Pattern Recognition 122 (2022) 108315

Contents lists available at ScienceDirect

Pattern Recognition

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

DDBN: Dual detection branch network for semantic diversity predictions

Qifeng Lin[a], Chengjiang Long[b], Jianhui Zhao[a,*], Gang Fu[a], Zhiyong Yuan[a,*]

[a] School of Computer Science, Wuhan University, Wuhan 430072, China
[b] Finance America Corporation, Mountain View, CA, USA

ARTICLE INFO

Article history:
Received 12 January 2021
Revised 25 August 2021
Accepted 9 September 2021
Available online 10 September 2021

2010 MSC:
00-01
99-00

Keywords:
Adjacent feature compensation
Dual detection branch network
Diversity enhancement strategy
Object detection

ABSTRACT

It is well known that detail features and context semantics are conducive to improving object detection performance. However, the current single-prediction detectors do not well incorporate these two types of information together. To alleviate the limitation of single-prediction on the use of multiple types of information, we propose a dual detection branch network (DDBN) with adjacent feature compensation and customized training strategy for semantic diversity predictions. Different from the conventional single-prediction models, our DDBN is in the form of a single model with dual different semantic predictions. In particular, two types of adjacent feature compensations are designed to extract detail and context information from different perspectives. Also, a specialized training strategy is customized for our DDBN to well explore the diversity of predictions for improving the performance of object detection. We conduct extensive experiments on three datasets, *i.e.*, DOTA, MS-COCO, and Pascal-VOC, and the experimental results strongly demonstrate the efficacy of our proposed model.

1. Introduction

As we know, the prediction is always accompanied by uncertainty [1]. For human visual cognitive systems, when there is uncertainty in identifying an object, humans often need more clues (such as internal detail information or external context information) to enhance the certainty of prediction. Similarly, if the object detection model can better grasp these two types of information at the same time, the detection accuracy of the model can be significantly improved.

The previous works have extensively explored these two types of information. Some models introduce part-level features [2,3] or fuse lower-level feature maps [4,5] to enhance the representation ability of the inner detail features. There are also some methods that leverage expanding the region of interest [6,7] or fusing higher-level semantics layers [8,9] to strengthen the ability of the detectors to perceive the surrounding context semantics of the object. However, none of the existing detectors are able to simultaneously and effectively use both types of information. This is mainly due to the fact that the current models [10,11] are all based on a single-prediction mechanism, *i.e.*, each region of interest (RoI) is predicted once based on one type of feature, as shown in Fig. 1(a). Such a single detection branch model fails to employ multiple types of features on a model at the same time. This motivates us to design a non-single-prediction model to incorporate these two types of information effectively.

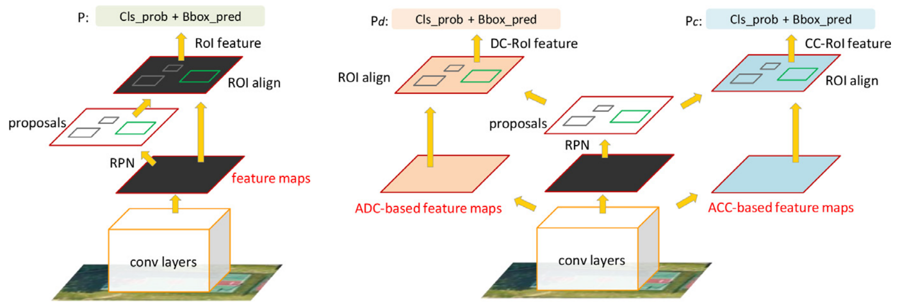
In this paper, we first propose an effective feature fusion method, named adjacent feature compensation (AFC), which leverages inherent adjacent features to perform two types of feature compensations. Our AFC includes adjacent detail compensation (ADC) and adjacent context compensation (ACC) to achieve different feature representations of these two types of information. Through our AFC, we can construct two types of features on the same model at the same time.

Then, we build the first non-single-prediction model in the community of object detection, *i.e.*, Dual Detection Branch Network (DDBN), as shown in Fig. 1(b). Each detection branch contains one type of feature and a specialized detection head based on this type of feature. Our detector with dual detection branches is able to interpret each RoI from the perspectives of detail features and context semantics, and then provide two different semantic predictions. Finally, a better prediction will be obtained for each RoI via our customized testing strategy (Voting Decision Strategy, see Algorithm 1), thereby improving the performance of object detection.

Training our dual detection branch network is very challenging. This is because the ground truth of the common object detection

[*] Corresponding authors.
E-mail addresses: jianhuizhao@whu.edu.cn (J. Zhao), yuanzywhu@gmail.com (Z. Yuan).

https://doi.org/10.1016/j.patcog.2021.108315
0031-3203/© 2021 Elsevier Ltd. All rights reserved.



