ISPRS Journal of Photogrammetry and Remote Sensing 141 (2018) 208–218

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

A light and faster regional convolutional neural network for object detection in optical remote sensing images

Peng Ding[a,b,c,d,e,*], Ye Zhang[a,e], Wei-Jian Deng[b], Ping Jia[a,c], Arjan Kuijper[d]

[a] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China
[c] Key Laboratory of Airborne Optical Imaging and Measurement, Chinese Academy of Sciences, Changchun 130033, China
[d] Fraunhofer Institute for Computer Graphics Research & TU Darmstadt, 64283 Darmstadt, Germany
[e] State Key Laboratory of Applied Optics, Chinese Academy of Sciences, Changchun 130033, China

ARTICLE INFO

Keywords:
Deep convolution neural network
Deep learning (DL)
Remote sensing images
Object detection

ABSTRACT

Detection of objects from satellite optical remote sensing images is very important for many commercial and governmental applications. With the development of deep convolutional neural networks (deep CNNs), the field of object detection has seen tremendous advances. Currently, objects in satellite remote sensing images can be detected using deep CNNs. In general, optical remote sensing images contain many dense and small objects, and the use of the original Faster Regional CNN framework does not yield a suitably high precision. Therefore, after careful analysis we adopt dense convoluted networks, a multi-scale representation and various combinations of improvement schemes to enhance the structure of the base VGG16-Net for improving the precision. We propose an approach to reduce the test-time (detection time) and memory requirements. To validate the effectiveness of our approach, we perform experiments using satellite remote sensing image datasets of aircraft and automobiles. The results show that the improved network structure can detect objects in satellite optical remote sensing images more accurately and efficiently.

1. Introduction

With the development of remote sensing technology, the resolution of optical remote sensing images has greatly improved and images have become largely available. Compared with other types of images, remote sensing images provide more details and a clearer texture. Thus, object detection using optical remote sensing images offers many advantages. Firstly, optical remote sensing images can be used to detect "radar stealth" objects that use surface coatings and special structures. Secondly, optical remote sensing images can provide more favorable features for detection (Cheng and Han, 2016). In the international classification competition in 2012, researchers used deep convolution neural networks (deep CNNs) to classify objects, and the precision of their approach was significantly higher than those of other methods (Guo et al., 2016). In this context, deep learning (Chen and Lin, 2014; Salakhutdinov, 2014), particularly deep CNN (LeCun et al., 2015; Schmidhuber, 2015) processing, has been applied in several fields ranging from object detection (Alshehhi et al., 2017; Fytsilis et al., 2016) to object classification (Paoletti et al., 2017; Szegedy et al., 2015; Zeiler and Fergus, 2013; Zhang et al., 2017) and tracking (Cui et al., 2016; Wang and Yeung, 2013). Different methods of reducing the network training complexity and overfitting have been presented. These include initialization from the original random distribution to those of Gauss and Xavier (Glorot and Bengio, 2010), as well as attempts to reduce the difficulty of training decline and improve convergence. Moreover, the BN (Ioffe and Szegedy, 2015) approach has been demonstrated to not only reduce training difficulty, but also the possibility of overfitting. The rectified linear unit (ReLU) and parametric ReLU (PReLU) (Glorot et al., 2011; Goodfellow et al., 2013; He et al., 2015c; Kim et al., 2015; Pan and Srikumar, 2015) activation functions have replaced the original sigmoid and tanh activation functions, and since these functions more closely resemble human biological activation, the precision of the results is greatly enhanced. In addition, the use of the dropout technique (Baldi and Sadowski, 2013; Srivastava et al., 2014) has added to the success of the deep CNN approach.

In this context we adopt in our study deep CNNs to detect objects (airplanes and automobiles) in our data sets. There are several frameworks in object detection based on deep CNNs, like Regions with CNN features (RCNN) (Girshick et al., 2014), Fast Region-based Convolutional Network (Fast RCNN) (Redmon et al., 2015), and others (Kabani and Elsakka, 2016; Sermanet et al., 2013; Zitnick and Dollar, 2014).

* Corresponding author at: Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China.
E-mail address: dingpeng14@mails.ucas.ac.cn (P. Ding).

https://doi.org/10.1016/j.isprsjprs.2018.05.005
Received 24 June 2017; Received in revised form 13 March 2018; Accepted 8 May 2018
Available online 14 May 2018

Among these frameworks, the Faster RCNN approach affords suitable precision for real-time object detection (Ren et al., 2016). In the field of remote sensing, many researchers have focused on airplane detection using deep CNNs. Some of them have designed their own frameworks. Wu et al. (2015) proposed the BING approach in combination with a CNN to perform aircraft detection. However, the average detection time (or test time) for test images with this approach is about 6.414 s. In addition, the precision is not that high. Along similar lines, Cao et al. (2016) performed airplane detection by means of RCNN, which is thought to perform poorer than Faster RCNN in terms of both precision and speed. Zhang et al. (2016) performed aircraft detection by using weakly supervised CNNs. This approach is similar to the RPN + Fast RCNN (Faster RCNN without feature sharing) approach.

In the field of remote sensing, many researchers have also performed vehicle detection using deep CNNs. Ammour et al. performed car detection by combining CNNs and support vector machines (SVMs), similar to the RCNN approach (Ammour et al., 2017). Tang et al. performed vehicle detection by using RCNNs and Hard Negative Example Mining (Tang et al., 2017), which is an improvement on the Faster RCNN. They performed vehicle detection by adapting ZF-Net as the baseline and using the RealBoost algorithm to replace the Fast RCNN.

Our work is different from these approaches, since we adopt a more advanced framework, the Faster RCNN (Ren et al., 2016) framework, and choose the VGG16 network (Simonyan and Zisserman, 2015), a very deep CNN network, as the base network to detect objects. So Faster RCNN forms the holistic framework and VGG16-Net is the base network used in this framework. To improve the precision and recall of the tests, we adopt specific measures to strengthen the capability of VGG16-Net. Since the computational cost is a major problem that restricts Faster RCNN applications, we propose the use of a fully convolutional neural network instead of the fully connected layers in the Faster RCNN framework. Through this approach, the memory requirements of the final model become significantly smaller. The test-time also reduces considerably. Moreover, the precision of the approach is still able to meet our requirements.

The main contributions of this paper are thus as follows:

1. 1 For the detection of dense objects in optical remote sensing images, we adopt dilated convolutions instead of traditional convolutions to improve precision.
2. As certain objects in satellite remote sensing images are small and difficult to detect, we adopt a bootstrapping strategy called Online Hard Example Mining (Shrivastava et al., 2016) for mining hard negative examples, and we add it to Faster RCNN.
3. We use a multi-scale representation and its combinations in a new manner.
4. We propose a fully convolutional neural network instead of the fully connected layers in the Faster RCNN framework.
5. The object detection accuracy and recall show significant improvement with our approach.

The rest of the paper is organized as follows: In the next section, we describe the basic principles of CNNs and the development and principles of Faster RCNN. The details of our method are explained in Section 3. Our analysis and comparison of experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related work

2.1. Principles of convolutional neural networks

Traditional CNNs are composed of multiple stages, with each stage consisting of a convolution layer, a feature pooling layer, and a fully connected (FC) layer (Krogh and Hertz, 1992; Lecun et al., 1998).

Convolution layers: At the convolution layer, the previous layer's feature maps X_i^{l-1} are convolved with learnable kernelsk_{ij}^l, a trainable bias parameterb_j is added and the result is processed by the activation function $f(\cdot)$ to form the output feature map. This process can be expressed as:

$$X_j^l = f\left(\sum_{i \in Mj} X_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{1}$$

Here, M_j represents a selection of input maps. In this work, we chose ReLU, which is called the rectifier activation function, as the activation function in the new layers since it works better than the logistic sigmoid and hyperbolic tangent functions (Glorot et al., 2011).

Feature pooling layer: This layer treats each feature map separately. In general, this layer is called the subsampling layer, and it produces down-sampled versions of the input maps. This means that the number of input and output maps is the same, but the output maps are smaller in size. The results are robust to small variations in the location of features in the previous layer. This process can be expressed as:

$$X_j^l = down(X_j^{l-1}) \tag{2}$$

Here, $down(\cdot)$ denotes a down-sampling operation. By means of down-sampling, we reduce the size of the input by summarizing neurons from a small spatial neighborhood (Scherer et al., 2010).

