

Automatic Velocity Picking Based on Improved Mask R-CNN

Yijun Yuan^{ID}, Ying Li, Xiaoxuan Fang^{ID}, and Fengfeng Shi

Abstract—Velocity spectrum analysis is the main method used to determine the normal moveout (NMO) velocity. However, conventional velocity spectrum analysis methods are inefficient and are typically affected by subjective factors, resulting in a deviation of the selected velocity from the actual velocity. Consequently, we developed a velocity-picking method that uses a mask region-based convolutional neural network (Mask R-CNN). The proposed method first generates a large number of training datasets, including velocity spectrum and labels, and then feeds them to the network for training. After the network is trained with the training data, it automatically extracts time–velocity functions from the velocity spectrum and outputs them for subsequent data processing. The proposed method converts the picking of velocity-energy clusters in seismic data processing into object detection in image processing. Therefore, to enable the method to accurately detect velocity-energy clusters in the velocity spectrum, according to the characteristics of velocity-energy clusters, the anchor scale and anchor ratio of the traditional Mask R-CNN are improved and a Mask R-CNN structure suitable for velocity-energy cluster detection is constructed. We solved the problem of object duplicate detection in the traditional Mask R-CNN by improving the nonmaximum suppression algorithm. The results of experiments conducted on synthetic and field data indicate that the proposed method quickly and accurately extracts time–velocity functions from the velocity spectrum. Furthermore, it obtains better results for velocity picking, NMO-corrected common midpoint gathers, and stacked sections than the traditional Mask R-CNN and the man–computer interaction velocity-picking (MIVP) method.

Index Terms—Mask region-based convolutional neural network (Mask R-CNN), object detection, velocity picking.

I. INTRODUCTION

IN SEISMIC data processing, the normal moveout (NMO) velocity is one of the most important parameters. The accuracy of NMO velocity affects the reliability of the stacked

Manuscript received 21 February 2023; revised 3 July 2023, 27 August 2023, and 7 November 2023; accepted 12 November 2023. Date of publication 28 November 2023; date of current version 1 December 2023. This work was supported by the National Natural Science Foundation of China under Grant 41974157. (*Corresponding author: Yijun Yuan.*)

Yijun Yuan is with the School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China and the Key Laboratory of Intraplate Volcanoes and Earthquakes (China University of Geosciences, Beijing), Ministry of Education, Beijing 100083, China (e-mail: yyj@cugb.edu.cn).

Ying Li is with the School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China, and also with BYD Ltd., Shenzhen 518000, China (e-mail: 1752675330@qq.com).

Xiaoxuan Fang is with the School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China (e-mail: 3032849646@qq.com).

Fengfeng Shi is with the Exploration and Development Research Institute, PetroChina Huabei Oilfield Company, Renqiu 062552, China (e-mail: wty_sff@petrochina.com.cn).

Digital Object Identifier 10.1109/TGRS.2023.3335250

and migrated data. The man–computer interaction velocity-picking (MIVP) method based on the velocity spectrum is the most commonly used velocity analysis method in the industry. The velocity-energy cluster is interactively selected in the velocity spectrum generated by the common midpoint (CMP) gathers to provide parameters for stack or migration processing. However, the MIVP method is susceptible to subjective factors [1]. When multiple solutions exist in the velocity spectrum, manual picking typically deviates from the true velocity, resulting in incorrect imaging. Additionally, MIVP is time-consuming and has difficulty in adapting to large-scale data processing [2], [3]. Therefore, a method to determine the NMO velocity quickly and accurately is needed for seismic data processing. Various automatic picking methods have been developed. They can be divided into two main categories: optimization search methods and neural network-based velocity analysis methods.

The optimization search method employs the maximum similarity criterion to automatically search for the global optimal solution in the velocity spectrum, that is, the maximum superimposed energy value, to build an accurate velocity field. Toldi [4] regarded velocity analysis as an optimization problem, used a predefined objective function to calculate the path curve with the maximum stacking energy, and then automatically built the velocity field through inversion. Lin et al. [5] proposed an automatic velocity-picking method based on the Viterbi algorithm. Their proposed method uses the Viterbi operator to perform forward integration of the high-energy peaks in the velocity spectrum and recursively calculated the optimal solution backward to automatically pick the stacking velocity. Later, Lin et al. [6] combined the Viterbi algorithm with a surface fairing to smooth the surface of the velocity spectrum, resulting in more precise velocity picking. Zhang et al. [7] introduced an intelligent velocity-picking method based on a nonlinear function that uses the determined nonlinear function as a constraint to realize intelligent picking of the velocity spectrum. In addition to the aforementioned methods, optimization search methods based on Monte Carlo algorithms [8], [9] have been applied to automatic velocity picking. Although these methods can automatically perform velocity spectrum analysis, they have limitations. For example, appropriate constraints must be set—when the given initial velocity deviates from the true velocity, the automatically picked velocity is incorrect.

The neural network-based velocity-picking method is an early method for automatically determining the NMO velocity. Fish and Kusuma [10] proposed a neural network-based automatic picking method that involves gridding the velocity

spectrum, feeding the picked points and their surrounding grids to a neural network for training, and automatically selecting possible velocity points near the curve according to a guidance function. Dong and He [11] realized the automatic picking of the stacking velocity using backpropagation neural networks. In recent years, various network algorithms have been developed for automatic velocity picking. Araya-Polo et al. [12] introduced a deep neural network that builds a 2-D velocity field. However, their method was only tested on synthetic data and was not validated using real data. Zhang et al. [2] used a combination of two deep neural networks—You Only Look Once (YOLO) and long short-term memory (LSTM)—to perform velocity picking. Biswas et al. [13] proposed a recurrent neural network-based NMO velocity estimation method. Their method estimates NMO velocities directly from CMP gathers. However, due to the presence of multiples and noises in CMP gathers, extracting velocity-related information is difficult [14]. Therefore, the accuracy of velocity estimation depends on the quality of the CMP gathers. Recently, with the successful application of convolutional neural networks (CNNs) in seismic exploration, such as noise attenuation [15], [16] and fault interpretation [17], CNN-based velocity picking has become an active research topic. Various applications of CNN-based velocity-picking methods have been reported. Ma et al. [18] proposed an automatic velocity-picking method based on CNNs. However, this method was tested only on a 2-D synthetic dataset. Ferreira et al. [3] used a hybrid regression approach based on CNNs to automatically select the velocity functions. Similar to the method proposed by Biswas et al. [13], this method requires CMP gathers to be fed to the network for training. In addition, they only evaluated the performance of their proposed method for high-signal-to-noise ratio (SNR) data; they did not verify its effectiveness for low-SNR data. In addition to the methods discussed above, various unsupervised machine-learning methods for automatic velocity estimation have been proposed [19], [20], [21], [22].

YOLO is an object-detection algorithm used in image processing. Velocity spectrum picking can be regarded as a problem of object detection in image processing. Therefore, YOLO can be applied to velocity spectrum picking [2]. However, due to YOLO's direct location and classification of all objects on the entire image, certain small objects cannot be accurately detected. A region-based CNN (R-CNN) is an effective object-detection method that combines CNNs with region proposal generation algorithms. Because proposal generation provides geometric information that is vital for accurate object localization, excellent results have been obtained, e.g., selective search [23]. However, the introduction of a region proposal increases the computational cost. To increase the efficiency of R-CNN, Girshick [24] proposed Faster R-CNN. Compared with R-CNN, Faster R-CNN no longer extracts CNN features from each region proposal but first feeds the image to the CNN for feature extraction, and then, the region proposal extracts the feature map. In addition, a region of interest (ROI) pooling layer is applied. However, ROI pooling typically makes the processed feature map inconsistent with the original image, reducing the detection accuracy. To address the shortcomings of Faster R-CNN,

He et al. [25] proposed Mask R-CNN. By improving the pooling rule, Mask R-CNN solves the problem of mismatching in various regions of the image and realizes the simultaneous detection and classification of objects. Mask R-CNN has been widely used in medicine, remote sensing, shipping, and autonomous driving. However, there are no reports on the application of Mask R-CNN to seismic data processing.

The velocity spectrum generated from CMP gathers can be regarded as a 2-D spatial matrix. The energy clusters in different spaces in the spectrum represent different velocity peaks—i.e., candidate NMO velocities—that increase with time. According to these characteristics, the extraction of the true velocity from candidate NMO velocities can be converted into the problem of object detection in image processing. In contrast to CMP gathers, noise has minimal impact on velocity spectra. Therefore, using velocity spectra to predict NMO velocity values can improve the network's antinoise ability [14]. In this study, we introduced Mask R-CNN to the field of seismic exploration and developed a Mask R-CNN-based velocity-picking method. We constructed a Mask R-CNN algorithm for velocity spectrum picking by improving the traditional anchor scale and ratio. In addition, we improved the nonmaximum suppression (NMS) algorithm to solve the problem of object duplication detection. The results of this study indicate that the proposed method can reduce the cost of seismic data processing by increasing the efficiency of velocity picking and can obtain accurate NMO velocities to improve seismic data processing. The remainder of this article is organized as follows. Section II presents the theory behind Mask R-CNN, describes the network structure, and discusses the generation of training data and network training. Section III outlines the performance of the proposed method that evaluated using synthetic-data and field-data examples. Section IV discusses the effects of the anchor scale on the velocity-energy cluster detection, the ramifications of varying the number of detected energy clusters on NMO stacking, the picking of simultaneously appearing multiple energy clusters, and the computation time of the proposed method.