(a)

(b)

Fig. 1. (a) Traditional single detection branch model (*e.g.*, Faster RCNN) leverages one type of feature to perform single-prediction. (b) Our dual detection branch model leverages adjacent detail compensation (ADC) and adjacent context compensation (ACC) to produce two types of features, and then performs different semantic predictions (P_d and P_c) for each RoI based on both types of features respectively.

datasets (like DOTA [12], MS-COCO [13], and Pascal Voc [14]) only includes the coordinates of bounding boxes and the corresponding category labels, without any branch-labels. However, our DDBN has two detection branches, so how to perform loss regression on our dual detection branches becomes a tricky problem. Obviously, adding branch-labels manually is a time-consuming and labor-intensive task, and excessive manual participation also greatly limits the usability of our model. Therefore, we do not adopt such a method with branch-labels for training.

For each RoI, our DDBN generates two predictions, which means that when performing loss calculation with the corresponding ground truth, we will obtain two loss-values. Now, there are two processing methods. One is to directly back-propagate the losses like the traditional regression method [10], *i.e.* these two losses are back-propagated to the corresponding detection branches respectively for loss learning. And another is to choose one of the detection branches for loss learning, which is similar to selective regression method (like MCL [15], FSAF [16]). However, we found that both these loss regression methods fail to give full play to the performance of our dual detection branches.

Considering the diversity of objects, for some objects that mainly depend on context information, the context semantics is conducive to enhancing the model's ability to identify these objects; for some objects with ample detail information, the detail feature is beneficial to detect these objects; for some objects that contain both context and detail information, both detection branches can improve detection accuracy. Therefore, to improve the ability to recognize various objects, the learning method of our DDBN also needs to vary from objects to objects. We customize a diversity enhancement strategy (DES) for training our DDBN. Since the dual detection branches conduct different loss regressions based on different samples during training, our dual detection branches can provide semantic diversity predictions in testing.

The contributions of the proposed DDBN can be summarized as follows:

1) We proposed a simple but effective feature representation method, adjacent feature compensation (AFC), to provide both detail and context information simultaneously.
2) We construct the first non-single-prediction model, Dual Detection Branch Network (DDBN), in the community of object detection. The dual detection branch of DDBN can make full use of the two types of feature generated by AFC, so that our detector can better grasp the detail feature and the context semantic of the object to enhance the object detection performance.
3) We customize a specialized diversity enhancement strategy for our DDBN, which can train each of our dual detection branches to be a detection expert on subsets of the dataset and then provide semantic diversity predictions.
4) Our DDBN has brought significant accuracy improvements on multiple benchmark datasets, which shows the generality and superiority of our model.

In our DDBN, both the two feature compensation methods and the dual detection branches are based on the parallel design, which has little effect on the inference time. Experiments show that our DDBN has almost the same inference time as the single detection branch model with single-prediction, but it obtains a significant accuracy improvement from dual predictions. To the best of our knowledge, our DDBN is the first dual/multi-prediction approach based on a single model. As a first multi-prediction model, the design of our model and the customized training strategy bring some new insights to the object detection community.

2. Related work

In this section, we briefly review related feature representation, detection pipeline, and learning methods.

2.1. Feature representation

Feature representation has always been a very important research task in machine learning. Object detection performance also depends heavily on the ability of features to represent the region of interest. Before the popularity of deep learning, the scale-invariant feature transform (SIFT) [17] and the histogram of oriented gradients (HOG) [18] methods were the main feature representation methods in object detection. The SIFT feature can effectively deal with the changes in scaling, panning, and rotation. And the HOG leverages the gradient intensity and distribution of gradient direction to represent the object of interest.

Since Hinton and Salakhutdinov [19] have made great progress in the field of deep learning, the ability of feature representation based on deep models has also improved significantly. Currently, in the field of object detection, there are mainly two ways to enhance the ability of feature representation. One is to construct or use an advanced backbone with strong ability of feature representation, such as GoogleNet [20], ResNet [21] and DenseNet [22] etc. The other is based on the advanced backbone to construct a feature layer with richer semantics via feature fusion methods [23–25]. Just like ION [26], PANet [27], NAS-FPN [8], and EfficientDet [28] methods, the ability of feature representation is greatly enhanced by various feature fusing methods. However, since the conventional single-prediction mechanism only accepts one type of feature, these advanced feature fusion methods [26] about the detail and context information ultimately employ one type of fused feature to describe all objects of interest. Obviously, it is difficult to use a type of feature to uniformly represent the objects with different semantics, such as internal rich-detail objects and external context-dependent objects. Therefore, this paper proposes an adjacent feature compensation (AFC) method to construct two types of features at the same time for conducting different feature representations separately.

