Multiple Heterogeneous P-DCNNs Ensemble With Stacking Algorithm: A Novel Recognition Method of Space Target ISAR Images Under the Condition of Small Sample Set

HONG YANG1, YASHENG ZHANG2, AND WENZHE DING1

1Graduate School, Space Engineering University, Beijing 101416, China
2Department of Aerospace Institute, Space Engineering University, Beijing 101416, China

Corresponding author: Hong Yang (yanghongearth@gmail.com)

This work was supported by the National Outstanding Youth Fund under Grant 2017-JCQ-ZQ-005.

ABSTRACT In this paper, a novel method of multiple heterogeneous pre-trained deep convolutional neural network models (P-DCNN) ensemble with stacking algorithm is proposed, which can realize automatic recognition of space targets in inverse synthetic aperture radar (ISAR) images with high accuracy under the condition of the small sample set. In this method, transfer learning (TL) is introduced into the recognition of space targets in ISAR images for the first time, and the automatic recognition of space target ISAR images under a small sample set is realized. Besides, the stacking algorithm is used to realize the ensemble of multiple heterogeneous P-DCNNs, which effectively overcomes the limitations of the single weights fine-tuned P-DCNN (FP-DCNN), such as weak robustness and difficulty in classification accuracy. Firstly, the space target ISAR image data set after despeckling and standardization is divided into specific parts, and the training set of each part is augmented based on ISAR image transformation such as contrast adjustment, small-angle rotation, azimuth scaling, and range scaling. Then, multiple heterogeneous P-DCNNs are taken as the base learners in the first layer of the stacking ensemble learning framework (SELF), and fine-tuning training is carried out for each heterogeneous P-DCNN by using the augmented ISAR image dataset. Thus, the meta-features of ISAR images of space targets with stronger generalization are proposed. Furthermore, the XGBoost classifier is used as the meta-learner in the second layer of SELF, and the extracted meta-features of training data are used to train the meta-learner. Finally, the trained meta-learner is used to realize the automatic recognition of space targets in ISAR images. The experiment results show that the stacking algorithm can effectively realize the ensemble of multiple heterogeneous P-DCNNs, and the classification performance of the SELF is better than any single FP-DCNN.

INDEX TERMS Space target, ISAR image, small sample set, transfer learning, data augmentation, ensemble learning.

I. INTRODUCTION ISAR plays an important role in the field of space situational awareness. It obtains a high range resolution by using wide-band signals and obtains azimuth resolution by using Doppler signals generated by the relative motion between the target and the radar. It can provide abundant target structure information for space targets. Therefore, using two-dimensional ISAR images for space target recognition has always been the research focus of automatic target recognition (ATR) in the field of space situation awareness [1], [2], [3].

ISAR image recognition can be divided into coarse-grained recognition and fine-grained recognition. Compared with the coarse-grained identification of aircraft, missile and other different types of targets in ISAR image, the recognition of space targets (such as A2100, Galaxy and Sea-sat satellites) in ISAR images which belongs to fine-grained identification pays more attention to the subtle differences among various satellites. Therefore, the recognition of space targets in ISAR images is more difficult and higher requirements are put forward for the identification methods.

Early use of ISAR images for space targets recognition mainly relies on manual feature extraction, and use engineering skills and professional knowledge to design feature.
descriptors for different types of tasks. Traditional feature extraction methods involve spatial, textural, morphological, statistical and other information. Typical features include texture description maps, gist features, scale invariant feature transform (SIFT), gradient histograms, local binary patterns (LBP) [4], etc. The method of artificial feature extraction is in line with intuitive understanding and has significant advantages in terms of calculation amount, but also has obvious defects: (1) due to the influence of clutter heterogeneity and operating environment and other factors, it is difficult to guarantee the robustness of features; (2) rich engineering experience and theoretical knowledge are needed to build a practical feature library; (3) feature dimensions are usually high, the calculation process is complex, and the relationship between multi-feature dimensions is unclear and lacks sufficient theoretical support; (4) lack of intelligent reasoning ability to learn and adapt to dynamic environment, resulting in poor generalization ability; (5) in the face of hierarchical features, irregular or complex decision-making problems, classification methods may become very tough. Generally speaking, the feature-based method is essentially a feature space sparse processing method, some hidden but possibly key features can not be effectively used, which limits the improvement of its classification performance. At the same time, the process of feature extraction is complicated, which needs a lot of manpower and material resources, and the timeliness is poor.

With the development and progress of artificial intelligence algorithms, deep learning based on big data provides a new technical route to recognize which with no need for manual feature extraction. In 2012, in the ImageNet Large Scale Visual Recognition Competition (ILSVRC), the deep learning method won the championship by its absolute advantage, which set off a boom in deep learning [5]. Among the various methods of deep learning, Convolution Neural Networks (CNN) have been successfully applied in target recognition, face detection, speech recognition, semantic segmentation, and other fields. CNN does not need to carry out complicated manual feature extraction and avoids complicated pre-processing of images. It can identify the potential features of images and has excellent image classification ability, which has attracted great attention from researchers in the field of radar image processing. In 2013, Ni et al. proposed an automatic SAR target recognition method based on the visual cortex system [6]. Although the method retains the traditional feature extraction steps, it is a ground-breaking attempt to use deep network structure to achieve the target recognition. In 2014, Chen et al. used single-layer CNN to automatically extract SAR image features [7]. However, the performance of the single-layer CNN is not as good as that of an artificial feature extraction method. Wagner combined CNN and SVM by using CNN for feature extraction and SVM instead of the fully connected multi-layer perceptron in the decision-making stage for classification. It not only improves the generalization ability of the classifier but also keeps the computation time lower [8]. In 2015, Wang et al. proposed to replace the full connection layer with the sparsely connected convolution layer, which improved the over-fitting problem caused by the limited training data set [9]. In the same year, Wagner, based on his previous work, proposed a method of pre-processing by morphological component analysis, which improved the accuracy of classification to 99% [10]. In 2016, Schwegmann et al. proposed to use Convolutional Highway Unit (CHU) to build a very deep network. Thanks to CHU’s adaptive gating mechanism, flexible network configuration can be achieved to prevent gradient attenuation across multiple layers [11]. In 2017, Cho et al. proposed a CNN structure based on feature generation to improve the robustness of the network under different attitudes, noise and rotation conditions [12]. He et al. firstly trained a shallow CNN with MSTAR data set to extract the output of the first layer in forwarding propagation. Then, the position of the targets in the images is determined by maximum sampling and clustering, so as to realize the rapid unsupervised detection of the SAR targets [13]. We can see that CNN has become an important development direction of radar ATR, so it is possible to develop an ISAR image automatic target recognition method based on CNN.

However, training a deep CNN model containing millions of parameters usually requires a training set containing millions of samples to properly constrain and optimize the training process. Since the ISAR imaging principle is complex and imaging time is limited, it is difficult to obtain a large amount of ISAR image data of space targets in practice. At present, there is no public data set about ISAR images. At the same time, the space target ISAR images must be labeled by professionals to ensure effectiveness. Those difficulties make it very difficult to train deep CNN models directly for space targets recognition.

To solve this problem, transfer learning (TL) provides a new solution. The core idea of TL is to transfer the deep CNN model that has been trained on the mass data set to the new classification task through weight fine-tuning or directly as a feature extractor, so as to solve the problem that the data volume in the target classification task is too small to directly train the deep CNN model. In the field of image recognition, Alexnet [5], GoogleNet [14], Resnet [15], etc., are all successful deep CNN architectures trained in the ImageNets database, which can be downloaded directly online and used as P-DCNN. For example, in reference [16], Alexnet, a pre-trained deep network architecture, was transferred to the task of alcohol intoxication identification through weight fine-tuning, and the identification accuracy reached 97.42%±0.95%. In reference [17], the features of blood slice images were extracted by the TL method and classified by the support vector machine (SVM). The classification results show that the method has an accuracy rate of over 99% in the diagnosis of leukemia. In reference [18], to solve the problem of insufficient labeled images in SAR target recognition, a TL strategy was adopted to initialize the weights of the single shot multibox detector (SSD) model by using the pre-trained VGGNet model. The experimental
results show that the false alarm rate is lower and the target positioning accuracy is higher than the traditional method.

However, in the process of TL, we cannot guarantee that there is a P-DCNN that performs well on any target dataset. At the same time, it is difficult to know in advance which kind of P-DCNN can get a better classification effect for a specific target dataset. For this problem, ensemble learning (EL) provides a solution. EL strategies can be divided into homogeneous ensemble and heterogeneous ensemble. The homogeneous ensemble refers to combining individual learners of the same type to form a more general classifier. Currently, the most popular homogeneous ensemble methods are AdaBoost, Random Forest, Random Subspace, Gradient Boosted Decision Trees (GBDT) and XGBoost. The heterogeneous ensemble refers to combining individual learners of different categories and making decisions based on the output of these classifiers (meta-feature). Considering that we want to combine different kinds of deep CNN models, we adopt the heterogeneous ensemble strategy. In the heterogeneous ensemble strategy, the combination method of meta-features can be divided into the fixed combination method and trainable combination method. The fixed combination method includes the average method and the voting method. Compared with the fixed combination method, the trainable combination method does not deal with the meta-features directly, but adds a layer of meta-learner, takes the meta-features as input, and relearns to get a discriminant model. It can realize the effective ensemble between multiple different models and effectively overcome the influence of subjective factors in the fixed combination method. The stacking algorithm, as the most representative method of trainable combination method, has strong robustness and good classification effect. At present, stacking EL method has been widely used in credit scoring, cataract detection and other fields [19], [20], and has achieved good results. Therefore, we propose to use the stacking EL method to realize the ensemble of multiple heterogeneous P-DCNNs. It can effectively overcome the limitations of single FP-DCNN, such as weak robustness and difficulty in guaranteeing classification accuracy, thus further enhancing the space target recognition performance.

To sum up, we propose a new recognition method of space target ISAR images, which realizes the ensemble of multiple heterogeneous P-DCNNs with the stacking algorithm. In this method, we use the TL method and data augmentation (DA) operation to transfer the large-scale state of the art (SOTA) models to the task of space target ISAR image recognition, and to a certain extent, we solve the problem that it is difficult to train the deep CNNs directly when the space target ISAR image data set is small. At the same time, to further enhance the recognition effect, the stacking method is used to realize the ensemble of multiple heterogeneous P-DCNNs, which effectively overcomes the problem of weak robustness and low classification accuracy of the single transferred model. Thus, the high precision recognition of the space targets in the ISAR images is realized under the condition that the ISAR image dataset is small. The main contributions of this method are the following three aspects:

1) For the first time, the TL method is introduced into the recognition of ISAR images of space targets. By fine-tuning the weights of the deep CNN model that was pre-trained on the source data set, the large-scale SOTA model is transferred to the task of space target ISAR image recognition. It effectively solves the problem that it is difficult to train the deep CNN models directly when the ISAR image data volume is small. Compared with the traditional recognition method based on artificial feature extraction, this method can realize the automatic recognition of space targets in ISAR images without manual feature extraction, which has strong real-time performance and high robustness.