II. METHOD

Mask R-CNN, proposed by He et al. [25], is one of the most effective object-detection methods. It extends Faster R-CNN [26], [27] by adding a branch for predicting segmentation masks in each ROI, increasing the accuracy of object detection. Mask R-CNN is successfully used in image processing.

A. Mask R-CNN Architecture

The Mask R-CNN architecture consists of a backbone network, a feature pyramid network (FPN), a region proposal network (RPN), ROI alignment, and a classifier. Fig. 1 shows a schematic of the Mask R-CNN architecture. As shown in Fig. 1, the Mask R-CNN first uses an RPN to generate candidate regions and then obtains accurate bounding boxes through ROI alignment and bounding box regression. Compared to YOLO, the Mask R-CNN can more accurately detect small objects in images through the addition of candidate regions.

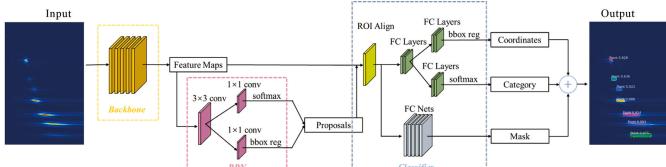


Fig. 1. Mask R-CNN architecture.

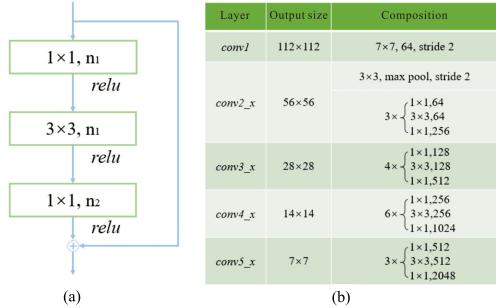


Fig. 2. Schematics of the: (a) block module and (b) ResNet-50 network architecture.

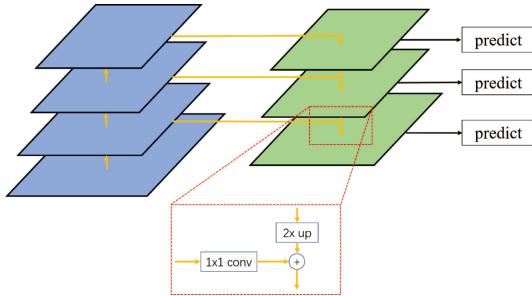


Fig. 3. Typical FPN architecture.

1) *Backbone Network*: In Mask R-CNN, the backbone network is used to extract the object features of the images. Commonly used backbone networks include VGG, ResNet, and DenseNet. In network training, with the deepening of the network, the training accuracies of VGG and DenseNet decreased. Therefore, we used ResNet-50 as the backbone network, where “50” refers to the number of network layers. Fig. 2 shows schematics of the block module and the ResNet-50 architecture.

Fig. 2(a) shows the block module. It consists of three convolutional layers and a short connecting structure, where n_1 and n_2 represent the numbers of convolution cores, and 1×1 and 3×3 are the sizes of the convolution cores. Fig. 2(b) shows the ResNet-50 network structure, where the first layer is the convolutional layer, the second layer is the pooling layer, and the last layer is a block module with 16 parameters.

2) *FPN*: An FPN is used to solve the problem of multiscale object detection. In object detection, the depth and step length of a network are typically contradictory. The scale of the energy cluster in the velocity spectrum is generally 3×3 or 5×5 . The step length in the commonly used network structure is 32, which is larger than that of the velocity-energy clusters. Using a larger step length makes it more difficult for the network to detect small objects. Fig. 3 shows the typical FPN structure. It consists of three parts: the first part is the path from bottom to top (blue part), which is used to extract image features; the second part is the path from top to bottom

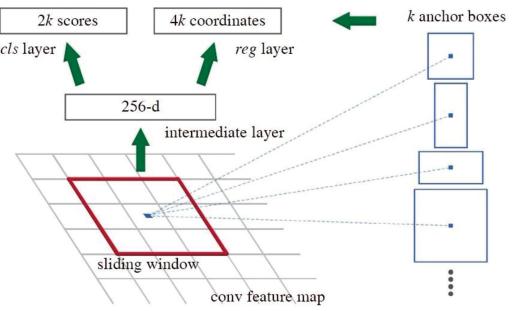


Fig. 4. Schematic of the RPN architecture.

(green part), which is used to upsample the feature maps; and the third part is the lateral connection (yellow line connecting the blue and green parts). The lateral connection can be used to fuse the feature maps.

3) *RPN*: The RPN is used to generate a region proposal. The image is scanned through a sliding window to determine the object region. At the position of each sliding window, the RPN provides k object proposals. After the object proposals are parameterized, they become k anchor boxes, where the anchor is located at the center of the sliding window, and the stride of each sliding window is determined according to the network parameters. In this study, the sliding-window stride was set as one pixel (anchor point). Each anchor point in the feature map can generate three anchors, where the three anchor sizes are scaled according to the length-width ratio (1:4, 1:1, and 4:1). In accordance with the maximum principle of intersection of union (IOU)—i.e., a larger IOU value corresponds to a more accurate detected object area, where the maximum IOU value is one, the anchor box surrounding the maximum IOU is selected as the object prediction box given by the RPN. Once all the anchors have completed the above operations, the object features in the anchor box are converted into low-dimensional feature vectors after undergoing maximum suppression, and they are used as the inputs of the regression and subclassification branches of the subbounding box formed by the fully connected network to judge the categories of the output proposals. Fig. 4 shows a schematic of the RPN.

4) *ROI Alignment*: Both ROI pooling and ROI alignment are used to obtain a fixed-size feature map from the generated region proposal. However, in contrast to ROI pooling, ROI alignment uses the bilinear interpolation method to obtain the pixel values of four fixed points. Therefore, it preserves the floating-point number in the process of obtaining the feature map and produces a more precise feature than ROI pooling.

Fig. 5 illustrates the principle of the ROI-alignment algorithm, where the orange box is a feature map with a dimension of 5×5 , and the black box is the ROI, which is scaled to a 2×2 feature map. The scaling process is described as follows. First, the ROI is divided into several small regions of the same size, as indicated by the red lines in Fig. 5(a). Here, there are four small black dots in each small region, whose coordinates can be calculated. Second, according to the coordinates of the black dots in each small region, a bilinear interpolation algorithm is used to calculate the eigenvalue of each black dot, and the maximum eigenvalue

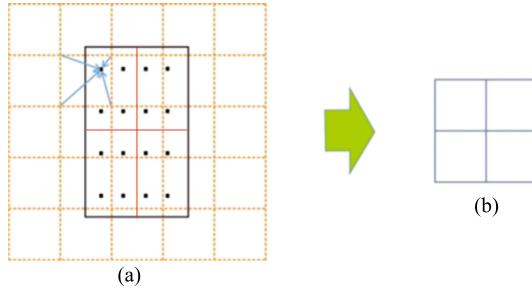


Fig. 5. Schematic of ROI alignment: (a) ROIs with the same size and (b) final output feature map.

of each small region is determined as the final eigenvalue of the corresponding small region. Because only one eigenvalue is reserved for each small region, the size of the final output feature map is 2×2 .

5) Classifier: The loss of Mask R-CNN is the sum of the losses of classification, regression, and mask prediction

$$L = L_{\text{cls}} + L_{\text{bbox}} + L_{\text{mask}} \quad (1)$$

where L_{cls} represents the classification loss, L_{bbox} represents the regression loss of the prediction box, and L_{mask} represents the mask regression loss.

B. Improvement of Mask R-CNN

Although the traditional Mask R-CNN has been successfully used for the detection of large objects, such as cars, planes, and horses, it is not suitable for the detection of small objects, such as velocity-energy clusters. Therefore, we developed a Mask R-CNN algorithm that is suitable for velocity-energy cluster detection. The RPN is an important part of Mask R-CNN and plays a key role in object detection. During network training, when the feature map is fed to the RPN, the RPN generates three anchor boxes with different length-width ratios at pixel points and uses these anchor boxes to select the object features in the feature map; thus, the region proposal box is obtained. The size and shape of the anchor box are determined by the anchor scale and anchor ratio, respectively. In the traditional Mask R-CNN, the anchor scales of the P2–P6 feature layers are 32×32 , 64×64 , 128×128 , 256×256 , and 512×512 , respectively, where P1 is the original image that does not generate an anchor box, and the anchor ratios are 0.5, 1, and 2, which represent the length-width ratios of the three different anchor boxes.

The above parameters indicate that the feature maps of different layers correspond to different anchor scales and that each anchor scale corresponds to three different length-width ratios. For example, in the traditional Mask R-CNN, the anchor scale of the P6 feature layer is 512×512 , and the corresponding length-width ratios are $(512 \times \sqrt{2}, 512/\sqrt{2})$, $(512, 512)$, and $(512/\sqrt{2}, 512 \times \sqrt{2})$. The anchor scale and anchor ratio, e.g., in automobiles and aircraft, are suitable for large-object detection. However, the energy clusters in the velocity spectrum are small—far smaller than 512×512 . Therefore, to enable the RPN to generate an anchor box that matches the shape of the energy clusters in the velocity spectrum, we improved the traditional anchor scale and ratio.