Fully connected (FC) layers: After data processing by several convolutional and subsampling layers, high-level reasoning in the neural network is performed via FC layers. Neurons in an FC layer have full connections to all activations in the previous layer. Their activations can hence be computed with a matrix multiplication followed by a bias offset. The flowchart of a CNN is shown in Fig. 1.

Training is performed by means of the backpropagation algorithm (Chen et al., 2008) to minimize the aberrations between the ideal output and the actual output of the CNNs. In general, for the purpose of detection, a CNN is followed by a classification module.

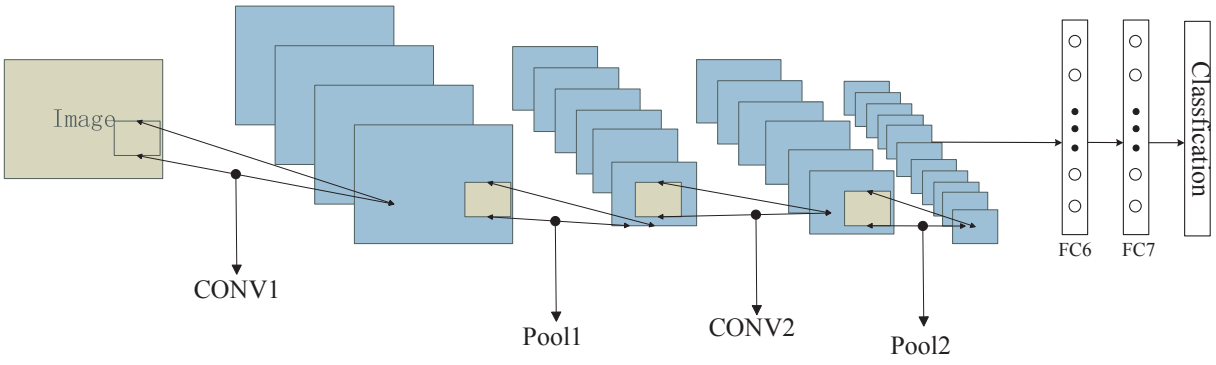

Fig. 1. Flowchart of convolutional neural network.

2.2. Development and basic principles of faster RCNN

The application of RCNNs (Girshick et al., 2014) is considered a remarkable achievement in object detection. The approach combines CNNs and a support vector machine (SVM) (Jiang et al., 2013) as well as bounding boxes (Zitnick and Dollar, 2014) to detect objects. RCNNs can be used to detect objects with high accuracy. However, the approach is time-consuming for each proposal region of different images to repeatedly undergo CNNs. Moreover, a proposal region needs to be cropped (or warped) to a fixed size for the FC layers. However, the cropped region may not contain the entire object and the warped content may result in unwanted geometric distortion. Consequently, the spatial pyramid pooling (SPP)-Net model (He et al., 2015), which uses an SPP layer to remove the fixed-size constraint, was proposed to address this issue. As the fixed-size constraint arises only from the FC layers, the pyramid pooling layer is added on top of the last convolutional layer. Moreover, with the use of SPP-Net, one can run the convolutional layers only once on the entire image. When compared with RCNN, SPP-Net exhibits significant improvements. However, SPP-Net still suffers from several disadvantages, as it is unable to update weights before the SPP layer and the training is still under a multistage pipeline. On the basis of SPP-Net, Ross proposed the Fast RCNN (Girshick and IEEE, 2015). With Fast RCNN, one can update all the network layers. The training now involves only a single stage via the use of multi-task loss. In addition, this model is faster and more precise than SPP-Net and RCNN. Importantly, there is no need of disk storage for feature caching, which is needed for the SVM. The SVM is replaced by the Softmax layer (Liu et al., 2016b), which can be inserted into the network directly. With this model, we can fine-tune all the networks, which directly aids us in finding reasonable parameters. While Fast RCNN has exhibited considerable improvements in terms of performance, the aspect of region proposal has become the bottleneck for real-time requirements Consequently, Faster RCNN was proposed to address this problem (Ren et al., 2016). In order to overcome the disadvantages of Fast RCNN, the approach uses a deep fully convolutional network called Region Proposal Network (RPN) (Ren et al., 2016) to propose regions. Subsequently, Fast RCNN uses the proposed regions to detect objects. RPN and Fast RCNN can share features, and this is speculated to aid in improving accuracy. The flowchart of Faster RCNN is shown in Fig. 2. The layers before ROI-pooling should be labeled Conv1-Conv5, but for simplicity, we only depict Conv1, Conv3, and Conv5. This simplification is also used in Figs. 3 and 4.

Among the abovementioned frameworks, Faster RCNN (FRCNN) affords several advantages, and researchers are constantly developing and refining this approach. In this work, we therefore also choose Faster RCNN as our framework. Since the VGG16-Net performs better than ZF-Net as a baseline, we choose it as our base net. Other base-models such as Resnet50 or Resnet101 are also suitable, but the memory needed during training in these cases is extremely large. Moreover, Faster RCNN with Resnet50-Net or Resnet101-Net is significantly harder to realize in practice; the results with this combination are not significantly better than those obtained with VGG16-Net in our datasets.

3. Proposed approach

3.1. Fine-tuning deep convolutional neural networks

When labeled data is scarce, there are two options. One approach is to use unsupervised pre-training followed by supervised fine-tuning. The other option is the use of a supervised pre-training model on a large auxiliary dataset, followed by domain-specific fine-tuning on the dataset (Girshick et al., 2014). In this work, we select the latter approach. The original Faster RCNN model is designed for 21 classes of the object. Here, since we only detect airplanes or cars, we only have two categories: the object and the background. Every object has a tuple, $and foraground-truthbounding-boxregression$, $wesetupthecategoriesas$, and for a ground-truth bounding-box regression, we set up the categories as$4*(K+1)$, where K denotes the number of categories.

3.2. Online hard example mining

Online Hard Example Mining (OHEM) is a type of bootstrapping technique (Shrivastava et al., 2016), which is applied to the standard Fast RCNN framework. We apply the technique to Faster RCNN because every dataset has certain examples that are hard to train. The Faster RCNN adopts the hard example mining by setting the ratio between the foreground ROIs and background ROIs as 3:1, which is valid for VOC datasets. However, this setting may not be appropriate for our datasets. Still, the OHEM can aid in solving this problem. The flowchart of Faster RCNN with OHEM is shown in Fig. 3. In our approach, we sort the input ROIs by loss and consider the examples on which the current network performs the poorest as hard examples. The network computes forward and backward passes only for the selected hard examples, accumulates the gradients, and passes them on to the convolutional network.

3.3. Dilation

In optical remote sensing images, many objects are usually small and dense. For example, when an object of 28 * 28 pixels passes through a basic VGG16-Net network, the output map has a resolution of only two pixels. Therefore, the output feature map cannot contain too much information, resulting in low accuracy and recall. Dilated convolutions are specially designed for dense prediction support expansion of receptive field without loss of resolution. They improve the receptive field and thus the accuracy and recall rate, especially for small objects. In order to enlarge the receptive field by using dilated convolution, the pool4 layer is omitted, and we extend all Conv5 filters to 2.

3.4. Multi-scale representation and combination

In general, the last layer of Conv5 is adopted to generate candidate

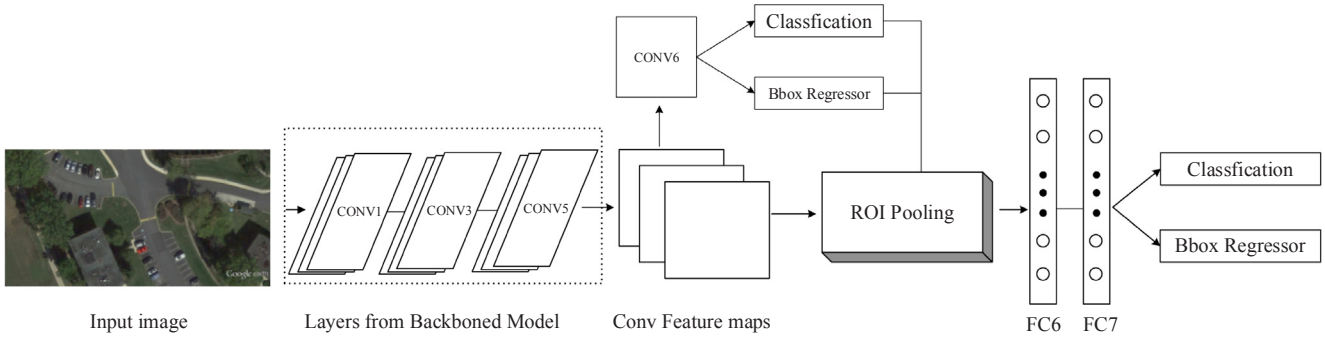

Fig. 2. Flowchart of Faster region convolutional neural network (Faster RCNN).

