Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

UAV swarm based radar signal sorting via multi-source data fusion: A deep transfer learning framework

Liangtian Wan [a], Rong Liu [a], Lu Sun [b,*], Hansong Nie [a], Xianpeng Wang [c]

[a] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province and School of Software, Dalian University of Technology, Dalian, 116620, China

[b] Department of Communication Engineering, Institute of Information Science Technology, Dalian Maritime University, Dalian 116026, China

[c] State Key Laboratory of Marine Resource Utilization in South China Sea, School of Information and Communication Engineering, Hainan University, Haikou 570228, China

ARTICLE INFO

Keywords:
Signal sorting
Deep transfer learning
Data fusion
UAV swarm

ABSTRACT

Traditional clustering algorithms can be applied for the pre-sorting step of radar signal sorting. It can effectively dilute the pulse stream and prevent the dense pulse stream from interfering pulse repetition interval (PRI) extraction. However, the pre-sorting deviation will cause interference and missing pulses during the main sorting process. To solve this problem, we deploy the unmanned aerial vehicle (UAV) swarm to monitor reconnaissance areas and put forward a novel deep transfer learning based signal sorting method. The UAV swarm can collect the pulses from different time and spatial domains, and interference and missing pulses in main sorting processing can be relieved dramatically. In our model, we pre-train our model with the data collected from multiple source areas, which corresponds to different areas detected by different parts of UAV swarms. Then we fine-tune our model with the data of the target area. The experimental results prove that the signal sorting accuracy of methods based on deep transfer learning, i.e., YOLO-MobileNet, F-RCNN and cascade RCNN, are higher than that of the baseline methods. In addition, the signal sorting accuracy of traditional methods based on deep learning can be greatly improved with the help of transfer learning.

1. Introduction

Modern information warfare is composed of various advanced techniques, and electronic warfare (EW) is the core of information warfare that determines the victory of warfare. EW contains three components: electronic assistance, electronic attack and electronic protection, which controls the electromagnetic spectrum and attacks enemies by using electromagnetic and directed weapons. Radar countermeasure plays an important role in electronic warfare, and reconnaissance receiver can be utilized to obtain the related parameters of the enemy's radars. The enemy's radar can be jammed, weakened and destroyed when the enemy's radar is detected correctly. Radar radiator signal sorting is an important part of reconnaissance receiver and it can separate the radar pulse signals from the densely overlapping signal pulse stream and extract relevant parameters.

Multi-source data fusion can integrate the incomplete information of the local environment collected from multiple channels and directions, so as to eliminate the redundant and contradictory information existing in the multiple sources of data. By exploiting multi-source data fusion, the description of the system environment becomes consistent and completed. Furthermore, the speed and accuracy of decision-making, and reflection of intelligent systems can be improved. The decision-making risks can be reduced as well. In general, the data of radar signal sorting is collected by one device or in specific position. The collected data may be insufficient for signal sorting with high accuracy. With the collected information from other areas, multi-source data fusion improves the signal sorting accuracy in the target area. However, the data fusion method needs to be designed in detail to achieve high-accuracy signal sorting.

In the early years, radar radiator signal sorting uses the pulse repetition interval (PRI) of the radar signal for sorting. The single parameter sorting algorithms include the associated expansion algorithm, histogram algorithm, plane transformation algorithm and PRI transformation algorithm. The histogram algorithm is divided into two categories: cumulative difference histogram algorithm and sequence difference histogram algorithm. In the scenario of a small number of radiation sources and a simple radar system, the single parameter

[*] Corresponding author.
E-mail addresses: wanliangtian@dlut.edu.cn (L. Wan), Rong_lau@outlook.com (R. Liu), sunlu@dlmu.edu.cn (L. Sun), hansong.nie@outlook.com (H. Nie), wxpeng1986@126.com (X. Wang).

https://doi.org/10.1016/j.inffus.2021.09.007
Received 31 December 2020; Received in revised form 7 July 2021; Accepted 15 September 2021
Available online 23 September 2021

sorting method is effective. With the increase of the pulse stream density and radiation sources, the multiple parameter sorting method has been developed to use other radar pulse parameters. The pulse width, operating frequency and signal type can be used for pre-sorting, and then PRI can be used for main sorting of radar signal. The theoretical research and hardware implementation of the single parameter sorting algorithm are relatively mature. However, it is invalided for varying PRI. In the environment with plenty of radiation sources and high pulse stream density, the real-time performance of signal sorting is reduced. Nowadays, there are many radiation source systems and complex waveforms in the modern battlefield environment, and the operating frequency band continues to widen. Different radiation sources overlap in an increasingly wide range of operating frequency bands. Single parameter radar signal sorting method has gradually become obsolete.

Since the radar system becomes complex and the pulse parameter diversification increases, new signal classification methods should be proposed to deal with these problems. The simplest idea is to use all available radar parameters for signal sorting, and thus multi-parameter signal classification algorithm have been proposed. The cluster sorting method is one of the most widely used signal sorting methods for multi-parameter sorting. The idea of multi-parameter sorting is that the pulse parameters of signals transmitted by the identical radiation source are highly similar, while the pulse parameters of different radar radiation sources are different from each other. Clustering algorithms, which includes fuzzy clustering, K-means clustering, support vector clustering, etc, group similar signals together. These clustering based sorting algorithms have their own advantages and disadvantages. The average clustering algorithm has excellent convergence performance. But it needs to determine the number of cluster centers in advance. Fuzzy clustering method has high real-time performance. However, it cannot sort radar signals with parameter jumping. The support vector clustering algorithm has good classification performance for linear inseparable signals. However, its computational complexity is large and the real-time performance is relatively low. Thus efficient multi-parameter signal sorting algorithm needs to be designed in detail to satisfy the requirements of high accuracy signal sorting.

Nowadays, deep learning techniques have been used in radar signal sorting and the sorting performance of deep learning based methods are better than that of other traditional methods. The general idea of deep learning based methods is to transform radar signal parameters to images, and then these methods can be applied for radar signal sorting. In this paper, we deploy the UAV swarms to reconnoiter the unknown scenarios. Different groups of UAVs are in charge of different areas. When the data collected by UAVs is not sufficient, the deep learning based methods cannot be trained well. Thus deep transfer learning, as a promising multi-source data fusion technology, can be utilized for radar signal sorting. The data collected by UAVs in other source areas can be used to pre-train the deep learning model, and then the data collected by UAVs in target area is utilized to fine-tune our model. The signal sorting accuracy of the target area can be greatly improved by deep transfer learning. The contributions of our paper are given as follows.

(1) The reconnaissance architecture constructed by UAV swarms are deployed for target detection by receiving the signals transmitted by enemy radiation sources.
(2) Deep transfer learning is used to improve the sorting accuracy by transferring the information collected by source areas to target area.
(3) Experiment results verify that the proposed deep transfer learning architecture preforms better than other traditional sorting methods. We also discuss and analyze the best training models of typical deep transfer learning methods for signal sorting based on UAV swarm.

2. Related work

2.1. Big data fusion

Nowadays, data storage and collection technologies have been developed rapidly, and thus big data is a promising technique to mine information in science and engineering areas. However, the characteristics of big data, such as multi-source, heterogeneous and large volume, bring many challenges to its application.

Wu et al. [1] have proposed a HACE theorem which characterizes the revolution of big data, and data fusion can deal with sparse, heterogeneous, uncertain, incomplete and multi-source data. Zheng [2] has summarized the data fusion methods, which contains stage, feature level and semantic meaning based methods. The author provides examples of data fusion and discusses the relationships and differences among various methods.

Sorber et al. [3] have proposed the structured data fusion (SDF) framework, which can be used for rapidly prototyping knowledge discovery in one or more possibly incomplete datasets. Tensor is a powerful tool for data fusion, and then all datasets can be factorized with tensor decomposition by SDF. Papalexakis et al. [4] have discussed the techniques of tensor decomposition in the field of data fusion. The authors also summarize the existing systems and discuss some future directions.

Machine-to-Machine communication (M2M) and Internet of Things (IoT) can produce a large number of heterogeneous, multi-source, dynamic and sparse data. It is necessary to improve the computational efficiency used to fuse and analyze large amounts of data. Ahmad et al. [5] have proposed an efficient and multidimensional big data analysis fusion model based on five-level architecture. It can analyze real-time and offline data and provide basis to make decisions. Ding et al. [6] have summarized the data fusion technologies in IoT and have proposed the requirements of data fusion. By using these requirements, the advantages and disadvantages of these technologies have been analyzed as well.

The fourth industrial revolution uses data fusion strategies and machine learning to predict abnormal behaviors and security issues in industrial processes. Diez-Olivan et al. [7] have completed a comprehensive review on the latest development of data fusion and machine learning in the field of industrial forecasting. This survey focuses on identifying research trends, opportunities and undeveloped challenges.

The combination of data fusion and deep learning (DL) technology has an important position in solving practical problems. Hossain et al. [8] have put forward a deep learning based emotion recognition system. This system uses two CNNs to process video data and audio data respectively, and uses two extreme learning machines (ELMs) fusion to feed support vector machine (SVM) to classify the emotion. Liu et al. [9] have summarized the deep learning based urban big data fusion methods, and have divided them into three classes: DL-output-based fusion, DL-input-based fusion and DL-double-stage-based fusion. The authors also discuss the challenges while handling urban big data.