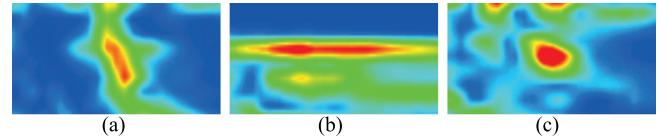


Fig. 6. Energy clusters with different shapes in the velocity spectrum: (a) and (b) inadequate focusing and (c) optimal focusing.

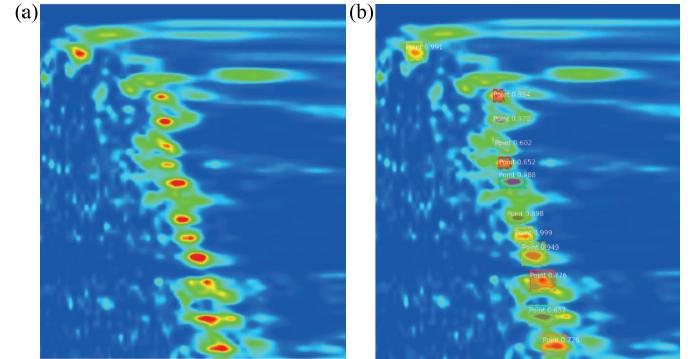


Fig. 7. Results obtained using: (a) traditional anchor scale and (b) improved anchor scale.

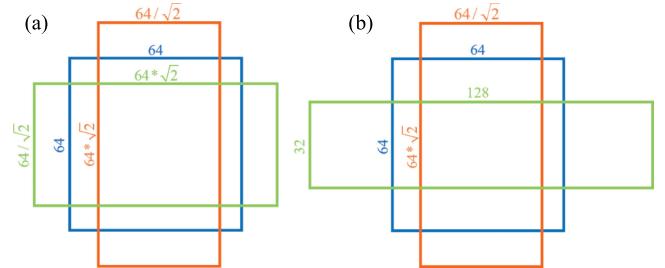


Fig. 8. Schematic of: (a) traditional anchor ratio of the P6 feature layer and (b) improved anchor ratio of the P6 feature layer.

1) Anchor Scale for Energy Cluster Detection: Fig. 6 shows three velocity spectra with different shapes extracted from the real. The energy cluster in Fig. 6(c) has optimal focusing, whereas those in Fig. 6(a) and (b) have inadequate focusing. According to the characteristics of the energy clusters in the velocity spectrum, we determined the anchor scale of the P2–P6 feature layers through experiments; specifically, 1×1 , 4×4 , 8×8 , 16×16 , and 64×64 .

Fig. 7(a) shows the results obtained using the traditional anchor scale. No energy clusters were detected. Fig. 7(b) shows the results obtained using the improved anchor scale. All the energy clusters were detected, and the prediction confidence was greater than 0.6.

2) Anchor Ratio for Energy Cluster Detection: Fig. 8(a) shows a schematic of the traditional anchor ratio corresponding to an anchor scale of 64×64 , where the anchor ratio is $(0.5, 1, 2)$. The green anchor box in Fig. 8(a) is laterally narrow and longitudinally wide. However, the energy cluster in the deep parts is typically wide in the horizontal direction, as shown in Fig. 6(b). Clearly, the anchor ratio in Fig. 8(a) does not match that of this energy cluster. Fig. 8(b) shows a schematic of the improved anchor ratio, where the anchor ratio is $(0.5, 1, 4)$. Compared with Fig. 8(a), we only changed the anchor ratio of the green box; the anchor ratios of the orange and blue boxes remained unchanged. This ensures that the energy cluster detection in the shallow parts is not affected.

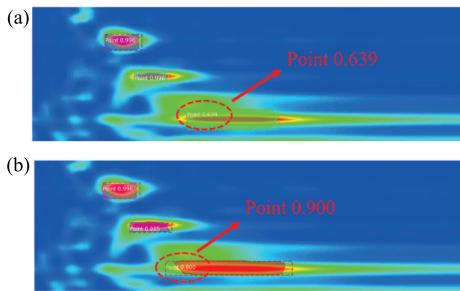


Fig. 9. Results obtained using: (a) traditional anchor ratio and (b) improved anchor ratio.

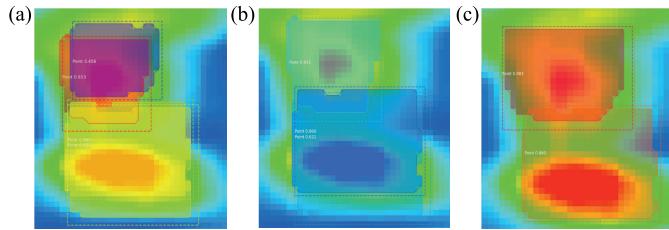


Fig. 10. Results obtained: (a) without NMS; (b) with the traditional NMS; and (c) with the improved NMS.

Fig. 9(a) shows the results obtained using the traditional anchor ratio. The energy cluster enclosed by a circle was not completely detected. The prediction confidence was 0.639. Fig. 9(b) shows the results obtained using the improved anchor ratio. The energy cluster enclosed by the circle was accurately detected, and the prediction confidence was greater than 0.9.

3) Improvement of NMS Algorithm: Duplicate detection is a common problem in R-CNN-based object detection. Because only one detection box is reserved in each object area, redundant detection boxes must be removed. The NMS algorithm can remove duplicate detection boxes. It is expressed as follows:

$$s_i = \begin{cases} s_i, & \text{IOU}(M, b_i) < N_t \\ 0, & \text{IOU}(M, b_i) \geq N_t \end{cases} \quad (2)$$

where N_t represents a threshold value, M denotes the pre-selected box, b represents the set of candidate boxes, IOU is the intersection over union of M and b , and s_i represents the score of the i th pre-selected box. If the IOU is less than N_t , the pre-selected box is retained. Otherwise, the pre-selected box is eliminated.

The threshold value N_t in (2) is typically set at 0.7 for detecting large objects. Fig. 10(a) shows the results obtained without NMS. Two energy clusters were selected by four anchor boxes simultaneously, and four anchor boxes overlapped with each other. Fig. 10(b) shows the preselection results obtained with the traditional NMS method. Compared with Fig. 10(a), only the preselection box with a score of 0.456 was removed; other duplicate candidate boxes remained, as shown in Fig. 10(b). Clearly, the traditional NMS failed to remove duplicate detections, as shown in Fig. 10(a). Therefore, we used the control-variable method to obtain a threshold value suitable for removing duplicate energy cluster boxes. We changed the threshold value N_t in the range [0, 1], where the interval was 0.05. By comparing the experimental results, we determined the optimal threshold value ($N_t = 0.25$). Fig. 10(c) shows the results obtained using the improved NMS.

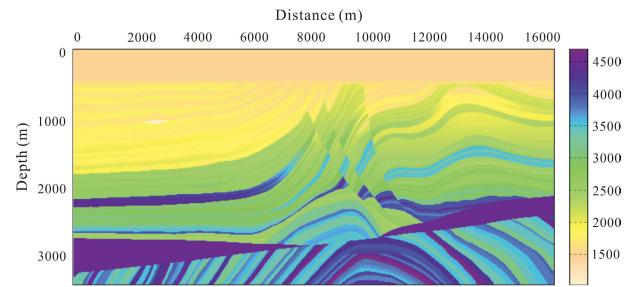


Fig. 11. Marmousi2 model for forward modeling.

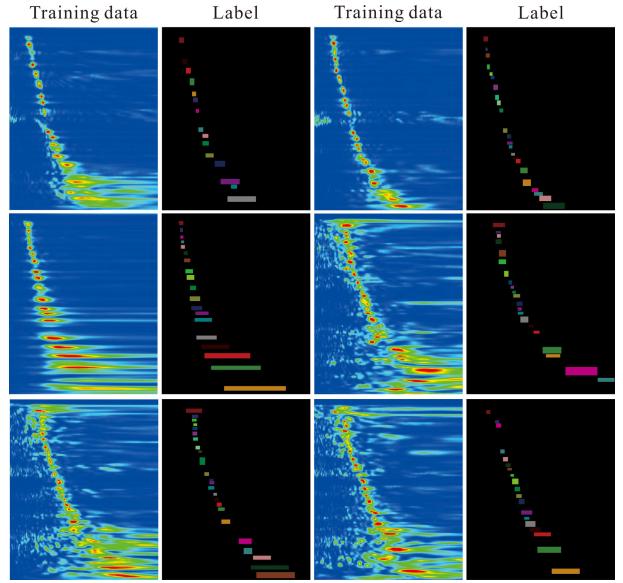


Fig. 12. Training datasets and corresponding labels.

Only one preselection box was reserved for each energy cluster, and the prediction confidence exceeded 0.86.

C. Training Data and Network Training

1) Generation of Training Data: Using the Mask R-CNN to detect the reflected velocity energy clusters in velocity spectra requires the preparation of a set of training data for use in network training. Therefore, we first generated numerous synthetic training datasets. These synthetic datasets consisted of the velocity spectrum generated from synthetic data and corresponding labels. Synthetic data were obtained using the acoustic equation based on the Marmousi2 model (Fig. 11). Through forward modeling, 851 common shot gathers (CSGs) were generated, and 1700 CMP gathers were obtained by sorting the CSGs. One velocity spectrum was generated for every five CMPs, and 320 velocity spectra were obtained. In addition, to make the training data more realistic and enable the network to accurately extract the characteristics of the energy clusters in real velocity spectra, we generated 60 velocity spectra by using different field data. Thus, we obtained 380 velocity spectra, of which 304 were used as training sets and 76 were used as test sets. Then, the energy clusters considered as NMO velocity in the 304 velocity spectra were manually marked with rectangular boxes, which were used as training labels. Finally, 304 velocity spectra and corresponding labels were fed to the network for training. Fig. 12 shows several training datasets and the corresponding labels.