2.2. Detection pipeline

The first popular deep learning based detection pipeline framework is based on RCNN [29], which originally extracted region features and then input the features into a linear SVM for classifying. To achieve higher detection speed, Girshick et al. proposed Fast RCNN [30] which shared the computational cost among candidate boxes in the same image and introduced a novel RoI-pooling operation to extract feature vectors for each region proposal. For further improving the speed of detection, an upgraded version of Fast RCNN was introduced in [10], the region proposal module and the classification module were combined, which can share the backbone of the Faster R-CNN framework. At present, there are also some methods (like YOLO [31] and SSD [32]) to remove the step of region proposals in the pipeline directly, and leverage the predefined anchor proposals to directly perform object classification and boundary regression. Almost all deep learning detectors [11,33] are based on the above detection pipelines. However, such pipelines only provide a prediction for each RoI (*i.e.*, each anchor/proposal). In order to obtain higher detection performance, our DDBN interprets each RoI separately from two different semantic perspectives: detail and context. That is, our detector will provide two different semantic predictions for each RoI. Therefore, compared with the conventional single-prediction mechanism (one detection pipeline, one prediction for each RoI), this paper designs the first multi-prediction architecture to achieve better detection performance.

2.3. Learning method

The learning method of the model is also a key factor that enables the detector to obtain high detection performance. Typically, by pre-training on the ImageNet classification dataset and fine-tuning on the target object detection dataset [29], the detection accuracy of the model has been greatly improved compared with that without pre-training. For avoiding internal covariate shift and accelerating deep network training, Batch normalization [34] was introduced to normalize each layer input of each mini-batch. To overcome the issue of sample imbalance between categories, some researchers [35] have conducted research on loss processing for learning better feature representation. To avoid scale-imbalance problem in anchor matching strategy, the work [36] proposed scale-balanced loss to enhance detection ability of small objects.

Although the above methods aim at training detectors with higher detection performance, they all directly perform loss regression for the traditional single output models. Since our models have dual outputs, we refer to the training method of multiple outputs (structured output) algorithms, such as MCL [15,37], FSAF [16]. The essence of these training methods is selective regression. However, whether the direct regression method or the selective regression method is adopted, it is not optimal for our model. Therefore, according to the characteristics of our dual detection branch model, we customize a special loss regression method, diversity enhancement strategy, for training our detector to obtain better detection performance. Experiments show that our training method can indeed effectively improve the detection performance of our model.

3. Proposed framework

As illustrated in Fig. 2, we propose a novel object detection framework (DDBN) to obtain higher detection performance. In this framework, we construct two types of adjacent feature compensations (including compensation of higher/context semantics and compensation of lower/detail features) for each scale to bring about the diversity of semantic representation. And then, the different RoI features (DC-RoI feature and CC-RoI feature) of the same RoI are input into the dual detection branches to perform semantic diversity predictions. We are going to describe the details in the following subsections.

3.1. The overall architecture

Note that many object detection frameworks [11,38] that achieve high performance are based on residual networks. Considering the excellent ability of feature extraction from the residual networks, we adopt the ResNet-101 [39] network as the backbone. To specify, we extract information from conv1 to conv5_3. We also convert the layers of the average pooling, FC and softmax of ResNet-101 to convolutional layers by subsampling their parameters, and these converted convolutional layers can generate more abstract semantic which is used to detect the larger object. Then we select conv2_3, conv3_4, conv4_23, conv5_3 and conv_fc as the detection layers. These layers are used as input to the Region Proposal Networks (RPN) [10] to generate multi-scale proposals which are also considered as regions of interest (RoI).

By reusing the feature maps of the backbone, our adjacent feature compensation (see Section 3.2) will generate two types of richer semantic features. And then the RoIs are mapped into these two types of features for extracting different semantic RoI features as inputs of our dual detection branches. Our dual detection branch network (see Section 3.3) will provide two different semantic interpretations (*i.e.*, two predictions: P_d and P_c) for each RoI. In order to further develop the detection performance of our detector, during training, we design a diversity enhancement strategy (DES, see Section 3.4) to make the branch parameters only learn the training samples that the input features of the branch are good at. An overview of our DDBN and customized training method is shown in Fig. 2