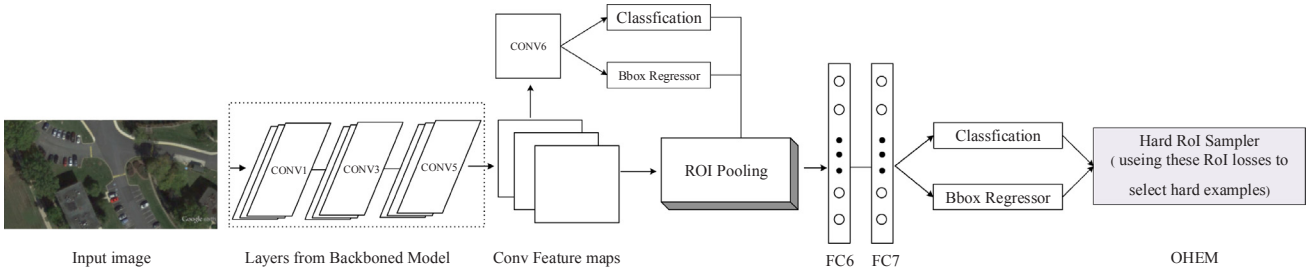

Fig. 3. Flowchart of Faster RCNN with OHEM.

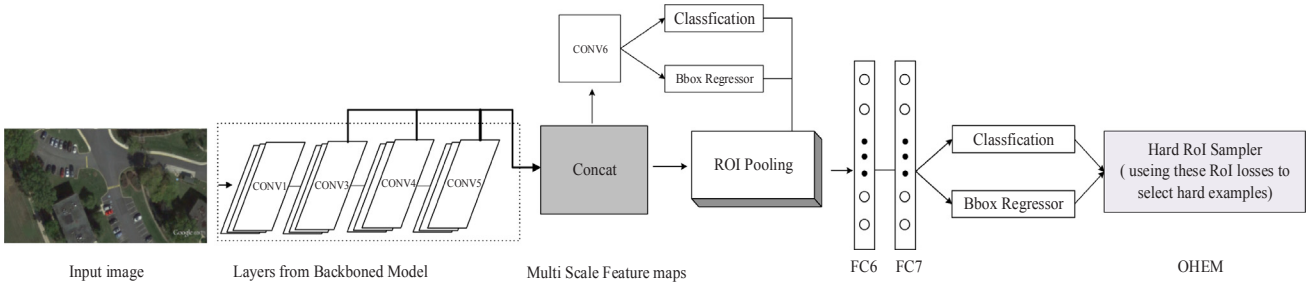

Fig. 4. Flow chart of Faster RCNN with OHEM and multi-scale prediction.

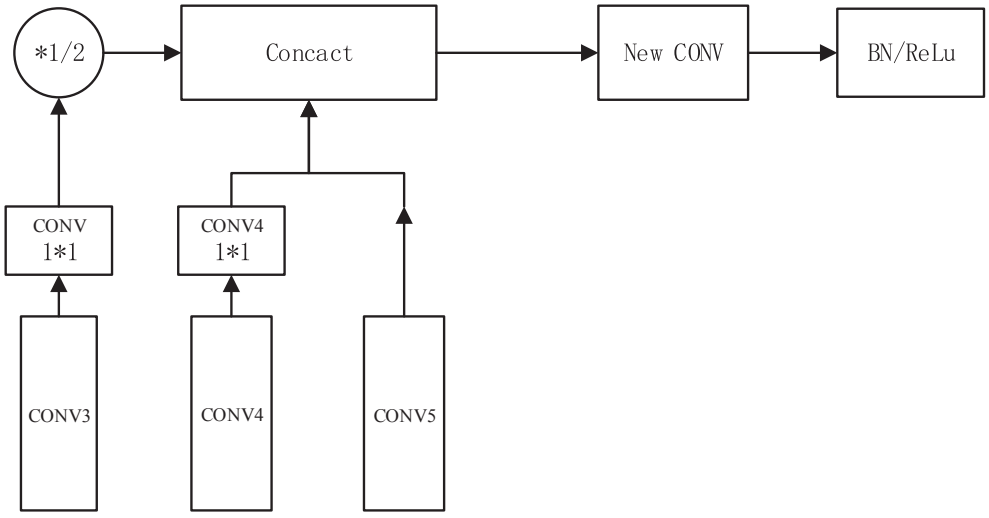

Fig. 5. Details of multi-scale representation and its combinations.

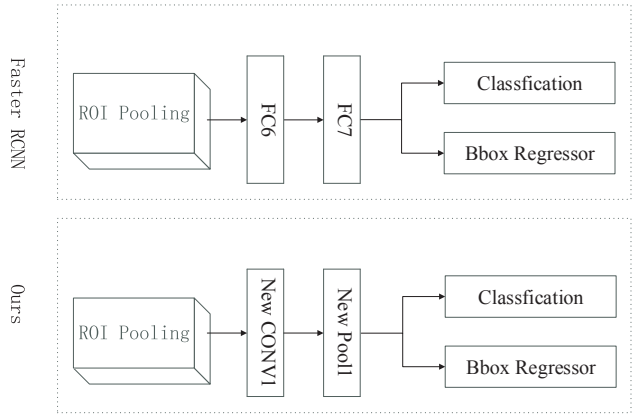

Fig. 6. Flowchart of our approach.

regions or detect objects. For many tasks this is not enough, because the shallow layers in CNNs include fine-grained details. These are conducive to detect small objects or blur objects. Nowadays, a multi-scale representation and its combinations such as Hyper-Net have been proven to be effective (Kong et al., 2016). This method just connects some layers to detect objects which would cause overfitting sometimes. Besides, Hyper-Net puts ROI-pooling into the RPN stage and adds some special layers which we consider as not necessary. In this paper, we also adopt the multi-scale representation and its combinations to detect objects, as our objects need information of shallow layers. We not just only connect some different layers, we also adopt lateral connections. By our approach, we can detect objects more accuracy. The total flow chart is shown in Fig.4. In order to deduce overfitting, for the new layer, we add Batch Normalization (Ioffe and Szegedy, 2015) and scale layer to help convergence. For this new convolution layer, we adopt an initialize method called Xavier (Glorot and Bengio, 2010; He et al., 2015c), which has been proven more helpful to convergence than Gaussian. Lastly, ReLu (Glorot et al., 2011; Gulcehre et al., 2016; He et al., 2015c; Pan and Srikumar, 2015; Xu. et al., 2015) activation is

Table 1
Comparison between improved VGG16-Net and original VGG16-Net for airplane dataset.

	Plane				
OHEM (O)	×	√	×	√	√
Dilation (d)	×	×	√	√	√
Multi-scale (M)	×	×	×	×	√
AP	0.8871	0.9038	0.8942	0.9065	0.9070
Recall	0.8875	0.9278	0.9227	0.9631	0.9685

needed for this new layer. Some papers use the top-down and lateral approach, but this requires too much GPU memory. Furthermore, they are too slow, both in training and in testing.

Details of our multi-scale representation and its combinations is shown in Fig.5.

3.5. Lighter and faster FRCNN

In this paper we adopt "Faster RCNN with enhanced VGG16-net", a framework that affords more accurate object detection than other frameworks. The memory requirement of the final model is large and the test-time is about 0.104 s for an image of size 1200 × 600 pixels (with the CuDNN opened using Tianx). Thus, the question of reducing the memory requirement of the final model and increasing the speed of detection becomes significant. Here, we propose an approach to reduce the detection time and the storage space required, which is important for practical applications. The ROI-pooling layer forms a significant aspect of Faster RCNN. However, the two following layers are FC layers, which require a large number of parameters. This leads to increased storage space and decreased detection speed. Thus, we adopt a convolutional layer instead of these two FC layers. In order to ensure that the weights can be updated, the size of the convolution kernel is set to 3 and the pad is set as 1. For the last classification, the input size of the feature map should be 1 * 1, and we therefore add a new pooling layer after the convolutional layer. Details of our "Lighter and Faster" FRCNN are illustrated in Fig. 6. For simplicity, we only display the modified part after the ROI-pooling layer without OHEM.