2) By DA operations such as contrast adjustment, small-angle rotation, azimuth scaling and range scaling of small space target ISAR image dataset before P-DCNNs being transferred, the over-fitting phenomenon in the transfer training process of P-DCNNs can be reduced to a certain extent. Furthermore, it can reduce the fine-tuning difficulty of the P-DCNNs and improve the classification ability of the FP-DCNNs.

3) For the first time, the stacking method is used to realize the ensemble of multiple heterogeneous P-DCNNs. The pre-trained models of Alexnet, Inception-V3, and Resnet50 are transferred by using the divided ISAR image dataset of space targets through the weight fine-tuning strategy, so as to extract the meta-features of the ISAR image dataset. Then, XGBoost is trained by the extracted meta-features to generalize the output of multiple heterogeneous FP-DCNNs models. It effectively overcomes the disadvantages of a single FP-DCNN model such as weak robustness and difficulty in ensuring classification accuracy and further improves the overall classification accuracy of space target ISAR images under a small sample set.

II. ISAR IMAGE ACQUISITION OF SPACE TARGET
A. ACQUISITION MODEL
Space target ISAR image is the distribution of the high-energy scattering centers of the space target on a two-dimensional plane. According to the electromagnetic scattering characteristics of the space target, the ISAR imaging system maps the components of the space target, such as the solar panel, antenna, and body, to a two-dimensional image projection plane, so as to obtain the ISAR image of the space target. Figure 1 shows a schematic diagram of the above process.

Figure 1 (b) depicts the projection of a three-dimensional space target on a two-dimensional imaging plane when the radar is illuminating the space target. In synthetic aperture time, the effective rotation vector of the space target is assumed to be constant. After time \( t_m \), a scattering center on the space target rotates from coordinate \( P(x, y) \) to \( P(x', y') \), with rotation angle \( \theta = \theta(t_m) \), where \( t_m \) is the slow time. The distance variation in the vertical direction is...
\[ \Delta y = -x \sin \theta (t_m) + y \cos \theta (t_m) - 1, \] then the slope distance from the radar to the scattering point is
\[ R(t_m) = R_0(t_m) + x \sin \theta (t_m) - y \cos \theta (t_m) \] (1)
where \( R_0(t_m) \) is the translation component, the distance from the radar to the center of the space target. Let the frequency modulation (LFM) signal transmitted by the radar be
\[ s(t) = \exp \left( j \pi \gamma t^2 \right) \times \exp (j2\pi f_c t) \times \text{rect} \left( t/T_p \right) \] (2)
where \( j \) is the imaginary component, \( \gamma \) is the modulation frequency, \( t \) is the fast time, \( f_c \) is the carrier frequency, \( \text{rect} (\cdot) \) is the window function, and \( T_p \) is the pulse width. The matched filter of the LFM signal is \( h(t) = s^*(t) \), where \( s^*(t) \) is the complex conjugate function. Considering that the time delay from radar to the scattering center is \( t_0 = 2R(t_m)/c \), \( c \) is the speed of light, then the fundamental frequency echo signal after the range compression can be described as
\[ \hat{s}(t, t_m) = \sum s(t) \exp \left( j \pi \gamma T_p \right) \times \text{rect}(t_m/T_m) \times \exp (-j4\pi R(t_m)/\lambda) \] (3)
where \( T_m \) is the azimuth coherence time, \( \lambda \) is the wavelength, and sinc(\( \cdot \)) is the sinc function. Then, envelope alignment and autofocus technology are used to realize the translational motion compensation (TMC) of the echo signal [21], [22]. After TMC, the instantaneous slope distance of the radar is rewritten as
\[ R(t_m) = r_0 + x \sin \theta (t_m) - y \cos \theta (t_m) \] (4)
where \( r_0 \) is the distance between the turntable center and the radar phase center. Under the small-angle assumption \( (3^\circ - 5^\circ) \), the following approximation can be obtained:
\[ \cos \theta (t_m) \approx 1 - \theta^2 (t_m)/2 = 1 - \omega^2 t_m^2/2 \] and \( \sin \theta (t_m) \approx \theta (t_m) = c t_m \), where \( \omega \) is the rotational angular velocity, then the instantaneous slope distance can be approximated as
\[ R(t_m) \approx r_0 - y + x c t_m + y \omega^2 t_m^2/2 \] (5)
Keystone transform is used to compensate for the migration through resolution cell (MTRC) caused by target rotation [23], [24]. Ignoring constant term and quadratic term, the echo signal after the range compression can be rewritten as
\[ \hat{s}(t, t_m) = \sum s(t) \cos \left( \gamma T_p (t - 2(r_0 - y)/c) \right) \times \text{rect}(t_m/T_m) \times \exp (-j4\pi x \omega t_m/\lambda) \] (6)
After the Fourier transform in the azimuth direction, the echo signal can be expressed as
\[ \hat{s}(t, f_m) = \sum s(t) \cos \left( \gamma T_p (t - 2(r_0 - y)/c) \right) \times \text{rect}(t_m/T_m) \times \exp (-j4\pi x \omega t_m/\lambda) \] (7)
where \( f_m \) represents the sampling frequency. Through coefficient substitution, the discrete expression of the signal is
\[ \hat{s}(\kappa_r, \kappa_a) = \sum A \cos \left( \beta_1 (\kappa_r - y) \right) \times \cos \left( \beta_2 (\kappa_a - x) \right) \] (8)
where \( \kappa_r = r_0 - c t/2 \), \( \kappa_a = [-M/2 : \rho_r : M/2] \), \( M \) is the size in range direction, \( \rho_r = c/(2B) \) is the resolution in range direction, \( B \) is the bandwidth of transmission signal; \( \kappa_a = \lambda f_m/2c, \kappa_a \in [-N/2 : \rho_a : N/2] \), \( N \) is the size in the azimuth direction, \( \rho_a = c/(2T_m \omega) \) is the resolution in the azimuth direction; the transformation coefficient \( A = \gamma T_p \kappa_m \beta_1 = -2\gamma T_p/c, \beta_2 = -2\omega T_m/\lambda \). Figure 1 (c) shows a schematic diagram of the two-dimensional ISAR image of the space target obtained by (8).

B. ISAR IMAGE PREPROCESSING
1) DESPECKLING
During ISAR imaging, the imaging system has limited resolution and the surface of the space target is rough relative to the wavelength of the signal, so each resolution unit contains many scattering centers, and the received space target signals are the vector superposition of the echoes of these scattering centers. Since the echo phase of each scattering center changes randomly, the amplitude and phase of each resolution unit on the ISAR image will change randomly after the vector superposition, and this random change forms the so-called speckle noise. Practice shows that the existence of speckle-noise seriously reduces the visual quality of the space target image.
ISAR image, and limits the effectiveness of subsequent interpretation operations such as target recognition [25], [26]. Therefore, it is necessary to preprocess the ISAR image to remove speckle noise, or despeckling, before using it for space target recognition.

In the process of despeckling, we should not only remove the speckle noise of the ISAR image to the greatest extent but also keep the details and edge information of space targets in the ISAR image complete. There are two ways to solve this problem. One is to suppress noise through incoherent multi-look processing, but this method has a great influence on the spatial resolution of the ISAR image. The second is filtering after imaging. At present, there are three kinds of filtering algorithms: spatial filtering, transform-domain filtering, and partial differential diffusion filtering. For spatial filtering, the algorithm is simple, effective, real-time, and has developed a large number of algorithms. Among them, the classical ones are mean filtering, median filtering, Sigma filtering, Lee filtering, Frost filtering, Gamma-MAP filtering, etc., [27]. Where, mean filtering and median filtering are realized by assigning the mean and median values of pixels in the filtering window to the center pixel of the window, respectively. However, in the smoothing process, noise and edge information cannot be effectively distinguished, resulting in reduced resolution of the edge region. Sigma filtering, Lee filtering, Frost and other filters have one common feature, that is, select the sliding filtering window on the image, take all the pixels in the window as the input value, take its local characteristics as the condition for filtering processing, and take the output as the filtering value of the central pixel of the window. And these algorithms are collectively called adaptive filtering algorithms based on local characteristics. The Lee filtering can not only suppress the speckle noise in the homogeneous region of the image but also effectively protect the edge texture and other information. And the algorithm is simple, effective and real-time. Therefore, we choose the Lee filtering algorithm for ISAR image despeckling.

The statistical characteristics of the intensity of the speckle image obey the negative exponential statistical law, in this sense, the speckle belongs to multiplicative noise [28]. The statistical model of speckle-noise is

\[ z_{i,j} = x_{i,j}v_{i,j} \]  \hspace{1cm} (9)

where \( z_{i,j} \) is the intensity of the ISAR image, \( x_{i,j} \) is the intensity of the noise-free ISAR image, \( v_{i,j} \) is the multiplicative noise with the mean value of 1 and the standard deviation of \( \sigma_v \). Assuming that the mean and variance of the sampled pixels are equal to the mean and variance of the samples in the Lee filtering window, we make Taylor expansion of (9), take the first-order approximation to linearize it, and then we can estimate the following formula according to the minimum mean square error criterion [29]

\[ \hat{x}_{Lee} = \bar{x}_{Lee} + k_{Lee} (z_{Lee} - \bar{x}_{Lee}) \]  \hspace{1cm} (10)

where \( \hat{x}_{Lee} \) is the pixel value of the ISAR image after despeckling, \( z_{Lee} \) is the signal of the noisy ISAR image in the filtering window, and \( \bar{x}_{Lee} \) is the mean value of the noise-free ISAR image in the filtering window. The adaptive filtering weight coefficient \( k_{Lee} \) in the formula (10) is

\[ k_{Lee} = \frac{\bar{z}^2_{Lee} - \hat{x}^2_{Lee}}{\bar{x}^2_{Lee} \sigma^4_{Lee} + 2 \bar{x}_{Lee} \sigma^2_{Lee} - 1 + \bar{z}^2_{Lee}} \]  \hspace{1cm} (11)

where \( \bar{z}_{Lee} \) is the mean value of the noisy ISAR image in the filtering window.

