Open Journal of Geology, 2024, 14, 578-593
https://www.scirp.org/journal/ojg
ISSN Online: 2161-7589
ISSN Print: 2161-7570

Review of Artificial Intelligence for Oil and Gas Exploration: Convolutional Neural Network Approaches and the U-Net 3D Model

Weiyan Liu

College of Geosciences, China University of Petroleum (Beijing), Beijing, China
Email: wyann@student.cup.edu.cn

How to cite this paper: Liu, W.Y. (2024) Review of Artificial Intelligence for Oil and Gas Exploration: Convolutional Neural Network Approaches and the U-Net 3D Model. *Open Journal of Geology*, **14**, 578-593.
https://doi.org/10.4236/ojg.2024.144024

Received: April 1, 2024
Accepted: April 22, 2024
Published: April 25, 2024

Abstract

Deep learning, especially through convolutional neural networks (CNN) such as the U-Net 3D model, has revolutionized fault identification from seismic data, representing a significant leap over traditional methods. Our review traces the evolution of CNN, emphasizing the adaptation and capabilities of the U-Net 3D model in automating seismic fault delineation with unprecedented accuracy. We find: 1) The transition from basic neural networks to sophisticated CNN has enabled remarkable advancements in image recognition, which are directly applicable to analyzing seismic data. The U-Net 3D model, with its innovative architecture, exemplifies this progress by providing a method for detailed and accurate fault detection with reduced manual interpretation bias. 2) The U-Net 3D model has demonstrated its superiority over traditional fault identification methods in several key areas: it has enhanced interpretation accuracy, increased operational efficiency, and reduced the subjectivity of manual methods. 3) Despite these achievements, challenges such as the need for effective data preprocessing, acquisition of high-quality annotated datasets, and achieving model generalization across different geological conditions remain. Future research should therefore focus on developing more complex network architectures and innovative training strategies to refine fault identification performance further. Our findings confirm the transformative potential of deep learning, particularly CNN like the U-Net 3D model, in geosciences, advocating for its broader integration to revolutionize geological exploration and seismic analysis.

Keywords

Deep Learning, Convolutional Neural Networks (CNN), Seismic Fault Identification, U-Net 3D Model, Geological Exploration

1. Introduction

Artificial intelligence, as a significant branch of computer science, was proposed in 1956. After more than sixty years of development, spanning early theoretical exploration, symbolic reasoning and expert systems, knowledge representation and machine learning, and deep learning, it has gradually integrated into production environments across various fields.

Today, artificial intelligence technologies in production environments are primarily dominated by two branches: machine learning and deep learning. Machine learning involves analyzing and learning from data, enabling computer systems to acquire knowledge from data and perform tasks such as decision-making and prediction based on the acquired knowledge. These algorithms typically rely on feature engineering, where human experts extract features from data and provide them as input to the model. These algorithms have relatively low complexity and apply to various scales and types of data. From a data perspective, traditional machine learning algorithms are somewhat limited by the quality and quantity of data and require appropriate feature engineering to improve model performance. They are primarily applied in structured data domains such as text classification, recommendation systems, and regression analysis. The theory of neural networks originated in the late 19th century with the neuroscientist Cajal's establishment of the neuron doctrine. Later, researchers drew inspiration from this theory and combined it with mathematical ideas to propose artificial neurons [1] (**Figure 1**) capable of processing multiple inputs. Building upon this, artificial neural networks, which simulate the workings of the human brain, were developed by connecting multiple artificial neurons. This development has significantly contributed to the advancement of deep learning.

Deep learning [2] is a specialized form of machine learning. When the shallow layers of neural networks in machine learning are deepened, they evolve into deep learning. Deep learning learns high-level features of data through multi-layer neural network structures and optimizes network parameters using the backpropagation algorithm. Deep learning requires large-scale data, especially in fields such as image recognition and speech recognition, where a large amount of annotated data is needed to train models. It performs well in handling unstructured

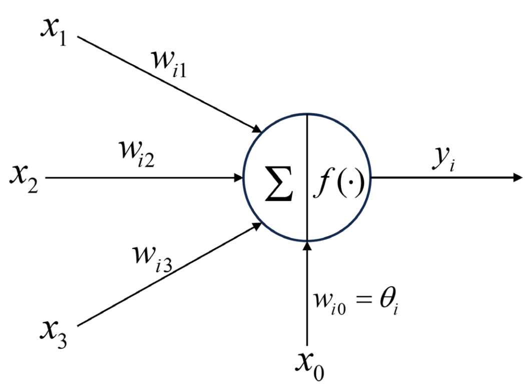

Figure 1. Artificial neuron model [1].

data such as images, speech, and natural language. A network that connects all artificial neurons layer by layer, as shown in Figure 2, is called a fully connected neural network [3] (FCNN). CNN is generally used for image-level classification, while FCN can perform pixel-level classification on images, thus addressing the problem of semantic segmentation at the semantic level. While FCNN are highly flexible, they tend to capture too many data features, resulting in high computational complexity, long training times, and susceptibility to overfitting, making the model overly sensitive to changes in data structure and difficult to generalize. In the early 1960s, researchers such as Hubel and Wiesel proposed the concept of receptive fields [4] through the study of the visual cortex system in cats. By the mid-1980s, Fukushima proposed the neocognitron [5] based on the concept of receptive fields and abstracted the receptive field into convolutional kernels, which can be considered the first implementation of convolutional neural networks [6]. Compared to fully connected neural networks, convolutional kernels achieve similar functionality to visual neural receptive fields through parameter sharing, enabling them to learn features from one part of the input data and apply them to another part. As the network deepens(Figure 3), CNN capture target features from low to high levels in unstructured data, ultimately extracting the necessary information, enhancing the ability to extract main features from data, and reducing the number of parameters needed for training. This has greatly propelled the development of the computer vision field.

In the field of oil and gas exploration, analyzing seismic data to differentiate

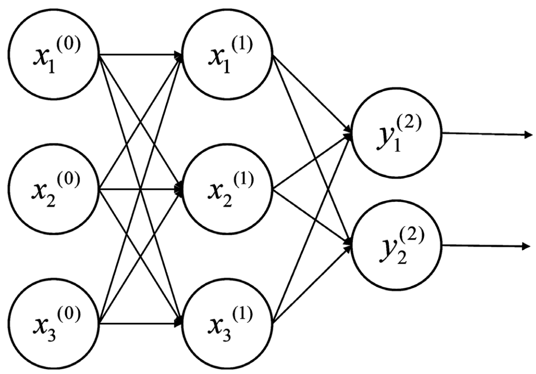

Figure 2. Three-layer artificial neural network.