3.2. Adjacent feature compensation

Generally, the adjacent feature layers contain complementary information [4] of each other, *i.e.*, the lower layer feature map with smaller receptive field contains the detail information of the higher layer, and the higher layer feature map with bigger receptive field contains the context information of the lower layer. Therefore, we adopt two methods of adjacent feature compensations:

- **Adjacent Detail Compensation (ADC)**
  ADC uses the lower adjacent feature layer with detail features to compensate for the current detection layer, as shown in Fig. 3(a). Then 14 × 14 RoI align [40] is conducted on this compensated feature map to extract the corresponding detail compensation based RoI feature (DC-RoI feature) for the detail detection branch, which makes the branch deal with the objects with ample detail information well.
- **Adjacent Context Compensation (ACC)**
  ACC uses the higher adjacent feature layer with context semantics to perform feature compensation of the current detection layer, as shown in Fig. 3(b). Then 7 × 7 RoI align is conducted on this compensated feature map to extract the corresponding context compensation based RoI feature (CC-RoI feature) for the context detection branch which can well explore the objects of rich context semantics.

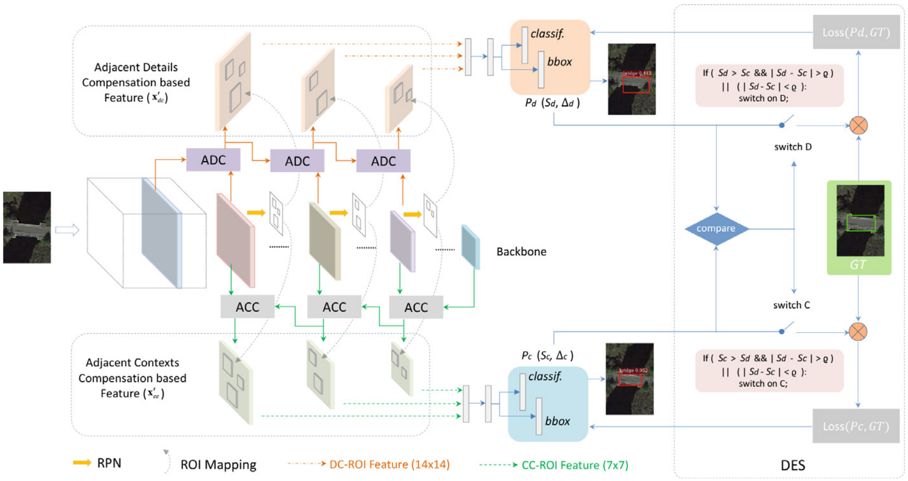

Adjacent Details Compensation based Feature ($\mathbf{x}'_{dc}$)
ADC
ADC
ADC
Backbone
ACC
ACC
ACC
Adjacent Contexts Compensation based Feature ($\mathbf{x}'_{cc}$)
classif.
bbox
P_d (S_d, Δ_d)
P_c (S_c, Δ_c)
classif.
bbox
bridge 0.412
bridge 0.952
RPN
ROI Mapping
DC-ROI Feature (14x14)
CC-ROI Feature (7x7)
Loss(Pd, GT)
If (Sd > Sc && | Sd - Sc | > ϱ) || (| Sd - Sc | < ϱ): switch on D;
switch D
compare
GT
switch C
If (Sc > Sd && | Sd - Sc | > ϱ) || (| Sd - Sc | < ϱ): switch on C;
Loss(Pc, GT)
DES

Fig. 2. The overall architecture of our dual detection branch model. It consists of ResNet-101 based backbone, Adjacent Feature Compensation based Layers (including Adjacent Details Compensation based Feature and Adjacent Contexts Compensation based Feature), and Dual Detection Branches. Our Diversity Enhancement Strategy (DES) is shown in the right, Switch D and switch C have three states: (1) switch on D only, (2) switch on C only, or (3) switch on D and C at the same time. Only when the switch is connected, the loss between the prediction and the ground truth is calculated, and then the calculated loss can be back-propagated to the corresponding detection branch at the time of regression.

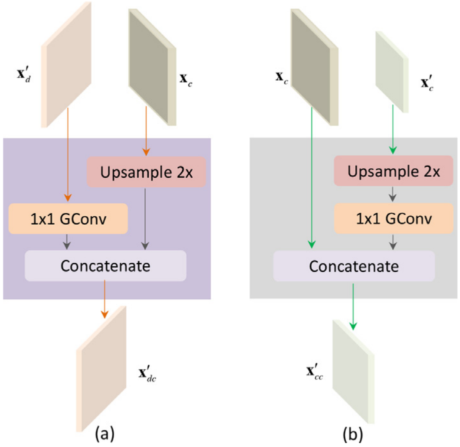


Fig. 3. Illustration of generating compensation feature: (a) Adjacent detail compensation (ADC), (b) Adjacent context compensation (ACC).