2) STANDARDIZATION

Due to the difference in distance between radar and target, there are often differences in signal level between different space target ISAR images. In order to reduce the impact of these differences on the recognition effect and increase the generalization ability of the deep CNNs, we use the Min-Pooling filter to standardize the ISAR image [30]. The calculation formula for image standardization is as follows:

\[ I_{pool} = \alpha' \times X + \beta' \times G (\sigma') \ast X + \gamma' \]  \hspace{1cm} (12)

where \( I_{pool} \) represents the normalized ISAR image, \( G (\sigma') \ast \) represents the Gaussian filter whose standard deviation is \( \sigma' \), and \( \ast \) represents the convolution operator. \( X \) represents the ISAR image after despeckling. The values of \( \alpha' \), \( \beta' \), \( \sigma' \), and \( \gamma' \) are empirically designed to be 4, -4, 10, and 128, respectively. Finally, the threshold of the ISAR image after standardization is limited. The pixel with amplitude greater than 255 is assigned to 255, and the pixel with an amplitude less than 0 is set to 0. Namely:

\[ I_{pool} (\tilde{x}, \tilde{y}) = \begin{cases} 
0, & I_{pool} (\tilde{x}, \tilde{y}) < 0 \\
255, & I_{pool} (\tilde{x}, \tilde{y}) > 255 
\end{cases} \]  \hspace{1cm} (13)

Figure 3 (a) and figure 3 (b) show examples of standardized ISAR images. On this basis, the data augmentation operation of the ISAR image will be further carried out.

3) DATA AUGMENTATION

To improve the transfer effect of deep CNN models and avoid over-fitting in the retraining process, we need a large number of training samples. However, due to the limitations of imaging principles and imaging conditions, it is difficult to obtain
a large ISAR image data set. In this case, data augmentation technology provides a new solution to overcome the above difficulties.

Data amplification technology is first applied to optical image target recognition with limited data samples. In recent years, due to the rise of deep learning, data augmentation has become an indispensable technology in the field of image recognition. The research results of SAR image data amplification prove the validity of data augmentation technology in the field of radar image recognition. For example, in 2016, J. Ding et al. introduced the data augmentation method of optical images into SAR images, carried out translation, noise addition and rotation of SAR images to expand the training set, and established the CNN model with the expanded training set to realize the target recognition of SAR images [31]. In 2017, B.Y. Ding et al. divided the SAR target into regions and reconstructed the images of each region of the SAR target by using the attribute scattering center, which expanded the SAR image sample data and improved the recognition rate of the support vector machine (SVM) classification algorithm [32]. In 2018, J.F. Pei et al. proposed a new data amplification method based on multi-perspective images, which realized multi-perspective SAR image data amplification by combining SAR images from different perspectives and realized multi-perspective SAR image target recognition by using a CNN with parallel structure [33].

In this paper, we consider that the real ISAR image is affected by many factors, such as projection plane, sampling rate, self-occlusion, nonlinear scattering mechanism, the relative attitude between the target and the radar, which results in strong variability of scattering point intensity and spatial distribution of ISAR image. Therefore, our augmentation operations for each original ISAR image mainly include: change the image contrast; rotate the image at different angles; use bilinear interpolation method to scale the ISAR image in azimuth and range direction.

When adjusting the contrast of the ISAR image, we can realize it by nonlinear transformation, which can be expressed as:

$$I_{DA_1} = I_{pool} + \epsilon (E - I_{pool}) \times I_{pool}$$

(14)

where $I_{pool}$ represents the standardized ISAR image and $I_{DA_1}$ represents the contrast adjusted ISAR image. $\epsilon$ is a user-predefined adjustable factor with a value between 0 and 1. $E$ is a unit matrix with the same dimension as $I_{pool}$.

Figure 3 (a) and Figure 3 (c) show an example of the ISAR image contrast adjustment.

In the rotation transformation of ISAR image, the mapping relationship between the coordinates after rotation and the original coordinates can be expressed as follows:

$$\begin{align*}
    x_{\text{rotate}} &= x_{\text{initial}} \cos \theta_{\text{rotate}} + y_{\text{initial}} \sin \theta_{\text{rotate}} \\
    y_{\text{rotate}} &= -x_{\text{initial}} \sin \theta_{\text{rotate}} + y_{\text{initial}} \cos \theta_{\text{rotate}}
\end{align*}$$

(15)

where $(x_{\text{rotate}}, y_{\text{rotate}})$ and $(x_{\text{initial}}, y_{\text{initial}})$ represent the coordinates of the scattering point after rotation and before rotation respectively. $\theta_{\text{rotate}}$ is the rotation angle. As for the problem of how much rotation angle to take, the author in literature [34] proves that the position and intensity of the backscatter characteristics of the target are rotationally invariant in a small angle range of at least 5°. So we choose the rotation angle as D, which not only can not change the radar scattering characteristics of space targets but also can achieve ISAR image augmentation. Figure 3 (a) and Figure 3 (d) show an example of the small-angle rotation transformation.

In the scale transformation of the ISAR image, we express the source ISAR image as $Q_{a_1 \times b_1}$, the target ISAR image as $Q_{a_2 \times b_2}$, and the scale transformation factors of azimuth and range direction as $a_2/a_1$ and $b_2/b_1$ respectively. If $a_2/a_1 = 1$, it means that we only scale ISAR image in range direction; if $b_2/b_1 = 1$, it means that we only scale ISAR image in the azimuth direction.

The pixel value of the target image at point $(x_{\text{pixel}}, y_{\text{pixel}})$ corresponds to the pixel value of the source image at position $(x'_{\text{pixel}} \times a_2/a_1, y'_{\text{pixel}} \times b_2/b_1)$, however, since both $a_2/a_1$ and $b_2/b_1$ may be floating-point numbers, $(x'_{\text{pixel}} \times a_2/a_1, y'_{\text{pixel}} \times b_2/b_1)$ may also be floating-point numbers. At this time, point $(x'_{\text{pixel}} \times a_2/a_1, y'_{\text{pixel}} \times b_2/b_1)$ does not exist in the source image. To get the pixel value of the target image in position $(x'_{\text{pixel}}, y'_{\text{pixel}})$, we need to use the four nearest neighbor pixels $(x'_{\text{near1}}, y'_{\text{near1}})$, $(x'_{\text{near1}} + 1, y'_{\text{near1}})$, $(x'_{\text{near1}}, y'_{\text{near1}} + 1)$, $(x'_{\text{near1}} + 1, y'_{\text{near1}} + 1)$ around point $(x'_{\text{pixel}} \times a_2/a_1, y'_{\text{pixel}} \times b_2/b_1)$ in the source image for bilinear interpolation and calculate the pixel value of the target image in position $(x_{\text{pixel}}, y_{\text{pixel}})$. Figure 2 shows the interpolation process.

In Figure 2, $\lambda$ and $h$ represent the corresponding weight values. The target image can be calculated by traversing every

**FIGURE 3.** Data augmentation example. (a) Despeckling; (b) standardization; (c) contrast adjustment; (d) small-angle rotation; (e) azimuth scaling; (f) range scaling.
point in the target image. The calculation formula is

\[ Q(x_{\text{pixel}}, y_{\text{pixel}}) = r_1 + r_2 + r_3 + r_4 \]

\[ r_1 = (1 - \lambda)(1 - h) \times Q'(x'_{\text{near1}} + 1, y'_{\text{near1}} + 1) \]

\[ r_2 = \lambda(1 - h) \times Q'(x'_{\text{near1}} + 1, y'_{\text{near1}}) \]

\[ r_3 = h(1 - \lambda) \times Q'(x'_{\text{near1}}, y'_{\text{near1}} + 1) \]

\[ r_4 = \lambda h \times Q'(x'_{\text{near1}}, y'_{\text{near1}}) \] (16)

Figure 3 (a), figure 3 (e), and figure 3 (f) show examples of azimuth scaling and range scaling.

The above-mentioned data amplification operation makes the number of training samples increase several times, which can meet the requirements of the sample data amount in the retraining of the deep convolution neural network model to a certain extent.

III. ENSEMBLE LEARNING OF MULTIPLE P-DCNNS BASED ON STACKING ALGORITHM

A. ENSEMBLE LEARNING FRAMEWORK CONSTRUCTION BASED ON STACKING ALGORITHM

Stacking is a model fusion technology that realizes the ensemble of multiple heterogeneous models to generate a new model. It generalizes the output of multiple models to improve the overall classification accuracy. In the stacking ensemble learning framework (SELF), we should not only consider the individual classification ability of each base learner but also analyze the ensemble effect of each base learner, so that the SELF can get the best classification effect.

When using the stacking method to realize the ensemble of the P-DCNNs, we first use the TL idea to fine-tune the weights of the P-DCNNs on the divided space target ISAR image data set to extract the meta-features. On this basis, the second layer of SELF is constructed, and the selected meta-learner in the second layer is trained by using the meta-features proposed in the first layer. Finally, the trained meta-learner is used to output the classification results of ISAR images of space targets. Figure 4 shows the schematic of the SELF.

The meta-features used for training meta-learners are generated by the base learners. If the training set of the base learners is directly used to generate meta-features, serious over-fitting may occur. Therefore, to prevent the training data from being repeatedly learned by double-layer learners and avoid the occurrence of the “over-fitting” effect, it is necessary to divide the training data reasonably. First of all, we divide the original data set into a training set and test set. According to the number of selected base learners, we divide the training data set into K data blocks randomly and ensure that the data IDs of each block do not overlap with each other. For each single base learner, one data block is used for testing, and the rest K-1 data blocks are used for training. After each round of training, each base learner can output a set of classification results for the data block used for testing. Then all the output results are combined to get the meta-features of the training set, whose size is the same as that of the training set.

The detailed ensemble process is as follows:

Firstly, we divide the space target ISAR image data set \( S = \{(y_i, x_i), i = 1, \ldots, N\} \) into training set \( S_{\text{train}} = \{(y_{tr}, x_{tr}), tr = 1, \ldots, N_{\text{train}}\} \) and test set \( S_{\text{test}} = \{(y_{te}, x_{te}), te = 1, \ldots, N_{\text{test}}\} \), where \( x_i \) is the i-th ISAR image sample in data set \( S \), and \( y_i \) is the corresponding space target category. \( x_{tr} \) is ISAR image sample in training set \( S_{\text{train}} \), \( y_{tr} \) is the corresponding space target category; \( x_{te} \) is ISAR image sample in test set \( S_{\text{test}} \), \( y_{te} \) is the corresponding space target category; and \( N_{\text{train}} + N_{\text{test}} = N \).