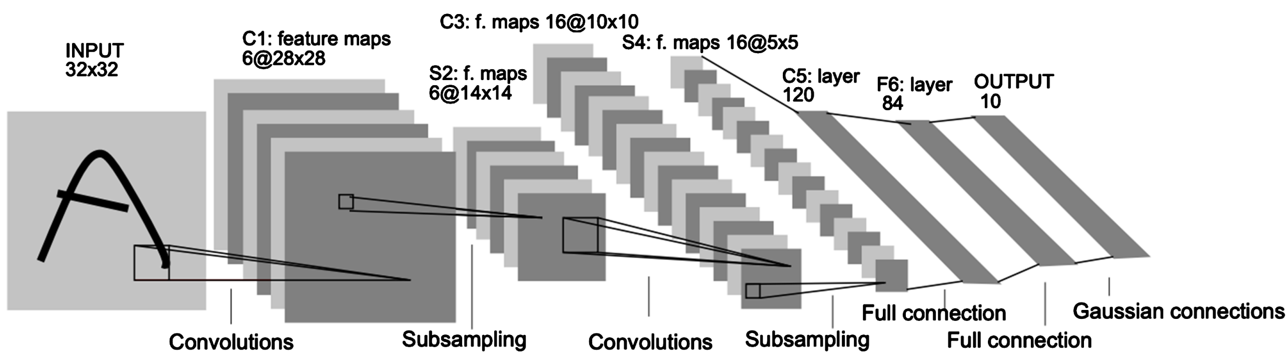

Figure 3. LeNet-5 convolution neural network [6].

between faults and non-faults and subsequently delineating the fault structures in the subsurface of a working area falls under the category of classification problems in the field of artificial intelligence machine learning. In the domain of deep learning, this task belongs to the category of semantic segmentation tasks in computer vision (as shown in **Figure 4**), typically using models based on convolutional neural networks. In traditional machine learning, in 1997, Dong *et al.* utilized adjacent trace cross-correlation coefficients, maximum variance norm, maximum amplitude, and seismic wave attributes to extract data features, which were then inputted into a BP neural network (a shallow neural network, for example, only has three layers.) model (as shown in **Figure 5**) to achieve intelligent identification of small faults. However, this method is sensitive to data and heavily relies on feature engineering to remove redundant data from the original dataset. Moreover, the model is relatively simple and cannot capture too much target information, resulting in weak generalization capability. In 2015, Tan *et al.* used various geometric properties of faults such as fault dip, dip angle, and throw as sample features inputted into a support vector machine (as shown in **Figure 6**). They used a Genetic Algorithm [7] (GA) for parameter tuning to

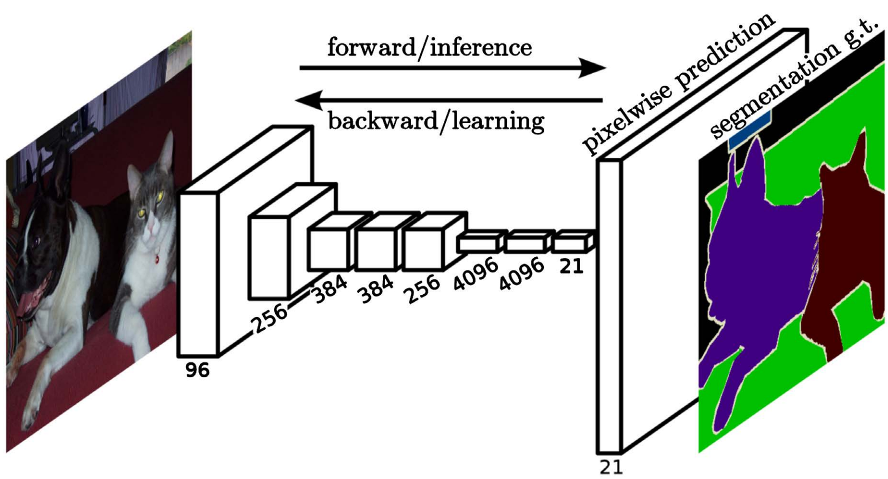

Figure 4. The semantic segmentation results obtained by the FCN network [8].

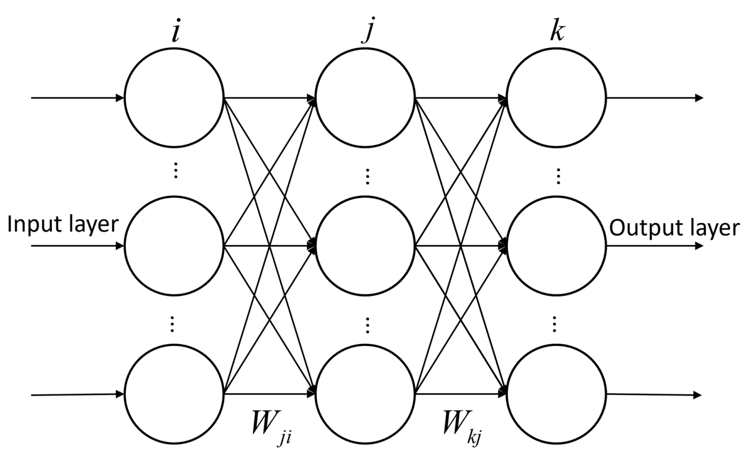

Figure 5. BP Neural Network basic architecture.