To avoid the compensation feature being over-weighted and the main information of object at the current detection scale being affected heavily, we introduce a variable λ to adjust the ratio of compensation feature to current feature based on the channel number of compensation feature layers, *i.e.*,

$$\lambda = \frac{N_{comp}}{N_{cur}} \tag{1}$$

where N_{comp} and N_{cur} are the channel numbers of compensation feature layer and the current feature layer respectively.

To control the channel number of the compensation feature with the least cost, we use Pointwise Grouped Convolution (1 × 1 GConv) [41]. To avoid information loss, we adopt upsampling for unifying the spacial size of feature maps of the compensation layer and the current layer. Let $\mathbf{x}_d^{'}$, $\mathbf{x}_c$, and $\mathbf{x}_c^{'}$ denote the adjacent detail feature map, the feature map of current detection layer, and the adjacent context feature map, respectively. Then by adjacent detail compensation, we get the feature map:

$$\mathbf{x}'_{dc} = F_{cat}\,(\,F_{pg}(\mathbf{x}'_d),\ F_{up}(\mathbf{x}_c)\,) \tag{2}$$

where F_{cat}, F_{pg} and F_{up} represent the operations of the Concatenation, the Pointwise Grouped Convolution, and the Upsampling respectively. Then by adjacent context compensation, we get the feature map:

$$\mathbf{x}'_{cc} = F_{cat}\,(\,\mathbf{x}_c,\ F_{pg}(F_{up}(\mathbf{x}'_c))\,) \tag{3}$$

Therefore, for each detection layer, we use feature layers with smaller receptive fields for detail compensation, and use feature layers with larger receptive fields for context compensation. In this way, two types of features based on different compensation information are obtained to conduct different feature representations on the same scale object.

3.3. Dual detection branches

With the extracted regional features via RoI operations (like RoI pooling [10] and RoI align [40]), most of the existing object detectors [10,11] just input the RoI feature into a single detection branch to predict both bounding boxes and corresponding class probabilities. These approaches have achieved good performance in detecting objects with ample detail information or rich context information. However, for simultaneously detecting these two types of objects through a single detection branch, the parameters of this single detection branch need to learn a trade-off between the features of ample details and rich contexts, which prevents the detector from achieving its optimal detection performance for both types of objects at the same time.

In this paper, we construct dual detection branches, which well incorporates the detail detection branch and the context detection branch. The detail detection branch is dedicated to exploring internal feature differences of the objects, while the context detection branch focuses on the impact of surrounding semantics on the detected targets. The dual detection branches help our detector to obtain the optimal detection performance of both types of objects at the same time.

Note that during forward inference, each region proposal (*i.e.* RoI) generated by the RPN [10] will be mapped to both the ADC-based feature map and the ACC-based feature map. Then we leverage RoI Align operator to extract the corresponding 14 × 14 detail compensation based RoI feature (DC-RoI feature) and 7 × 7 context compensation based RoI feature (CC-RoI feature) respectively. The 14 × 14 DC-RoI feature is input to the detail detection branch, and the 7 × 7 CC-RoI feature is used as the input of the context detection branch, then the dual detection branches will produce different semantic predictions, as shown in Fig. 2. Therefore, our model outputs two predictions ($P_d(s_d, \Delta_d)$ and $P_c(s_c, \Delta_c)$) for each RoI, where each prediction contains both prediction scores s and bounding box offsets Δ of the RoI.

3.4. Diversity enhancement strategy

As mentioned in Section 3.3, our model provides two predictions for each RoI, which also means that when we compare these two predictions with ground truth (GT), two loss-values will be generated. Although we can also directly back-propagate the two losses to their respective detection branches as done in traditional regression methods [10], it is not optimal for our model. Therefore, in order to further improve the detection ability of our DDBN, we customize a diversity enhancement strategy for training our DDBN. At the training stage, there are the following two cases to be considered.