2.2. Transfer learning

Transfer learning plays an important role in avoiding expensive data-labeling efforts and improving learning effect, which can be utilized for data fusion.

Pan et al. [10] have summarized the development trend of transfer learning and have divided it into three categories: transductive transfer learning, inductive transfer learning and unsupervised transfer learning. The authors have discussed potential problems and the relationships between transfer learning and machine learning technologies as well. Ding et al. [11] have put forward an incomplete multisource transfer learning by cross-domain transfer and cross-source transfer. The incomplete multiple sources can be used to transfer knowledge effectively to promote the learning task in the target domain. Long et al. [12] have proposed the joint adaptation networks (JAN). Based on the joint maximum mean discrepancy (JMMD) criterion, JAN learns the transfer network by adjusting the joint distributions of multiple domain-specific layers. Cao et al. [13] have realized that the label space is generally assumed to be completely shared across domains. The partial transfer learning can be utilized for a new selective adversarial network, which relaxes the assumption of shared label space and considers that the target label space is a subspace of the source label space. Ying et al. [14] have proposed a novel framework for transfer learning called Learning to Transfer (L2T). It can make full use of the previous transfer learning experience and automatically determine the best transfer learning strategy.

Transfer learning has played an important role in applications. Yuan et al. [15] have realized that the existing methods in the field of hyperspectral image superresolution need to observe the same scene of low-resolution hyperspectral image repeatedly. To solve this problem, a new framework has been put forward for improving the hyperspectral image resolution by using natural image based on transfer learning. Gong et al. [16] have found that human parsing models are hardly adapted to other human parsing tasks without a large number of retraining. To this end, the authors have put forward a novel universal human parsing agent called Graphonomy, which can predict all labels in a system without increasing complexity. Transfer learning can also be used to realize and accelerate the training of deep neural network, so as to achieve high accuracy machine fault diagnosis [17]. Chen et al. [18] have used transfer learning to solve the conflict between the lack of the deep neural network data and the lack of labeled data in RGB-d salient object detection. Due to the problem of over fitting, meta learning under the few-shot learning setting can only use shallow neural network, which greatly limits the performance of meta learning. To this end, Sun et al. [19] have put forward meta-transfer learning (MTL), which uses transfer learning to adapt deep neural networks to a small number of shot learning tasks. In [20], the ImageNet is used to pre-train the deep learning model, and then the data received by the receiver is used to fine-tune the model. This method is different from our proposed method. The data for pre-training are ImageNet and the data collected by source domain for the reference method and the proposed method, respectively. In [21], the deep transfer learning and linear weight decision fusion has been utilized to recognize the unknown radar waveform. However, this method is complex compared with the pure deep transferring learning method.

2.3. Signal sorting

In the present electronic warfare, the high precision sorting of radar emitter signals is a key technology, which can sort the pulses of emitters from the crowded radar pulse stream. The traditional methods are based on five parameters: pulse repeat interval (PRI), radio frequency (RF), angle of arrival (AOA), pulse width (PW) and pulse amplitude (PA). The latest researches can be divided into two categories: the method using intelligent sorting and the method using new sorting parameters [22].

The accuracy of the traditional signal sorting methods using five parameters is relatively low. To solve this problem, Guo et al. [23] have proposed G features, which is a new kind of periodic change feature hidden in the environment of radar pulse sequence. Experimental results show that the G features can distinguish the periodically changing pulse sequence signal sorting from the complex and variant radar pulse signal stream. Xue et al. [22] have proposed a coherent characteristics of coherent radar based on feature-extraction method and have proved the feasibility of signal sorting with coherent characteristics as feature parameters.

Due to the low sorting rate and sensitivity to signal-to-noise ratio (SNR) of the commonly used signal sorting methods, Chen et al. [24] have put forward a novel sorting algorithm based on bispectrum, which improves the stability of the algorithm in low SNR scenarios. Sheng et al. [25] have proposed an extract pulse clustering algorithm based on SVC, which greatly reduces the running time of CPU without losing accuracy.

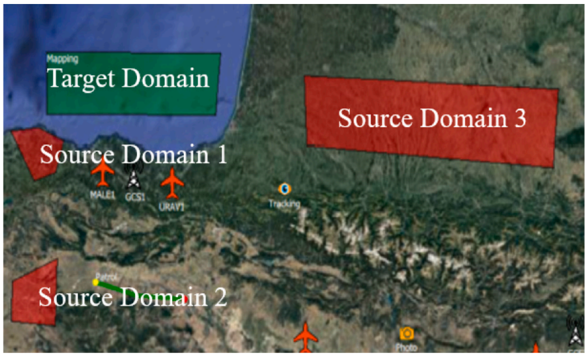

Fig. 1. Signal sorting based on UAV swarm.

Clustering and random forest are commonly used algorithms in researches of intelligent sorting algorithms. Wang et al. [26] have discovered that the time complexity of traditional support vector clustering (SVC) based on radar signal sorting method is high, and the disadvantage of traditional validity index is that it cannot effectively indicate the best sort. For this reason, a novel cone cluster labeling (CCL) based on sorting method and a new index (similitude entropy, SE) have been proposed. The random selection of the initial clustering center makes it more likely to sink into local optimal clustering, which is not applicable to non-spherical data. To solve this problem, the k-means clustering radar signal sorting algorithm is proposed [27], and the initial cluster center and cluster number can be achieved automatically. Cao et al. [28] have proposed the density-based fuzzy C-means multi-center re-clustering (DFCMRC) radar signal algorithm. It overcomes the disadvantage of the non-spherical sample set clustering. In the random forest based signal sorting algorithm, how to select a part of random forest to form a better ensemble classifier with lower complexity is a key problem. Zhang et al. [29] have proposed a modified teaching–learning-based optimization (TCTLBO) algorithm to search decision trees in random forests. With the development of transfer learning, it can also be used in signal sorting.

3. System architecture

Unmanned Aerial Vehicle Swarm has the characteristics of low stand-alone cost, large number of UAVs and high swarm intelligence, which brings unlimited possibilities for future combat styles. In this paper, the UAV swarm do not transmit electromagnetic wave to detect the enemies' targets. It means that the UAV swarm is difficult to detect by the enemies' radars and other detection devices. Compared with single UAV, UAV swarm can detect the enemies' targets in a large area. Although several UAVs are destroyed by the enemy, other UAVs can finish the mission as well. In addition, the data collected from time–frequency-spatial domains by the distributed UAV swarm is more abundant compared with single UAV, which means that the UAV swarm outperform single UAV for signal sorting with abundant data.

In this paper, UAV swarm has been deployed to detect the unknown areas, and each UAV is equipped with a passive receiver. As shown in Fig. 1, four areas are needed to be detected by UAV swarms. The data collected in the source domain is enough, but the data collected in the target domain is rare. The data collected in the source domain can be used for improving the signal sorting accuracy of the target domain, which is known as transfer learning. We solve the signal sorting problem based on transfer learning method in this paper.

The data collected by distributed UAV swarm should be delivered to a fusion center, and then the data can be used for signal sorting for different areas. For example, the data collected by source domain 1 can be utilized to pre-train the deep learning model, and then the data collected by target domain can be utilized to fine-tuning the model. The data can be delivered to the fusion center from UAVs by wireless link such as 5G link, WiFi, etc. The data can be delivered to the fusion center from UAVs by the relay stations which are performed by some UAVs as well.

4. Proposed deep transfer learning based method

In this paper, UAV swarm radar pulse is represented by two parameters: RF (radio frequency, unit:GHz) and PW (pulse width, unit:us). Classifying or recognizing a combination of PW and PF is challenging. We transform the task of processing point set sequences into the task of objectives detection on images. The field of computer vision develops rapidly. Different objective detection methods based on deep convolution neural network emerge in an endless stream. Due to the relevance and independently identically distribution between source data and target data, transfer learning in deep neural networks becomes possible. The following parts will introduce two types of objective detection models briefly and illustrate our deep transfer learning method for signal sorting.

4.1. Existing method

One-Stage objective detection model: YOLO [30] is the acronym of “You Only Look Once", which is an object detection algorithm based on a convolutional neural network. A series of methods of RCNN need to generate potential bounding boxes using region proposals methods, and then run classifiers on those bounding boxes. After classification, we need to fine tuning the bounding box, eliminate duplicate bounding boxes, and reposition the bounding box according to other targets. These processes are complex, slow and difficult to optimize. However, YOLO regards object detection task as a single regression problem from image pixels to boundary box coordinates.