4. Experiment and discussion

All experiments in this study were performed on two commonly used remote sensing datasets: the airplane and car datasets (Zhu et al., 2015). The aircraft and car datasets are both proposed by the University of the Chinese Academy of Science. The airplane dataset is composed of 1000 optical remote sensing images, with about 7000 objects in total. The car dataset is composed of 500 optical remote sensing images, with about 7000 objects in total. all the experiments were performed on the GTX-Tianx, Intel(R) Core(TM) i7-6850K CPU with Caffe (Jia et al., 2014), MATLAB2014a, and certain other software such as Opencv3.0. We choose VGG16-Net (Simonyan and Zisserman, 2015) as the baseline, which is a good choice for its detection ability. In terms of precision, it is better than ZF-Net (Krizhevsky et al., 2012). Further, in terms of speed, it performs significantly better than ResNet (He et al., 2015a). Our framework was run for 40000 stochastic gradient descent (SGD) iterations (Bottou, 2012; Duchi et al., 2011; Kingma and Ba, 2014; Krogh and Hertz, 1992; Zeiler, 2012). The learning rate was set as 0.001 with the momentum set as 0.0005 (Duchi et al., 2011; Kingma and Ba, 2014; Qian, 1999; Zeiler, 2012).

We adopt the two commonly used objective criteria of average precision (AP) and recall to evaluate the performance of our approach. These parameters defined as follows:

$$average{-}precision = \frac{True\ positive}{True\ positive + True\ negative} \tag{3}$$

$$recall = \frac{True\ positive}{True\ positive + False\ negative} \tag{4}$$

These two criteria are generally used in deep learning. "False negative" is a test result indicating that a condition does not hold, while in fact it does. "True negative" is a test result indicating that a condition does not hold and in fact it does not. "True positive" is a result indicating that a given condition exists and it does exist. We evaluate all the images in the test set instead of evaluating the objects in a single image. This approach is different from the traditional evaluation that involves evaluating the result with a single image.

4.1. Precision and recall analysis

Accuracy and recall rate form our foremost priorities. It is meaningful to improve the speed of the convolution network on the basis of

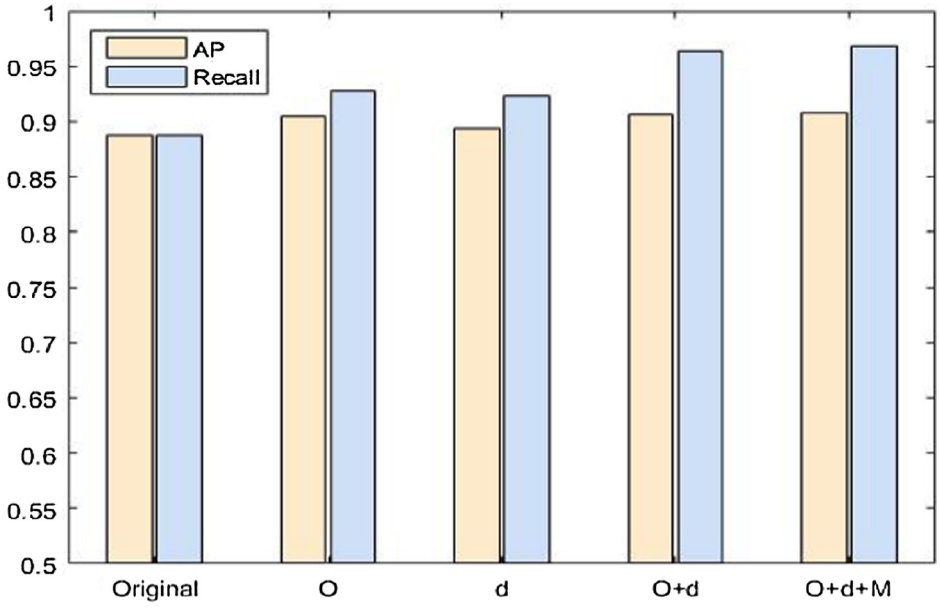

Fig. 7. Column charts depicting improved VGG16-Net performance on aircraft dataset.

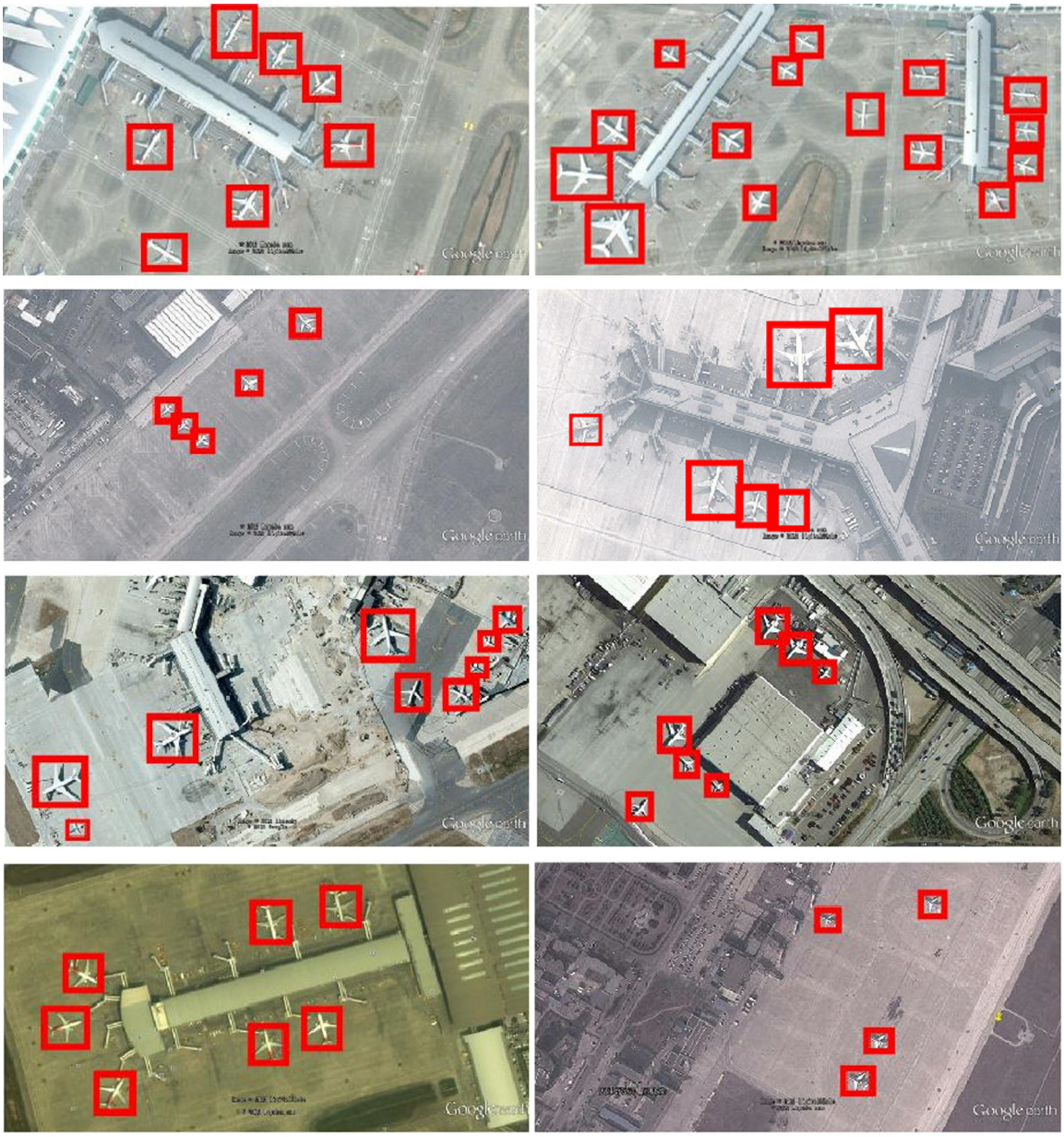

Fig. 8. Some representative results of our approach for plane dataset.