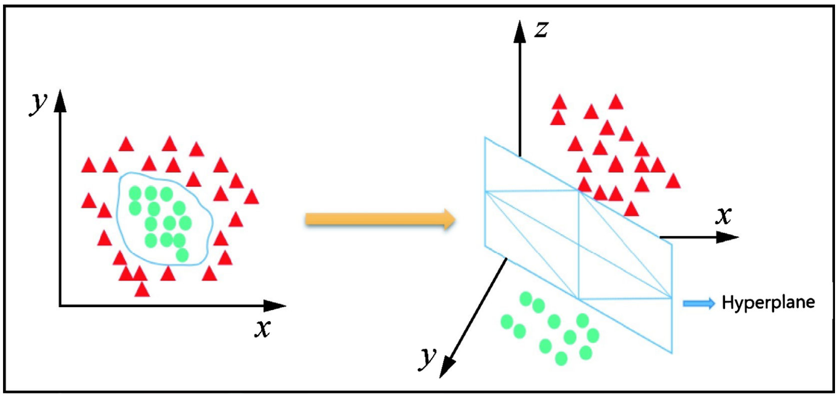

Figure 6. SVM model.

automatically identify faults. However, this method is suitable for a small volume of data samples; once the data volume increases, the model's effectiveness will be significantly reduced. In 2017, Chen *et al.* utilized unsupervised learning methods such as similarity propagation clustering algorithms and Principal Component Analysis (PCA) to identify faults. This method is relatively easier in terms of preprocessing data and does not require much data manipulation or manual labeling. However, due to the lack of human intervention, the model's recognition accuracy is generally average.

With breakthroughs in deep learning theory and computational power, computers have significantly enhanced their ability to process and analyze data. Technical personnel no longer need to perform extensive manual analysis and intervention on data, reducing the technical barriers and time costs associated with data processing. Moreover, they can capture more and deeper features of the data based on large datasets. In 2017, Huang *et al.* [9] applied Convolutional Neural Networks to three-dimensional seismic data, achieving certain effectiveness in fault recognition, thus confirming the feasibility of their approach through internal workstations. In 2018, Zhao *et al.* [10] integrated the SEAM model with CNN and applied principles of image processing to sharpen and smooth three-dimensional data during the recognition process, demonstrating its reliability in the offshore basin of New Zealand. In 2019, Wu *et al.* [11] drew inspiration from the U-Net model designed by Ronneberger *et al.* [12] for cell recognition in medical two-dimensional images and reconstructed it into a simplified end-to-end U-Net structure suitable for three-dimensional seismic data. This structure exhibited good performance in fault range recognition but still requires improvement for identifying brittle rock structures. Following the introduction of the residual module by He *et al.* [13] in 2015, which effectively addressed the degradation of deep learning networks with increasing depth, Liu *et al.* [14] in 2020 and Zhou *et al.* in 2021 separately added residual modules to the U-Net 3D model, enhancing its depth and ability to capture high-dimensional features. Yang *et al.* combined the simplified U-Net 3D model with more complex residual modules and used the Gaussian Importance Map for weighted stitching to eliminate edge effects and achieve continuous fault results. In summary, convolutional neural network-based deep learning models have strong applicability and utility in analyzing three-dimensional seismic data and intelligently interpreting faults, enjoying high recognition in the field of geological exploration.

2. Convolutional Neural Networks

With the deepening of network depth, when the input data is visual, the number of parameters that the network needs to learn is enormous. To address recognition problems in the field of computer vision, Convolutional Neural Networks [15] have emerged based on deep neural networks. CNN is a type of deep learning model commonly used for processing matrix data with grid structures but unstructured, such as images. One of the key innovations of CNN is its ability to directly accept raw grid matrix data as input, automatically extracting features without manual feature extraction. Additionally, due to the existence of convolutional kernels, the network does not need to learn too many redundant parameters, greatly reducing the time required for training. Below are the main organizational structures of convolutional neural networks: 1) Convolutional Layer: The convolutional layer is one of the most important components of CNN. In the convolutional layer, features of the input data are extracted by applying a series of convolutional kernels (or filters) to the input data. Each convolutional kernel performs convolutional operations with the input data, producing a set of output feature maps. 2) Activation Layer: Typically, following each convolutional layer, there is an activation layer, such as the ReLU activation function. Its purpose is to introduce nonlinearity, allowing the network to learn more complex features. 3) Pooling Layer: The pooling layer is used to reduce the spatial dimensions of the feature maps, thereby decreasing the model parameters and computational complexity, while enhancing the robustness of the features. Common pooling operations include Max Pooling and Average Pooling. 4) Fully Connected Layer: The fully connected layer flattens the outputs of the convolutional and pooling layers and connects them to the neural network's output layer. It is typically used for performing final classification or regression tasks.

With the development of deep learning theory, depending on the application scenario, convolutional neural networks can appropriately add layers such as Dropout Layer [16] and Batch Normalization Layer [17] to improve the model's performance.

2.1. Receptive Field

Receptive Field: Inspired by the biological visual nervous system, it refers to the range that a neuron or a layer of neurons can "see" or influence input information. It describes the size of the local region from which a neuron receives information, aiding in understanding the extent to which each neuron in a neural network perceives local information from the input. It is a part of the input space, and its information can influence the output of that neuron through the convolution and pooling layers of the network. The receptive field of each subsequent layer's neuron includes a larger range from the previous layers. The receptive field has several important aspects:

1) Local Connectivity: In the primary convolutional layers, the receptive field is typically small (such as a 3×3 or 5×5 pixel region), allowing it to capture local features. This design enables the network to recognize simple visual patterns like edges, color patches, and so on.

2) Hierarchy: As shown in **Figure 7**, as the layers deepen, more and more shallow features are extracted and converged together, compressing the data. For the initial data, the receptive field of each neuron in the deeper layers gradually expands, covering larger areas of the input image. This allows the network to detect more complex, global features in higher layers, such as parts or the overall shape of objects.

The concept of receptive field helps to understand how convolutional kernels extract features from input data and achieve hierarchical decomposition of complex functions in the network structure. For example, in applications of image processing, lower layers may focus on textures and contours, while deeper layers may focus on recognizing specific objects and scenes. By designing networks with appropriate receptive fields, models can better adapt to different data features and learning tasks.