YOLO is an end-to-end object detection model, which is developed on the basis of RCNN. The main step of YOLO algorithm is given as follows. First, the input features are extracted through the feature extraction network to obtain the feature graph of a specific size. The image is divided into $S \times S$ grid cells. If the central coordinate of an object in the real box falls in a grid cell, the grid cell can predict the object. The backbone network of YOLO v1 refers the network structure of GoogLeNet22. Compared with traditional region convolutional neural network (RCNN), the detection speed of YOLO v1 is very fast for detecting images. However, it has obvious disadvantages as well. For example, the position accuracy is low, and it cannot detect relatively small objects. The calculation of basic network is relatively large as well. In view of the shortcomings of YOLO v1, YOLO v2 has made specific improvements. First, the backbone network is replaced with the network structure of DartNet-19, which accelerates the calculation speed of the basic network. Then the batch normalization upgrading model is used for convergence, which achieves certain regularization effect and reduces the over-fitting of the model. The accuracy of YOLOv2 has been significantly improved. The low detection accuracy of small target is its disadvantage. The backbone network of YOLOv3 is replaced with Dartknet-53, and it uses the concept of feature pyramid networks and imports multi-scale detection. YOLOv3 is suitable for small targets.

Two-Stage objective detection model: Faster RCNN [31], has shown strong vitality in the field of object detection. Faster RCNN is the basis of many object detection algorithms. RCNN (Regions with CNN Features) uses the excellent feature extraction ability of convolutional neural network to greatly improve the effect of target detection. Fast-RCNN and Faster-RCNN further optimize the object detection problem based on RCNN. They greatly improve the implementation mode, speed and accuracy. When the traditional object detection reaches the bottleneck period, RCNN appears. RCNN can be seemed as the beginning of object detection by using deep learning. However, RCNN also has many disadvantages. Because the region proposals on an image can have a large number of overlaps, there would be a large amount of redundancy when these region proposals are sent into the neural network for computing CNN features, resulting in a waste of resources. Moreover, the extraction of region proposals, features and classification regression are separated in RCNN, so the space requirement of training is very large. Fast-RCNN has made improvements to address these issues. First, it sends the whole image into the neural network after normalization, and then adds the region proposals information in the last layer. Only one feature computing is needed for the whole image. The region proposals computed by selective search is mapped to the feature map, and then the proposal feature maps can be obtained. This process avoids unnecessary double-counting. The loss function uses multi-task loss function to add boundary regression to the CNN network for training, which can save the training time. Faster-RCNN has made many improvements aimed at how to generate region proposals efficiently, and it uses the RPN instead of selective search method to generate region proposals. The network of generating region proposals shares convolution with the network of object detection, and this process makes the number of region proposals reduce from the original about 2000 to 300. The quality of the region proposals has obvious improvement as well.

Faster-RCNN contains four parts. First is convolutional layers. As an object detection method based on CNN network, it uses a set of basic multi-layer neural network to compute the feature map of the image. This feature map can be shared for subsequent RPN layers and the full connection layers. Second is region proposal network, which is used to generate region proposals. It utilizes the softmax function as an activation function to determine whether the anchor values are positive or negative. The bounding box regression is then used to modify anchor values to obtain the accurate proposals. Third is ROI pooling layer. This layer integrates the feature maps and proposals, calculates the proposal feature maps and sends them to the subsequent full connection layer to determine the categories of the proposal. Finally, the proposed category is calculated by using the proposed feature map, and the bounding box regression is carried out again to obtain the final accurate position of the region proposal.

4.2. Transfer learning method

Because the above deep learning models are end-to-end models, all hidden layers of these models are neural networks. Nowadays, almost all new methods are proposed by adding some different tricks or using some combinations of them. We abstract the mentioned objective detection models as shown in Fig. 2.

In the convolution neural network models, the closer to the bottom of them, the more abstract and universal the features are, vice versa. The top fully-connected layer classifier is usually trained to solve specific problems, which means that transfer learning of top fully-connected layers only works when using the same dataset. But the convolution layers at the bottom have many common features, which can be reused by other models using transfer learning method. If the convolution neural network is trained from scratch, its initial weights are random, and it needs a lot of time to converge. Moreover, the optimal solution cannot be reachable because of too many local minima in multidimensional space. Researchers often use transfer learning to train models. Because the features of the bottom convolution layers (such as color, contour, etc.) are more generic, we can use some excellent pre-trained weights (such as the model trained by Imagenet, VOC, COCO, etc.) to speed up model training and improve model accuracy. Although the target data and the source data observed by naked eyes are very different, their abstract local features maybe have similarities through high dimensional mapping. Therefore, transfer learning usually performs well. If the source data of our transfer learning is the radar signals, which have high relevance between source data and target data, the effect of transfer learning will be better.

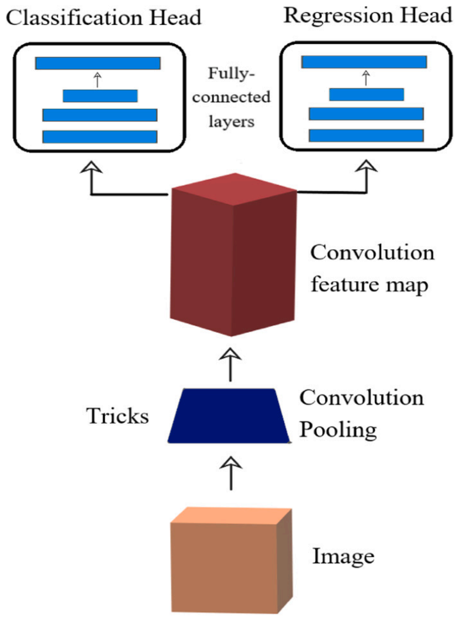

Fig. 2. Objective detection abstraction model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We choose the image to classification head fork to illustrate our transfer learning based method for signal sorting. The image to regression head fork has four outputs and is similar with former fork. Computer can transform an image as a three dimension array as follows

$$Image \Leftrightarrow (width, height, deep), \tag{1}$$

where $(width, height, deep)$ is the shape of the array, as shown in Eq. (1). The deep of color images and grayscale image is 3 and 1, respectively. In this way, computer can recognize an image by the float numbers in the three dimension array.

Based on local features, the raw image can be mapped into K convolution feature maps by convolution operation as follows

$$\begin{aligned} &Feature\ map \Leftrightarrow (width, height, features),\\ &width = \frac{width_{last} - F + 2P}{S} + 1,\\ &height = \frac{height_{last} - F + 2P}{S} + 1,\\ &features = K, \end{aligned} \tag{2}$$

where K is the number of convolution kernels, and (F, F) is the shape of each convolution kernel. Therefore, the total number of mapping weight parameters is $K * F * F$. The stride and padding size of each kernel are S and P, respectively. Besides, variables with subscript $last$ represent that they are the weights of last network layer or the variables of last iterative function. At this time, the shape of three dimension array is changed into $(width, height, features)$ by Eq. (2). The core of convolution operation is a local information extraction operation. Computer reconstructs the image by its K local features.

The operation after the convolution operation is pooling operation, which is given by

$$\begin{aligned} &Feature\ map(Pooling) \Leftrightarrow (width, height, features),\\ &width = \frac{width_{last} - F}{S} + 1,\\ &height = \frac{height_{last} - F}{S} + 1,\\ &features = features_{last}, \end{aligned} \tag{3}$$

where the size and stride of pooling kernel are defined as (F, F) and S, respectively. Pooling operation makes the feature maps smaller than the same number of features by Eq. (3). The aim of pooling operation is information compression.

As shown in Fig. 2, iterating convolution operation and pooling operation can construct the backbone of object detection models, which can convert the raw image into convolution feature maps, which are marked as blue area. The final convolution feature maps have small width and height, but lots of local features. The function principle of them is similar to human, which can recognize overall objectives through their local features. The final convolution feature is mapped into a n-dimension vector, which can be written as

$$\begin{aligned} &n = width * height * features,\\ &Feature\ map \Leftrightarrow (x_1, x_2, \dots, x_n)^T,\\ &X_n = (x_1, x_2, \dots, x_n)^T, \end{aligned} \tag{4}$$

where n is the total number of parameters, and we can use the vector $X_n = (x_1, x_2, \dots, x_n)^T$ to represent it.

Furthermore, if we have m target predict labels, the weight W_{mn} and the bias b_m to calculate the vector Y_m can be defined as

$$W_{mn} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \dots & \dots & & \dots \\ w_{m1} & w_{m2} & \dots & w_{mn} \end{pmatrix}, \tag{5}$$

$$b_m = (b_1, b_2, \dots, b_m)^T, \tag{6}$$

where weight W_{mn} and bias b_m are the parameters learned by computer itself like the weight of filter in convolution operation through stochastic gradient descent with momentum algorithm shown in Eq. (14)

$$Y_m = W_{mn} X_n + b_m = (y_1, y_2, \dots, y_m)^T, \tag{7}$$

where Y_m is the result of linear regression using W_{mn}, X_n and b_m we mentioned above.

At last, the relative probability of every target label i using the vector Y_m through $SoftMax$ function to predict the label of image can be calculated by

$$\begin{aligned} &p_i = SoftMax(Y_i) = \frac{e^{Y_i}}{\sum_j^m e^{Y_j}},\\ &P_m = (p_1, p_2, \dots, p_m)^T,\\ &Label_{predict} = Index(Max(P_m)), \end{aligned} \tag{8}$$

where p_i is the predicted relative probability of label i, and P_m is the vector composed of m probability p_i. The index of maximum relative probability value is the category index predicted by the model, which can convert to $Label_{predict}$.