Table 2
Comparison between improved VGG16-Net and original VGG16-Net for vehicle data set.

	Car				
OHEM (O)	×	√	×	√	√
Dilation (d)	×	×	√	√	√
Multi-scale (M)	×	×	×	×	√
AP	0.7519	0.7729	0.7973	0.8388	0.8790
Recall	0.7836	0.8205	0.8629	0.8759	0.8846

precision and recall rate, and therefore, we first analyze the impact of different improvement schemes on accuracy and recall rate.

4.1.1. Aircraft datasets

As mentioned before, the aircraft dataset includes 1000 optical remote sensing images, with about 7000 objects. For higher resolution and larger size, Faster RCNN with the base VGG16-Net can yield good precision and recall, but with our improved methods, we can further improve the precision and recall, particularly the recall. The precision and recall of the different improvement schemes (OHEM, Dilation, and Multi-scale on or off) are listed in Table 1.

A more intuitive comparison is shown in Fig. 7.

Several effects of the different improvement schemes on the results can be observed. The comparison between the different frameworks and the baselines was introduced in Section 4.3. With regard to the results listed in Table 1 and Fig. 7, we note that the different improvement schemes indeed improve the result. The final precision is slightly better than that of the original framework. This improvement is not very significant. This is due to the fact that the size of the aircraft is not very small, whereas our improvements are more suitable for small objects. One can see that some smaller aircraft can be detected very accurately. This is clear from Fig. 8, where some representative results of our approach are shown. It is thus worth noting that the recall rate increases substantially. This is because the receptive field is extended by the dilation and the multi-scale representation. A small object that has been neglected in the original framework is learned in our approach.

In Fig. 8, one can see that both large and small airplanes can be

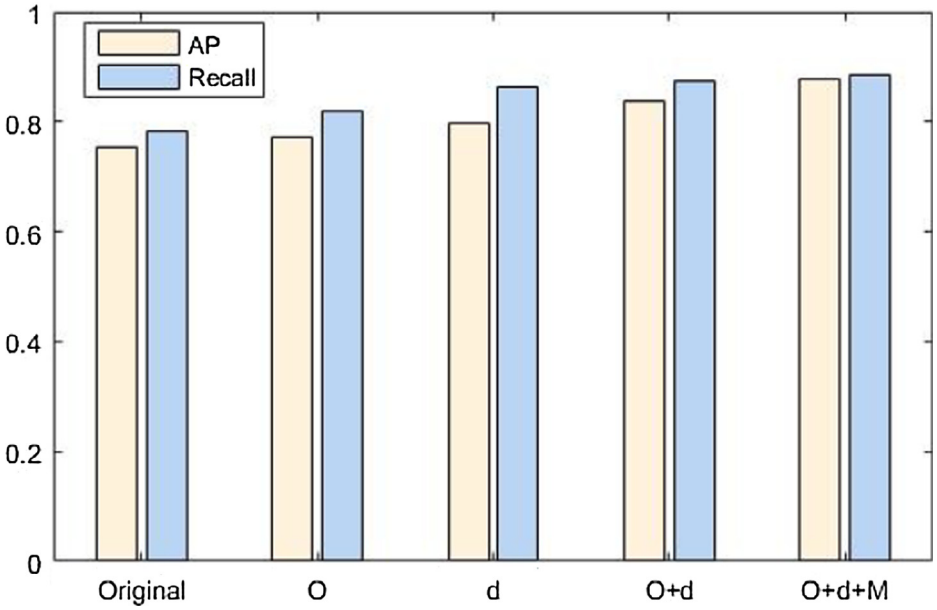

Fig. 9. Column charts showing improvement in VGG16-Net performance on car dataset.

detected precisely. From these images, it can be noted that our method is able to obtain very effective results, which makes the method suitable for object detection in optical remote sensing images. Even in environments such as a foggy environment, objects can still be detected.

4.1.2. Car datasets

The car dataset is composed of 500 optical remote sensing images, with about 7000 objects in total. With the Faster RCNN and the base VGG16-network, we found the precision to be very low and the recall to be even lower. This was because the resolution (that is, the size) of cars is less than that of airplanes. However, by applying our improvement schemes, the recall and precision improved significantly. the recall and precision subsequently improved significantly. Here, we also discuss the effect of different improvement schemes on the results; the precision and recall obtained with the various schemes are listed in Table 2.

A more intuitive comparison is shown in Fig. 9.

The accuracy rate for car detection has increased by 13%, the recall rate by 10%. This has two reasons. Firstly, dilation is a good approach to extend the receptive field, which can aid in detecting small objects. With a larger receptive field, more features can be extracted by deep CNNs. Secondly, deconvolution and the lateral connection, which are used in our multi-scale representation and its combinations, also aid in extending the receive field without loss of resolution. Furthermore, OHEM is a suitable approach to address examples that are hard to train. The stronger the ability of the approach is to cope with difficult examples, the better the features are that can be learned by CNNs. Overall, our approach thus offers better feature learning, and therefore the accuracy rate and recall rate both increased by more than 10% in our experiments. Some representative results of our approach are shown in Fig. 10.

As can be seen, cars are detected very accurately and the bounding-box regression works very well. One can pinpoint the location of the object very accurately. Thus, our method is very suitable for the detection of small objects. Objects can be suitably acquired in dense situations involving road vehicles or in residential areas.

4.2. Analysis of lighter and faster FRCNN

Besides the accuracy and recall rates, the detection speed is also an important aspect. In this section, we compare our Lighter and Faster FRCNN with FRCNN without the inclusion of our improvement schemes. The average accuracy and test-time as well as the required memory for different channels are listed in Table 3, where ******* stands for no channels. As the image sizes of the two datasets are the same, the average test time of the images is almost the same.

As can be observed from the results in Table 3, we chose the number of channels as 32, which is suitable for our datasets. We note that our Lighter and Faster RCNN performs significantly better than Faster RCNN in terms of both speed and memory requirements, while the accuracy is also very high. The convolution layer is more powerful than the FC layer. A more detailed analysis is provided in Figs. 11 and 12 (where the memory requirement is normalized, we just provide detailed analysis of airplane).

From Figs. 11 and 12, the influence of the different channels appears more intuitive and clear. From the discussion above, we note that with our approach the speed can increase by up to 13 frames per second and the memory required becomes significantly smaller.

4.3. Comparison with other popular frameworks

In Sections 4.1 and 4.2, we discussed the effect of different improvement schemes on the detection results. Here, we discuss the advantages of our approach over other frameworks. The recent years have seen the proposal of several frameworks for object detection. Among these, we chose some well-known ones such as Faster RCNN, YOLO9000 (Redmon and Farhadi, 2016), SSD (Liu et al., 2016a) and RFCN (Dai et al., 2016; Xie et al., 2016) for our comparison. Moreover, many deep CNNs such as ZF-Net (Zeiler and Fergus, 2014), VGG-Net, Resnet50, and Resnet101 (He et al., 2015a; Xie et al., 2016) have also been proposed. These frameworks and deep CNNs are effective in many detection tasks. Here, we mainly analyze the precision and recall of different frameworks and different baselines. These results are listed in Table 4.