2.2. Convolutional Kernel

The convolutional kernel, also known as a filter, is essentially the embodiment of the receptive field concept in CNN. It is a parameter matrix whose size is specified based on the specific problem. In the convolution operation, the kernel slides over the input data with a certain stride and performs element-wise multiplication with the local region of the input data. The results are then summed to obtain the output. This process effectively extracts local features from the

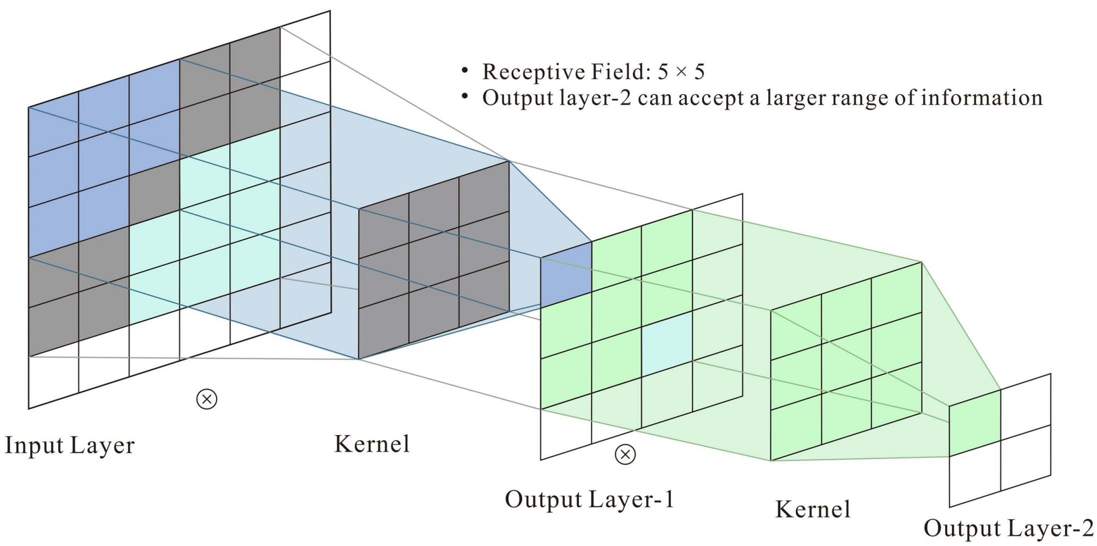

Figure 7. Feature extraction layer by layer [18].

input data, and different kernels can extract different features. It has the following characteristics:

1) Size: The convolutional kernel is a small matrix. For data represented as flat surfaces (such as images), a two-dimensional convolutional kernel is used; for data represented as volumes (such as three-dimensional seismic data), a three-dimensional convolutional kernel is used. This has the following implications: a) Receptive Field Size: As shown in **Figure 7**, the size of the convolutional kernel determines the receptive field of the neuron. A larger convolutional kernel can cover a larger area, capturing more extensive features, but may lose some details. Smaller convolutional kernels focus on smaller areas, capturing finer features. b) Parameter Count: A larger convolutional kernel size will increase the number of parameters in the model, which may lead to overfitting, especially when the training data is limited. Additionally, larger convolutional kernels also increase computational complexity. In contrast, smaller convolutional kernels, while reducing the number of parameters and computational load, may require more layers to cover the same receptive field range. c) Feature Learning Capability: Larger convolutional kernels may be more suitable for learning global features, such as overall shape and structure, while smaller convolutional kernels are better suited for learning local details, such as edges and textures. By carefully selecting the convolutional kernel sizes, the network can effectively extract useful features while maintaining lower computational complexity.

Common convolutional kernel sizes include 1, 3, 5, and 7 units. Convolutional kernels with a size of 1 unit are typically used for cross-channel information fusion and dimensionality reduction without altering the spatial dimensions of the feature maps. Kernels with a size of 3 units are one of the most used kernel sizes, providing a good balance between effectively extracting local features while maintaining lower computational complexity. Kernels with sizes of 5 and 7 units can capture a wider range of features but increase computational complexity.

2) Depth: The depth of a convolutional kernel is an important concept in Convolutional Neural Networks, and it, along with the size (width and height) of the kernel, determines the overall dimensionality of the kernel. The depth of a convolutional kernel refers to its size along the channel dimension of the input data, essentially representing the depth or number of channels considered in the convolution operation. In the context of processing color images (composed of red, green, and blue color channels in the RGB color space, typically represented by three channels), as shown in **Figure 8**, the convolutional kernel also has three channels, with each channel dedicated to processing the corresponding channel of the input image. Its functionality and impact are as follows: a) Feature Integration: By applying different channels of the convolutional kernel to the corresponding channels of the input data and summing the results, the convolution operation can integrate feature information from different input channels. This mechanism enables CNN to effectively handle multi-channel data, such as color images, while capturing inter-channel feature correlations. b) Parameter Features: The increase in the depth of the convolutional kernel directly affects the

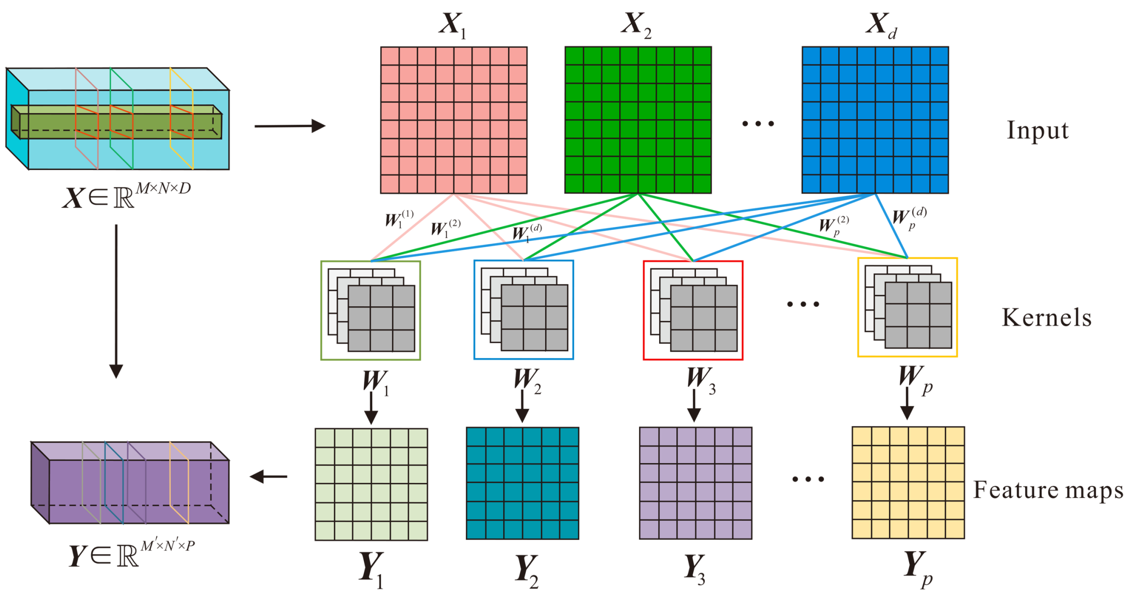

Figure 8. Extracting features from a color image using convolutional kernels [19].

