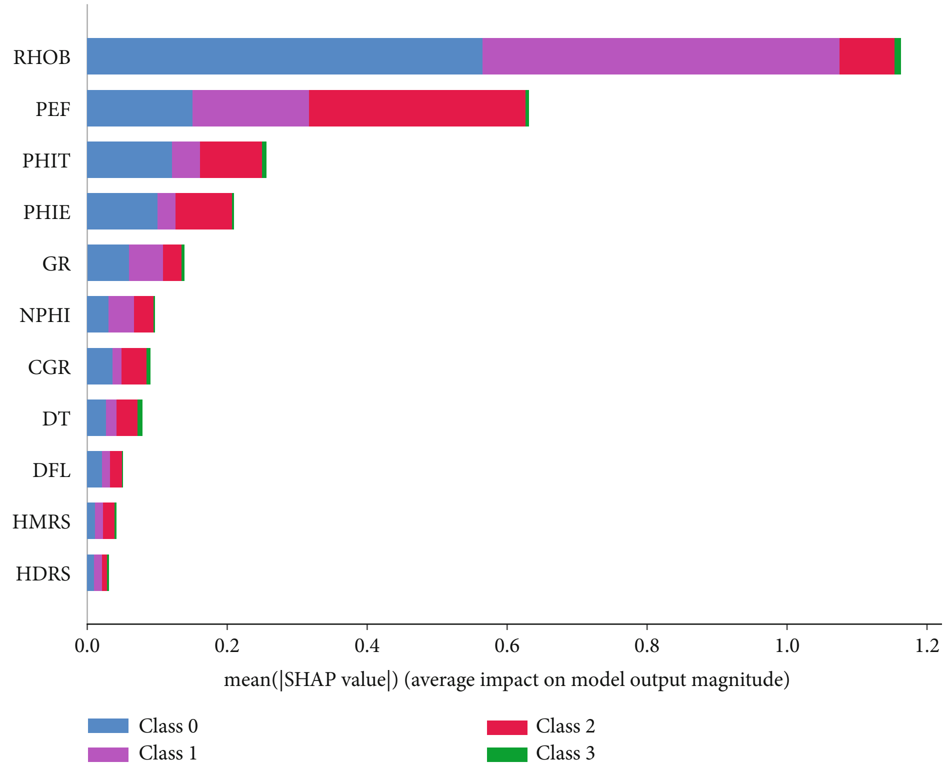
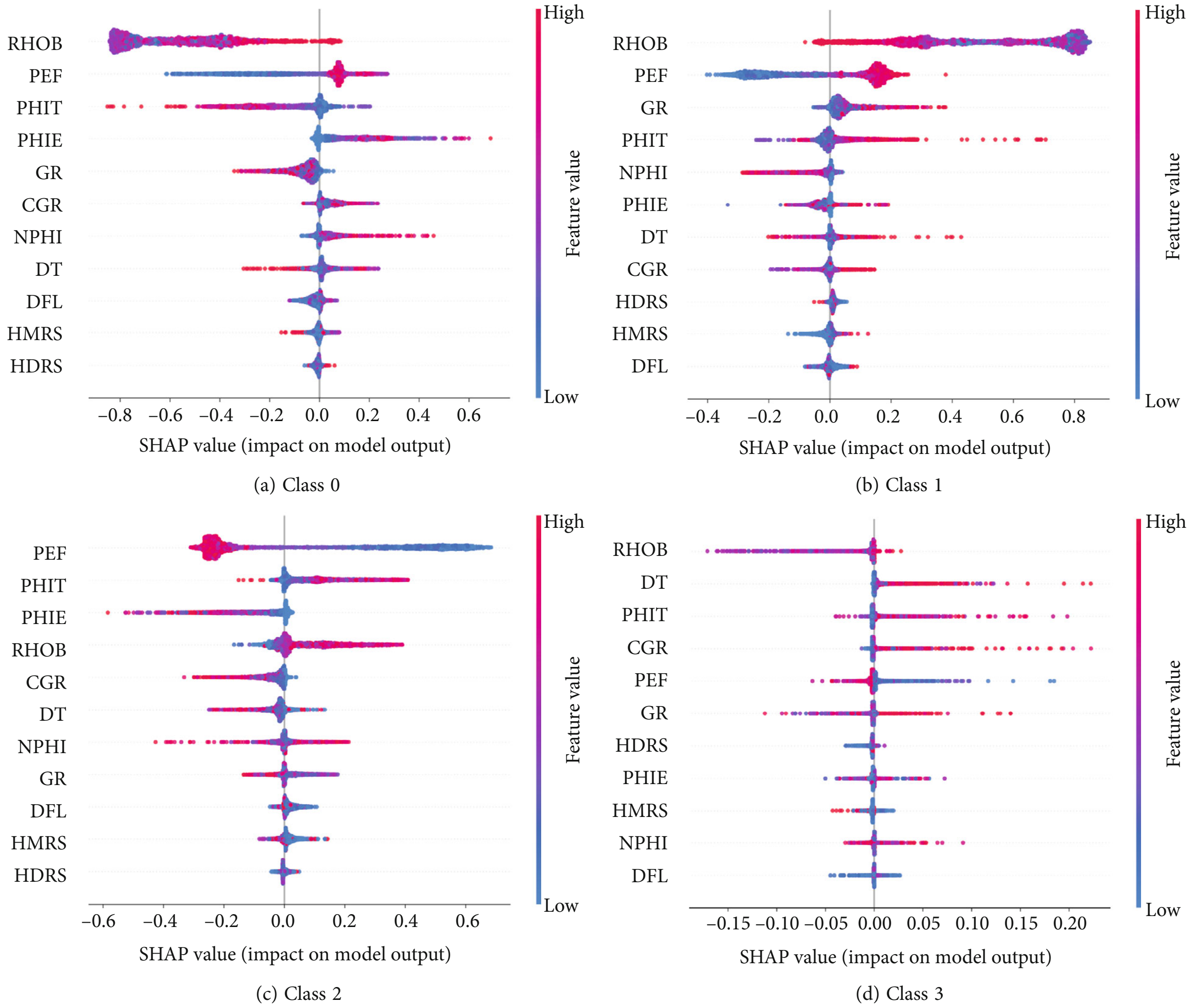