In conclusion, the whole process is a series of transition as follows

$$Image \overset{W}{\Rightarrow} Feature\ map \overset{W}{\Rightarrow} X_n \overset{W}{\Rightarrow} Label_{predict}. \tag{9}$$

where each transition has its own weights W, and the distance $Loss$ always exists between the predicted label $Label_{predict}$ and the real label $Label_{real}$. We can abstract the whole mapping as follows

$$Loss(W) = Label_{predict} - Label_{real}. \tag{10}$$

In order to eliminating the effect of negative numbers and ensuring its differentiability, we modify Eq. (10) as follows

$$Loss(W) = (Label_{predict} - Label_{real})^2. \tag{11}$$

In order to prevent over-fitting, L2 regularization penalty is added after Eq. (11). L2 regularization penalty prefers small and discrete weight tensor, which encourages the filter to utilize the features of all dimensions instead of some specific dimensions.

$$Loss(W) = (Label_{predict} - Label_{real})^2 + \lambda \sum_m \sum_n W_{m,n}^2, \tag{12}$$

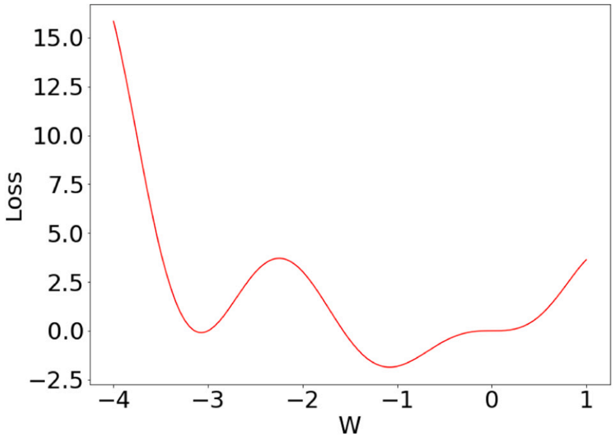

Fig. 3. $f(x) = (x^3 + 2x^2)sin(2x)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

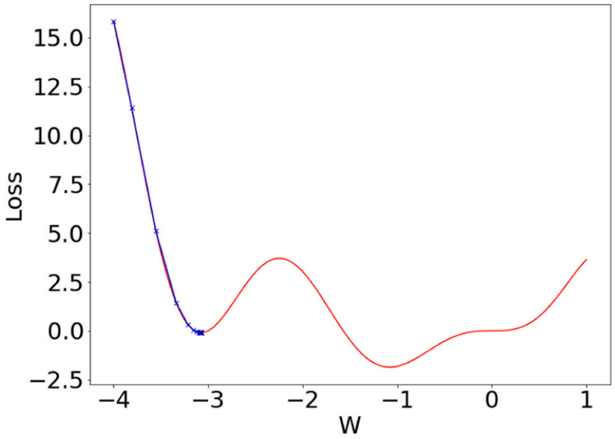

Fig. 4. Stochastic gradient descent.

where $\lambda \sum_m \sum_n W_{m,n}^2$ is the L2 regularization penalty, and λ is the coefficient of L2 regularization penalty, which is the hyperparameter that we need to select.

The accuracy of model increases as the value of $Loss$ function descends. Hence we can calculate a certain weight W with the minimum $Loss$ based on optimization method. Ensuring all processes are differentiable, we can calculate the gradient of $Loss$ function and make W move along the descending direction of gradient, which is called stochastic gradient descent (SGD) algorithm.

$$W = W_{last} - \eta \frac{\partial Loss}{\partial W_{last}}, \tag{13}$$

where $\frac{\partial Loss}{\partial W_{last}}$ is the gradient of $Loss$ function with respect to variable W, and its negative $-\frac{\partial Loss}{\partial W_{last}}$ is the gradient decent direction. η is the learning rate(another hyperparameter).

For example, we regard the function $f(x) = (x^3 + 2x^2)sin(2x), x \in (-4, 1)$ as an example, which is represented as the red line in Fig. 3, to represent the real differentiable multivariate function.

There are a local minimum and a global minimum as shown in Fig. 3. The best weight that we expect is the global minimum, instead of the local minimum. By calculating Eq. (13), we plot the trail of W as the blue line and use blue "×" to label the W value of each step. As shown in Fig. 4, it can be seen that W stays in the local minimum.

Inspired by the dynamic method, we hypothesize that W is a ball rolling on the surface. The ball W with its momentum is more easily to rush out of the local minimum cavity. Therefore, we modify Eq. (13) as Eq. (14), where V is the velocity of ball W and $momentum$ is the momentum ball W carried, and its effect is shown in Fig. 5. The ball W rushes out of the local minimum and the minimum goes astray. But with the momentum, it eventually comes back to the global minimum. This optimization algorithm will be used in our experiment for verifying the practicality of transfer learning.

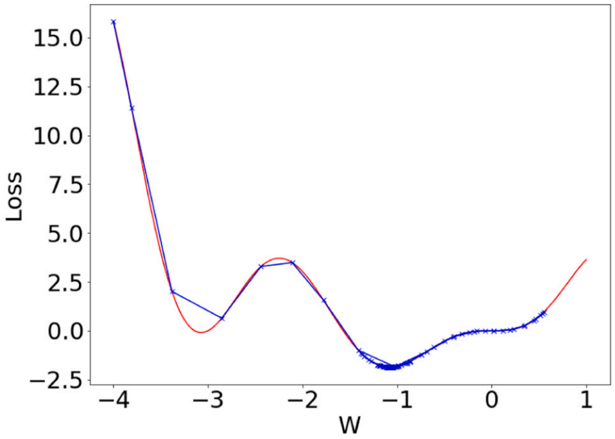

Fig. 5. Stochastic gradient descent with momentum.

$$V = momentum * V_{last} - \eta \frac{\partial Loss}{\partial W_{last}},$$
$$W = W_{last} + V. \tag{14}$$

If we train deep learning models from scratch, the weight parameters will be randomly initialized. Because the number of weights is too numerous, the random weights and the model-needed weights always differentiate greatly. The value of $Loss$ function is very high and it needs more training time to converge. Besides, the actual $Loss$ function is a multivariate function instead of bi-variate function. Its "topography" is more complex and the ball W is more inclined to stop in the local minimum. The transfer learning idea is that if using the same model structure, we can use pre-trained weights of other data to train, instead of random weights. It can improve the speed of convergence and accuracy, because the pre-trained weights is meaningful and represents the features of trained data. With the improving of the relevance between source data and target data, the similarity between initial weights and final weights is high (The distance between predict label and real label is small). Therefore, the probability of falling into local minimum cavity is smaller and the speed of convergence and accuracy are better.

5. Experiment

5.1. Dataset

5.1.1. Generation

We will introduce how to generate the dataset used in the experiment in this section. In our experiment, a radar pulse is represented by two parameters: RF (radio frequency, unit: GHz) and PW (pulse width, unit: us). We can use rectangular coordinate system to represent the pulses, where the horizontal axis represents PW and the vertical axis represents RF. Because a radar emitter can transmit many different pulses with different RFs and PWs, we suppose that pulse parameters are changed within a relatively fixed range. The shape of the area which is composed of the pulses generated by the same radar emitter is similar with the rectangle in the coordinate system. That is, one radar emitter is represented uniquely as one pulse area. Once given a batch of pulses, we can convert the pulse scatters into 256 × 256 gray-scale images, and these images construct our dataset.

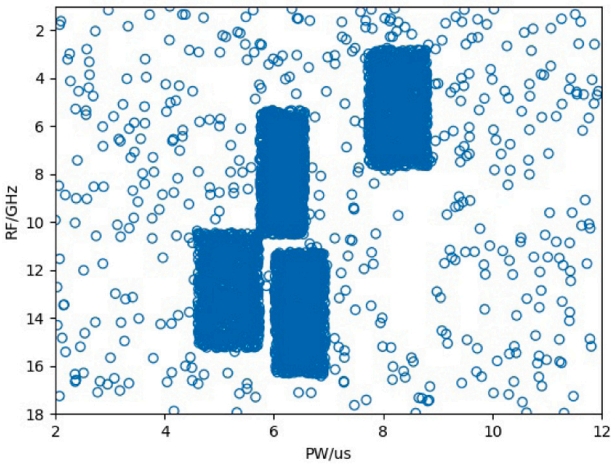

Fig. 6. Sample of the pulse scatter.

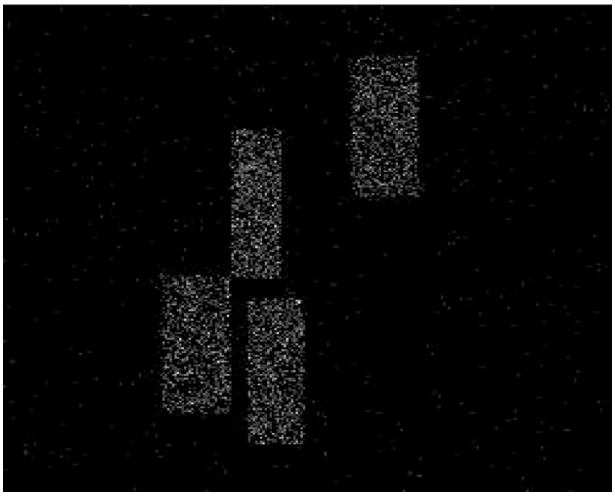

Fig. 7. Sample of the pulse scatter.