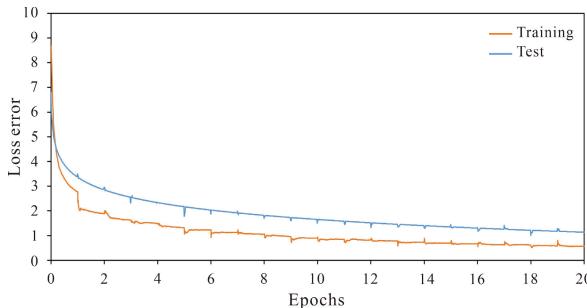


Fig. 13. Curves of the loss error.

2) *Network Training*: During the network training, we performed several experiments to optimize the tradeoff among the training parameters, including the number of epochs, batch size, and learning rate. The parameters for network training were set as follows: the initial learning rate of network training was 0.001, the batch size was three, and 100 iterations were performed for each epoch. The end of the network training depended on the convergence of the loss function. When the loss function stabilized, the network training was terminated. Fig. 13 presents the learning curves for 20 epochs. As shown, the loss error decreased sharply after two epochs and became smooth after 16 epochs. When the number of epochs reached 20, the loss function converged to a small value. Therefore, the network training was terminated.

To quantitatively evaluate the effectiveness of the network training, we plotted the root-mean-square error (RMSE) and prediction accuracy (PA) of the trained model, as shown in Fig. 14(a) and (b), respectively. As indicated by Fig. 14(a), during the period of one–eight epochs, the RMSE decreased sharply from approximately 21 to 2. After eight epochs, the curve of the RMSE stabilized, and the final error value was approximately one. As shown in Fig. 14(b), with an increase in the number of epochs, the PA value increased, and the maximum value was approximately 0.8. Fig. 14(a) and (b) indicates that the trained model achieved good convergence and a high PA.

III. NUMERICAL RESULTS

A. Synthetic Data

We evaluated the performance of the proposed method with a synthetic velocity spectrum extracted from test sets. Fig. 15(a) shows the synthetic velocity spectrum. Fig. 15(b) shows the velocity curve obtained via MIVP. Considering previous experimental results, 13 velocity-energy clusters were selected. Fig. 15(c) presents the results obtained using the proposed method—23 energy clusters were detected. In contrast to MIVP, the weak energy clusters at 0.6, 1.1, 1.4, and 2.3 s were detected by the proposed method, and the prediction confidence was greater than 0.68. Fig. 15(d) shows the velocity curve connecting all the detection points in Fig. 15(c). The velocity curve in Fig. 15(d) is consistent with that of Fig. 15(b). To evaluate the effectiveness of the proposed method, we applied the traditional Mask R-CNN to the velocity spectrum in Fig. 15(a). Fig. 15(e) shows the results detected by the traditional Mask R-CNN. Fourteen velocity-energy clusters were identified, where three energy

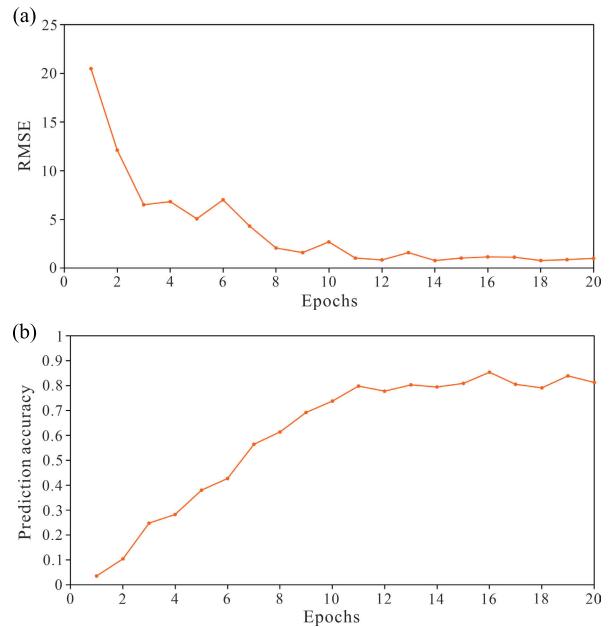


Fig. 14. Evaluation indices of the trained model: (a) RMSE and (b) PA.

clusters were repeatedly detected and the confidence level of six detection points was less than 0.6. In addition, several key energy clusters in the range of 1.5–2.5 s were not detected. Fig. 15(f) shows the velocity curve connecting the detection points in Fig. 15(e). Due to the low confidence level of the detection points and key velocity points not being detected, the velocity curve in Fig. 15(f) is inconsistent with those in Fig. 15(b) and (d).

Fig. 16(a)–(c) shows the 2-D velocity field obtained via MIVP, the proposed method, and the traditional Mask R-CNN, respectively. The velocity fields in Fig. 16(a) and (b) are highly congruent, whereas that in Fig. 16(c) differs significantly. The lateral velocity variation in Fig. 16(c) is uneven—particularly in complex structures.

To evaluate the performance of the velocity field obtained via the proposed method, we applied the velocity fields shown in Fig. 16(a)–(c) to the synthetic 2-D dataset. Fig. 17(a)–(c) shows the stacked sections corresponding to the velocity field in Fig. 16(a)–(c), respectively. In Fig. 17(b), the continuity of reflection events marked by the arrow is better than that in Fig. 17(a). The stacked section in Fig. 17(c) is worse than those in Fig. 17(a) and (b)—particularly the reflection events in the box. This indicates that the velocity field of the proposed method is better than those of MIVP and the traditional Mask R-CNN.

B. Field Data

To evaluate the effectiveness of the proposed method for real data, we applied it to a velocity spectrum generated using a 3-D seismic dataset. The CSGs in the 3-D seismic data consisted of 12 receiver lines with 1440 traces per receiver line. The receiver interval was 50 m along the inline direction and 150 m along the crossline direction, and the bin size was 25×25 m.

Fig. 18(a) shows the velocity spectrum generated by the field data. Fig. 18(b) shows the CMP gather corresponding

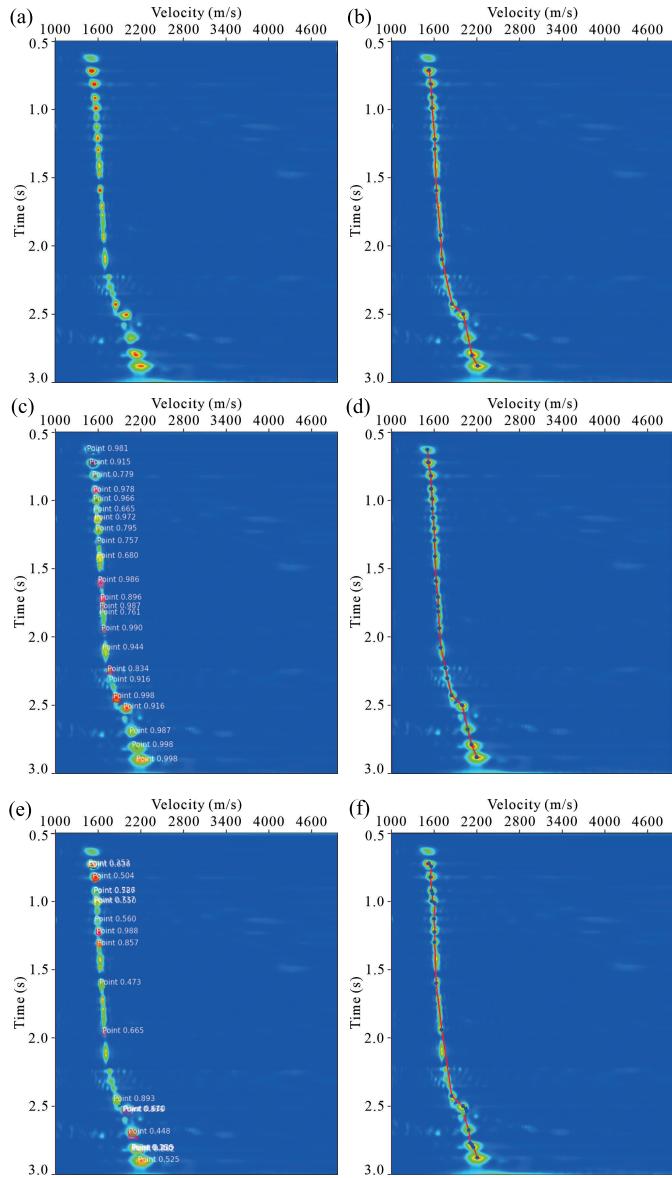


Fig. 15. Results of velocity picking using different methods: (a) velocity spectrum generated by CMP gathers; (b) MIVP; (c) proposed method; (d) velocity curve connecting the detection points in (c); (e) detection results of the traditional Mask R-CNN; and (f) velocity curve connecting the detection points in (e).

to the velocity spectrum shown in Fig. 18(a). As shown in Fig. 18(b), the reflected energy after 2.0 s was weak, resulting in inadequate focusing of the energy cluster after 2.0 s in Fig. 18(a). Before applying the trained network to the velocity spectrum in Fig. 18(a), the time and velocity range must be input to the network, which is used to convert the coordinates of the detected energy clusters into time–velocity values. Here, the time and velocity ranges are 0.1–3.5 s and 1000–6500 m/s, respectively. Fig. 18(c) shows the results obtained using the traditional Mask R-CNN. This method failed to detect the energy clusters after 2.0 s, and several energy clusters were repeatedly detected. Fig. 18(d) shows the results obtained using the proposed method. Compared with Fig. 18(c), all the energy clusters in Fig. 18(a) were detected, and the prediction confidence was greater than 0.9. Fig. 18(e) presents the velocity curve obtained using MIVP.