Figure 25: The bar chart displays the prioritization of eleven NWL variables' importance determined by the average absolute SHAP values. The classes zero, one, two, and three correspond to the ANHYDR, CALCITE, DOLOM, and ILLITE facies, respectively.



(a) Class 0

(b) Class 1

(c) Class 2

(d) Class 3

Figure 26: SHapley Additive exPlanation values (SHAP) are provided for each class, ranking the most important features from top to bottom. Blue dots represent small variable values, while red data points represent large variable values.

The improvement in $F1$-scores for all classes demonstrates the positive impact of SMOTE on the learning process. However, SMOTE impact on model performance can vary depending on the dataset and classification problem. While similar improvements cannot always be assumed, it has been beneficial in this case. It is essential to evaluate its effectiveness on a case-by-case basis. The integration of SMOTE into this lithology classification task has effectively resolved the problem of class imbalance leading to improved performance metrics for individual classes as well as overall accuracy.

5.6. Impact of Attention Mechanism. Attention is a mechanism in neural networks that enables the model to focus on specific parts of the input sequence when making predictions [56]. The key principle behind attention is to allocate different levels of importance or weights to different elements in the input sequence, which allows the model to weigh the contributions of each element differently. In the context of image classification or computer vision, attention mechanisms can be employed spatially to focus on specific pixels of an image [57, 58].

Table 6 provides insights into the performance of the DIs-ResCNN 2D model before the implementation of an attention mechanism. The effect of the attention mechanism on the DIs-ResCNN 2D model has been investigated, and its results are shown in Table 8.

Before using the attention mechanism, the model achieved a $F1$-score of 0.74 for class 0 (ANHYDR) and 0.55 for class 3 (ILLITE). After implementing the attention mechanism, there is an improvement, particularly in classes 0 and 3. The $F1$-score for class 0 increased to 0.75, and for class 3, it has improved significantly to 0.64. The macro $F1$ -score also showed an enhancement by rising from 0.79 to 0.82 after incorporating the attention mechanism. This improvement in macro $F1$-score indicates a better balance in the model's ability to correctly classify all classes, emphasizing a reduction in the bias towards specific classes.