1) The two prediction scores of the RoI differ greatly, *i.e.*, the prediction scores differ by more than ϱ. It is obvious that the detection branch with a higher prediction score is better at predicting the object of the current region proposal, while the other detection branch with a lower prediction score is not good at predicting the current region object. To make two detection branches play a greater role in the field that they are good at, without being affected by the prediction that they are not good at, we do not allow the detection branch with a lower prediction score to learn from the loss, which is also good for parameter stability of this detection branch. Thus, we only conduct loss calculation and back-propagation on the detection branch with a higher prediction score. This selective learning method can promote the diversity of semantic representation of each branch. Selecting only one branch for learning is beneficial to each branch to be trained as a specialist on one particular data subset.
2) If the difference between the two prediction scores is smaller than ϱ, we argue that both branches are all good at predicting the current RoI object, and this small difference in prediction scores may come from somewhat randomness. Therefore, we let both branches calculate the loss and perform back-propagation. Meanwhile, the operation of allocating losses to two detection branches can increase the number of assigned samples for each detection branch, which helps reduce the possibility of overfitting. Because if one branch is always selected for loss regression, it may lead to few samples allocated to another branch, which may cause this branch to overfit to these few assigned samples. Furthermore, when our model learns about these two losses, our model has to accept a double penalty, especially for the backbone, which will break the balance of the training samples. Therefore, in this case, we average the two loss-values.

In summary, the loss function for our diversity enhancement strategy can be defined as:

$$\mathcal{L}(D) = \sum_{i}^{N} \text{loss}\,(y_i, f_d(x_i), f_c(x_i)) \tag{4}$$

where N is the number of RoI in training images and the corresponding loss of the RoI (x_i) can be defined as:

$$loss(y_i, f_d(x_i), f_c(x_i)) = \begin{cases} \min\limits_{m\in[d,c]} l(y_i, f_m(x_i)) & |s_d - s_c| \geqslant \varrho \\ \sum_{m\in[d,c]} \frac{1}{2} l(y_i, f_m(x_i)) & |s_d - s_c| < \varrho \end{cases} \tag{5}$$

where the y_i is the corresponding ground truth of the RoI (x_i), and the $f_d(x_i)$ and $f_c(x_i)$ are the prediction results via the detail inference function (f_d, *i.e.*, the detail detection branch) and the context inference function (f_c, *i.e.*, the context detection branch) respectively. The multi-task loss (*i.e.*, $l(.)$, defined in [30]) is used to calculate the difference between the prediction and the ground truth. And the s_d and s_c are the prediction scores from the detail detection branch and the context detection branch, respectively.

Discussion: It is worth mentioning that the design of our loss function encourages the dual detection branch network to generate different semantic interpretations for each RoI, which is mainly due to the different detection branches learning different training samples. Furthermore, compared to traditional regression methods, our customized regression strategy not only makes our network parameters more stable but also makes it unnecessary for our network parameters to learn a trade-off between categories with large semantic differences. This is beneficial to the convergence of the network in training, and also helps to enhance the detection performance of each branch in our detector.

For Eq. 5, when ϱ is 0, our loss function is a selective loss function, that is, only one detection branch is selected for loss regression. When ϱ is 1, our loss function is a mean method of multiple losses in traditional regression. We find that regressing our DDBN with the selective loss function (*i.e.*, ϱ equals 0) can bring about an improvement in detection accuracy. However, it is still not optimal. When ϱ takes a value between 0 and 1, it can have a better detection performance. It is worth noting that, in the initial stage of training, to avoid randomly assigning training samples to different detection branches, we make the parameter (ϱ) of our DES gradually decay from 1 to the value (like 0.1) we set, the decay ratio of the used natural exponential method [42] is set as 0.5.

In the testing stage, with an input image, our DDBN outputs two prediction sets, one prediction set comes from the detail detection branch and the other one is from the context detection branch. Instead of directly selecting a prediction set as the output of this image, we leverage a voting decision strategy (VDS) to automatically produce a better prediction as the final prediction for each RoI. Similar to training, we also handle two predictions for each RoI in two cases. 1) When the predicted score difference is greater than ϱ, we argue that the prediction with a higher score is more trustworthy. So we choose the prediction with the higher score as the final prediction. 2) When the difference between the two prediction scores is smaller than ϱ, we think that the two predictions are not much different and are both trustworthy. So we take the average of the two predictions. For details, please see Algorithm 1.

Algorithm 1 Voting Decision Strategy.

```
Input: the predictions of $P_d(s_d, \Delta(dx_d, dy_d, dw_d, dh_d))$
  and $P_c(s_c, \Delta(dx_c, dy_c, dw_c, dh_c))$
Output: the final prediction $P_{final}(s_{final}, \Delta(dx, dy, dw, dh))$
  if $|s_d - s_c| \geqslant \varrho$:
  if $s_d > s_c$ :
    $s_{final} = s_d$
    $\Delta(dx, dy, dw, dh) = \Delta(dx_d, dy_d, dw_d, dh_d)$
  else:
    $s_{final} = s_c$
    $\Delta(dx, dy, dw, dh) = \Delta(dx_c, dy_c, dw_c, dh_c)$
else:
  if $s_d > s_c$ :
    $s_{final} = s_d$
  else:
    $s_{final} = s_c$
  $\Delta(dx, dy, dw, dh) = \frac{\Delta(dx_d, dy_d, dw_d, dh_d) + \Delta(dx_c, dy_c, dw_c, dh_c)}{2}$
```