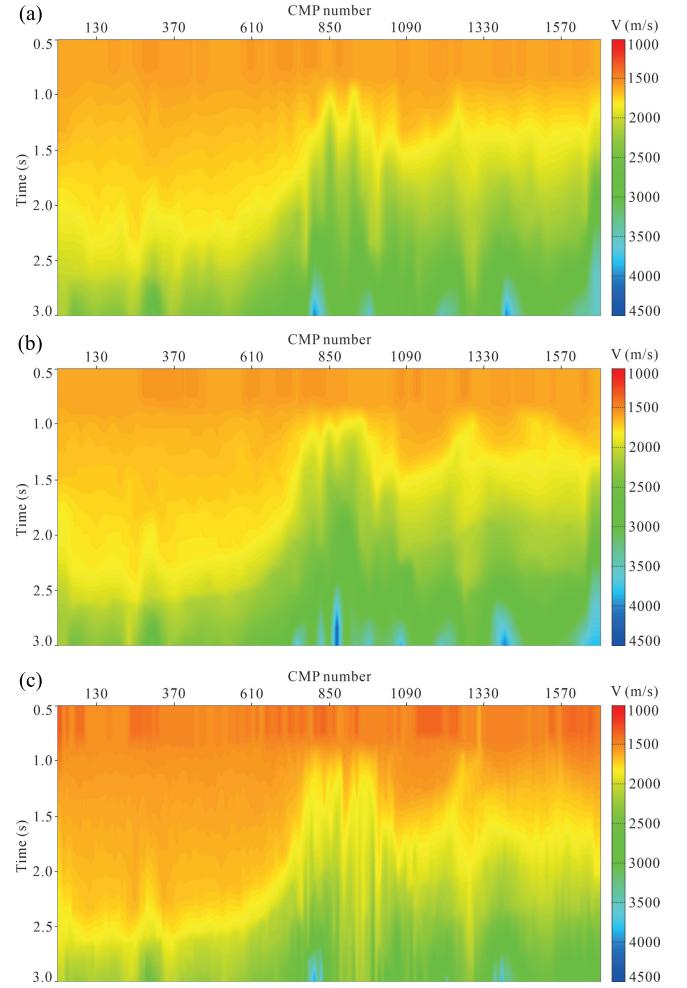


Fig. 16. Velocity field obtained via: (a) MIVP; (b) proposed method; and (c) traditional Mask R-CNN.

As shown, 11 energy clusters were selected. Fig. 18(f) shows the NMO-corrected CMP gathers obtained using MIVP velocities. The reflected events marked by the red arrow were not straightened. This indicates that the velocity obtained via MIVP deviated from the actual velocity. Fig. 18(g) shows the velocity curve connecting the detected points in Fig. 18(c), and Fig. 18(h) shows the NMO-corrected CMP gathers. Because an inaccurate NMO velocity is used, the far offset reflection events after 2.0 s in Fig. 18(h) are upward. Fig. 18(i) shows the velocity curve connecting the detected points in Fig. 18(d), and Fig. 18(j) shows the NMO-corrected CMP gathers. All the reflection events in Fig. 18(j) are straightened, indicating that the NMO velocity obtained using the proposed method is close to the true velocity.

Fig. 19(a)–(c) presents the velocity field of inline 120 datasets obtained using MIVP, the proposed method, and the traditional Mask R-CNN, respectively. As shown in Fig. 19(a), the velocity field in the lateral direction was unstable after 1.0 s. Compared with the velocity field in Fig. 19(a), the velocity field in Fig. 19(b) is more stable in both the lateral and longitudinal directions. The velocity field in Fig. 19(c) is inconsistent with those in Fig. 19(a) and (b).

We also applied the velocity field in Fig. 19(a)–(c) to the inline 120 datasets. Fig. 20(a)–(c) shows part of the stacked

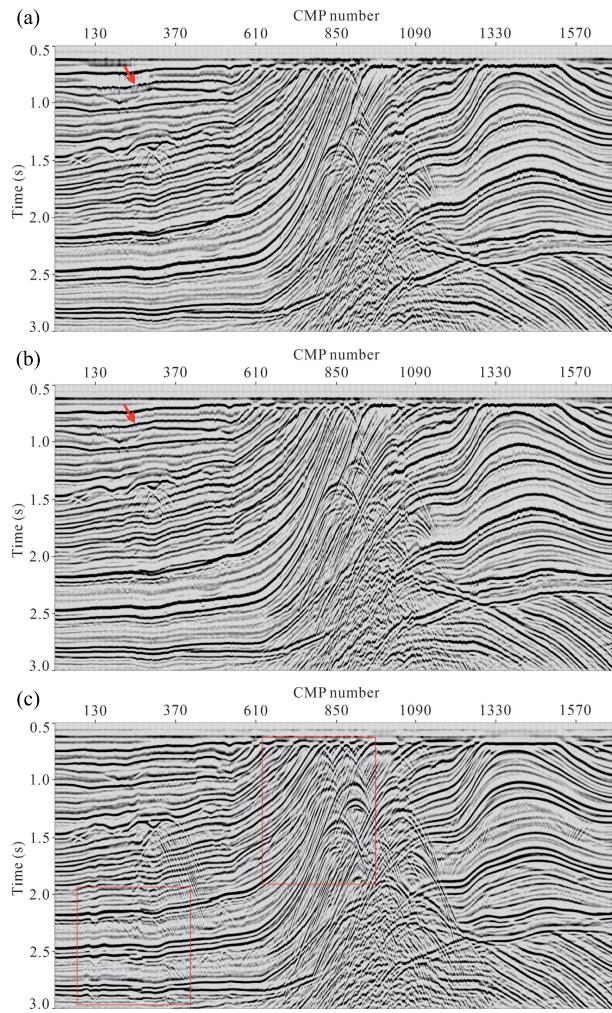


Fig. 17. Stacked sections obtained using the velocity fields in: (a) Fig. 16(a); (b) Fig. 16(b); and (c) Fig. 16(c).

sections corresponding to MIVP, the proposed method, and the traditional Mask R-CNN, respectively. The continuity of the reflection events marked by the red arrow in Fig. 20(b) is better than that in Fig. 20(a). The continuity of the reflection events in Fig. 20(c) is worse than that in Fig. 20(b)—particularly the reflections marked by the blue arrow, and the structural characteristics differ significantly from those in Fig. 20(a) and (b). The stacked sections indicate that the velocity field obtained via the proposed method is better than those obtained via MIVP and the traditional Mask R-CNN.

The velocity spectrum is generated using multitrace data in a CMP. The more traces the CMP contains, the more prominent are the reflected energy clusters in the velocity spectrum; thus, the reflected energy clusters can be accurately detected. To build an accurate 3-D velocity field, the velocity spectrum is typically generated according to a certain grid. The smaller the grid, the higher the accuracy of the predicted velocity field is. For this field example, we used 10×10 grids to generate velocity spectra, that is, every ten CMPs in the crossline direction and ten lines in the inline direction. Fig. 21 shows the 3-D velocity field predicted by the proposed method. Fig. 22 shows the 3-D stacked data volume obtained by using the velocity field in Fig. 21. As shown in Fig. 21, the precision of the proposed method in predicting velocity values within each

velocity spectrum prevents abrupt velocity changes between neighboring velocity spectral points, thus ensuring the lateral stability of the predicted velocity field.

To verify the generalization of the proposed method in other areas, we applied the trained Mask R-CNN to another 3-D field dataset. Before applying the trained network to the data, we did not adjust the network settings of the anchor scale and ratio because the shape of energy clusters in field data is virtually identical. Fig. 23(a) shows the velocity curve obtained via MIVP. Fig. 23(b) shows the velocity curve obtained via the proposed method. The velocity curve in Fig. 23(b) is consistent with that in Fig. 23(a). Fig. 24(a) shows the stacked section after the application of the velocity field built via MIVP to the field data. Fig. 24(b) shows the stacked section after the application of the velocity field built via the proposed method to the field data. The reflection events marked by the red arrow in Fig. 24(b) are better than those in Fig. 24(a). This example indicates that the trained model can be generalized to the 3-D dataset.

IV. DISCUSSION

A. Effect of Anchor Scale on Velocity-Energy Cluster Detection

The anchor scale of an RPN is a key parameter for object detection. If the anchor scale is set to an extremely large or small value, then objects cannot be accurately detected. As shown in Fig. 7(a), because the anchor scales of the P2–P6 feature layer (32×32 , 64×64 , 128×128 , 256×256 , and 512×512) were far larger than the energy clusters in the velocity spectrum, no energy clusters were detected. To obtain an anchor scale that matched the size of the energy clusters, we performed several anchor-scale experiments. Next, we present two of the experimental results obtained. Fig. 25(a) shows the results obtained using anchor scales of 1×1 , 3×3 , 10×10 , 50×50 , and 100×100 . With a reduction in the anchor scale, certain energy clusters were detected. However, three energy clusters still leaked, as shown in Fig. 25(a). This indicates that the anchor scale did not match the sizes of some of the energy clusters. Fig. 25(b) shows the results obtained using anchor scales of 1×1 , 2×2 , 4×4 , 8×8 , and 16×16 . Only two energy clusters were detected because the set anchor scale was too small. Clearly, for Mask R-CNN-based object-detection methods, an appropriate anchor scale is a prerequisite for accurate object detection.