Choose \( P \) P-DCNNs as the base learners of the first layer of SELF. Then divide the training set \( S_{\text{train}} \) randomly into \( K = P \) equal size training data blocks \( S_1, S_2, \ldots, S_K \). Take the \( k \)-th training data block \( S_k(k = 1, 2, \ldots, K) \) as the test data and the remaining \( K-1 \) data blocks (represented by \( S_{-k} \)) as the training data, and \( S_{-k} = S - S_k \). Then augment the \( S_{-k} \) according to the data augmentation method. We use the augmented \( S_{-k} \) to fine-tune the weights of \( P \) P-DCNN base

![Figure 4. Schematic of SELF.](image-url)
learners in turn to obtain the fine-tuned base learners $L_{k,p}(p = 1, 2, \ldots, P)$. Base learners $L_{k,p}(p = 1, 2, \ldots, P)$ have then used to output the recognition probability $z_{k,p}(p = 1, 2, \ldots, P)$ and $z_{\text{test},k,p}(p = 1, 2, \ldots, P)$ of the training data block $S_t$ and test set $S_{\text{test}}$ respectively. Repeat the above operations until $k = K$.

Combine the classification probabilities of all training data blocks under each type of base learner, and then splice the combined results of all $P$ groups to get the meta-features $\{(z_{1,1}; z_{2,1}; \ldots; z_{1,k}); (z_{1,2}; z_{2,2}; \ldots; z_{2,k}); \ldots; (z_{1,p}; z_{2,p}; \ldots; z_{k,p})\}, p = 1, \ldots, P$ of the training set; at the same time, add and average all the classification probabilities corresponding to test set under each type of base learner, and then splice the averaged results of all $P$ groups to get the meta-features

$$\left\{ \left( \sum_{k=1}^{K} z_{\text{test},k,1}/K; \sum_{k=1}^{K} z_{\text{test},k,2}/K; \ldots; \right. \sum_{k=1}^{K} z_{\text{test},k,p}/K \bigg), p = 1, \ldots, P \right\}$$

of the test set.

Input the meta-features of the training set to the second layer of SELF to train the meta-learner. We represent the trained meta-learner as $L_{\text{new}}$. SELF’s configuration mode enables the classification results of the base learners in the first layer to be used for the training of the meta-learner in the second layer, so as to find and correct the classification errors of the base learners and improve the classification accuracy of the model. Algorithm 1 gives the detailed ensemble steps under SELF.

### B. TRANSFER TRAINING OF BASE LEARNERS

In this section, we use the weight fine-tuning TL strategy to retrain the base learners in the first layer of SELF. On this basis, we realize the automatic feature extraction of the space target ISAR images and avoid the construction of the artificial feature.

TL refers to using the existing source domain information to solve the related target domain problems, to apply the useful information learned from the source domain to the target domain. That is, for a given task $\Gamma_T$ in the target domain $D_T$, the purpose of TL is to reduce the difficulty of solving $\Gamma_T$ by using the knowledge learned when solving task $\Gamma_S$ in the source domain $D_S$. For space target ISAR image recognition, if we reconstruct a new deep CNN model and only use the small set of space target ISAR images to train from scratch, the model training will take a long time and be easy to overfit. Therefore, referring to the TL idea, we transfer the deep CNN pre-trained on the source domain, i.e. large-scale dataset ImageNets to the space target ISAR image recognition task, and then use the target domain, i.e. space target ISAR image dataset to retrain the P-DCNN. On this basis, we realize the automatic recognition of space targets and solve the difficulty of training deep CNN from scratch.

To make the ensemble model get better classification performance, we also need to choose the P-DCNN models with a large difference as the base learners in the first layer of SELF. This is because sample data could be observed at different spatial and structural angles for different models.

#### Algorithm 1 Ensemble Method

**1. Input:**
- Space target ISAR image data set $S = \{(y_i, x_i), i = 1, \ldots, N\}$.

**2. Data Partitioning**

1) Divide the space target ISAR image data set $S$ into the training set $S_{\text{train}} = \{(y_{tr}, x_{tr}), tr = 1, \ldots, N_{\text{train}}\}$ and test set $S_{\text{test}} = \{(y_{te}, x_{te}), te = 1, \ldots, N_{\text{test}}\}$.

2) Divide the training set $S_{\text{train}}$ into $K$ equal size training data blocks $S_1, S_2, \ldots, S_K$, where $K = P, P$ is the number of base learners, and $S_{-k} = S - S_k$.

**3. Construct the first layer of SELF**

1) Select $P$ P-DCNNs as base learners;

2) For $(k = 1; k \leq K; k + +)$

   - Augment the $S_{-k}$ according to the data augmentation method;

   - Use the augmented $S_{-k}$ to fine-tune the weights of $P$ P-DCNNs (base learners);

   - Use fine-tuned base learners $L_{k,p}$ to output the recognition probability $z_{k,p}$ of the test data block $S_k$;

   - Use fine-tuned base learners $L_{k,p}$ to output the recognition probability $z_{\text{test},k,p}$ of the test set $S_{\text{test}}$;

3) Combine the classification probabilities of all training data blocks under each type of base learner, and then splice the averaged results of all $P$ groups to get the meta-features

$$\left\{ \left( \sum_{k=1}^{K} z_{\text{test},k,1}/K; \sum_{k=1}^{K} z_{\text{test},k,2}/K; \ldots; \right. \sum_{k=1}^{K} z_{\text{test},k,p}/K \bigg), p = 1, \ldots, P \right\}$$

of the test set.

**4. Construct the second layer of SELF**

Input the meta-features of the training set to the second layer of SELF to train the meta-learner, and the trained meta-learner is represented as $L_{\text{new}}$.

**5. Output:**

- Use the trained meta-learner $L_{\text{new}}$ to classify the test set, and output the classification accuracy.
Therefore, the selection of P-DCNNs with large differences can best reflect the advantages of different P-DCNNs, so that different P-DCNNs can complement each other. In this paper, for the P-DCNNs, we select three network structures: Alexnet, Inception-V3, and Resnet50. These three kinds of P-DCNNs have completely different structures and all have certain representativeness. From Alexnet to Resnet50, the number of layers of the network is increased, and Inception-V3 makes the network “wider”, and Resnet50 adds “cross-layer connection”. The image size of the input layer of Alexnet is 227 pixels × 227 pixels, that of Inception-V3 is 299 pixels × 299 pixels, and that of Resnet50 is 224 pixels × 224 pixels.

There are two TL strategies for P-DCNN. One is to use P-DCNN as a feature extractor to extract the deep feature of ISAR images of space target directly; the other is to retain the weights of P-DCNN’s feature extraction layers, randomly initialize the weights of P-DCNN’s classification layer, set different learning rates for the feature extraction layers and classification layer, and then retrain the P-DCNN with the space target ISAR image data set to obtain a new deep CNN. The first TL strategy does not need to change the structure and weights of P-DCNN and takes less time. However, due to the large difference between the samples in the ImageNet library and the ISAR images of space target, the direct extraction of ISAR image features for classification will lead to the limitation of classification performance. To obtain a better TL effect, we adopted the second TL strategy, which uses the ISAR image data set of space targets to fine-tune the weights of three kinds of P-DCNN.

The three kinds of P-DCNN mentioned above have been fully trained on ImageNet, and the features extracted from the shallow and middle layers (such as edge and corner features) will have certain universality. Meanwhile, compared with the tens of millions of weights in the three P-DCNNs, the amount of data in the space target ISAR image dataset is only in the order of hundreds. Based on the above considerations, we fix the weights of the convolution modules in the shallow and middle layers, set the weights of the deep parts near the classification layer to the trainable state, and then retrain the P-DCNN by the space target ISAR image data set. In this way, P-DCNN can adaptively adjust the deep weights according to the characteristics of ISAR image samples of space targets and then enhance the global generalization ability of the transferred deep CNN models. At the same time, the convergence rate of P-DCNN retraining is accelerated to a certain extent. The above TL process is shown in Figure 5.

Since the three kinds of P-DCNN structure we adopted are quite different, when we use the target ISAR image data to retrain the P-DCNN, the network layers which chose to be fine-tuned is also different. Alexnet is an 8-layer network structure, which consists of 5 convolutional layers and 3 full-connection layers. The convolution kernel size used in the convolution layer is 11 × 11 and 3 × 3. Here we name the later three full connection layers fc6, fc7 and fc8 respectively. During the fine-tuning training for Alexnet, keep the weights of all convolutional layers and fine-tune the later 3 full-connection layers fc6-fc8. We design three kinds of training strategies for the full connection layer, which are: [(fc6, fc7), fc8], [fc6, (fc7, fc8)], and [(fc6, fc7, fc8)]. Where [(fc6, fc7), fc8] means to retain the weights of fc6 and fc7, only fine-tuning the weights of fc8 layer, others are similar to this. Figure 6(a) shows the above three fine-tuning training methods.

Inception-V3 generally consists of convolutional layers, pooling layers, and 3 Inception module groups. The Inception module group makes the network “wider”, which results in a significant reduction in the number of weights for the entire network. When we fine-tune the weights of Inception-V3, we study two schemes. One is to use the space target ISAR image data set to train only the last full connection layer, while the weights of other layers are all frozen. Another option is to add a full connection layer to the Inception-V3 model, retrain the two full connection layers, and the weights of other layers are all frozen. Figure 6(a) shows the above two fine-tuning training methods.

Resnet50 is a network formed by the stacking of residual learning modules. During the training, the deep error is directly transmitted to the shallow layer through the residual module. These Shortcut Connections ensure smooth data flow in the middle of the network, thereby avoiding some of the under-fitting that occurs when gradients disappear. Thus, Resnet50 can effectively enhance the expression ability, reduce the gradient dispersion phenomenon, and enhance the feature learning ability and recognition performance while deepening the network layer. In the fine-tuning training of resnet50, we only train the last full connection layer. Figure 6(c) shows the above fine-tuning training method.