5.7. Feature Importance. The SHapley Additive exPlanations (SHAP) technique is used to assess the importance of features in ML and DL models. One of its advantages is that it provides a comprehensive framework to interpret different types of models [59]. It effectively handles interactions between variables and provides accurate estimates of feature importance. Unlike methods like permutation importance, which struggle with complexity, SHAP is particularly useful for tree-based models and neural networks. It can evaluate both the importance of features across the entire dataset and their specific contributions, in individual cases. An interesting aspect of SHAP is that it assigns scores to indicate how features impact predictions [60, 61].

Figure 25 presents a bar chart that showcases the prioritization of the importance of eleven well-logs related to NWLs. The significance of these variables is determined based on their SHAP values. This bar chart allows us to easily identify features, across the entire dataset. According to the SHAP values, the two most crucial features for lithology classification based on well-logs are RHOB and PEF. These features play a role in identifying the lithology and mineralogy of rocks which holds importance in the oil and gas industry.

Figure 26 displays the SHAP values for each class in lithology classification based on well-logs. These SHAP values represent the contribution of each feature across all combinations of features. The features are ranked from top to bottom in this visualization. In this representation, blue dots represent low values, while red data points indicate high variable values. Analyzing this information enables us to gain an understanding of how the model behaves and responds to combinations of features for each class in lithology classification. This knowledge is extremely valuable when developing models for lithology classification based on well-logs. For example, in Figure 26(c), the impact of model features on class 2 (DOLOM) is highlighted. The PEF and PHIT features have a significant impact, while the HDRS feature has a small influence. It is interesting to note that as the PEF increases, the SHAP value decreases, but when the PEF decreases, the SHAP value increases.

6. Conclusion

In this comprehensive research, we have designed an advanced deep learning framework specifically developed for the accurate identification of rock facies in the oil and gas industry. Our distinctive methodology utilizes well-log data obtained during logging operations, significantly improving the accuracy and efficiency of rock facies classification.

The main elements of our framework include the use of imaging techniques, namely, shallow images and deep images, which expertly transform sensor data into synthetic images. These transformed data visualizations serve as feeds for our deep learning models, assisting them to identify complex patterns during the learning stage. This leads to a more successful classification of rock facies compared to traditional methods as well as approaches devised by human engineers. The following outlines the key discoveries from our study:

(i) SI and DI methods were used to convert NWLs into images that capture the complex relationships between well-log features, improving our understanding of subsurface geology

(ii) Incorporating residual and bottleneck structures significantly improved the performance of our lithology classification model. We have demonstrated the potential of the DIs-ResCNN 2D architecture with balanced data in geological applications by outperforming other models (SIs-ResCNN 2D and NWLs-ResCNN 1D) in terms of accuracy, achieving an impressive accuracy of 93%. Furthermore, our model excelled in precision and recall, resulting in an $F1$-score of 80%, highlighting its robustness and suitability for challenging geological classification tasks

(iii) The SHAP method provided valuable insight into feature importance and model interpretability by identifying RHOB and PEF as the most influential input features of NWLs in CNN predictions

(iv) The DIs-ResCNN 2D demonstrates a good level of stability and robustness against Gaussian noise at different noise levels

(v) As part of future research, we will explore generative adversarial networks (GANs) to generate realistic well-logs and use transfer learning methods to enhance model training and generalization. These efforts are aimed at improving the capabilities of DL in addressing complex geological challenges

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Seyed Hamid Reza Mousavi was responsible for conceptualization, methodology, software, and writing the original draft preparation. Seyed Mojtaba Hosseini-Nasab was responsible for supervision, technical advice, technical approaches, writing, reviewing, and editing.

Acknowledgments

We express our gratitude to Iran University of Science and Technology (IUST) for their provision of access to their cloud server infrastructure. Additionally, we appreciate the National Iranian Oil Company (NIOC) for providing the well-log data of the gas field in Iran. Their valuable contributions greatly improved our research. This study has been financially supported by the Iran University of Science and Technology (IUST), with no involvement from any other funding institution. The funding for this study is not international.

References

[1] T. S. Bressan, M. Kehl de Souza, T. J. Girelli, and F. Chemale Junior, “Evaluation of machine learning methods for lithology classification using geophysical data,” *Computers & Geosciences*, vol. 139, article 104475, 2020.

[2] I. M. Mohamed, S. Mohamed, I. Mazher, and P. Chester, “Formation lithology classification: insights into machine learning methods,” in *SPE Annual Technical Conference and Exhibition*, Calgary, Alberta, Canada, September 2019.