B. Effect of the Number of Detected Energy Clusters on the Accuracy of NMO Stacking

In NMO stack processing, it is essential for all sampling points in seismic data to possess corresponding time–velocity pairs. However, as velocity picking is performed at the energy cluster point in velocity spectrum picking, only a limited number of time–velocity pairs can be obtained through velocity-picking methods. To obtain the velocity for all points in time, linear interpolation is employed between the detected time–velocity pairs in NMO stack processing. According to the interpolation principle, a greater number of detected velocity points leads to smaller

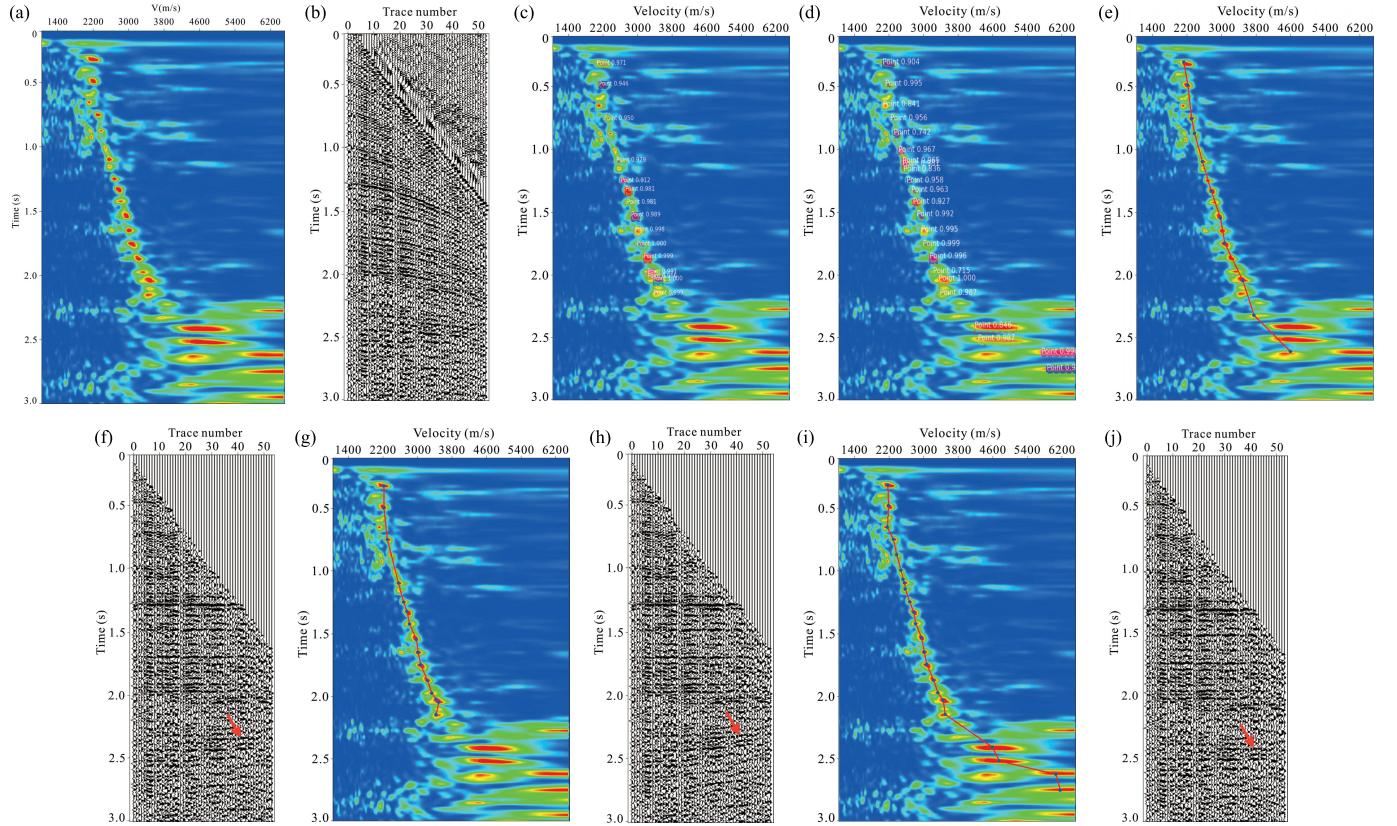


Fig. 18. (a) Velocity spectrum generated using real data. (b) CMP gathers used to generate the velocity spectrum in (a). (c) Results obtained via the traditional Mask R-CNN. (d) Results obtained via the proposed method. (e) Stacking velocity obtained via MIVP. (f) NMO-corrected CMP gathers obtained using the velocity in (e). (g) Stacking velocity obtained via the traditional Mask R-CNN. (h) NMO-corrected CMP gathers using the velocity in (g). (i) Stacking velocity obtained via the proposed method. (j) NMO-corrected CMP gathers using the velocity in (i).

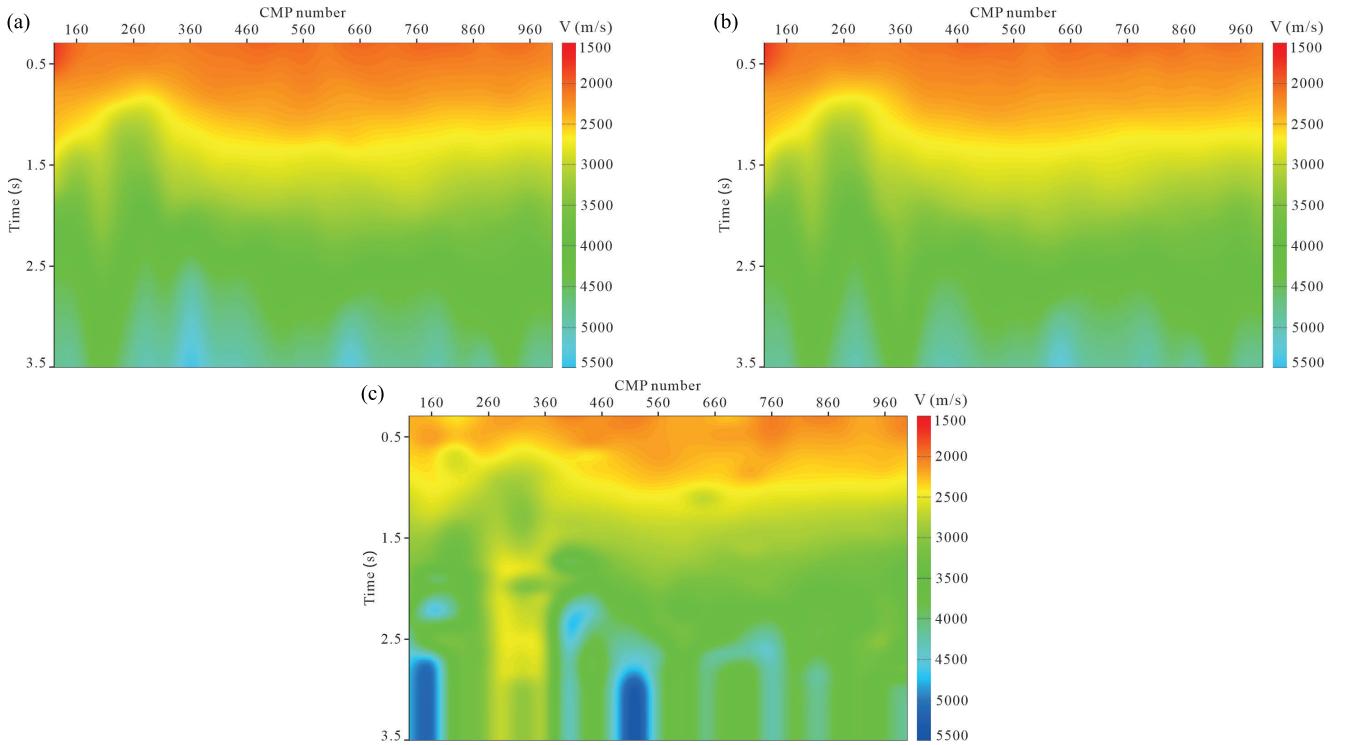


Fig. 19. Velocity fields built via: (a) MIVP; (b) proposed method; and (c) traditional Mask R-CNN.

errors between the interpolation velocity and actual velocity. As shown in Figs. 15(d), 18(d), and 23(b), the velocity points detected via the proposed method are greater in number than

those selected via MIVP. Therefore, the proposed method obtained more precise interpolation velocities compared to MIVP, resulting in better results in NMO stack processing.