Based on the above training methods and algorithm 1, we construct the classification model in the first layer of SELF. The classification model in the first layer is used to extract the meta-features of the space target ISAR image data set for the training of the subsequent meta-learner. It should be noted that in the classification task of this paper, we use the softmax layer in each base learner to output the probability value of the corresponding category of the sample image.
Where the softmax function is expressed as

\[
h_{\tilde{\theta}}(x^{(i)}) = \begin{bmatrix}
p(y^{(i)} = 1|x^{(i)}; \tilde{\theta}) \\
p(y^{(i)} = 2|x^{(i)}; \tilde{\theta}) \\
p(y^{(i)} = 3|x^{(i)}; \tilde{\theta}) \\
\vdots \\
p(y^{(i)} = n|x^{(i)}; \tilde{\theta})
\end{bmatrix}
\]

\[
= \frac{1}{\sum_{j=1}^{n} \exp (\tilde{\theta}_j^T x^{(i)})} \begin{bmatrix}
\exp (\tilde{\theta}_1^T x^{(i)}) \\
\exp (\tilde{\theta}_2^T x^{(i)}) \\
\exp (\tilde{\theta}_3^T x^{(i)}) \\
\vdots \\
\exp (\tilde{\theta}_n^T x^{(i)})
\end{bmatrix}
\]

(17)

where \(\sum_{j=1}^{n} \exp (\tilde{\theta}_j^T x^{(i)})\)^{-1} is to normalize the probability distribution; \(n\) is the number of space target categories; \(x^{(i)}\) is the input variable of the \(i\)-th sample after calculation; \(\tilde{\theta}\) is the model parameter; \(\tilde{\theta}^T\) stands for the transposition of the \(\tilde{\theta}\).

**C. META-LEARNER CONSTRUCTION BASED ON XGBOOST**

The meta-learner in the second layer of SELF should be selected as the model with a strong generalization ability to correct the bias of multiple base learners on the training data. As an ensemble learning algorithm using the Boosting strategy to carry out tree Boosting, XGBoost [35] has achieved
excellent performance in classification, regression, sorting and many other problems. It is one of the most popular algorithms in Kaggle and other data competitions and is widely used in academia and industry. So we choose XGBoost as the meta-learner in the second layer of SELF.

The newly added tree model  \( \hat{y}_{tr} \) is the meta-leaf node. For the multi-target recognition problem, we use softmax as the loss function as far as possible. Next, we will briefly introduce the optimization principle of the XGBoost, and the specific optimization process can be seen in literature [35].

The optimization process can be seen in literature [35].

The optimization principle of the XGBoost is a multi-classification problem, we use softmax as the loss function. \( \hat{y}_{tr} \) is the predicted value of the \( tr \)-th sample. For the multi-target recognition problem, the dimension of \( \hat{y}_{tr} \) is equal to the number of categories of space targets. Each dimension value of \( \hat{y}_{tr} \) represents the discrimination probability of the space target corresponding to that dimension.

Similar to other optimization processes, such as cost function optimization in literature [36] and \( H_\infty \) function optimization in literature [37], the optimization process of XGBoost is to minimize the following loss function:

\[
L(F) = \sum_{tr=1}^{N_{\text{train}}} l(\hat{y}_{tr}, y_{tr}) + \sum_{t=1}^{T} \Gamma(f_t) = \bar{\gamma}V + 0.5\lambda \|w\|^2 \tag{19}\]

where \( l(\hat{y}_{tr}, y_{tr}) \) is a differentiable convex loss function used to measure the difference between the predicted value \( \hat{y}_{tr} \) and the true value \( y_{tr} \). Since multi-target recognition is a multi-classification problem, we use softmax as the loss function; \( N_{\text{train}} \) is the number of training samples; \( \Gamma(f_t) \) is the regularizer used to control the complexity of the model; \( V \) is the number of leaf nodes in the tree; \( \bar{\gamma} \) is the regularization parameter of the number of leaves; and \( \lambda \) is the regularization parameter of the weights of leaves.

In the process of sequence minimal optimization in (19), the newly added tree model \( f_t \) should minimize the loss function as far as possible. Next, we will briefly introduce the optimization principle of the XGBoost, and the specific optimization process can be seen in literature [35].

The loss function of the \( t \)-th round can be written as:

\[
L^{(t)} = \sum_{tr=1}^{N_{\text{train}}} l(\hat{y}_{tr}^{(t-1)}, y_{tr}) + f_t(z_{tr}), y_{tr}) + \Gamma(f_t) + \sum_{t=2}^{T-1} \Gamma(f_{t-1}) \tag{20}\]

Through the second-order Taylor expansion of (20), we can get:

\[
L^{(t)} \simeq \sum_{tr=1}^{N_{\text{train}}} \left( l(\hat{y}_{tr}^{(t-1)}, y_{tr}) + g_{tr}f_t(z_{tr}) + 0.5h_{tr}f_t^2(z_{tr}) \right) + \Gamma(f_{t+1}) \tag{21}\]

where \( g_{tr} = \partial_l l(\hat{y}_{tr}^{(t-1)}, y_{tr}) \), \( h_{tr} = \partial^2 l(\hat{y}_{tr}^{(t-1)}, y_{tr}) \).

When the \( t \)-th tree is added, the structure of the previous \( t-1 \) trees is fixed. So the loss \( \sum_{tr=1}^{N_{\text{train}}} l(\hat{y}_{tr}^{(t-1)}, y_{tr}) + \sum_{t=2}^{T-1} \Gamma(f_{t-1}) \) is constant, and \( I_v = \{ tr | q(z_{tr}) = v \} \) is the set of data samples that belong to the \( v \)-th leaf node. After the constant term is removed, the approximate expression of \( L^{(t)} \) can be obtained:

\[
\hat{L}^{(t)} \simeq \sum_{tr=1}^{N_{\text{train}}} \left( g_{tr}f_t(z_{tr}) + 0.5h_{tr}f_t^2(z_{tr}) \right) + \Gamma(f_t) = \sum_{i=1}^{V} \left( \sum_{tr \in I_v} g_{tr}w_r + 0.5(\sum_{tr \in I_v} h_{tr} + \hat{\lambda})w_r^2 \right) + \tilde{\gamma}V \tag{22}\]

where \( w_v \) represents the weight of the \( v \)-th leaf node.

Let \( \partial_q \hat{L}^{(t)} = 0 \), then for the fixed CART structure \( q(x) \), the optimal weight of the \( v \)-th leaf node is

\[
w_v^{\text{opt}} = -\sum_{tr \in I_v} g_{tr} / \sum_{tr \in I_v} h_{tr} + \hat{\lambda} \tag{23}\]

Substitute \( w_v^{\text{opt}} \) into (22), then the optimal loss function can be expressed as:

\[
\hat{L}_v^{\text{opt}} = -0.5 \sum_{i=1}^{V} \left( \sum_{tr \in I_v} g_{tr}^2 \right) \left( \sum_{tr \in I_v} h_{tr} + \hat{\lambda} \right) - \tilde{\gamma}V \tag{24}\]

Therefore, we need to minimize (24) to obtain the optimal structure \( q_v^{\text{opt}} \) of the \( t \)-th CART. Assume that \( I_l \) is the sample set before splitting, and \( I_L \) and \( I_R \) are the sample sets of the left and right subtrees after splitting, that is, \( I = I_l + I_R \). Then, the gain \( L_{\text{Gain}} \) of the loss function after splitting can be obtained as:

\[
L_{\text{Gain}} = 0.5 \left( \sum_{tr \in I_L} g_{tr}^2 \right) \left( \sum_{tr \in I_L} h_{tr} + \hat{\lambda} \right) - \sum_{tr \in I_R} g_{tr}^2 \left( \sum_{tr \in I_R} h_{tr} + \hat{\lambda} \right) - \tilde{\gamma} \tag{25}\]

To get the optimal tree structure \( q_v^{\text{opt}} \), the maximum gain is calculated by (25) in each iteration. And the specific iterative optimization process can be seen in literature [35].

At this point, we can use the meta-features obtained by the base learners in the first layer of SELF to adjust the weights of the meta-learner in the second layer of SELF.

In summary, we show the overall algorithm flow chart of space target ISAR image recognition based on SELF in Figure 7.
IV. EXPERIMENT AND PERFORMANCE ANALYSIS

A. EXPERIMENTAL SETUP AND PARAMETER ENVIRONMENT

When building the ISAR image data set of the space targets, we first use 3DMAX software to build the 3D mesh models of 5 types of space targets: Satellite1, Satellite2, Satellite3, Satellite4, and Satellite5. The mesh cell size is 0.1m, and the specific structure of the 3D mesh models are shown in figure 8.

Then we set up the ISAR imaging simulation environment according to the normal state of the satellite in orbit. In the simulation environment of the satellite tool kit (STK), the space targets are placed on the circular orbit with an orbital altitude of 788.9km, an orbital inclination of 98.57°, and a right ascension of ascending node of 99.44°. Meanwhile, the radar is placed at 29.5° N and 119° E. The space targets are illuminated by LFM signal with pulse frequency $f_c = 10$GHz, bandwidth $B = 1$GHz, and pulse width $T_p = 1 \times 10^{-5}$s. The radar echoes are collected with the sampling frequency $f_m = 1 \times 10^7$Hz. Then, according to the ISAR image acquisition method, we process the received radar echoes to obtain 1325 initial ISAR images of space targets. According to the despeckling method of the ISAR image, we continue to suppress the speckle of the initial ISAR images and show some ISAR images after despeckling in Figure 9.
FIGURE 9. Some ISAR image samples of five kinds of satellites after despeckling.
Then the ISAR image samples are standardized and the processing results are shown in Figure 10 (b). The standardized ISAR image samples are first randomly divided into the training data and test data at a ratio of 4:1, and then training data are randomly divided into the training dataset and verification dataset at a ratio of 9:1. According to the data augmentation method, we expand the size of the training dataset to 8595. Figure 10 shows an example of the data augmentation operation for each type of space target ISAR image. In table 1, the size of the verification dataset, test dataset and augmented training dataset are given for each type of space target.

Under the TL strategy of fine-tuning weights, we need to set the super-parameters of the P-DCNNs, which mainly include learning rate, batch size, and epoch. The learning rate refers to the degree of network weights update. Setting too high will cause the model to fail to converge, and setting too low will cause the network to converge too slowly. Therefore, we set the learning rate to 0.001. Batch size refers to the number of samples selected during each training process. Increasing the batch size will reduce the vibration amplitude during network training, make the network easier to converge, but also increase the memory consumption. So we set the batch_size of to 64. Epoch refers to the number of times the training set is repeatedly trained in the whole training process. To control the overall training time, we set epoch to 10.

The hardware environment of the experiment is mainboard DELL INC.0F8T29, CPU Intel Core i7-8850h, graphics card Intel(R) UHD Graphics 630, memory SK HYNIX 16GB.