[3] W. Wenhua, W. Zhuwen, H. Ruiyi, X. Fanghui, Q. Xinghua, and C. Yitong, “Lithology classification of volcanic rocks based on conventional logging data of machine learning: a case study of the eastern depression of Liaohe oil field,” *Open Geosciences*, vol. 13, no. 1, pp. 1245–1258, 2021.

[4] S. Dong, L. Zeng, D. Xiangyi, J. He, and F. Sun, “Lithofacies identification in carbonate reservoirs by multiple kernel fisher discriminant analysis using conventional well logs: a case study in a oilfield, Zagros basin, Iraq,” *Journal of Petroleum Science and Engineering*, vol. 210, article 110081, 2022.

[5] Y. Liu, Y. Zhang, X. Mao et al., “Lithological facies classification using attention-based gated recurrent unit,” *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 1206–1218, 2024.

[6] Y. Sun, S. Pang, J. Zhang, and Y. Zhang, “DRSN-GAF: deep residual shrinkage network (DRSN) for lithology classification through well logging data transformed by gram angle field,” *IEEE Geoscience and Remote Sensing Letters*, vol. 21, 2023.

[7] Y. Xie, L. Jin, C. Zhu, and S. Wu, “A semi-supervised coarse-to-fine approach with Bayesian optimization for lithology identification,” *Earth Science Informatics*, vol. 16, no. 3, pp. 2285–2305, 2023.

[8] C. Yuan, Y. Wu, Z. Li, H. Zhou, S. Chen, and Y. Kang, “Lithology identification by adaptive feature aggregation under scarce labels,” *Journal of Petroleum Science and Engineering*, vol. 215, article 110540, 2022.

[9] E. L. Faria, J. M. Coelho, T. F. Matos et al., “Lithology identification in carbonate thin section images of the Brazilian pre-salt reservoirs by the computational vision and deep learning,” *Computational Geosciences*, vol. 26, no. 6, pp. 1537–1547, 2022.

[10] F. Alzubaidi, P. Mostaghimi, P. Swietojanski, S. R. Clark, and R. T. Armstrong, “Automated lithology classification from drill core images using convolutional neural networks,” *Journal of Petroleum Science and Engineering*, vol. 197, article 107933, 2021.

[11] M. B. Valentín, C. R. Bom, J. M. Coelho et al., “A deep residual convolutional neural network for automatic lithological facies identification in Brazilian pre-salt oilfield wellbore image logs,” *Journal of Petroleum Science and Engineering*, vol. 179, pp. 474–503, 2019.

[12] S. Chaki, A. Routray, and W. K. Mohanty, “A probabilistic neural network (PNN) based framework for lithology classification using seismic attributes,” *Journal of Applied Geophysics*, vol. 199, article 104578, 2022.

[13] R. Feng, S. M. Luthi, D. Gisolf, and E. Angerer, “Reservoir lithology classification based on seismic inversion results by hidden Markov models: applying prior geological information,” *Marine and Petroleum Geology*, vol. 93, pp. 218–229, 2018.

[14] Y. He, W. Li, Z. Dong et al., “Lithologic identification of complex reservoir based on PSO-LSTM-FCN algorithm,” *Energies*, vol. 16, no. 5, p. 2135, 2023.

[15] Z. Wang, X. Nie, C. Zhang, M. Wang, J. Zhao, and L. Jin, “Lithology classification and porosity estimation of tight gas reservoirs with well logs based on an equivalent multi-component model,” *Frontiers in Earth Science*, vol. 10, article 850023, 2022.

[16] M. Matinkia, R. Hashami, M. Mehrad, M. R. Hajsaeedi, and A. Velayati, “Prediction of permeability from well logs using a new hybrid machine learning algorithm,” *Petroleum*, vol. 9, no. 1, pp. 108–123, 2023.

[17] O. Raheem, W. Pan, C. Torres-Verdín, and M. M. Morales, “Best practices in automatic permeability estimation: machine-learning methods vs. conventional petrophysical models,” in *SPWLA Annual Logging Symposium*, Lake Conroe, TX, USA, June 2023.

[18] A. F. Ibrahim, S. Elkatatny, Y. Abdelraouf, and M. Al Ramadan, “Application of various machine learning techniques in predicting water saturation in tight gas sandstone formation,” *Journal of Energy Resources Technology*, vol. 144, no. 8, article 083009, 2022.