As proposed by He, for Resnet50/Resnet101, ROI pooling should be performed before conv5. On this ROI-pooled feature, all layers of Conv5 and beyond are adopted for each region, and they play the roles of VGG-16's FC layers. We call this modification Faster RCNN*. Further, RFCN is another famous framework proposed by Dai (Dai et al., 2016; Xie et al., 2016). This framework is more compatible with Resnet50/Resnet101 for both speed and precision. Further, although Resnet50/Resnet101 afford better classification than VGG16-net, our results compare to RFCN with Resnet50/Resnet101 and Faster RCNN* with

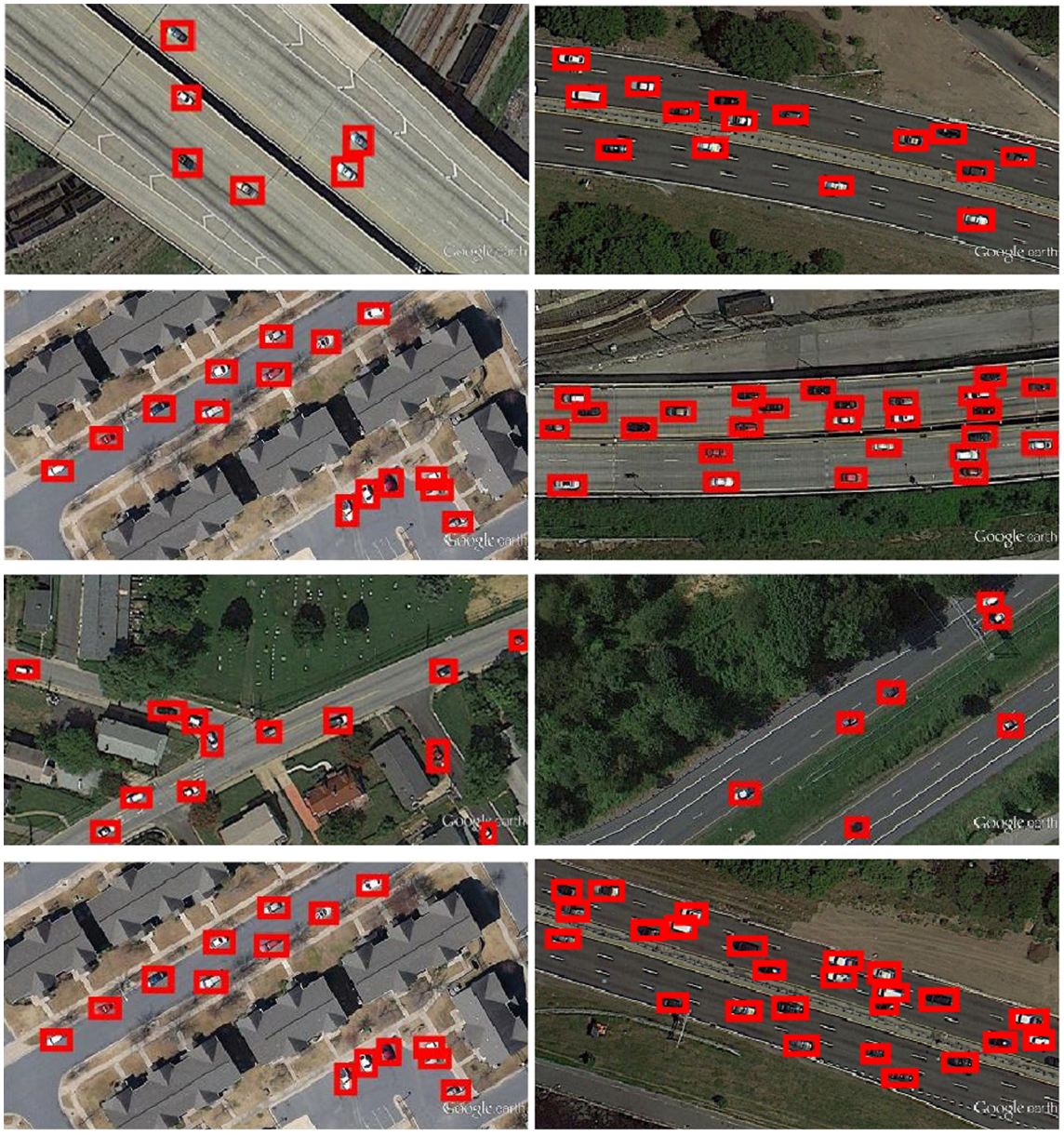

Fig.10. Some representative results of our approach for the car dataset.

Table 3
Average memory, precision, and test-time required for the different channels.

Frame	Channels	Memory (M)	AP(Plane/Car)	Test-time (s)
Faster RCNN	4096 (Fc layer)	512.0	0.887/0.752	0.104
Faster RCNN(No FC)	*******	58.9	0.667/0.653	0.086
Lighter and faster FRCNN	1024	77.8	0.887/0.752	0.096
	512	68.3	0.887/0.752	0.089
	256	63.6	0.887/0.752	0.087
	128	61.2	0.887/0.752	0.086
	64	60.0	0.887/0.752	0.085
	32	59.5	0.887/0.752	0.083
	16	59.2	0.887/0.745	0.082
	8	59.0	0.887/0.739	0.081
	4	58.9	0.887/0.736	0.081
	2	58.9	0.856/0.679	0.081

Resnet50/Resnet101. Faster RCNN and RFCN includes to two-stage detectors. Compared to Faster RCNN and RFCN, one-stage detectors such as SSD and YOLO9000 are applied over a regular, dense sampling of objects locations, scales, and aspect rations. Because the Faster RCNN contains a size conversion operation, for a more equitable comparison, we change the input size of SSD and YOLO9000 to 800 * 600. YOLO9000 works very well for for VOC datasets, but may be not suitable for our datasets. One-stage detectors must process a large amount of candidate object locations which would cause class imbalance. Class imbalance would affect the precision. We can see that newest SSD's performance is much better, because it uses many tricks: much richer data expansion; the use of multi-layer network information and hard example mining. They do hard example mining by setting the ratio between the foreground ROIs and background ROIs as 3:1. Although they use of multi-layer network information, the feature of different layers is unrelated. Moreover, we change the input size of SSD and YOLO9000 to 800 * 600 rather than 300 * 300 (448 * 448 for YOLO9000). This operation is helpful to improve precision.

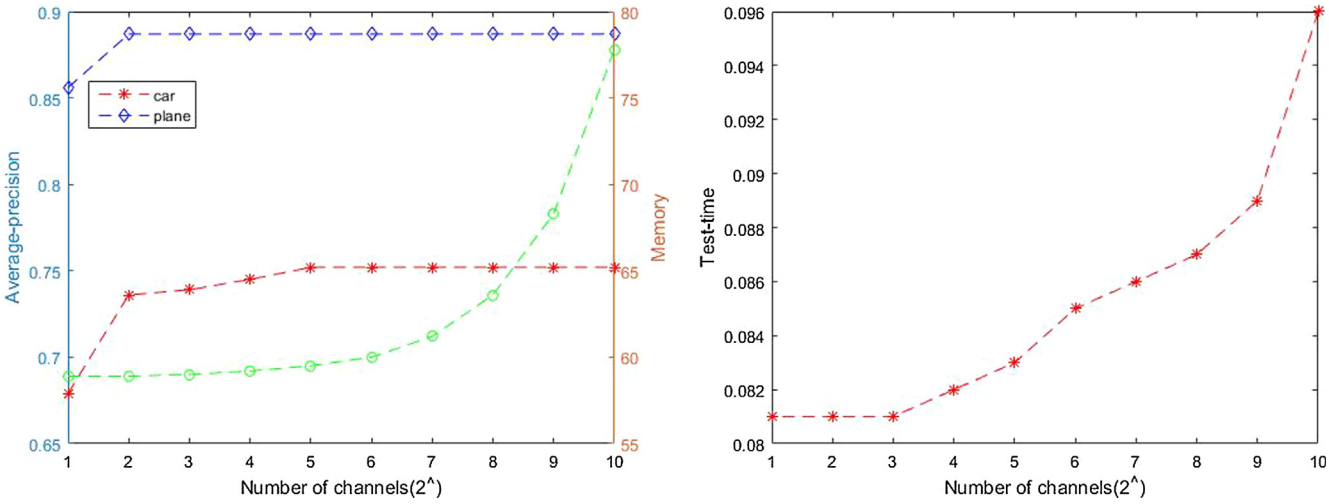

Fig. 11. Plots of precision, test-time, and memory requirement with Lighter and Faster RCNN (Left plane/right car).