number of parameters in the model. For a given size of the convolutional kernel, the greater the depth, the more parameters it contains, which may lead to an increase in computational cost. However, it also enhances the model's ability to capture inter-channel features. c) Feature Map Depth: In the convolutional layer, the depth of the output feature map is determined by the number of convolutional kernels in that layer, rather than the depth of the kernels. Each convolutional kernel generates one feature map, so using more kernels can produce deeper feature maps, enhancing the model's ability to capture different features.

The depth of the convolutional kernel is an important hyperparameter in the design of convolutional neural networks. Unlike parameters, which are typically optimized during network training, kernel depth influences the number of parameters, computational costs, and the model's feature extraction capabilities. By carefully designing the size, depth, and number of convolutional kernels, efficient and powerful CNN models can be constructed.

3) Quantity: The quantity of convolutional kernels refers to the number of kernels used in the convolutional layer of a Convolutional Neural Network. In CNN, multiple different kernels are typically used at each layer, with each kernel responsible for extracting specific features. In **Figure 8**, there are at least four kernels depicted. By employing multiple kernels, the network can learn features at different scales and abstraction levels, thereby enhancing its understanding of the input data. For instance, in image classification tasks, the kernels in the first layer might learn to extract low-level features such as edges and textures, while kernels in subsequent layers might learn to extract higher-level features such as shapes and object parts.

The quantity of convolutional kernels is typically specified by the user and needs to be adjusted based on the complexity of the task and the characteristics of the dataset when designing the network architecture. Increasing the number of kernels can increase the network's parameter count and computational complexity, but it can also enhance the network's feature extraction and representation capabilities. Therefore, appropriate trade-offs and adjustments need to be made in practice.

2.3. Convolutional Layer

The convolutional layer is the core component of Convolutional Neural Networks, and the convolutional kernel is the key to enabling the powerful feature extraction capability of the convolutional layer. It is used to automatically extract features from the input grid-like data matrices. Taking the example of two-dimensional grid data, such as images in the context of image processing, the convolutional layer applies a series of learnable convolutional kernels to each region of the input image in a sliding manner to identify different spatial hierarchical structures such as edges, color patches, and textures.

As shown in **Figure 9**, taking image data as an example (with only height, width, and channels), the computation of feature extraction by the convolutional kernel (or filter) involves a process of weighted summation of the input data, typically including a bias term. Eventually, a numerical value is computed, representing a pixel point after convolution. This process can be simplified and represented by the following formula:

$$Y[i,j]=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}X[i+m,j+n]\cdot K[m,n]+b \tag{1}$$

where $Y[i,j]$ represents the output feature map at the i-th row and j-th column, M and N are the number of rows and columns of the convolutional kernel K respectively, $X[i+m,j+n]$ denotes the element at the ($i+m$)-th row and ($j+n$)-th column of the input data X, $K[m,n]$ refers to the element at the m-th

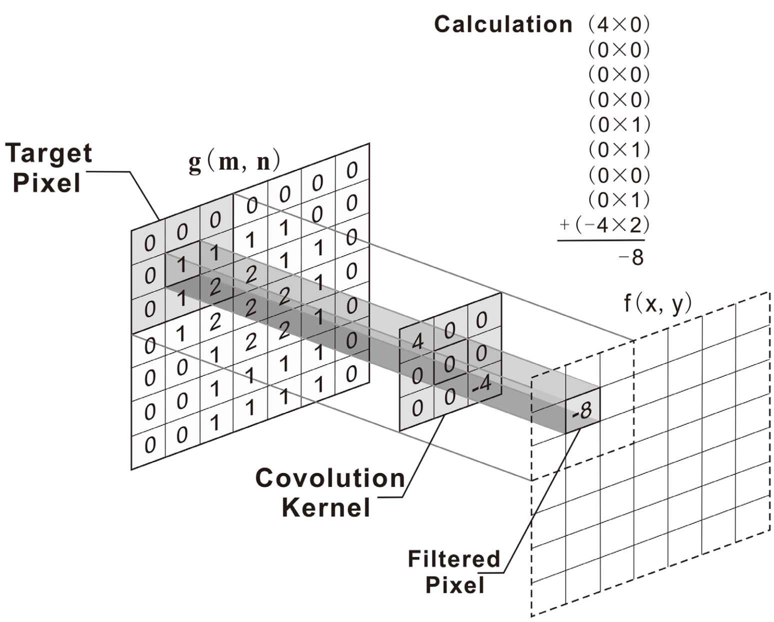

Figure 9. Feature extraction by weighted sum of convolutional kernels [20].

row and n-th column of the convolutional kernel K, and b is the bias term.

This formula represents the basic principle of convolution operation: it involves element-wise multiplication of the local region of the input data with the convolutional kernel, followed by summation of the results, resulting in the value of the corresponding element in the output feature map.

As the convolutional kernel slides over the data matrix, it sequentially captures feature information from different regions. During the feature extraction process, as more features are extracted, the number of channels increases, while the data volume decreases. Taking an image as an example, suppose the size of the input data is $W_{in} \times H_{in} \times D_{in}$, and the spatial dimensions of the output feature map are defined. Here, W_{in} represents the width of the input data, H_{in} represents the height of the input data, and D_{in} represents the depth of the input data (number of channels). Let's assume the size of the convolutional kernel is $F \times F \times D_{in}$, where F is the width and height of the kernel, P is the amount of padding (to mitigate edge effects), and S is the stride length, The convolution kernel moves horizontally first along the width dimension. When one row is finished, it moves vertically along the height dimension, opening a new row. The formula for the reduced volume is as follows:

$$W_{out} = \frac{W_{in} - F + 2 \times P}{S} + 1 \tag{2}$$

$$H_{out} = \frac{H_{in} - F + 2 \times P}{S} + 1 \tag{3}$$

where W_{out} is the width of the output data, H_{out} is the height of the output data, and D_{out} is the depth of the output data (number of channels).