Since dual detection branches' predictions just provide bounding box (bbox) offset, we need to get the final bbox offset (the Δ of final prediction) from Algorithm 1 to adjust the RoI coordinate to

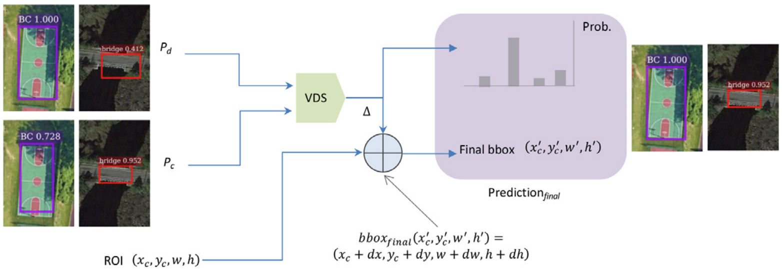


Fig. 4. Demonstration of the testing process and two typical examples of our DDBN. The two predictions are used as inputs to the voting decision strategy (VDS) to obtain the final prediction. The final prediction score is regarded as the confidence in the prediction of the current RoI, and the final bounding box offset $\Delta(dx, dy, dw, dh)$ is used to adjust the original RoI coordinate (x_c, y_c, w, h) to obtain the final bounding box of the object of interest.

produce the final prediction coordinate of RoI. The overall test process based on P_d and P_c is shown in Fig. 4. And Fig. 4 also provides two typical examples that different detection branches are good at. The detail detection branch is better at detecting objects with rich internal features, such as a Basketball Court with ample internal textures. The context detection branch is better at detecting objects with complex surrounding semantics, such as the Bridge as a part of the road, which has a similar appearance to the road, but the surrounding environment of the Bridge is obviously different from that of road.

3.5. Implementation details

The implementation of our DDBN is based on the original FPN [11] in Detectron [1]. Except for our three innovations, other hyperparameters and sub-modules (such as RPN) are largely based on the original FPN. The modifications based on our three innovations are as follows:

1) We replace the Feature Pyramid in FPN with the feature generation method of our AFC, which can produce two types of features as the input of the dual detection branches.
2) Unlike the FPN, which has only one detection head, we use both 14x14 and 7x7 detection heads to perform dual branch detection.
3) We customize the diversity enhancement learning method to replace the traditional direct loss regression method to train our DDBN.

As [29], we utilize network backbone, ResNet101 model, with its publicly available pre-trained model on the ImageNet classification set [43], and then fine-tune on the target detection dataset. We run our approach on a PC machine with an i5-7640X CPU (with 32 GB memory) and two NVIDIA GTX 1080Ti GPUs (with 11 GB memory). At the training stage, we adopt synchronized SGD on 2 GPUs. Due to the large size of some images of datasets, a mini-batch is assigned with 1 image for each GPU. For other hyper-parameters, we set the momentum as 0.9 and the weight decay as 0.0005. The learning rate is 0.005 for the first 480K iterations, 0.0005 for the next 160K, and 0.00005 for the last 80K. The size of 1024 × 1024 is set as the maximum scale of our model input.

4. Experiment and analysis

In this section, we firstly conduct ablation experiments and analyze the effectiveness of the components of our model. Then we further analyze how different categories benefit from different semantic predictions of our dual detection branch model. We also comprehensively evaluate our model performance on three benchmark datasets (DOTA [12], MS-COCO [13], Pascal-VOC [14]) by comparison with the baseline framework (*i.e.*, FPN). Finally, we compare the accuracy and inference time with the state-of-the-art detectors on the common COCO dataset to evaluate the advancedness of our DDBN.

Table 1
The effects among the feature compensation methods, the number of detection branch, and the training methods.

models mAP ϱ	1	0	0.05	0.1	0.15	0.2
WFC-S	70.1	-	-	-	-	-
WFC-D	70.9	71.3	71.3	71.5	71.3	71.0
ADC-S	71.4	-	-	-	-	-
ADC-D	72.2	72.7	73.1	73.3	73.0	72.8
ACC-S	71.2	-	-	-	-	-
ACC-D	71.9	72.4	72.5	72.7	72.2	72.2
AFC-D	73.1	74.7	75.3	75.8	75.1	74.9

Due to the special imaging environment of remote sensing images, the remote sensing images (DOTA) are not as clear as natural images (MS-COCO and Pascal-VOC) [12]. The prediction of the less clear image is usually accompanied by greater uncertainty, so the capture of detail and context information is more beneficial to improving the accuracy of remote sensing images (in Table 3, DOTA obtains the best accuracy improvement), which can better highlight the roles of different detection branches in our DDBN. Therefore, we select the DOTA dataset for ablation experiments and analysis.