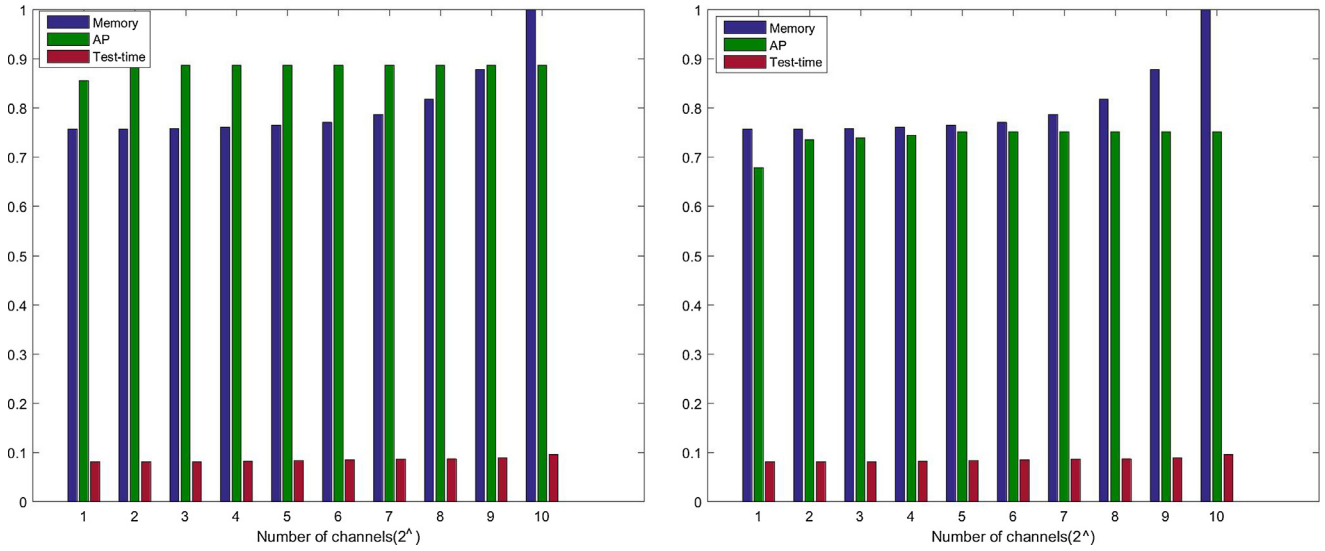

Fig. 12. Column charts depicting precision, test-time, and memory requirements with Lighter and Faster RCNN (Left plane/right car).

Table 4
Average precision and recall of different frames and baselines.

Frames	AP (Plane/car)	Recall (Plane/car)	Memory	Test-time
SSD	0.891/0.813	0.945/0.837	95.0M	0.049s
YOLO9000	0.885/0.669	0.922/0.735	268.2M	0.022s
Faster RCNN (ZF)	0.841/0.647	0.861/0.652	233.2M	0.070 s
Faster RCNN (VGG)	0.887/0.752	0.888/0.784	512.0M	0.104 s
Faster RCNN* (Resnet50)	0.897/0.759	0.930/0.781	94.4M	0.416 s
Faster RCNN* (Resnet101)	0.898/0.761	0.931/0.792	170.6M	0.527 s
RFCN (Resnet50)	0.898/0.784	0.944/0.793	108.5M	0.119 s
RFCN (Resnet101)	0.899/0.806	0.946/0.822	184.7M	0.139 s
Ours* (FC)	0.907/0.879	0.968/0.885	549.3M	0.133 s
Ours (Faster and lighter)	0.907/0.879	0.968/0.885	71.5M	0.112 s

The corresponding precision–recall curves are shown in Fig. 13. The precision–recall curve is an important indicator of the performance of a given model.

From the results, we note that the precision and recall rates of our approach are both considerably better than those of other frameworks and other baselines. The corresponding column charts of the test-time and memory requirement are shown in Fig. 14 (where the memory requirement is normalized).

In terms of speed, our approach performs highly satisfactorily. Importantly, the storage space required in our (faster and lighter) approach is significantly less than that for the other frameworks. Overall, our approach thus performs significantly better than other frameworks and baselines.

5. Conclusions

The task of object detection in optical remote sensing images has attracted considerable research attention in recent years. At the same time, significant progress has been made in object detection by the application of deep learning, particularly deep CNNs. In this work, we propose several improvements for deep CNNs in optical remote sensing. For dense objects in optical remote sensing images, we adopt dilated convolutions instead of traditional convolutions to improve precision. As certain objects in satellite remote sensing images are small and difficult to detect, we adopt a bootstrapping strategy called OHEM for mining hard negative examples. We implement OHEM in Faster RCNN for object detection. Moreover, multi-scale representation and combinations are utilized in a novel manner in this work. The computational cost of the method is a primary issue that restricts its application. We

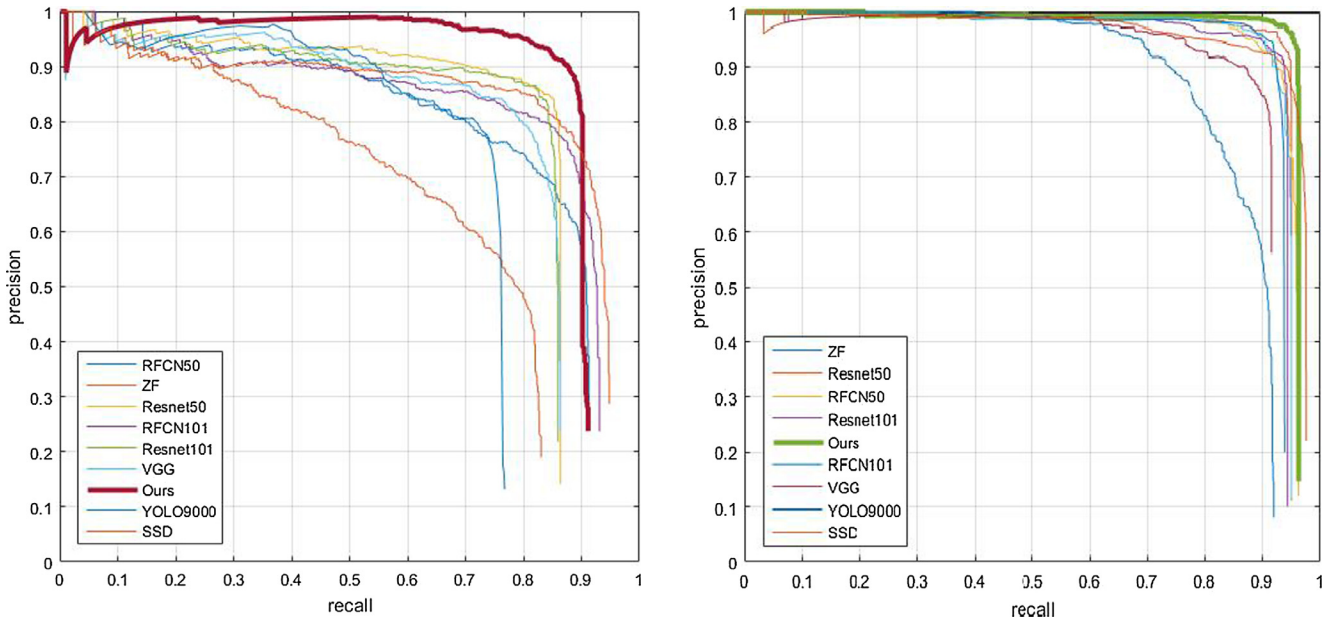

Fig. 13. Precision–Recall curve of different frameworks for aircraft dataset (right) and car dataset (left).

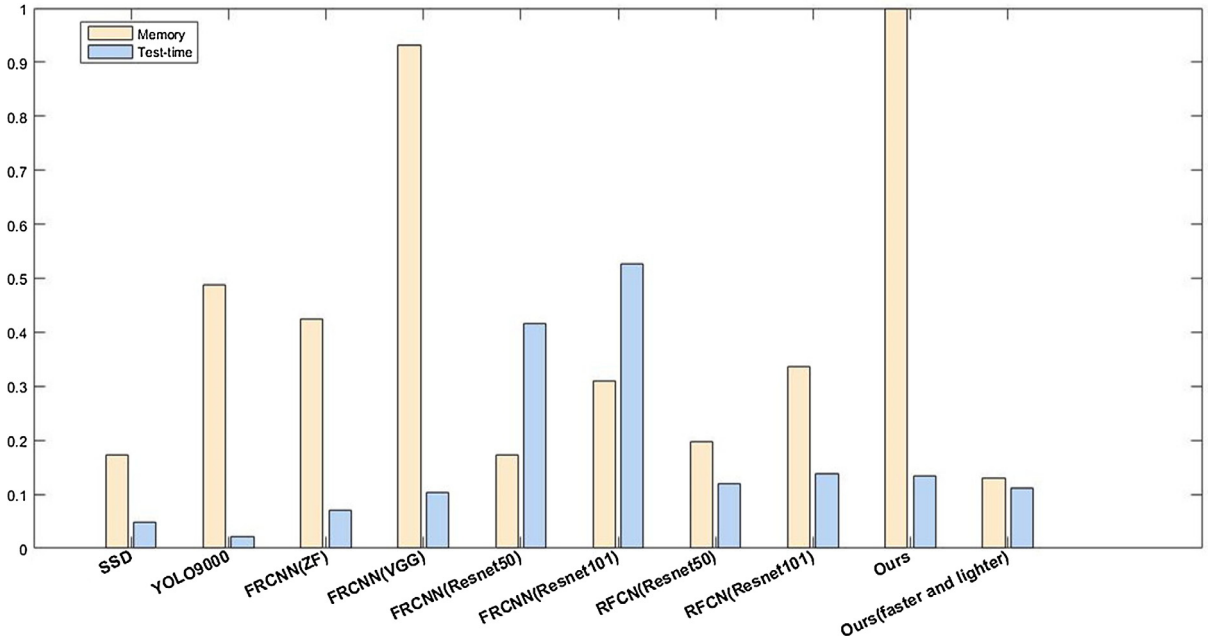

Fig. 14. Column charts of test-time and memory requirement for schemes compared in the study.