Suppose the scope of X-axis is from $2us$ to $12us$ (PW), and the scope of Y-axis is from 1 GHz to 18 GHz (RF), there are 4 different zones in the reconnaissance scenario. Ten radars exist in each areas. The height and width of the pulse area in coordinate system are denoted as rf and pw, respectively. The shape of pulse area is used to distinguish different radars, so we need to generate 40 parameter pairs $(pw_i, rf_i), i = 0, 1, \dots, 39$. In order to illustrate the shape of pulse area, the scale of this area is expressed as

$$scale_i = \frac{rf_i}{18-1} : \frac{pw_i}{12-2} \in \left[\frac{1}{4}, 1\right], \tag{15}$$

where $rf_i \in [3, 6], i = 0, 1, \dots, 39$. When $scale_i$ and rf_i are given, we can obtain pw_i according to the above equation. We sample 40 scale points from the mentioned range uniformly, sort them in ascending order and choose one rf for each scale to compute pw. After that, the scales are averagely split into 4 groups.

According to the parameter pair $(pw_i, rf_i), i = 0, 1, \dots, 39$, we generate some pulses in the area. 5000 gray-scale images are produced for each zone. To generate the gray-scale image, pulse scatter is needed. As mentioned above, each zone has 10 radar emitters. First, sampling with replacement is used from 6 to 10 times to obtain several radar emitters from the 10 radar parameters. Then we generate 5000 pulses for one pulse scatter where 10% pulses are noise. The scatter is produced as shown in Fig. 6.

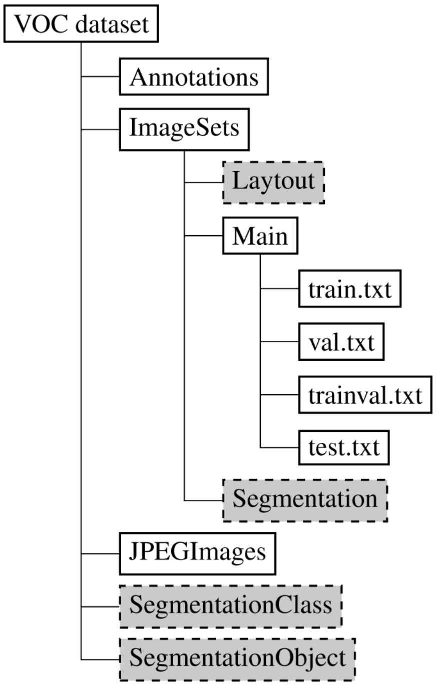

Fig. 8. VOC dataset file tree structure.

We divide the scatter into 256 × 256 grids and each grid is a pixel. The number of pulses in grids are counted to represent pixel values. In addition, we normalize the statistics by min–max normalization. The converted image is shown in Fig. 7.

5.1.2. PASCAL VOC format

This section will introduce how to transfer the data generated above to the PASCAL VOC format. It can be used to feed neural network to train or evaluate.

The Pascal Visual Object Classes (PASCAL VOC) is a project. It not only provides standardized image data sets for object recognition, but also provides tools for accessing datasets and annotations. In addition, the PASCAL VOC project is also facing the challenge of evaluating the performance of object class recognition (from 2005–2012, it has been completed), and has realized the evaluation and comparison of different methods. The PASCAL VOC faces two kinds of challenges:

(i) A publicly available image dataset, as well as ground truth annotation and standardized assessment software;
(ii) An annual competition and workshop. There are total five challenges: classification, detection, segmentation, action classification, and person layout [32].

The PASCAL VOC project challenge motivates people to propose novel methods to conquer it. Gradually, the format of dataset becomes more and more popular and dataset using VOC format becomes the input of many objective detection models.

VOC dataset file tree structure is shown in Fig. 8:

Our task is objective detection, so we ignore the directory (gray-masked) related to objective segmentation and human layout. JPEGImages directory stores the image files as jpg format. Annotations directory stores the label files as an XML format files with the same file name as the image name. ImageSets directory stores split files of dataset.

We first read the files in Annotations or JPEGImages directory as a list. Then we shuffle the files to ensure the randomness of data. Moreover, we divide the data into five parts: the first three parts are training sets, the fourth part is validation set, and the fifth part is test set. So far, we have completed the preparation of VOC data set.

5.2. Experimental environment

To verify our proposed transfer learning based method conveniently, we adopt paddle deep learning structure as backbend. Paddle deep learning structure provides some popular methods as YAML format, which consists of model structures, optimizer functions, loss functions and hyper parameters (such as learning rate, batch size, epoch, etc.) to choose and modify.

Because the training time and inferring time are the relative values based on hardware equipment, we make our computer hardware metrics public: GPU (Telsa V100), Video Memory (16 GB), CPU (4 Cores), RAM (32 GB).

5.3. Evaluate metrics

In the task of multiple label image classification, there are more than one labels in the task. Therefore, the evaluation cannot use the single label image classification metric, e.g., average accuracy. The mean average precision (mAP) should be used. To define the mAP, we should define some prerequisite concepts.

Intersection over union (IOU) is used to measure the overlap of two detection boxes (Box_p is the predicted box, and Box_{gt} is the ground truth box). The equation is shown as:

$$IOU = \frac{area(Box_p \bigcap Box_{gt})}{area(Box_p \bigcup Box_{gt})}. \tag{16}$$

In general, IOU threshold is set to be 0.5.

$$IOU_{threshold} = 0.5. \tag{17}$$

When comparing predicted boxes with ground truth boxes, $IOU > IOU_{threshold}$ means that it is a hit, otherwise it is a fail. For each class (for example class i), we can calculate four accuracy counts as follows

(i) True Positive (TP): a proposal is made for class i and there actually is an object of class i.
(ii) False Positive (FP): a proposal is made for class i, but there is no object of class i.
(iii) True Negative (TN): a proposal is made for other classes, but there actually is an object of class i.
(iv) False Negative (FN): a proposal is made for other classes and there actually is an object of other classes.

Then we can calculate two metrics to evaluate the model: precision and recall. The two equations are given as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{18}$$

$$Recall = \frac{TP}{TP + FN}. \tag{19}$$

Precision is the percentage of true positives in the retrieved results. Recall is the percentage of the class i that the model retrieves.

Definitely, we hope these two metrics to be high simultaneously, but they are a pair of contradictory metrics. It means that they cannot be high simultaneously, so we need to discover a balance of them. We take $Recall$ as X-axis while $Precision$ as Y-axis. Every evaluated class has its own Precision–Recall curve ($curve_{P-R}$), and the area under the $curve_{P-R}$ is Average Precision (AP). The equation is given as follows

$$AP = \int curve_{P-R}. \tag{20}$$

The mean AP for all classes is mAP, which is given as

$$mAP = \frac{1}{n}\sum_{i}^{n} AP_i. \tag{21}$$

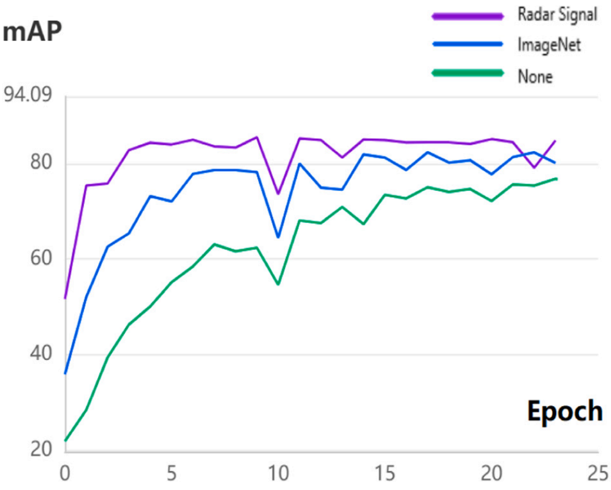

Fig. 9. $YOLO\ v3\ MobileNet\ v1$.

5.4. Experimental results

In the experiment, we select four different deep learning models for horizontal comparisons. Each model uses three different transfer learning tactics (using radar signal pre-trained weights, using ImageNet pre-trained weights and using none of pre-trained weights) for vertical comparison to verify the usability and practicality of transfer learning in signal sorting.

The four deep learning models contain two one-stage objective detection models (YOLO V3 [33] based on MobileNet V1 [34] and YOLO V3 based on MobileNet V3 [35]) and two two-stage objective detection models (Faster Rcnn [31] based on ResNet 50 [36] and Cascade Rcnn [37] based on ResNet 50 with FPN).