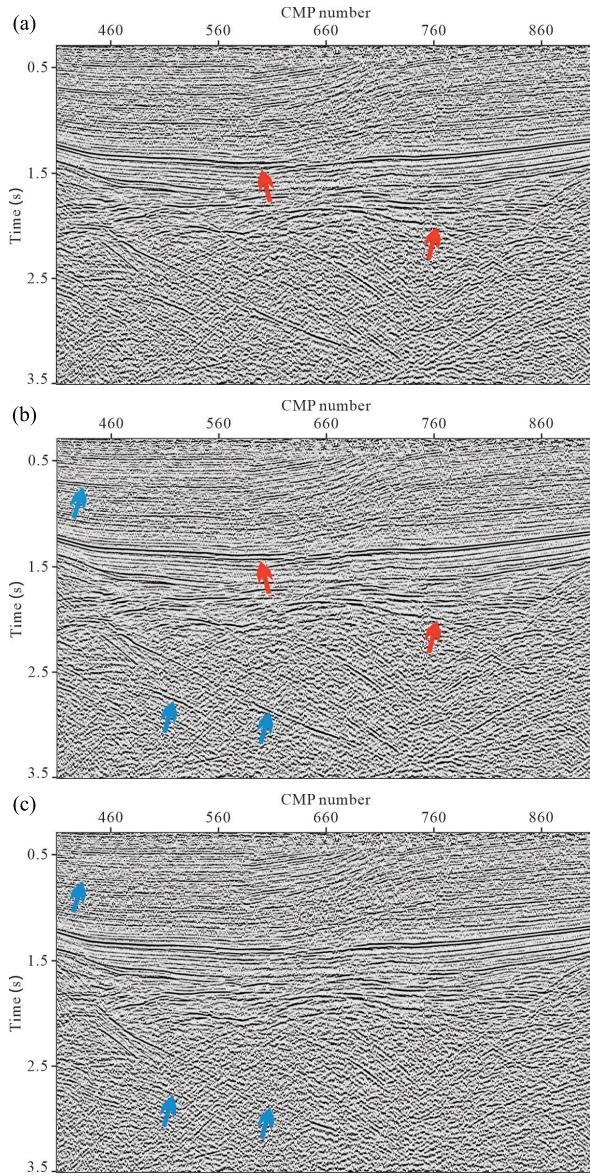


Fig. 20. Stacked sections obtained using the velocity field in: (a) Fig. 19(a); (b) Fig. 19(b); and (c) Fig. 19(c).

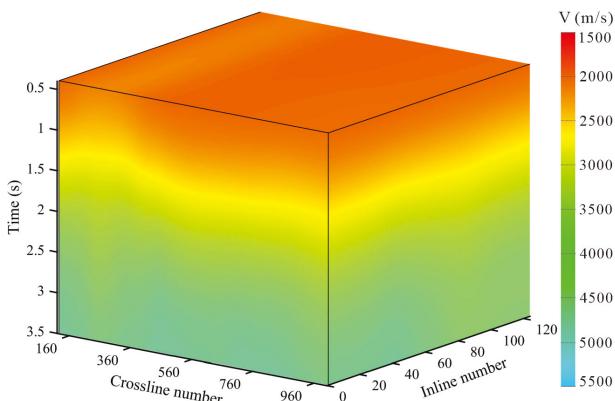


Fig. 21. 3-D velocity volume predicted by the proposed method.

The comparison of stacked sections in Figs. 20 and 24 indicates that the quality of NMO stacking is proportional not only to the accuracy of velocity picking, but also to the number of velocity-picking points.

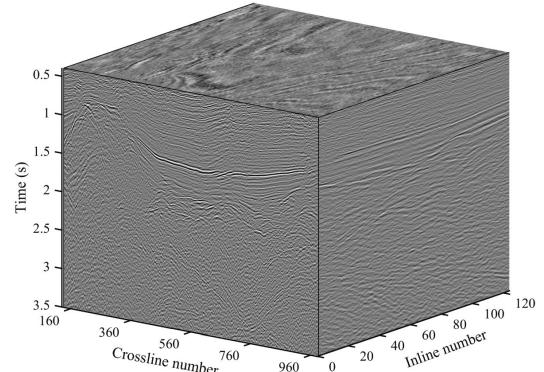


Fig. 22. 3-D stacked data volume obtained by using the velocity field in Fig. 21.

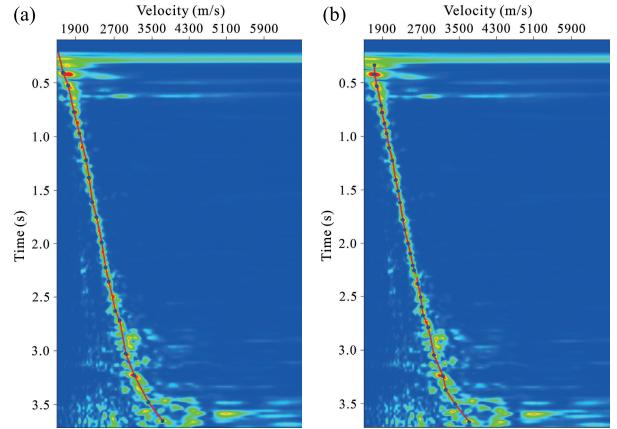


Fig. 23. Velocity curves obtained via: (a) MIVP and (b) proposed method.

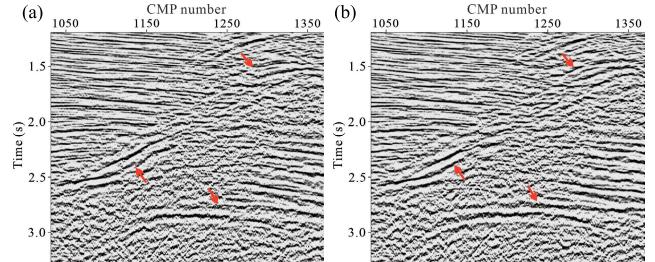


Fig. 24. Stacked section after the application of: (a) velocity field built via MIVP and (b) velocity field built via the proposed method to the field data.

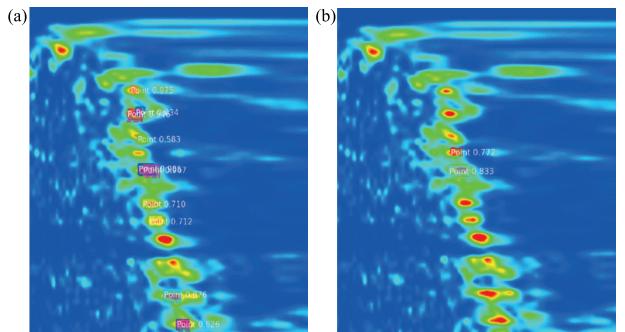


Fig. 25. (a) Results obtained using the anchor scales 1×1 , 3×3 , 10×10 , 50×50 , and 100×100 . (b) Results obtained using the anchor scales 1×1 , 2×2 , 4×4 , 8×8 , and 16×16 .

C. Picking Energy Clusters When Multiples and Reflections Appear Simultaneously in the Velocity Spectrum

In manual picking, the energy clusters from multiples in the velocity spectrum often interfere with velocity picking.

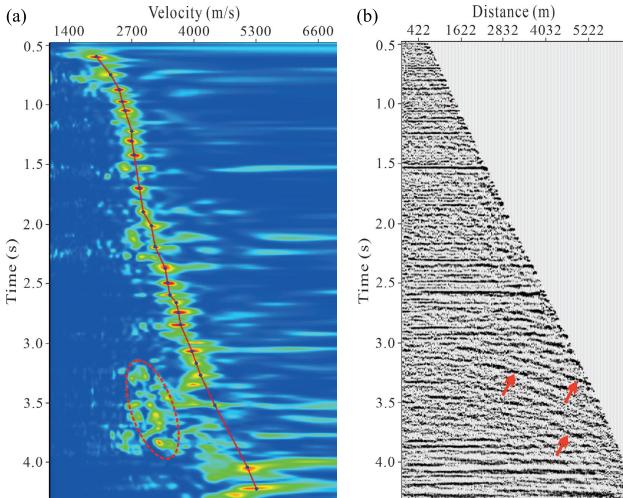


Fig. 26. (a) Velocity spectrum containing multiples. (b) NMO-corrected CMP gathers using the picked velocity in (a).

However, the proposed method is not sensitive to the energy clusters from multiples. When the energy clusters of multiples and reflections appear simultaneously in the velocity spectrum, the proposed method can distinguish true energy clusters from fake ones.

Fig. 26(a) shows a velocity spectrum containing multiples. The reflected energy clusters in Fig. 26(a) increase with time. By contrast, the energy clusters from multiples enclosed by the red circle are located to the left side of the reflected energy clusters at the same time, and these clusters are located far away from the reflected energy clusters in the velocity spectrum because the velocity of multiples is much smaller than that of reflections at the same time. The red curve in Fig. 26(a) shows the predicted velocity. The network picked the reflected energy clusters—not the energy clusters in the red circle—because the reflected energy clusters in Fig. 26(a) essentially indicate an increasing trend in time, which are similar to those in the label of Fig. 12. Fig. 26(b) shows the NMO-corrected gathers using the predicted velocity. The multiples marked by red arrows appear as a downward curved shape due to insufficient NMO corrections. This trend further indicates that when multiples and reflections simultaneously appear in the velocity spectrum, the network picks the reflected velocity.

D. Computation Time Comparison

MIVP has historically played an important role in seismic data processing. However, this approach has been criticized for its time consumption and inefficiency. Compared with MIVP, the deep learning-based velocity-picking method can significantly increase the efficiency of building a velocity field. We evaluated the computation time by performing experiments on the aforementioned synthetic and field data.