thus propose the use of a fully convolutional neural network instead of fully connected layers in the Faster RCNN framework. Through this approach, the memory requirement of the final model significantly reduces along with the test-time relative to the corresponding ones of the original framework. Importantly, the resulting precision fulfills our requirements. Our approach is useful for the application of deep CNNs to object detection in remote sensing images. In future, we plan to further improve the network's detection ability.

References

Alshehhi, R., Marpu, P.R., Woon, W.L., Mura, M.D., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. ISPRS-J. Photogramm. Rem. Sens. 130, 139–149.

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., Zuair, M., 2017. Deep learning approach for car detection in UAV imagery. Rem. Sens. 9, 312.

Baldi, P., Sadowski, P., 2013. Understanding Dropout, Neural Information Processing Systems.

Bottou, L., 2012. Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade, second ed. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 421–436.

Cao, Y.S., Niu, X., Dou, Y., 2016. Region-based convolutional neural networks for object detection in very high resolution remote sensing images. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (Icnc-Fskd), pp. 548–554.

Chen, W., Wei, X., Zhao, T., 2008. Product Schemes Evaluation Method Based on Improved BP Neural Network. In: International Conference on Intelligent Computing, pp. 99–106.

Chen, X., Lin, X., 2014. Big data deep learning: challenges and perspectives. IEEE Access 2, 514–525.

Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. ISPRS-J. Photogramm. Rem. Sens. 117, 11–28.

Cui, Z., Xiao, S., Feng, J., Yan, S., 2016. Recurrently Target-attending Tracking, Computer Vision and Pattern Recognition, pp. 1449–1458.

Dai, J., Li, Y., He, K., Sun, J., 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks.

Duchi, J.C., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159.
Fytsilis, A.L., Prokos, A., Koutroumbas, K.D., Michail, D., Kontoes, C.C., 2016. A methodology for near real-time change detection between Unmanned Aerial Vehicle and wide area satellite images. ISPRS-J. Photogramm. Rem. Sens. 119, 165–186.
Girshick, R., Donahue, J., Darrell, T., Malik, J., IEEE, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York, pp. 580–587.
Girshick, R., IEEE, 2015. Fast R-CNN, 2015 IEEE International Conference on Computer Vision. IEEE, New York, pp. 1440–448.
Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. 249–256.
Glorot, X., Bordes, A., Bengio, Y., 2011. Deep Sparse Rectifier Neural Networks, AISTATS, pp. 315–323.
Goodfellow, I.J., Wardefarley, D., Mirza, M., Courville, A., Bengio, Y., 2013. Maxout Networks, International Conference on Machine Learning.
Gulcehre, C., Moczulski, M., Denil, M., Bengio, Y., 2016. Noisy Activation Functions, International Conference on Machine Learning.
Guo, Y.M., Liu, Y., Oerlemans, A., Lao, S.Y., Wu, S., Lew, M.S., 2016. Deep learning for visual understanding: a review. Neurocomputing 187, 27–48.
He, K., Zhang, X., Ren, S., Sun, J., 2015a. Deep Residual Learning for Image Recognition, Computer Vision and Pattern Recognition, pp. 770–778.
He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J., 2015b. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37, 1904–1916.
He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J., IEEE, 2015c. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: 2015 IEEE International Conference on Computer Vision. IEEE, New York, pp. 1026–1034.
Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, International Conference on Machine Learning.
Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional Architecture for Fast Feature Embedding, acm multimedia, pp. 675-678.
Jiang, S.Q., Jin, H.Z., Wei, F.M., Ieee, 2013. LS-SVM application for ship course model predictive control. In: 2013 IEEE International Conference on Mechatronics and Automation (Icma), pp. 1615–1619.
Kabani, A., Elsakka, M.R., 2016. Object Detection and Localization Using Deep Convolutional Networks with Softmax Activation and Multi-class Log Loss. In: International Conference on Image Analysis and Recognition, pp. 358–366.
Kim, J., Kim, S., Lee, M., 2015. Convolutional Neural Network with Biologically Inspired ON/OFF ReLU, International Conference on Neural Information Processing.
Kingma, D., Ba, J., 2014. Adam: a method for stochastic optimization. Comput. Sci.
Kong, T., Yao, A., Chen, Y., Sun, F., 2016. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection, Computer Vision and Pattern Recognition, pp. 845–853.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, Neural Information Processing Systems, pp. 1097–1105.
Krogh, A., Hertz, J.A., 1992. A simple weight decay can improve generalization, Neural Information Processing Systems.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324.
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C., 2016a. SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision, pp. 21–37.
Liu, W., Wen, Y., Yu, Z., Yang, M., 2016b. Large-Margin Softmax Loss for Convolutional Neural Networks. In: International Conference on Machine Learning.
Pan, X., Srikumar, V., 2015. Expressiveness of rectifier networks. In: International Conference on Machine Learning.
Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., 2017. A new deep convolutional neural network for fast hyperspectral image classification. ISPRS-J. Photogramm. Remote Sens.
Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural Netw. 12, 145–151.
Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A., 2015. You Only Look Once: Unified, Real-Time Object Detection. CoRR abs/1506.02640.
Redmon, J., Farhadi, A., 2016. YOLO9000: Better, Faster, Stronger. CoRR abs/1612.08242.
Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell pp. 1-1.
Salakhutdinov, R., 2014. Deep Learning, Knowledge Discovery and Data Mining.
Scherer, D., Muller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In: International Conference on Artificial Neural Networks.
Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Netw. 61, 85–117.
Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y., 2013. OverFeat: integrated recognition, localization and detection using convolutional networks. Comput. Sci.
Shrivastava, A., Gupta, A., Girshick, R., 2016. Training region-based object detectors with online hard example mining. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 761–769.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR).
Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, computer vision and pattern recognition, pp. 1–9.
Tang, T., Zhou, S., Deng, Z., Zou, H., Lei, L., 2017. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. Sensors 17, 336.
Wang, N., Yeung, D., 2013. Learning a deep compact image representation for visual tracking, neural information processing systems, pp. 809–817.
Wu, H., Zhang, H., Zhang, J.F., Xu, F.J., IEEE, 2015. Fast aircraft detection in satellite images based on convolutional neural networks. In: 2015 IEEE International Conference on Image Processing. IEEE, New York, pp. 4210–4214.
Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K., 2016. Aggregated Residual Transformations for Deep Neural Networks. CoRR abs/1611.05431.
Xu., B., Wang., N., Chen, T., 2015. Empirical Evaluation of Rectified Activations in Convolutional Network.
Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. Comput. Sci.
Zeiler, M.D., Fergus, R., 2013. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833.
Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I. Springer International Publishing, Cham, pp. 818–833.
Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., Atkinson, P.M., 2017. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. ISPRS-J. Photogramm. Rem. Sens.
Zhang, F., Du, B., Zhang, L.P., Xu, M.Z., 2016. Weakly supervised learning based on coupled convolutional neural networks for aircraft detection. IEEE Trans. Geosci. Rem. Sens 54, 5553–5563.
Zhu, H.Q., Chen, X., Dai, W., Fu, K., Ye, Q., Jiao, J., 2015. Orientation robust object detection in aerial images using deep convolutional neural network. In: International Conference on Image Processing, pp. 3735–3739.
Zitnick, C.L., Dollar, P., 2014. Edge boxes: locating object proposals from edges. In: European Conference On Computer Vision, pp. 391–405.