[19] J. Zhang, Y. He, Y. Zhang, W. Li, and J. Zhang, “Well-logging-based lithology classification using machine learning methods for high-quality reservoir identification: a case study of Baikouquan formation in Mahu area of Junggar basin, NW China,” *Energies*, vol. 15, no. 10, p. 3675, 2022.

[20] C. Deng, H. Pan, S. Fang, A. A. Konaté, and R. Qin, “Support vector machine as an alternative method for lithology classification of crystalline rocks,” *Journal of Geophysics and Engineering*, vol. 14, no. 2, pp. 341–349, 2017.

[21] S. Yin, X. Lin, Z. Zhang, and X. Li, “A class-rebalancing self-training semi-supervised learning for imbalanced data lithology identification,” *Geophysics*, vol. 88, no. 6, pp. 1–42, 2023.

[22] T. Merembayev, R. Yunussov, and A. Yedilkhan, “Machine learning algorithms for classification geology data from well logging,” in *2018 14th International Conference on Electronics Computer and Computation (ICECCO)*, pp. 206–212, Kaskelen, Kazakhstan, November-December2018.

[23] H. Ruiyi, W. Zhuwen, W. Wenhua, X. Fanghui, Q. Xinghua, and C. Yitong, “Lithology identification of igneous rocks based on XGBoost and conventional logging curves, a case study of the eastern depression of Liaohe basin,” *Journal of Applied Geophysics*, vol. 195, p. 104480, 2021.

[24] Z. Li, Y. Kang, D. Feng et al., “Semi-supervised learning for lithology identification using laplacian support vector machine,” *Journal of Petroleum Science and Engineering*, vol. 195, article 107510, 2020.

[25] Y. Ao, H. Li, L. Zhu, S. Ali, and Z. Yang, “Logging lithology discrimination in the prototype similarity space with random forest,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 687–691, 2019.

[26] P. Wang, X. Chen, B. Wang, J. Li, and H. Dai, “An improved method for lithology identification based on a hidden Markov model and random forests,” *Geophysics*, vol. 85, no. 6, pp. -IM27–IM36, 2020.

[27] Y. Imamverdiyev and L. Sukhostat, “Lithological facies classification using deep convolutional neural network,” *Journal of Petroleum Science and Engineering*, vol. 174, pp. 216–228, 2019.

[28] C. Jiang, D. Zhang, and S. Chen, “Lithology identification from well-log curves via neural networks with additional geologic constraint,” *Geophysics*, vol. 86, no. 5, pp. IM85–IM100, 2021.

[29] S. Shi, M. Li, W. Gao, G. Shi, J. Bai, and J. Zuo, "Improved Unet in lithology identification of coal measure strata," *Lithosphere*, vol. 2022, Article ID 4087265, 15 pages, 2022.

[30] L. Zhou, S. Lu, Y. Liu, and X. Liu, "Lithology identification based on multi-scale residual one-dimensional convolutional neural network," in *Proceedings of the 3rd International Conference on Big Data Technologies*, pp. 167–171, New York, NY, USA, September 2020.

[31] K. Zhu, Y. Du, Q. Wang, N. Ji, and L. Zhang, "A comparative study of five networks for reservoir classification based on geophysical logging signals," *IEEE Access*, vol. 8, pp. 197776–197786, 2020.

[32] D. T. dos Santos, M. Roisenberg, and M. dos Santos Nascimento, "Deep recurrent neural networks approach to sedimentary facies classification using well logs," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[33] L. Zeng, W. Ren, L. Shan, F. Huo, and F. Meng, "Lithology spatial distribution prediction based on recurrent neural network with kriging technology," *Journal of Petroleum Science and Engineering*, vol. 214, article 110538, 2022.

[34] R. Zhou, X. Hu, B. Yuan, and X. Quanwei, "Lithology classification system for well logging based on bidirectional gated recurrent unit," in *2021 4th International conference on artificial intelligence and big data (ICAIBD)*, pp. 599–603, Chengdu, China, May 2021.

[35] W. Y. Ayele, "Adapting CRISP-DM for idea mining: a data mining process for generating ideas using a textual dataset," *International Journal of Advanced Computer Sciences and Applications*, vol. 11, no. 6, pp. 20–32, 2020.

[36] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021.

[37] S. Studer, T. B. Bui, C. Drescher et al., "Towards CRISP-ML(Q): a machine learning process model with quality assurance methodology," *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, pp. 392–413, 2021.