**TABLE 2. Super-parameter settings.**

| P-DCNN  | Fine-tuning Layer | Learning Rate | Batch size | Epoch |
|--------|-------------------|---------------|------------|-------|
| Alexnet | fc8               | $10^{-3}$     | 64         | 10    |
|        | fc7,fc8           |               |            |       |
|        | fc6,fc7,fc8       |               |            |       |
|        | last layer        |               |            |       |
| Inception-V3 | fc+ last layer  | $10^{-3}$     | 64         | 10    |
| Resnet50 | last layer       | $10^{-3}$     | 64         | 10    |

In table 2, the fine-tuning training of Alexnet includes three ways: a) train the fc8 full connection layer, and it is hereinafter referred to as “Alexnet-fc8”; b) train the fc7 and fc8 full connection layers, and it is hereinafter referred to as “Alexnet-fc7,8”; c) train the fc6, fc7, and fc8 full connection layers, and it is hereinafter referred to as “Alexnet-fc6,7,8”. Inception-V3 is fine-tuned in two ways: a) train the last full connection layer, and it is hereinafter referred to as “Inception-V3-a0”; b) add a new full connection layer in front of the last full connection layer, train the next two full connection layers, and it is hereinafter referred to as “Inception-V3-a1”. When Resnet50 is fine-tuned, only retrain the last full connection layer, and it is hereinafter referred to as “Resnet50-a0”.

**B. PERFORMANCE METRICS**

In the experiment, we use the classification accuracy ACC, cross-entropy loss LOSS, precision ratio PPV, sensitivity TPR, and comprehensive evaluation index $F_1$ to evaluate the
classification performance of the model. The calculation method is as follows:

$$\text{Acc} = \frac{N_{\text{correct}}}{M_{\text{total}}}$$

$$\text{Loss} = -\sum_{i=1}^{M_{\text{total}}} \sum_{j=1}^{K_{\text{class}}} y_{ij} \log \hat{y}_{ij}$$

$$\text{PPV} = \frac{T_P}{T_P + F_P} \times 100\%$$

$$\text{TPR} = \frac{T_P}{T_P + F_N} \times 100\%$$

$$F_1 = \frac{2\text{PPV} \times \text{TPR}}{\text{PPV} + \text{TPR}} \times 100\%$$

(26)

where $M_{\text{total}}$ is the total number of samples; $N_{\text{correct}}$ is the number of correctly predicted samples; $K_{\text{class}}$ is the number of categories of samples; $y_{ij} = \{0, 1\}$. In the $i$-th sample event, if $y_{ij}$ is class $j$, it is 1; otherwise, it is 0; $\hat{y}_{ij}$ is the output of class $j$ in the $i$-th sample event. $T_P$ (True Positive) represents the number of samples that are positive and predicted to be positive, $F_N$ (False Negative) represents the number of samples that are positive and predicted to be negative, and $F_P$ (False Positive) represents the number of samples that are negative but predicted to be positive. $F_1$ is a weighted average of precision ratio PPV and sensitivity TPR, between 0 and 1. The higher the $F_1$ is, the better the comprehensive performance of the classification model is.

C. SELECTION OF BASE LEARNERS

1) MODEL TRAINING

a: ALEXNET’S TRAINING EFFECT UNDER THREE FINE-TUNING TRAINING WAYS

First of all, we use the divided data to do fine-tuning training for Alexnet under three kinds of fine-tuning ways and evaluate the training process. The number of training iterations is 1490. The training accuracy, verification accuracy, and cross-entropy loss in the training process are shown in Figure 11. The fine-tuning training of “Alexnet-fc8” takes 8 min 8 s, “Alexnet-fc7,8” takes 12 min 55 s, and “Alexnet-fc6,7,8” takes 15 min 23 s.

From Figure 11 (a) - (d), we can see that with the increase of iterations, the test accuracy and verification accuracy are increasing, the training cross-entropy loss and test cross-entropy loss are decreasing, and they are gradually stable. After 1490 iterations of training, the training effect of “Alexnet-fc6,7,8” fine-tuning way is relatively best. The training accuracy, training cross-entropy loss, verification accuracy, and verification cross-entropy loss are 84.10%, 0.49, 94.29% and 0.20 respectively. The training effect of “Alexnet-fc8” is the worst. The training accuracy, training cross-entropy loss, verification accuracy, and verification cross-entropy loss are 68.57%, 0.74, 82.86% and 0.48 respectively. The training effect of “Alexnet-fc7,8” is very close to that of “Alexnet-fc6,7,8”. The training accuracy, training
cross-entropy loss, verification accuracy, and verification cross-entropy loss are 88.45%, 0.29, 90.48% and 0.23 respectively.

b: INCEPTION-V3’s TRAINING EFFECT UNDER TWO FINE-TUNING TRAINING WAYS

Based on the divided data, the fine-tuning training of Inception-V3 under the two fine-tuning ways is carried out, and the training process is evaluated. The number of training iterations is 1490. The training accuracy, verification accuracy, and cross-entropy loss in the training process are shown in Figure 12. The fine-tuning training of “Inception-V3-a0” takes 74 min 20 s, and “Inception-V3-a1” takes 75 min 26 s.

From Figure 12 (a) - (d), we can see that with the increase of iterations, the test accuracy and verification accuracy are increasing, the training-cross entropy loss and test cross-entropy loss are decreasing, and they are gradually stable. The training accuracy, training cross-entropy loss, verification accuracy, and verification cross-entropy loss of “Inception-V3-a1” fine-tuning way are 75.68%, 0.73, 64.76% and 0.94 respectively. Compared with “Inception-V3-a1”, the “Inception-V3-a0” fine-tuning way has faster convergence speed and better training effect. Its training accuracy, training cross-entropy loss, verification accuracy, and verification cross-entropy loss are 81.12%, 0.59, 70.48% and 0.82 respectively.

c: RESNET50’s TRAINING EFFECT

Fine-tune the Resnet50 with the partitioned data and evaluate the training process. The number of training iterations is 1490. The training accuracy, verification accuracy, and cross-entropy loss in the training process are shown in Figure 13. And the fine-tuning training takes 51 min 33 s.

From Figure 13 (a) - (d), we can see that with the increase of iterations, the test accuracy, and verification accuracy are increasing, the training cross-entropy loss and test cross-entropy loss are decreasing. At the end of the training, the training accuracy, and verification accuracy are 87.92% and 82.56% respectively, and the training cross-entropy loss and verification cross-entropy loss are 0.33 and 0.51 respectively.

2) MODEL TESTING

Next, we use the above FP-DCNNs to classify the test data. The test confusion matrices of the fine-tuned models of Alexnet, Inception-V3, and Resnet50 are shown in Figure 14, Figure 15 and Figure 16, respectively. The final classification results are given in Table 3.

Table 3 shows the training time and corresponding test results of each FP-DCNN under different fine-tuning training ways, in which Macro Average represents the average value of the evaluation indexes of five types of satellite. For the three fine-tuning training ways of Alexnet, the corresponding
FIGURE 13. ResNet50’s transfer training process in 10 epochs. (a) Training accuracy; (b) training cross-entropy loss; (c) verification accuracy; (d) verification cross-entropy loss.

FIGURE 14. AlexNet’s test results under three fine-tuning training ways. (a) Test confusion matrix under “Alexnet-fc8” fine-tuning training way; (b) test confusion matrix under “Alexnet-fc7,8” fine-tuning training way; (c) test confusion matrix under “Alexnet-fc6,7,8” fine-tuning training way.

In terms of training time, “Alexnet-fc7,8” is 2 min 28 s faster than “Alexnet-fc6,7,8”. At the same time, the FP-DCNN obtained by using “Alexnet-fc7,8” is higher than that of “Alexnet-fc6,7,8” in Macro Average PPV and Macro Average F1. And the index of Macro Average TPR is only 0.36% lower than that of “Alexnet-fc6,7,8”. So in the next model ensemble experiment, we choose Alexnet-fc7,8 as the first base learner of SELF. For the two fine-tuning training ways of Inception-V3, “Inception-V3-a1” is better than “Inception-V3-a0” in classification effect and training time. Therefore, we use Inception-V3-a0 as the second base learner of the SELF. ResNet50-a0 is used as the third base learner.

D. MODEL ENSEMBLE EXPERIMENT
1) SUB-DATASET DIVISION OF TRAINING DATA
In this section, we redivide the training data and test data. According to the number of base learners selected, we randomly divide the training data into three sub-datasets, and each sub-dataset ID is not repeated with each other, as shown...
FIGURE 15. Inception-V3’s test results under two fine-tuning training ways. (a) Test confusion matrix under “Inception-V3-a0” fine-tuning training way; (b) test confusion matrix under “Inception-V3-a1” fine-tuning training way.

TABLE 3. Test results of FP-DCNNs.

| Fine-tuning Ways | Training Time | Performance Metrics | Target | Macro Average(%) | Total Acc (%) |
|------------------|---------------|---------------------|--------|------------------|---------------|
|                  |               | PPV (%)             | Satellite1 | 97.78          | 87.50         |
| Alexnet-fc8      | 8 min 8 s     | TP (%)              | Satellite2 | 83.02          | 82.64         |
|                  |               | F1 (%)              | Satellite3 | 89.80          | 89.72         |
|                  |               | PPV (%)             | Satellite4 | 98.00          | 98.04         |
|                  |               |                     | Satellite5 | 90.73          | 90.74         |
| Alexnet-fc7,8    | 12 min 55 s   | TP (%)              | Satellite1 | 92.45          | 93.98         |
|                  |               | F1 (%)              | Satellite2 | 95.14          | 95.05         |
|                  |               | PPV (%)             | Satellite3 | 90.23          | 90.74         |
|                  |               |                     | Satellite4 | 94.55          | 93.96         |
|                  |               |                     | Satellite5 | 91.13          | 94.51         |
| Alexnet-fc6,7,8  | 15 min 23 s   | TP (%)              | Satellite1 | 96.23          | 93.44         |
|                  |               | F1 (%)              | Satellite2 | 94.45          | 93.43         |
|                  |               | PPV (%)             | Satellite3 | 72.73          | 76.24         |
|                  |               |                     | Satellite4 | 80.00          | 76.22         |
| Inception-V3-a0  | 74 min 20 s   | TP (%)              | Satellite1 | 60.34          | 76.01         |
|                  |               | F1 (%)              | Satellite2 | 65.96          | 76.49         |
|                  |               | PPV (%)             | Satellite3 | 70.45          | 76.47         |
|                  |               |                     | Satellite4 | 70.18          | 76.01         |
| Inception-V3-a1  | 75 min 26 s   | TP (%)              | Satellite1 | 58.49          | 70.15         |
|                  |               | F1 (%)              | Satellite2 | 63.92          | 69.43         |
|                  |               | PPV (%)             | Satellite3 | 82.61          | 69.43         |
|                  |               |                     | Satellite4 | 70.97          | 69.43         |
| Resnet50-a0      | 51 min 33 s   | TP (%)              | Satellite1 | 71.70          | 80.00         |
|                  |               | F1 (%)              | Satellite2 | 76.77          | 79.91         |

FIGURE 16. Test results of Resnet50 after fine-tuning training.

in Table 4. For any base learner, use one sub-dataset for testing and the other two for training. During training, the data used for training is randomly divided into a training dataset and verification dataset according to the proportion of 9:1, and the training dataset is augmented. The grouping results of the training data are shown in Table 5.