2.4. Pooling Layer

The Pooling Layer is an important component of Convolutional Neural Networks, implementing downsampling in deep learning. It is used to reduce the size and parameter count of feature maps while retaining the most important feature information regardless of geometric transformations applied to the original data. Typically following convolutional layers, the Pooling Layer is independently applied to each depth slice of the feature maps, reducing dimensionality and sampling the features extracted by the convolutional layers. The size of the pooled feature maps is usually determined by the pooling size and stride.

The main functions of the Pooling Layer include the following aspects:

1) Dimensionality reduction and parameter reduction: Pooling operations reduce the size of feature maps, thereby decreasing the computational complexity of subsequent layers. This helps to reduce the number of model parameters and computational burden, while enhancing the training efficiency of the model.

2) Maintaining feature invariance: Pooling operations, often performed using methods like max pooling or average pooling, compute the maximum or average value within local regions, thereby preserving important features. This helps the network to exhibit a degree of invariance to transformations such as translation,

rotation, and scaling of the input data. Max pooling focuses on capturing the predominant features within the receptive field. The formula is as follows:

$$Y[i, j] = \max_{m,n}\left(X[i \times S + m, j \times S + n]\right) \tag{4}$$

where $Y[i, j]$ denotes an element in the output feature map, S represents the stride, and m and n respectively represent the position of the pooling window.

The average pooling focuses on capturing global changes within the receptive field, and the formula is as follows:

$$Y[i, j] = \frac{1}{F \times F}\sum_{m=0}^{F-1}\sum_{n=0}^{F-1}\left(X[i \times S + m, j \times S + n]\right) \tag{5}$$

where $Y[i, j]$ represents an element in the output feature map, S denotes the stride, F indicates the single-side length of the pooling window, and m and n respectively represent the position of the pooling window.

3) Improve model generalization: Pooling operation effectively reduces the size of the feature maps while retaining the most significant features, thereby reducing the risk of overfitting and enhancing the model's generalization capability. This helps the model adapt better to new, unseen data.

In summary, convolutional neural networks can be seen as an extension of basic deep neural networks. They capture the most significant features in the raw data with minimal parameters through training. After passing through convolutional and pooling layers, the specific output data type required for the target task is adjusted. Figure 10 depicts the classification operation after pooling and flattening the data.

3. U-Net 3D Model

U-Net [12] is a convolutional neural network initially designed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015 for medical image segmentation tasks. The architecture of U-Net (as shown in Figure 11) is specifically

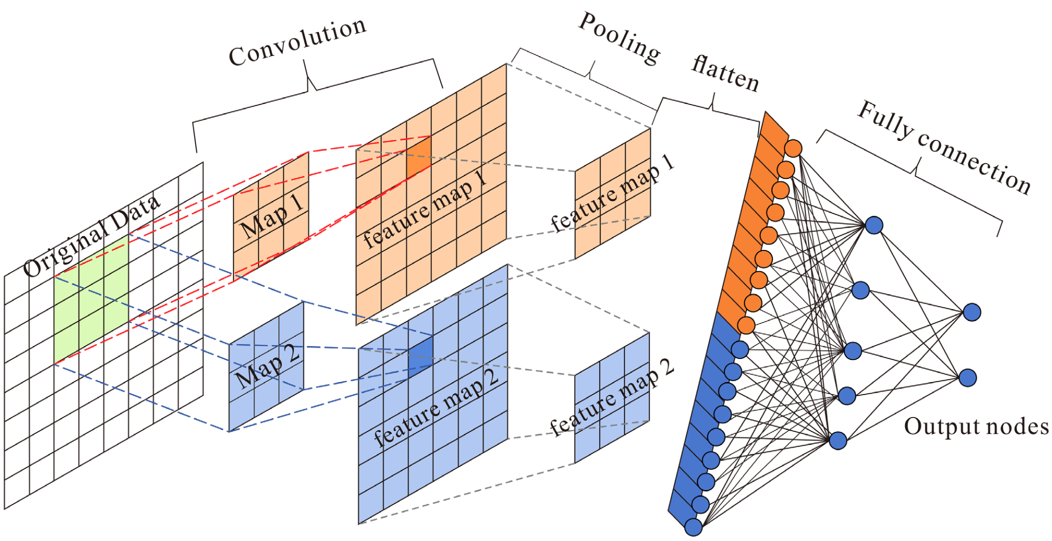

Figure 10. Working principle of CNN [21].