For fairness of comparison, four models have identical training strategy (using identical hyper parameters) as follows:

- $iterative\ epochs = 24$; $\eta = 0.001$ and $momentum = 0.9$ in Eq. (14); $\lambda = 0.0001$ in Eq. (12).
- image augmentation: the ratio between horizontally flip images and the total images is 50%; image rotation is not necessary because all objectives are rectangles.
- batch size: the batch size of one-stage model and two-stage model is 16 and 1, respectively. (the batch size is usually a multiple of 2, which satisfies the maximum video memory to reduce training time.)

5.4.1. Comparison between transfer learning tactics

In these two one-stage models, the mAP-Epoch figures of $YOLO\ v3\ MobileNet\ v1$ and $YOLO\ v3\ MobileNet\ v3$ training process are shown in Figs. 9 and 10, respectively.

Because the size of models is small and the number of parameters in the models is less as well. The models pretend to go astray. The mAP curve of the training process is more likely to fluctuate up and down, and more local minimums are needed to rush out.

In these two two-stage models, the mAP-Epoch figures of $Faster\ Rcnn\ r50$ and $Cascade\ Rcnn\ r50\ Fpn$ training process are shown in Figs. 11 and 12, respectively. Two-stage models have more parameters compared with one-stage models, and thus their mAP curves of training process are more stable.

All of these models have the same characteristics: convergence speed and accuracy (mAP) of the models using transfer learning method are higher than those of the model training without transfer learning, and both of them will increase as the relevance between pre-trained source data and target data increases. The details of accuracy (mAP) improvement are shown in Table 1. That is the reason why the

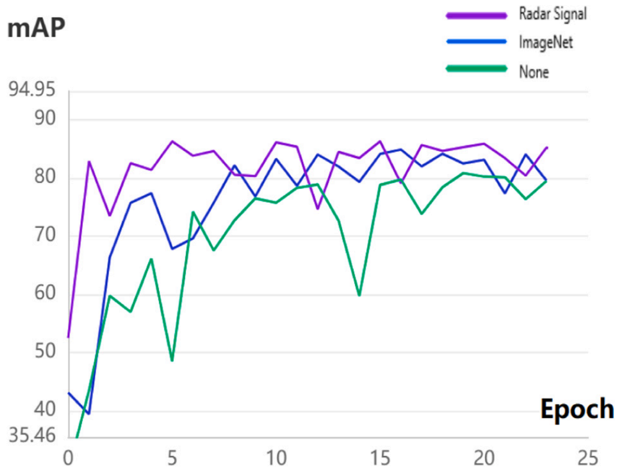

Fig. 10. *YOLO v3 MobileNet v3.*

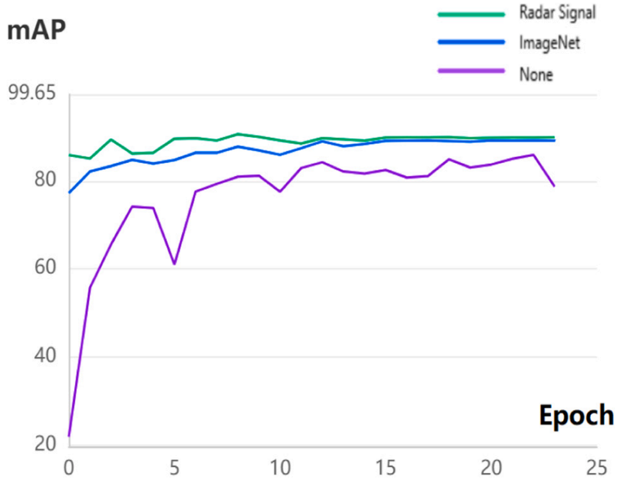

Fig. 11. *Faster Rcnn r50.*

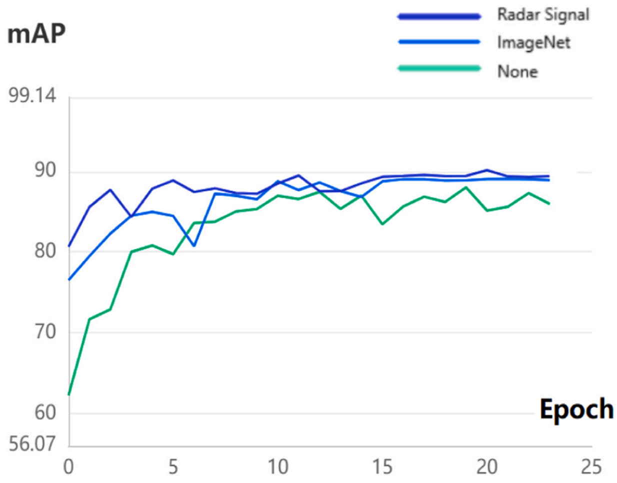

Fig. 12. *Cascade Rcnn r50 Fpn.*

performance of model using weights pre-trained by radar signal is better than that of the model using weights pre-trained by ImageNet and the model without using transfer learning.

Table 1
Transfer learning tactics comparison.

mAP	Transfer learning tactics		
Model	None	ImageNet	Radar Signal
yolov3_mobilenet_v1	76.831	82.373	**85.533**
yolov3_mobilenet_v3	80.835	84.891	**86.322**
faster_rcnn_r50_1x	85.898	89.209	**90.587**
cascade_rcnn_r50_fpn_1x	88	89.509	**90.125**

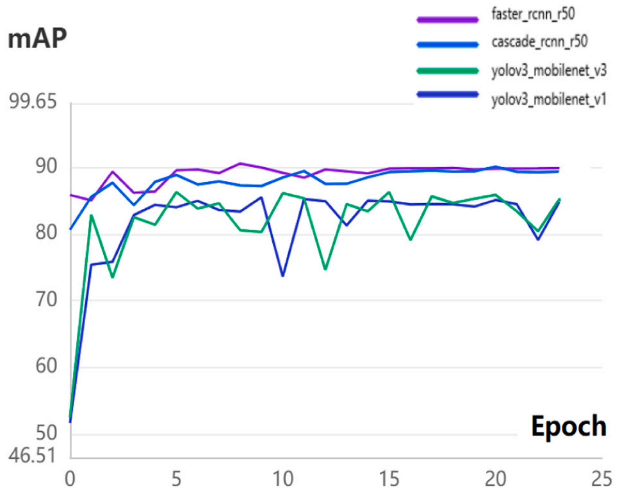

Fig. 13. The comparison of four optimal deep learning models.

5.4.2. Comparison between deep learning models

According to the figures in Section 5.4.1, the optimal model among the three transfer learning tactics is the model using radar signal pre-trained weights. Hence we compare four deep learning models with the radar signal pre-trained weights. The mAP-Epoch curves of four optimal deep learning models are shown in Fig. 13.

Overall, the convergence speed, accuracy and stability of two-stage models are all better than those of the one-stage models. It can be seen that one-stage models are less effective. However, since the figure above is independent of time, we need to combine with Table 2 below for a comprehensive consideration.

While $YOLO\ v3\ MobileNet\ v1$ has the lowest accuracy of all, the inference time of it is very short. However, sometimes such exaggerated inference speed is slow, because the frame rate of video is 30 fps to 75 fps generally. $YOLO\ v3\ MobileNet\ v3$ is the modification of $YOLO\ v3\ MobileNet\ v1$. It keeps inference time above 100 fps, improves the sorting accuracy and reduces the size of the model simultaneously.

$Faster\ Rcnn\ r50$ is based on ResNet convolution backbone, which is a deep network. It has a slow inference and training speed for exchange. However, it has the highest accuracy as well. Without FPN, $Faster\ Rcnn\ r50$ can use the last feature map for object classification and regression. Because the size of the last feature map is very small through a series of maxpooling functions, some features are lost. It is unsatisfied in detecting objects with small height or width. Similarly, $Cascade\ Rcnn\ r50\ FPN$ is the modification of $Faster\ Rcnn\ r50$. It has an FPN network for the combination of feature maps in different layers. Hence, it is adaptable for objects with small height or width. Meanwhile, it modifies the header networks of classification and regression, and sacrifices small model size for faster inference and training time. Nowadays, as hardware equipment develops rapidly, it is ordinary practice to exchange time with space to improve the real time in real applications.

The characteristics of the four models mentioned above have their own advantage and disadvantage. YOLO series models have small sizes

Table 2
Optimal model comparison.

	yolov3_mobilenet_v1	yolov3_mobilenet_v3	faster_rcnn_r50_1x	cascade_rcnn_r50_fpn_1x
Accuracy (mAP)	85.533	86.322	90.587	90.125
Training time (fps)	140.762	112.808	3.180	10.305
Inference time (fps)	212.724	117.355	5.165	24.886
Model size (M)	140	134	191	396

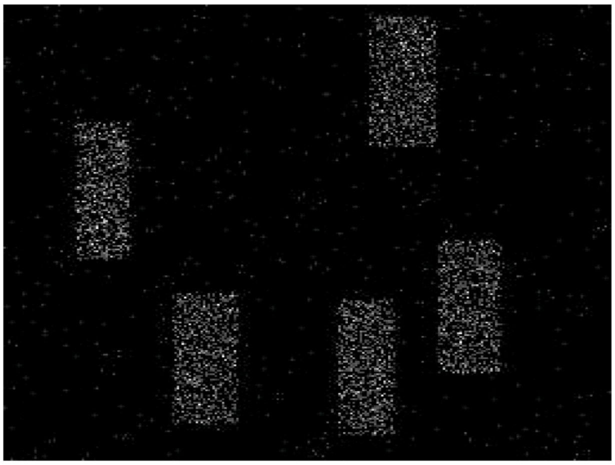

Fig. 14. Raw discrete objective detection image.