Table I presents the computation times of MIVP and the proposed method for synthetic velocity spectrum picking. As shown, the proposed method required approximately 300 min to train 304 velocity spectra. However, it only required approximately 0.4 min to complete the picking of 76 velocity spectra. For MIVP, although no extra training

TABLE I
COMPARISON OF COMPUTATION TIME

Methods	Training time (minutes)	Testing time (minutes)
MIVP	0	80
Proposed method	300	0.4

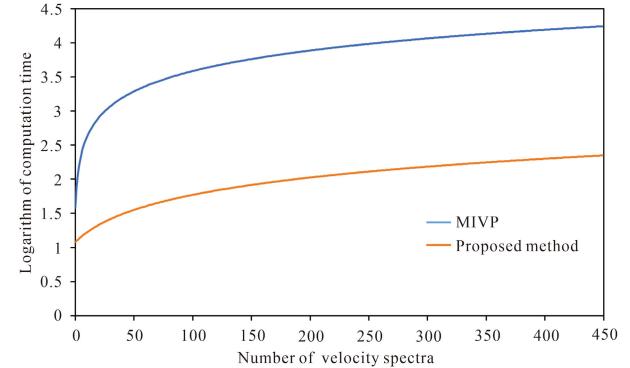


Fig. 27. Computation times of MIVP and the proposed method. The orange line indicates the logarithm of the computation time of the proposed method. The blue line indicates the logarithm of the computation time of MIVP.

computations were required, it required approximately 80 min to complete the picking of 76 velocity spectra. When network training was not considered, the computation time of the proposed method was 0.004 times that of MIVP.

We also measured the time required to build the 3-D velocity field of field-data examples using MIVP and the proposed method. For MIVP, the picking of 450 velocity spectra was performed in approximately 292 min. Because the proposed method required no extra network training computation, it required only approximately 3.7 min to complete the picking of the 450 velocity spectra. Fig. 27 shows the comparison of the computation times of MIVP and the proposed method. Because the computation time of the proposed method is much shorter than that of MIVP, the logarithm of the computation time is presented. As shown in Fig. 27, a larger amount of velocity spectrum picking corresponded to a larger computation-time difference between MIVP and the proposed method. Clearly, MIVP requires significantly more time compared to the proposed method. However, considering the time required to prepare the training data and train the network, the proposed method is inferior to MIVP.

V. CONCLUSION

We developed a Mask R-CNN-based automatic velocity-picking method. It has improved the anchor scale and ratio of the traditional Mask R-CNN and NMS algorithms; thus, the proposed method accurately extracted time–velocity information from the velocity spectrum.

We evaluated the performance of the proposed method using synthetic data and field data in comparison with MIVP and the traditional Mask R-CNN. In tests involving velocity spectrum picking, NMO-corrected CMP gathers, and stacked sections, our method outperformed the other two methods in terms of the accuracy of velocity picking. Compared with MIVP, our method can not only build a reliable velocity field,

but also enhance the efficiency of velocity picking. When network training is not considered, the computation time of the proposed method was <0.01 times that of MIVP. The proposed method significantly reduced the amount of time needed for velocity picking, thus reducing the cost of seismic data processing.

For supervised deep learning methods, the generalization of trained models is the main concern. In future research, to improve the applicability of the trained model to more survey areas, we intend to use numerous datasets with various characteristics to train the network. We expect the proposed method to be applied in industry and replace manual picking.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers and editors for their helpful comments and suggestions.

REFERENCES

- [1] D. Wang, S. Yuan, H. Yuan, H. Zeng, and S. Wang, "Intelligent velocity picking based on unsupervised clustering with the adaptive threshold constraint," *Chin. J. Geophys.*, vol. 63, no. 3, pp. 1048–1060, Mar. 2021.
- [2] H. Zhang, P. Zhu, Y. Gu, and X. Li, "Automatic velocity picking based on deep learning," in *Proc. SEG Tech. Program Expanded Abstr.*, Aug. 2019, pp. 2604–2608.
- [3] R. S. Ferreira, D. A. B. Oliveira, D. G. Semin, and S. Zaytsev, "Automatic velocity analysis using a hybrid regression approach with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4464–4470, May 2021.
- [4] J. L. Toldi, "Velocity analysis without picking," *Geophysics*, vol. 54, no. 2, pp. 191–199, Feb. 1989.
- [5] N. Lin, H. Liu, and J. Li, "Automatic picking velocity by the Viterbi algorithm for the complex geological case," *Prog. Geophys.*, vol. 19, no. 2, pp. 311–316, Jun. 2004.
- [6] N. Lin, H. Liu, G. Li, J. Tang, and L. Wei, "Auto-picking velocity by path-integral optimization and surface fairing," *Chin. J. Geophys.*, vol. 56, no. 1, pp. 246–254, Jan. 2013.
- [7] J. Zhang, N. Lin, D. Zhang, C. Zhao, and X. Yang, "Intelligent picking of velocity spectrum based on nonlinear function," *Prog. Geophys.*, vol. 31, no. 2, pp. 856–860, Apr. 2016.
- [8] D. E. Lumley, "Monte Carlo automatic velocity picks," Stanford Explor. Project Rep. 75, Nov. 1997, pp. 1–25.
- [9] H. Choi and J. Byun, "Automatic velocity analysis by using an high-resolution bootstrapped differential semblance method," *Geophys. Explor.*, vol. 16, no. 4, pp. 225–233, Nov. 2013.
- [10] B. C. Fish and T. Kusuma, "A neural network approach to automate velocity picking," in *Proc. SEG Tech. Program Expanded Abstr.*, Jan. 1994, pp. 185–188.
- [11] L. Dong and X. He, "Automatic velocity spectrum picking using artificial neural networks," *Oil Geophys. Prospecting*, vol. 31, pp. 98–103, Jun. 1996.
- [12] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, "Deep-learning tomography," *Lead. Edge*, vol. 37, no. 1, pp. 58–66, Jan. 2018.
- [13] R. Biswas, A. Vassiliou, R. Stromberg, and M. K. Sen, "Estimating normal moveout velocity using the recurrent neural network," *Interpretation*, vol. 7, no. 4, pp. 819–827, Nov. 2019.
- [14] H. Zhang, S. Yuan, H. Zeng, H. Yuan, Y. Gao, and S. Wang, "Automatic velocity analysis using interpretable multimode neural networks," *Geophys. J. Int.*, vol. 235, no. 1, pp. 216–230, May 2023.
- [15] Y. Zhao, Y. Li, X. Dong, and B. Yang, "Low-frequency noise suppression method based on improved DnCNN in desert seismic data," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 5, pp. 811–815, May 2019.
- [16] X. Si, Y. Yuan, T. Si, and S. Gao, "Attenuation of random noise using denoising convolutional neural networks," *Interpretation*, vol. 7, no. 3, pp. 269–280, Aug. 2019.
- [17] X. Wu, L. Liang, Y. Shi, and S. Fomel, "FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation," *Geophysics*, vol. 84, no. 3, pp. 35–45, May 2019.
- [18] Y. Ma, X. Ji, T. W. Fei, and Y. Luo, "Automatic velocity picking with convolutional neural networks," in *Proc. SEG Tech. Program Expanded Abstr.*, Aug. 2018, pp. 2066–2070.
- [19] P. Zhang and W. Lu, "Automatic time-domain velocity estimation based on an accelerated clustering method," *Geophysics*, vol. 81, no. 4, pp. 13–23, Jul. 2016.
- [20] S. Wei, O. Yonglin, Z. Qingcai, H. Jiaqiang, and S. Yaying, "Unsupervised machine learning: K-means clustering velocity semblance autopicking," in *Proc. 80th EAGE Conf. Exhib.*, Jun. 2018, pp. 1–5.
- [21] Y. Chen, "Automatic semblance picking by a bottom-up clustering method," in *Proc. SEG Workshop, SEG Maximizing Asset Value Through Artif. Intell. Mach. Learn.*, Beijing, China, Dec. 2018, pp. 17–19.
- [22] W. Wang, G. A. McMechan, J. Ma, and F. Xie, "Automatic velocity picking from semblances with a new deep-learning regression strategy: Comparison with a classification approach," *Geophysics*, vol. 86, no. 2, pp. 1–13, Feb. 2021.
- [23] K. Lenc and A. Vedaldi, "R-CNN minus R," 2015, *arXiv:1506.06981*.
- [24] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.
- [26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Conf. Workshop Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 1–9.
- [27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.



Yijun Yuan received the Ph.D. degree in geophysics from the China University of Geosciences, Beijing, China, in 2005.

In 2005, he joined the School of Geophysics and Information Technology, China University of Geosciences, where he has been doing research on seismic data processing and interpretation. His research interests include seismic signal analysis and processing, noise suppression, and seismic interpretation.



Ying Li received the M.Sc. degree in geological engineering from the China University of Geosciences, Beijing, China, in 2022.

In 2022, she joined BYD Ltd., Shenzhen, China, where she has been doing research on silicon carbide epitaxy. Her research interest is in the area of silicon carbide epitaxy and deep learning.



Xiaoxuan Fang is currently pursuing the M.Sc. degree in geodetection and information technology with the China University of Geosciences, Beijing, China.

Her research interest is the application of deep learning in seismic data processing.



Fengfeng Shi received the M.Sc. degree in geodetection and information technology from the China University of Geosciences, Beijing, China, in 2010.

In 2010, he joined the Exploration and Development Research Institute, PetroChina Huabei Oilfield Company, Renqiu, China, where he has been doing research on seismic data processing. His research interest is in the area of seismic data processing.