2) BASE LEARNER TRAINING AND META-FEATURE GENERATION

In this section, we extract the meta-feature of the space target ISAR images according to Algorithm 1. When extracting meta-features, we first train the base learners. Refer to Table 5 for the use of data during training. The learning rate, batch_size and the number of epochs are all consistent with Table 2. Therefore, when using DataSet2+DataSet3 and DataSet1+DataSet3 for training, the number of training iterations is 990. When using DataSet1+DataSet2 for training,
After the training of the base learners, the meta-features of the training data are extracted. In order to show the spatial differentiability of the meta-features of different targets more intuitively, we use the t-SNE algorithm [38] to carry out two-dimensional visualization of the meta-features. Figure 18 shows the two-dimensional visualization results of the original training data and its meta-features.

By comparing Figure 18 (a) - (d), we can see that the meta-features of the training data extracted by the three kinds of base learners are more distinguishable than the distribution of the original training data in two-dimensional space. At the same time, by comparing figure 18 (b) - (e), we can see that after splicing the meta-features extracted by the three kinds of base learners according to algorithm 1, the spliced meta-features are more distinguishable in two-dimensional space than the meta-features extracted by any kind of the base learners.

We also extract the meta-features of test data. We use the t-SNE algorithm to carry out two-dimensional visualization of the meta-features. Figure 19 shows the two-dimensional visualization results of the original test data and its meta-features.

By comparing Figure 19 (a) - (d), we can see that the meta-features of the test data extracted by the three kinds of base learners are more distinguishable than the distribution of the original test data in two-dimensional space. At the same time, by comparing figure 19 (b) - (e), we can see that after splicing the meta-features extracted by the three kinds of base learners according to algorithm 1, the spliced meta-features are more distinguishable in two-dimensional space than the meta-features extracted by any kind of the base learners.

3) XGBOOST CLASSIFICATION EXPERIMENT

After extracting the meta-features of ISAR images, we continue to use XGBoost as the meta-learner in the second layer of SELF and train the meta-learner with the extracted meta-features. XGBoost adopts the tree model for iteration. We set the tree depth to 5, the number of trees to 1000, and the learning rate to 0.01. When the sum of the sample weights of leaf nodes is less than 1, the segmentation process is ended. The random sample sampling ratio to generate each tree is 0.8.

Under the condition of the above super-parameter settings, we use the extracted training data meta-features to train XGBoost and use the trained XGBoost model to classify the test data. The test confusion matrix is shown in Figure 20. The final classification results are given in Table 6.

Considering that the clarification algorithm based on the stacking method will be affected by the classification performance of the base learners, to further improve the

| Performance Metrics | Target | Satellite1 | Satellite2 | Satellite3 | Satellite4 | Satellite5 | Macro Average(%) | Total Acc (%) |
|---------------------|--------|------------|------------|------------|------------|------------|------------------|---------------|
| PPV (%)             |        | 94.00      | 92.59      | 86.44      | 100        | 94.34      | 93.47            | 93.58         |
| TPR (%)             |        | 88.68      | 94.34      | 96.23      | 94.34      | 94.34      | 93.59            | 93.58         |
| F1 (%)              |        | 91.26      | 93.46      | 91.07      | 97.09      | 94.34      | 93.44            | 93.58         |
classification effect, we change the fine-tuning training way of the pre-trained Inception-V3 model, which has the weakest classification performance. As we fine-tune the pre-trained Inception-V3 model, we also fine-tune some of its convolutional layer weights. According to Figure 6 (b), the pre-trained Inception-V3 model consists of three Blocks. To improve the training effect and save the training time as much as possible, here we keep the weights of Blocks1 and Blocks2 frozen, and fine-tune the weights of Blocks3 and the last full connection layer at the same time. The learning-rate, batch_size, and epoch size remained unchanged when the pre-trained Inception-V3 model was retrained. Figure 21 shows the retraining process.

Comparing Figure 21 (a) - (c) with Figure 17 (d) - (f), we can see that after changing the fine-tuning training way of the pre-trained Inception-V3 model, the weights needed to be fine-tuned in the training process increases, and the training time increases by 25 min 51 s, 26 min 4 s and 34 min 6 s, respectively. But at the same time, the training accuracy increased by 19.32%, 22.16%, and 22.21% respectively, the verification accuracy increased by 21.43%, 21.43%, and 28.57% respectively, the training cross-entropy loss decreased by 0.60, 0.62, and 0.61 respectively, and the verification cross-entropy loss decreased by 0.29, 0.33, and 0.49 respectively.

To show the spatial distribution characteristics of the meta-features extracted by the newly transferred Inception-V3 model more intuitively, we still use the t-SNE algorithm for two-dimensional visualization of the meta-features.
Comparing Figure 22 (a) with Figure 18 (c) and Figure 22 (b) with Figure 19 (c), we can see that the meta-features of different targets extracted by the new transferred Inception-V3 model are significantly more distinguishable in two-dimensional space after the weights of Blocks3 is adjusted. We use this meta-feature to replace the part of the original spliced meta-feature extracted by the transferred Inception-V3 model. In Figure 23, we reshow the two-dimensional spatial distribution of the training data meta-feature and the test data meta-feature after re-splicing.

Comparing Figure 23 (a) with Figure 18 (e) and Figure 23 (b) with Figure 19 (e), we can see that the distinguishability of the re-spliced meta-features in the two-dimensional space has been enhanced to some extent. Keeping the super-parameters of XGBoost unchanged, we then train XGBoost by using the meta-features of the training data and then classify the test data by using the trained XGBoost model. The test confusion matrix is shown in Figure 24. The final classification results are given in Table 7.

Comparing the classification results in Table 7 and Table 6, we can find that the Total Acc of the satellites increased by 3.78%. At the same time, PPV, TPR, and F1 of the five types of satellites have also been significantly improved: For PPV, satellite 1, satellite 2 and satellite 3 increased by 6.00%, 1.85%, and 8.00% respectively, while satellite 4 remained 100% and satellite 5 increased by 3.74%; for TPR,
satellite 1 and satellite 2 increased by 9.43% and 1.89%, satellite 3 maintained at 96.23%, and satellite 4 and satellite 5 increased by 5.66% and 1.89%, respectively; and for evaluation index $F_1$, satellite 1 to satellite 5 increased by 7.79%, 1.91%, 4.3%, 2.91%, and 2.81% respectively. Meanwhile, Macro Average $PPV$, Macro Average $TPR$ and Macro
Average $F_1$ of the five types of satellites increased by 3.93%, 3.77%, and 3.95% respectively. The above results show that when the classification performance of the transferred Inception-V3 model is improved, the overall classification effect can be effectively improved.

4) COMPARATIVE ANALYSIS

To test the advantages of the methods proposed in this paper, in this section, we compare the classification effect of the stacking ensemble model with that of the FP-DCNNs obtained by “Alexnet-fc7,8”, “Inception-V3-a0” and “Resnet50-a0” fine-tuning ways respectively. In order to ensure the fairness of the comparison, we use the same training data, validation data, and test data as in Table 4 and Table 5 for transfer training and testing. Keep the learning rate and batch_size unchanged. The epoch is taken as 7, and the corresponding number of iterations is 1043. Figure 25 shows the transfer training process.

After the transfer training, in order to show the training effect of the above three models more intuitively, we show the feature map of the space target ISAR images learned in the last full connection layer of the transferred Alexnet, Inception-V3 and Resnet50 models in Figure 26 to Figure 28 respectively.

Finally, we use the above FP-DCNNs to classify the test data. Figure 29 (a) (b) (c) respectively show the test confusion matrix of the three FP-DCNNs. The final classification results are given in Table 8. From Figure 29 and Table 8, we can see that the classification effect of the stacking ensemble model on space target ISAR images is significantly better than that of three single FP-DCNNs. Firstly, compared...
FIGURE 23. Two-dimensional visualization of the meta-features after re-splicing. (a) Meta-features of the training data; (b) meta-features of the test data.

TABLE 8. Classification effect comparison between the stacking ensemble model and the three single FP-DCNNs.

| Classification Model          | Performance Metrics | Target  |          | Macro Average(%) | Total Acc (%) |
|------------------------------|---------------------|---------|---------|------------------|---------------|
|                              | PPV (%)             | Satellite1 | Satellite2 | Satellite3 | Satellite4 | Satellite5 |
| Stacking Ensemble Model      | 100                 | 94.44   | 94.44   | 100         | 98.08       | 97.39      |
|                              | TPR (%)             | 98.11   | 96.23   | 96.23      | 100         | 96.23      | 97.36      | 97.36    |
|                              | F1 (%)              | 99.05   | 95.37   | 95.37      | 100         | 97.15      | 97.39      |
|                              | PPV (%)             | 95.83   | 70.00   | 92.02      | 96.23       | 92.16      | 89.25      |
| Transferred Alexnet Model    | TPR (%)             | 86.79   | 92.45   | 75.47      | 96.23       | 88.68      | 87.92      | 87.92    |
|                              | F1 (%)              | 91.09   | 79.67   | 82.93      | 96.23       | 90.39      | 88.06      |
|                              | PPV (%)             | 73.21   | 74.00   | 69.81      | 86.21       | 85.42      | 77.73      |
| Transferred Inception-V3 Model | TPR (%)          | 77.36   | 69.81   | 69.81      | 94.34       | 77.36      | 77.74      | 77.74    |
|                              | F1 (%)              | 75.23   | 71.84   | 69.81      | 90.09       | 81.19      | 77.63      |
|                              | PPV (%)             | 89.36   | 75.56   | 65.79      | 96.15       | 93.33      | 84.04      |
| Transferred Resnet50 Model   | TPR (%)             | 79.25   | 64.15   | 94.34      | 94.34       | 79.25      | 82.27      | 82.26    |
|                              | F1 (%)              | 84.00   | 69.39   | 77.52      | 95.24       | 85.72      | 82.37      |

FIGURE 24. Test confusion matrix of the retrained XGBoost model.

with the transferred Alexnet model, the stacking ensemble model’s classification accuracy, Macro Average $PPV$, Macro Average $TPR$, and Macro Average $F_1$ increased by 19.62%, 19.66%, 19.62%, and 19.76%, respectively. Lastly, compared with the transferred Resnet50 model, the classification accuracy, Macro Average $PPV$, Macro Average $TPR$, and Macro Average $F_1$ increased by 15.10%, 13.35%, 15.09%, and 15.02%, respectively. In conclusion, the stacking ensemble model gives full play to the advantages of each FP-DCNNs and effectively improves the classification performance of any FP-DCNNs.