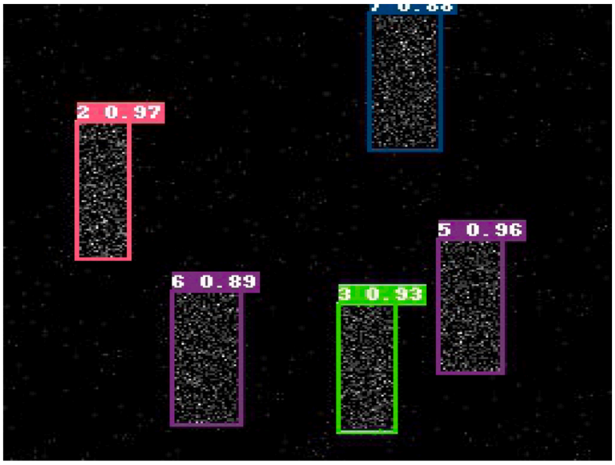

Fig. 15. Discrete objective detection by $YOLO\ v3\ MobileNet\ v1$.

and fast inference time. They are adaptable for real time objective detection tasks and mobile embedded applications. RCNN series models have higher sorting accuracy. They excel on the server models and tasks in the pursuit of precision without real time detection.

5.5. Application examples

We have already trained four different optimal deep transfer learning models. Now we apply them into real applications to illustrate their performance by randomly choosing two images: an image with discrete objectives as a simple objective detection task and an image with overlapping objectives as a complicated task.

In the following figures, the rectangular boxes are the objective positions, and text above the boxes is target labels of objectives and confidence (the scope of probability values is from 0 to 1) of target labels. All of them are predicted by the test model.

5.5.1. Discrete objective detection

The raw image shown as Fig. 14 is the input of the discrete objective detection task. The rectangle objectives are discrete. The output images predicted by four different models are Fig. 15, Fig. 16, Fig. 17 and Fig. 18 respectively.

In the task of discrete objective detection, all models correctly classify the target label and the position of each objective. But the confidence of two-stage models is better than that of one-stage models, which verifies the performance difference of different methods in Table 2.

5.5.2. Overlapping objective detection

The raw image shown in Fig. 19 is the input of overlapping objective detection task. The rectangle objectives are overlapping, and thus it is a difficult objective detection task. The output images predicted by four different models are Fig. 20, Fig. 21, Fig. 22 and Fig. 23, respectively.

In the task of overlapping objective detection, there are some objectives that people cannot distinguish easily. This is a good chance to discriminate the performance of different models. $YOLO\ v3\ MobileNet\ v1$ can classify the target label and the position of each objective with

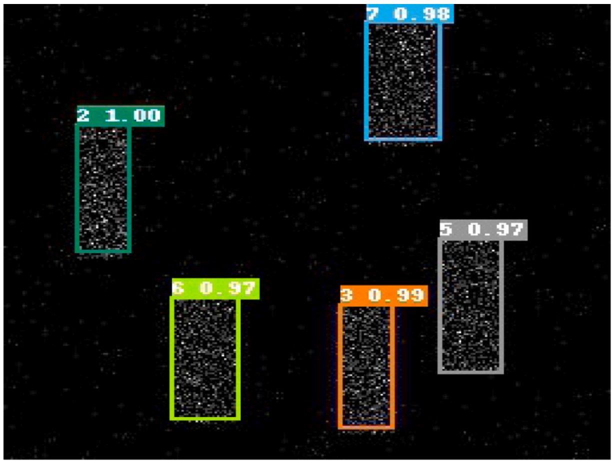

Fig. 16. Discrete objective detection by $YOLO\ v3\ MobileNet\ v3$.

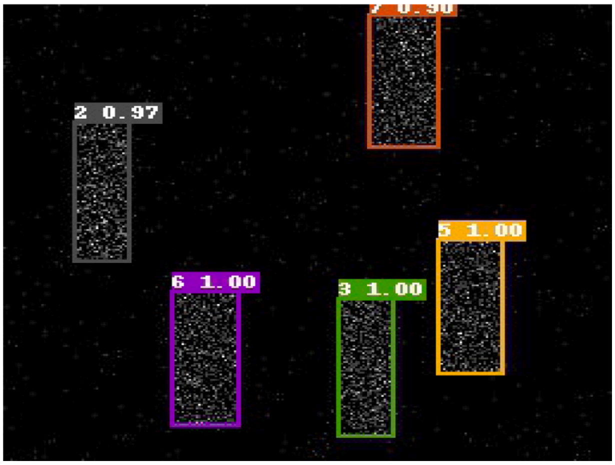

Fig. 17. Discrete objective detection by $Faster\ Rcnn\ r50$.

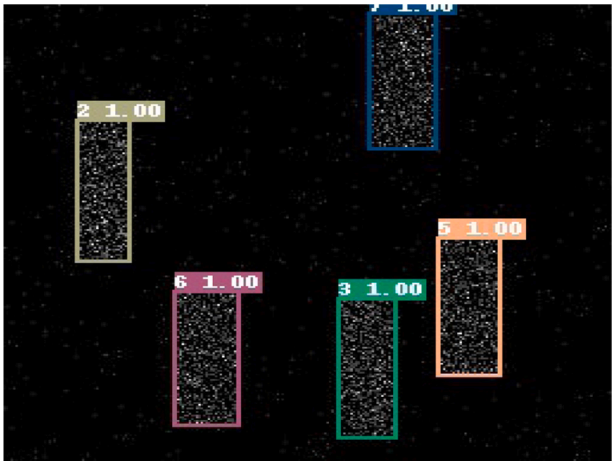

Fig. 18. Discrete objective detection by $Cascade~Rcnn~r50~FPN$.

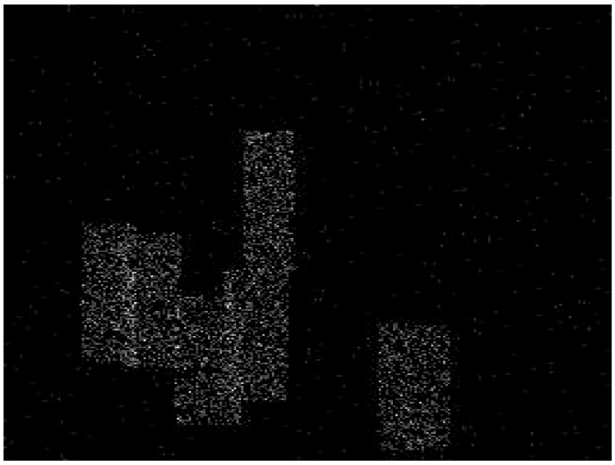

Fig. 19. Raw overlapping objective detection image.

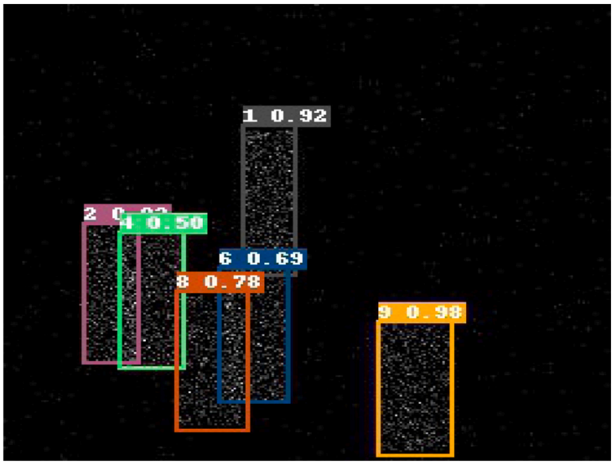

Fig. 20. Overlapping objective detection by $YOLO~v3~MobileNet~v1$.

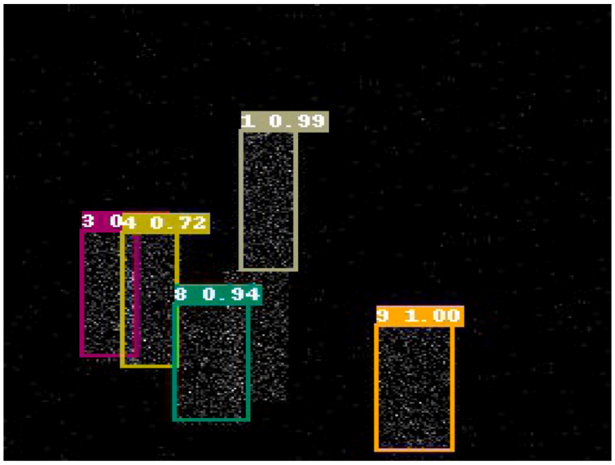

Fig. 21. Overlapping objective detection by $YOLO~v3~MobileNet~v3$.