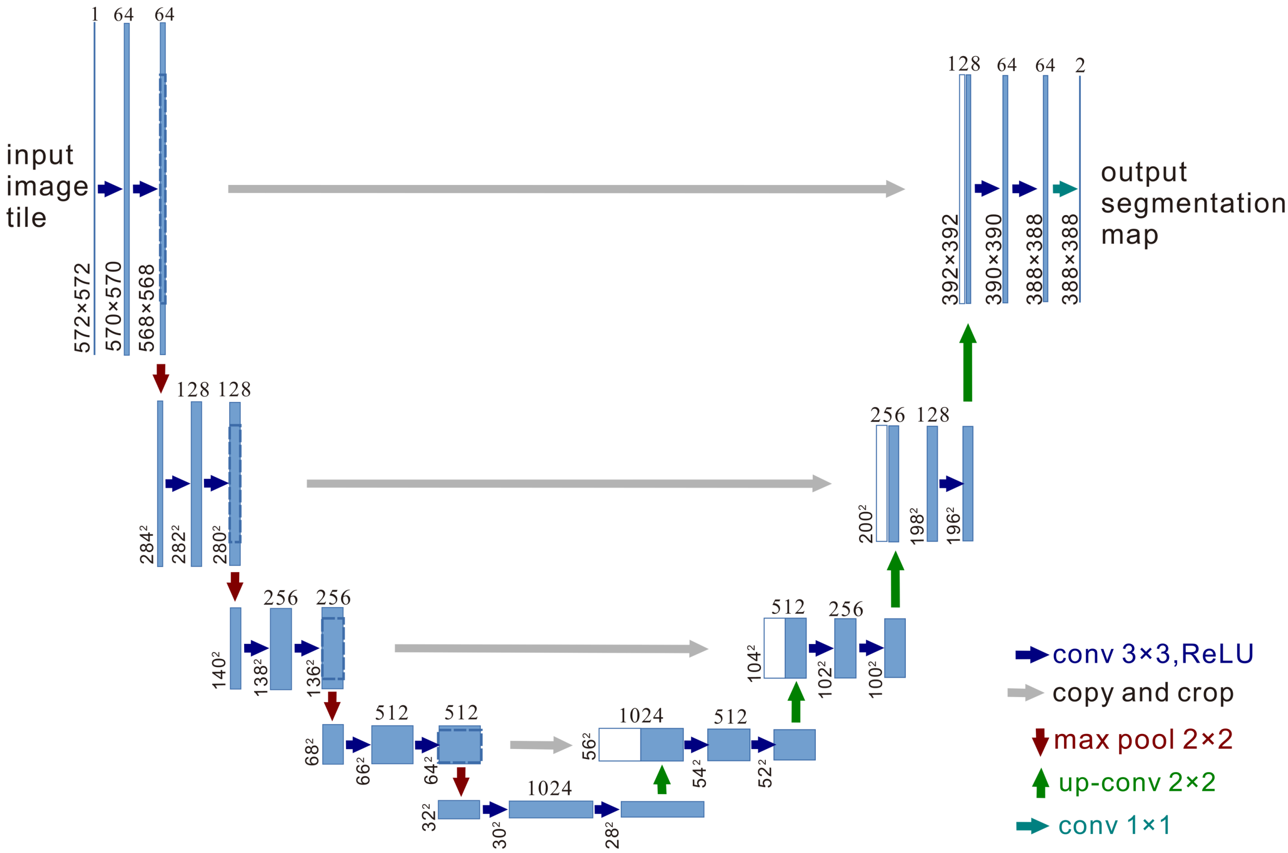

Figure 11. U-Net basic model structure [12].

designed to effectively handle small datasets, hence its name. It improves model performance and generalization by employing extensive data augmentation techniques. U-Net belongs to the category of semantic segmentation models, aiming to differentiate between target and non-target points in the original data (as depicted in **Figure 4**). The original U-Net model consists of three main components: the encoder, the decoder, and skip connections. The encoder typically comprises multiple convolutional layers and pooling layers, aimed at progressively reducing the spatial dimensions of the feature maps while increasing their depth. This helps the network capture contextual information from the input images. The decoder consists of a series of upsampling operations and convolutional layers, used to gradually restore the spatial dimensions of the feature maps while reducing their depth. The goal of the decoder is to reconstruct the precise details of the images and segmentation boundaries. Skip connections combine the feature maps from the encoder with those from the corresponding layers in the decoder. Through this mechanism, the model can preserve more detailed information in the output segmentation maps.

In 2019, Wu *et al.* [11] simplified the traditional U-Net model by reducing the number of skip connections from 4 layers to 3 layers. They also modified the original input data from 2D images to 3D seismic data volumes (as shown in **Figure 12**) and compressed the edge length of the data from 572 to 128. The U-Net 3D

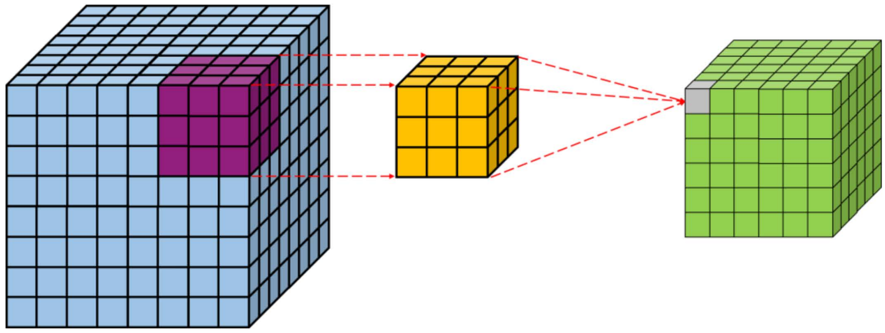

Figure 12. 3D Convolution.

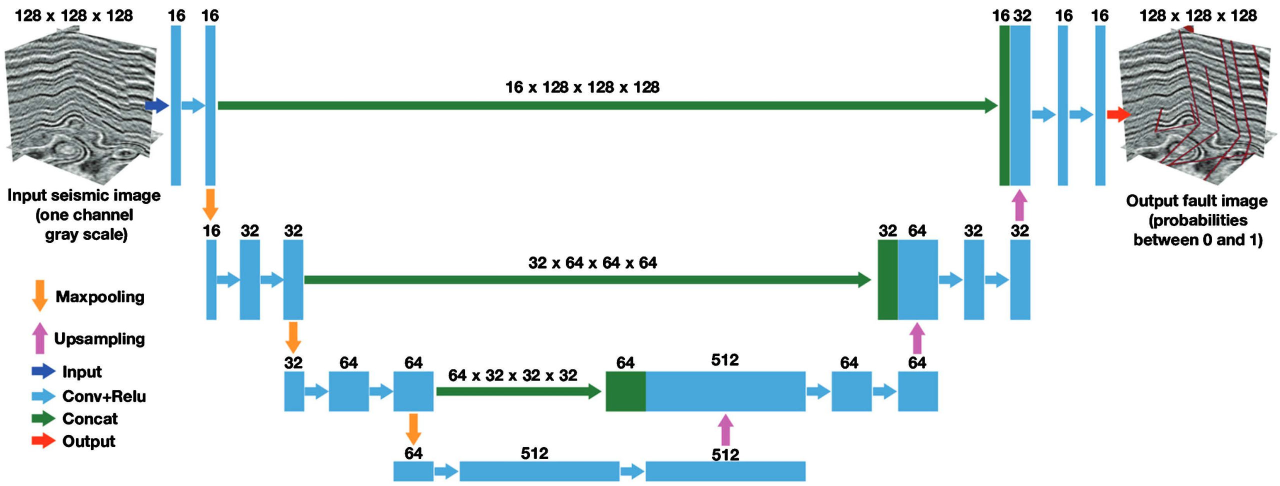

Figure 13. End-to-end U-Net model structure [11].

model (Figure 13) exhibited good performance in constructing relatively simple working areas.