Besides, we also train a simple CNN model from scratch to classify the test data (including two convolutional layers, two regularization layers, two rectified linear unit layers, two maximum pooling layers, one full connection layer, one Softmax layer, and one classification layer). At the same time, we also use the pre-trained Alexnet, Inception-V3, and Resnet50 as feature extractors, and use them with the Softmax to classify the test set. Figure 30 (a) (b) (c) (d) shows the test confusion matrix of the four methods, respectively. The final classification results are given in Table 9.
FIGURE 25. The transfer training process of the three P-DCNNs. (a) The transfer training process of the pre-trained Alexnet model; (b) The transfer training process of the pre-trained Inception-V3 model; (c) The transfer training process of the pre-trained Resnet50 model.

FIGURE 26. Feature maps of ISAR images of five types of satellites learned in the last layer of the transferred Alexnet model. (a) Feature map of satellite 1; (b) feature map of satellite 2; (c) feature map of satellite 3; (d) feature map of satellite 4; (e) feature map of satellite 5.

FIGURE 27. Feature maps of ISAR images of five types of satellites learned in the last layer of the transferred Inception-V3 model. (a) Feature map of satellite 1; (b) feature map of satellite 2; (c) feature map of satellite 3; (d) feature map of satellite 4; (e) feature map of satellite 5.

TABLE 9. Classification effect comparison between the stacking ensemble model and the four methods.

| Classification Model          | Performance Metrics | Target   | Satellite1 | Satellite2 | Satellite3 | Satellite4 | Satellite5 | Macro Average(%) | Total Acc (%) |
|------------------------------|---------------------|----------|------------|------------|------------|------------|------------|------------------|---------------|
| Stacking Ensemble Model      | PPV (%)             |          | 100        | 94.44      | 94.44      | 100        | 98.08      | 97.39            |               |
|                             | TPR (%)             |          | 98.11      | 96.23      | 96.23      | 100        | 96.23      | 97.36            | 97.36         |
|                             | F1 (%)              |          | 99.05      | 95.37      | 95.37      | 100        | 97.15      | 97.39            |               |
|                             | PPV (%)             |          | 87.50      | 63.16      | 79.55      | 97.78      | 77.27      | 81.05            |               |
| Trained CNN Model            | TPR (%)             |          | 92.45      | 90.57      | 66.04      | 83.02      | 64.15      | 79.25            | 79.25         |
|                             | F1 (%)              |          | 89.91      | 74.42      | 72.17      | 89.80      | 70.10      | 79.28            |               |
|                             | PPV (%)             |          | 97.50      | 56.76      | 46.07      | 91.30      | 66.04      | 71.53            |               |
| Alexnet Model                | TPR (%)             |          | 73.58      | 39.62      | 77.36      | 79.25      | 66.04      | 67.17            | 67.17         |
|                             | F1 (%)              |          | 83.87      | 46.67      | 57.75      | 84.85      | 66.04      | 67.84            |               |
|                             | PPV (%)             |          | 43.24      | 40.81      | 37.50      | 43.16      | 47.73      | 42.49            |               |
| Inception-V3 Model           | TPR (%)             |          | 30.19      | 37.74      | 28.30      | 77.36      | 39.62      | 42.64            | 42.64         |
|                             | F1 (%)              |          | 35.56      | 39.22      | 32.26      | 55.41      | 43.30      | 41.15            |               |
|                             | PPV (%)             |          | 55.32      | 52.24      | 57.50      | 60.27      | 60.53      | 57.17            |               |
| Resnet50 Model               | TPR (%)             |          | 49.06      | 66.04      | 43.40      | 83.02      | 43.40      | 57.02            | 56.98         |
|                             | F1 (%)              |          | 52.00      | 58.33      | 49.46      | 69.84      | 50.55      | 56.04            |               |

From Figure 30 and Table 9, we can see that the classification effect of the stacking ensemble model on the space target ISAR images is significantly better than the other four methods. Compared with the simple CNN model trained from scratch, the classification accuracy, Macro Average PPV, Macro Average TPR, and Macro Average F1 increased by
18.11%, 16.34%, 18.11%, and 18.11%, respectively. The above results further prove the correctness and superiority of our method.

V. CONCLUSION
In this paper, a novel method that realizes the ensemble of multiple P-DCNNs with the stacking algorithm is proposed.
This method can automatically recognize the space targets in ISAR images with high precision under the condition of a small sample set. Firstly, the space target ISAR images after despckling and standardization are divided into training data and test data according to a certain proportion. The training dataset divided from the training data is augmented by contrast adjustment, small-angle rotation, azimuth scaling, and range scaling operations. Then, several heterogeneous SOTA models are fine-tuned by using the weight fine-tuning strategy to realize the end-to-end automatic extraction of meta-features of space target ISAR image data sets. Finally, XGBoost is adopted as the meta-learner in the second layer of SELF, and the extracted meta-features are used to train the meta-learner. Thus high-precision classification of the space target ISAR images is realized.

The simulation example shows that:
(1) Fast and automatic recognition of space target ISAR images in a small sample set can be realized by using the TL method. Choosing different P-DCNNs and different fine-tuning ways can make the recognition effect vary greatly.
(2) Multiple heterogeneous P-DCNNs can be integrated by stacking algorithm, and the classification effect of the stacking ensemble model is better than any single FP-DCNN.

REFERENCES

[1] H. Yang, Y. Zhang, and W. Ding, “A fast recognition method for space targets in ISAR images based on local and global structural fusion features with lower dimensions,” Int. J. Aerosp. Eng., vol. 2020, pp. 1–21, Feb. 2020, doi: 10.1155/2020/3412582.
[2] Z. Ji, H. Jin, Y. Zang, and E. Hu, “Space objects recognition based on ISAR image,” in Proc. CPS, Harbin, China, 2017, pp. 1215–1222.
[3] B. Xue and N. Tong, “DIOD: Fast and efficient weakly semi-supervised deep complex ISAR object detection,” IEEE Trans. Cybern., vol. 49, no. 11, pp. 3991–4003, Nov. 2019.
[4] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” Proc. IEEE, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017.
[6] J. C. Ni and Y. L. Xu, “SAR automatic target recognition based on a visual cortical system,” in Proc. 6th Int. Congr. Image Signal Process. (CISP), Hangzhou, China, Dec. 2013, pp. 778–782.
[7] S. Chen and H. Wang, “SAR target recognition based on deep learning,” in Proc. Int. Conf. Data Sci. Adv. Analytics (DSAA), Shanghai, China, Oct. 2014, pp. 541–547.
[8] W. Simon, “Combination of convolutional feature extraction and support vector machines for radar ATR,” in Proc. 17th Int. Conf. Inf. Fusion, Salamanca, Spain, Jul. 2014, pp. 1–6.
[9] H. Wang, S. Chen, F. Xu, and Y.-Q. Jin, “Application of deep-learning algorithms to MSTAR data,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Milan, Italy, Jul. 2015, pp. 3743–3745.
[10] S. Wagnier, “ Morphological component analysis in SAR images to improve the generalization of ATR systems,” in Proc. 3rd Int. Workshop Compressed Sens. Theory Appl. Radar. Sonar Remote Sens. (CoSeRa), Pisa, Italy, Jun. 2015, pp. 46–50.
[11] C. P. Schwegmann, W. Kleynhans, B. P. Salmon, L. W. Mdakane, and R. G. V. Meyer, “Very deep learning for ship discrimination in synthetic aperture radar imagery,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Beijing, China, Jul. 2016, pp. 104–107.
[12] J. H. Cho and C. G. Park, “Additional feature CNN based automatic target recognition in SAR image,” in Proc. 4th Asian Conf. Defence Technol.-Jpn. (ACDT), Tokyo, Japan, Nov. 2017, pp. 1–4.
[13] H. He, S. Wang, D. Yang, and S. Wang, “SAR target recognition and unsupervised detection based on convolutional neural network,” in Proc. Chin. Autom. Congr. (CAC), Jinan, China, Oct. 2017, pp. 435–438.
[35] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), San Francisco, CA, USA, 2016, pp. 785–794.

[36] X.-H. Chang, R. Huang, and J. H. Park, “Robust guaranteed cost control under digital communication channels,” IEEE Trans. Ind. Informat., vol. 16, no. 1, pp. 319–327, Jan. 2020.

[37] X. Chang and G. Yang, “Nonfragile H∞ filtering of continuous-time fuzzy systems,” IEEE Trans. Signal Process., vol. 59, no. 4, pp. 1528–1538, Apr. 2011.

[38] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, pp. 2579–2605, Nov. 2008.

HONG YANG was born in 1991. She received the B.Eng. degree from Harbin Engineering University (HEU), China, in 2014. She is currently pursuing the Ph.D. degree with Space Engineering University, Beijing, China. Her research interest is space target recognition based on ISAR images.

YASHENG ZHANG was born in 1974. She received the B.Eng. and M.Eng. degrees from the National University of Defense Technology (NUDT), China, in 1996 and 1999, respectively, and the Ph.D. degree from Space Engineering University, Beijing, China, in 2006.

She is currently a Professor with Space Engineering University. She has published more than 60 articles and more than ten monograph publications. Her research interests include spacecraft orbit dynamics, design of constellation, and space mission analysis. She received the Young Science and Technology Award of China, the National Science and Technology Progress Award, and the Space Fund Award of China.

WENZHE DING was born in 1992. He received the B.Eng. degree from Harbin Engineering University (HEU), China, in 2014. He is currently pursuing the Ph.D. degree with Space Engineering University, Beijing, China. His research interests include interception orbit design and machine learning.

HONG YANG

WENZHE DING

75570

VOLUME 8, 2020