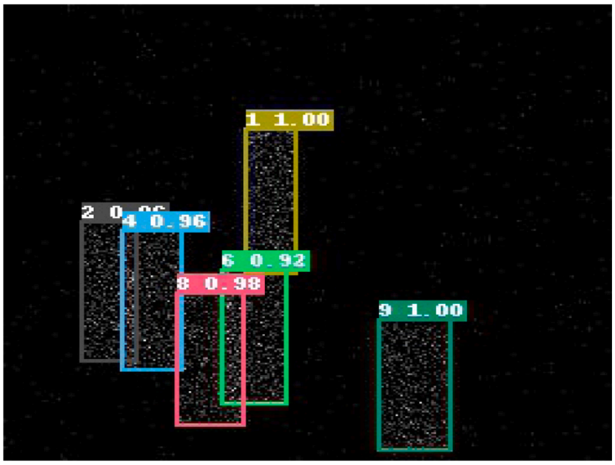

Fig. 22. Overlapping objective detection by $Faster~Rcnn~r50$.

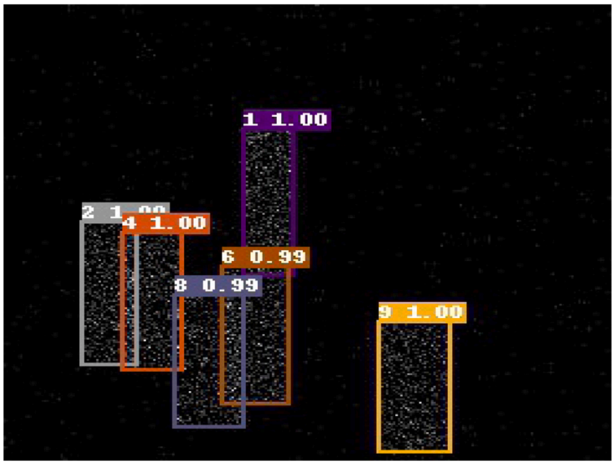

Fig. 23. Overlapping objective detection by $Cascade~Rcnn~r50~Fpn$.

high accuracy, but the confidence of recognizing overlapping objectives declines significantly. $YOLO~v3~MobileNet~v3$ still distinguishes the discrete objectives, but there is a wrong and less recognition of overlapping objectives. $Faster~Rcnn~r50$ is similar to $YOLO~v3~MobileNet~v1$, but there is a slight decline for recognizing overlapping objectives. The best model is $Cascade~Rcnn~r50~Fpn$. It recognizes all the objectives and their positions certainly with confidence close to 100%.

6. Conclusion

We propose a UAV swarm architecture based on multi-source data fusion for radar signal sorting in this paper. The knowledge of multiple source areas can be used to improve the signal sorting accuracy of the target area by exploiting deep transfer learning. Our model is pre-trained based on the data received by the UAV swarm covering source areas, and then the model is fine-tuned based on the data received by the UAV swarm covering the target area. We utilize four deep learning models, i.e., $YOLO\ v3\ MobileNet\ v1$, $YOLO\ v3\ MobileNet\ v3$, $Faster\ Rcnn\ r50$, $Cascade\ Rcnn\ r50\ Fpn$ to distinguish the actual radar signals from complex interference and noise. In addition, the proper training model parameters are given and discussed as well. Experiments show that even if the data of the target area is insufficient, these four deep transfer learning methods can be adopted to improve the signal sorting accuracy by exploiting UAV swarm.

CRediT authorship contribution statement

Liangtian Wan: Methodology, Writing – original draft, Supervision. **Rong Liu:** Data curation, Writing – original draft, Software. **Lu Sun:** Investigation, Writing – review & editing. **Hansong Nie:** Software, Validation. **Xianpeng Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants 61801076, the National Key Research and Development Program of China under Grants 2020YFC1511700, Fundamental Research Funds for the Central Universities (DUT20JC29), Radar Signal Processing National Defense Science and Technology Key Laboratory Fund (6142401200101).

References

[1] X. Wu, X. Zhu, G.Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2013) 97–107.

[2] Y. Zheng, Methodologies for cross-domain data fusion: An overview, IEEE Trans. Big Data 1 (1) (2015) 16–34.

[3] L. Sorber, M. Van Barel, L. De Lathauwer, Structured data fusion, IEEE J. Sel. Top. Sign. Proces. 9 (4) (2015) 586–600.

[4] E.E. Papalexakis, C. Faloutsos, N.D. Sidiropoulos, Tensors for data mining and data fusion: Models, applications, and scalable algorithms, ACM Trans. Intell. Syst. Technol. (TIST) 8 (2) (2016) 1–44.

[5] A. Ahmad, A. Paul, M. Rathore, H. Chang, An efficient multidimensional big data fusion approach in machine-to-machine communication, ACM Trans. Embedded Comput. Syst. 15 (2) (2016) 1–25.

[6] W. Ding, X. Jing, Z. Yan, L.T. Yang, A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion, Inf. Fusion 51 (2019) 129–144.

[7] A. Diez-Olivan, J. Del Ser, D. Galar, B. Sierra, Data fusion and machine learning for industrial prognosis: Trends and perspectives towards Industry 4.0, Inf. Fusion 50 (2019) 92–111.

[8] M.S. Hossain, G. Muhammad, Emotion recognition using deep learning approach from audio–visual emotional big data, Inf. Fusion 49 (2019) 69–78.

[9] J. Liu, T. Li, P. Xie, S. Du, F. Teng, X. Yang, Urban big data fusion based on deep learning: An overview, Inf. Fusion 53 (2020) 123–133.

[10] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.

[11] Z. Ding, M. Shao, Y. Fu, Incomplete multisource transfer learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (2) (2016) 310–323.

[12] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 2208–2217.

[13] Z. Cao, M. Long, J. Wang, M.I. Jordan, Partial transfer learning with selective adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2724–2732.

[14] W. Ying, Y. Zhang, J. Huang, Q. Yang, Transfer learning via learning to transfer, in: International Conference on Machine Learning, 2018, pp. 5085–5094.

[15] Y. Yuan, X. Zheng, X. Lu, Hyperspectral image superresolution by transfer learning, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (5) (2017) 1963–1974.

[16] K. Gong, Y. Gao, X. Liang, X. Shen, M. Wang, L. Lin, Graphonomy: Universal human parsing via graph transfer learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7450–7459.

[17] S. Shao, S. McAleer, R. Yan, P. Baldi, Highly accurate machine fault diagnosis using deep transfer learning, IEEE Trans. Ind. Inf. 15 (4) (2018) 2446–2455.

[18] H. Chen, Y. Li, D. Su, Discriminative cross-modal transfer learning and densely cross-level feedback fusion for RGB-D salient object detection, IEEE Trans. Cybern. (2019).

[19] Q. Sun, Y. Liu, T.S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 403–412.

[20] G. Wang, S. Chen, J. Huang, D. Huang, Radar signal sorting and recognition based on transferred deep learning, Comput. Sci. Appl. 9 (09) (2019) 1761–1778.

[21] A. Lin, Z. Ma, Z. Huang, Y. Xia, W. Yu, Unknown radar waveform recognition based on transferred deep learning, IEEE Access 8 (2020) 184793–184807.

[22] J. Xue, L. Tang, X. Zhang, L. Jin, Radar signal sorting method based on radar coherent characteristic, Electronics 9 (7) (2020) 1144.

[23] Q. Guo, Z. Qu, C. Wang, Pulse-to-pulse periodic signal sorting features and feature extraction in radar emitter pulse sequences, J. Syst. Eng. Electron. 21 (3) (2010) 382–389.

[24] C. Chen, M. He, J. Xu, J. Han, A new method for sorting unknown radar emitter signal, Chin. J. Electron. 23 (3) (2014) 499–502.

[25] Y. Sheng, C. Hou, W. Si, Extract pulse clustering in radar signal sorting, in: 2017 International Applied Computational Electromagnetics Society Symposium-Italy, ACES, IEEE, 2017, pp. 1–2.

[26] Z. Wang, D. Zhang, D. Bi, S. Wang, Multiple-parameter radar signal sorting using support vector clustering and similitude entropy index, Circuits Systems Signal Process. 33 (6) (2014) 1985–1996.

[27] X. Feng, X. Hu, Y. Liu, Radar signal sorting algorithm of k-means clustering based on data field, in: 2017 3rd IEEE International Conference on Computer and Communications, ICCC, IEEE, 2017, pp. 2262–2266.

[28] S. Cao, S. Wang, Y. Zhang, Density-based fuzzy C-means multi-center re-clustering radar signal sorting algorithm, in: 2018 17th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2018, pp. 891–896.

[29] Z. Mengmeng, L. Yian, Signal sorting using teaching-learning-based optimization and random forest, in: 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science, DCABES, IEEE, 2018, pp. 258–261.

[30] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016.

[31] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149, http://dx.doi.org/10.1109/TPAMI.2016.2577031.

[32] M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vis. 111 (1) (2015) 98–136.

[33] J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, CoRR, abs/1804.02767.

[34] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, CoRR, abs/1704.04861.

[35] A. Howard, M. Sandler, G. Chu, L.C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1314–1324.

[36] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[37] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.