In practice, despite the significant potential demonstrated by U-Net in seismic fault identification, it still faces some challenges in real-world applications: 1) Data Preprocessing: The quality and consistency of seismic data have a significant impact on the performance of the model, requiring effective data preprocessing to enhance the model's generalization ability. 2) Acquiring annotated data: High-quality annotated data is crucial for training accurate models, but manually annotating faults in seismic data is a time-consuming and specialized task. 3) Model generalization: Seismic data may vary significantly across different regions and conditions, posing a challenge to developing models with good generalization capability.

4. Conclusions

In conclusion, our comprehensive review of convolutional neural networks (CNN), particularly focusing on the U-Net 3D model for fault identification in seismic data, highlights a significant shift towards leveraging advanced deep learning techniques in the realm of oil and gas exploration. This shift is not merely a reflection of technological advancement but marks a pivotal change in how geoscientists approach data interpretation and analysis.

The evolution from manual feature extraction and traditional machine learning algorithms to sophisticated CNN architectures signifies a broader acceptance of AI's potential to transform complex, labor-intensive tasks into more efficient, automated processes. Our examination of various CNN approaches, culminating in the adaptability and effectiveness of the U-Net 3D model, underscores deep learning's remarkable ability to handle voluminous, unstructured seismic datasets, enhancing fault detection accuracy while reducing human intervention. This review elucidates the importance of ongoing research and development in deep learning for geoscience applications. It emphasizes the need for continual improvement in models to address challenges such as data quality, annotation scarcity, and generalization across diverse geological settings. Our discussion on the inception of CNN, their structural nuances, and their pivotal role in automating seismic fault interpretation serves as a testament to the transformative impact of deep learning in geosciences. Looking forward, future work must explore more intricate network architectures, innovative training strategies, and comprehensive models tailored to diverse geological conditions. By doing so, we can further enhance fault identification performance, contributing to more effective exploration strategies and a deeper understanding of the Earth's subsurface. The promising results reviewed here lay a solid foundation for such advancements, steering the geoscience community towards a more integrated, AI-driven future in geological exploration and analysis.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] McCulloch, W.S. and Pitts, W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*, **5**, 115-133. https://doi.org/10.1007/BF02478259

[2] Zhang, R., Li, W. and Mo, T. (2018) Review of Deep Learning. https://doi.org/10.48550/arXiv.1804.01653

[3] Zeng, P. (1996) Artificial Neural Networks Principle for Finite Element Method. *Zeitschrift für Angewandte Mathematik und Mechanik*, **76**, 565-566.

[4] Hubel, D.H. and Wiesel, T.N. (1962) Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology*, **160**, 106-154. https://doi.org/10.1113%2Fjphysiol.1962.sp006837

[5] Hinton, G.E. and Salakhutdinov, R.R. (2006) Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**, 504-507. https://doi.org/10.1126/science.1127647

[6] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**, 2278-2324. https://doi.org/10.1109/5.726791

[7] Katoch, S., Chauhan, S.S. and Kumar, V. (2021) A Review on Genetic Algorithm: Past, Present, and Future. *Multimedia Tools and Applications*, **80**, 8091-8126.

https://doi.org/10.1007/s11042-020-10139-6

[8] Long, J., Shelhamer, E. and Darrell, T. (2015) Fully Convolutional Networks for Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 3431-3440.

[9] Huang, L., Dong, X. and Clee, T.E. (2017) A Scalable Deep Learning Platform for Identifying Geologic Features from Seismic Attributes. *The Leading Edge*, **36**, 249-256. https://doi.org/10.1190/tle36030249.1

[10] Zhao, T. and Mukhopadhyay, P. (2018) A Fault Detection Workflow Using Deep Learning and Image Processing. *SEG International Exposition and Annual Meeting*, Anaheim, 14-19 October 2018, SEG-2018. https://doi.org/10.1190/segam2018-2997005.1

[11] Wu, X., Liang, L., Shi, Y. and Fomel, S. (2019) FaultSeg3D: Using Synthetic Data Sets to Train an End-to-End Convolutional Neural Network for 3D Seismic Fault Segmentation. *Geophysics*, **84**, IM35-IM45. https://doi.org/10.1190/geo2018-0646.1

[12] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention-MICCAI* 2015: 18*th International Conference*, Munich, 5-9 October 2015, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[13] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. http://arxiv.org/abs/1512.03385

[14] Liu, N., He, T., Tian, Y., Wu, B., Gao, J. and Xu, Z. (2020) Common-Azimuth Seismic Data Fault Analysis Using Residual UNet. *Interpretation*, **8**, SM25-SM37. https://doi.org/10.1190/INT-2019-0173.1

[15] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. https://doi.org/10.1038/nature14539

[16] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, **15**, 1929-1958. https://dl.acm.org/doi/10.5555/2627435.2670313

[17] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *International Conference on Machine Learning*, Lille, 7-9 July 2015, 448-456. http://arxiv.org/abs/1502.03167

[18] Gong, H., Liu, L., Liang, H., Zhou, Y. and Cong, L. (2023) A State-of-the-Art Survey of Deep Learning Models for Automated Pavement Crack Segmentation. *International Journal of Transportation Science and Technology*, **13**, 44-57. https://doi.org/10.1016/j.ijtst.2023.11.005

[19] Yang, K., Yang, T., Yao, Y. and Fan, S.D. (2021) A Transfer Learning-Based Convolutional Neural Network and Its Novel Application in Ship Spare-Parts Classification. *Ocean & Coastal Management*, **215**, Article ID: 105971. https://doi.org/10.1016/j.ocecoaman.2021.105971

[20] Aparna, S. and Naidu, M.E. (2016) Applying FIR and IIR Digital Filters over Video Image Processing. *International Journal of Applied Engineering Research*, **11**, 7624-7632.

[21] Maeda, K., Takahashi, S., Ogawa, T. and Haseyama, M. (2017) Automatic Estimation of Deterioration Level on Transmission Towers via Deep Extreme Learning Machine Based on Local Receptive Field. 2017 *IEEE International Conference on Image Processing* (*ICIP*), Beijing, 17-20 September 2017, 2379-2383. https://doi.org/10.1109/ICIP.2017.8296708