[38] P. Shammas, "Iran: review of petroleum developments and assessments of the oil and gas fields," *Energy Exploration & Exploitation*, vol. 19, no. 2-3, pp. 207–260, 2001.

[39] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1307–1310, New York, NY, USA, October 2015.

[40] F. Alzubaidi, P. Mostaghimi, G. Si, P. Swietojanski, and R. T. Armstrong, "Automated rock quality designation using convolutional neural networks," *Rock Mechanics and Rock Engineering*, vol. 55, no. 6, pp. 3719–3734, 2022.

[41] J. Gupta, S. Pathak, and G. Kumar, "A hybrid optimization-tuned deep convolutional neural network for bare skinned image classification in websites," *Multimedia Tools and Applications*, vol. 81, no. 18, pp. 26283–26305, 2022.

[42] J. Gupta, S. Pathak, and G. Kumar, "Deep learning (CNN) and transfer learning: a review," *Journal of Physics: Conference Series*, vol. 2273, article 012029, 2022.

[43] F. Alzubaidi, P. Makuluni, S. R. Clark, J. E. Lie, P. Mostaghimi, and R. T. Armstrong, "Automatic fracture detection and characterization from unwrapped drill-core images using mask R–CNN," *Journal of Petroleum Science and Engineering*, vol. 208, article 109471, 2022.

[44] N. J. Alqahtani, Y. Niu, Y. D. Wang et al., "Super-resolved segmentation of X-ray images of carbonate rocks using deep learning," *Transport in Porous Media*, vol. 143, no. 2, pp. 497–525, 2022.

[45] E. Hatır, M. Korkanç, A. Schachner, and İ. İnce, "The deep learning method applied to the detection and mapping of stone deterioration in open-air sanctuaries of the Hittite period in anatolia," *Journal of Cultural Heritage*, vol. 51, pp. 37–49, 2021.

[46] M. E. Hatir and I. Ince, "Lithology mapping of stone heritage via state-of-the-art computer vision," *Journal of Building Engineering*, vol. 34, article 101921, 2021.

[47] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: an overview of the concepts and a survey of the state of the art with focus on MRI," *Journal of Magnetic Resonance Imaging*, vol. 49, no. 4, pp. 939–954, 2019.

[48] D. Riquelme and M. A. Akhloufi, "Deep learning for lung cancer nodules detection and classification in ct scans," *AI*, vol. 1, no. 1, pp. 28–67, 2020.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[50] N. Shahadat and A. S. Maida, "Deep residual axial networks," 2023, https://arxiv.org/2301.04631.

[51] B. Hall, "Facies classification using machine learning," *The Leading Edge*, vol. 35, no. 10, pp. 906–909, 2016.

[52] L. Owen, *Hyperparameter Tuning with Python: Boost Your Machine Learning model's Performance Via Hyperparameter Tuning*, Packt Publishing, 2022.

[53] *François Chollet et al. Keras*, 2015, https://keras.io.

[54] S. Picek, A. Heuser, A. Jovic, S. Bhasin, and F. Regazzoni, "The curse of class imbalance and conflicting metrics with machine learning for side-channel evaluations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2019, no. 1, pp. 1–29, 2018.

[55] P. K. Jadwal, S. Jain, S. Pathak, and B. Agarwal, "Improved resampling algorithm through a modified oversampling approach based on spectral clustering and SMOTE," *Microsystem Technologies*, vol. 28, no. 12, pp. 2669–2677, 2022.

[56] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[57] F. An, X. Li, and X. Ma, "Medical image classification algorithm based on visual attention mechanism-MCNN," *Oxidative Medicine and Cellular Longevity*, vol. 2021, Article ID 6280690, 12 pages, 2021.

[58] J. Li, K. Jin, D. Zhou, N. Kubota, and Z. Ju, "Attention mechanism-based cnn for facial expression recognition," *Neurocomputing*, vol. 411, pp. 340–350, 2020.

[59] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., pp. 4765–4774, Curran Associates, Inc., 2017.

[60] S. M. Lundberg, G. Erion, H. Chen et al., "From local explanations to global understanding with explainable AI for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[61] S. M. Lundberg, G. G. Erion, and S.-I. Lee, *Consistent individualized feature attribution for tree ensembles*, 2019, https://arxiv.org/abs/1802.03888.