For the training set and validation set of DOTA-v1.0, the original images with ground truth are provided publicly. However, the testing set only provides original images, thus we should send our predicting results of the testing set to the DOTA-v1.0 server to obtain the detection accuracy (including AP of each category and mAP). Therefore, in our ablation experiments, we use the training set to train and test on the validation set. When comparing with other detection models, we use the training set and the validation set to train and test on the testing set.

The short names for some categories of the DOTA dataset are defined as: BD-Baseball Diamond, GTF-Ground Track Field, SV-Small Vehicle, LV-Large Vehicle, TC-Tennis Court, BC-Basketball Court, ST-Storage Tank, SBF-Soccer Ball Field, RA-Roundabout, SP-Swimming Pool, and HC-Helicopter.

4.1. Ablation study

In ablation studies, we evaluate the effectiveness of our model from three perspectives: feature compensation method, the number of detection branches, and training method.

1) For feature compensation, we verify 4 compensation methods, that is, without feature compensation (**WFC**, as a baseline), adjacent detail compensation (**ADC**), adjacent context compensa-

[1] https://github.com/facebookresearch/Detectron

Table 2
The comparison of mAP improvements from different values of the parameter λ.

λ	WFC-D	ADC-D	ACC-D	AFC-D
1	71.49	72.92 (1.43↑)	72.45 (0.96↑)	74.42 (2.93↑)
1/2	71.49	73.32 (1.83↑)	72.71 (1.22↑)	75.78 (4.29↑)
1/3	71.49	73.15 (1.66↑)	72.52 (1.03↑)	75.00 (3.51↑)
1/4	71.49	72.68 (1.19↑)	72.30 (0.81↑)	74.15 (2.66↑)

Table 3
The accuracy comparison of our DDBN with FPN on three different datasets.

models mAP DateSets	DOTA	MS-COCO	PASCAL VOC2012
FPN	75.4	38.8	80.5
Our DDBN	79.3	42.3	83.4
Gain	+3.9	+3.5	+2.9

tion (**ACC**), and adjacent feature compensation (**AFC**). Note that our **AFC** includes **ADC** and **ACC**.

2) For the number of detection branches, we verify the difference between the baseline (traditional single detection branch, denoted as **-S**) and the dual detection branches (denoted as **-D**).

3) For the training method, we perform different loss learning strategies by controlling the variable ϱ in Eq. 5. When ϱ equals 1, it is the traditional regression method which directly back-propagates the losses to corresponding detection branches. When ϱ is 0, it is the selective regression method, that is, there is one detection branch to learn the loss in each time. When ϱ takes a value between 0 and 1, it is our diversity enhancement strategy.

Now, we denote WFC-S, ADC-S, and ACC-S as single detection branch models based on WFC, ADC, and ACC, respectively. And WFC-D, ADC-D, ACC-D, and AFC-D are denoted as dual detection branch models based on WFC, ADC, ACC, and AFC, respectively. Note that, since AFC generates two types of features, there is only one state with dual detection branches based on AFC (*i.e.*, AFC-D), without the situation with a single detection branch based on AFC (*i.e.*, no AFC-S). And the WFC, ADC, ACC only output one type of feature, for performing dual branches detection (*i.e.*, WFC-D, ADC-D, ACC-D), we conduct the operations of both 7 × 7 RoI Align and 14 × 14 RoI Align on the same feature, and then output 7 × 7 RoI features and 14 × 14 RoI features into the dual branches to conduct ablation detection.

Adjacent feature compensation is useful. By observing Table 1, we can get the following conclusions: (1) Compared to WFC, no matter how many detection branches, our ADC and ACC can bring about 1% improvement in accuracy (*i.e.*, WFC-S vs ADC-S and ACC-S, WFC-D vs ADC-D and ACC-D), which means that our two feature compensation methods (ADC and ACC) are useful. (2) The AFC with both ADC and ACC has greater accuracy improvements, especially that the dual detection branch models based on both diversity enhancement strategy and AFC can significantly improve our detection accuracy (when ϱ is 0.1 the accuracy of AFC-D is over 4% higher than that of WFC-D). Therefore, the adjacent feature compensation method can bring more diverse semantics input for our dual detection branches, which is beneficial to generate semantic diversity predictions for improving the detection performance of our detector.

To further analyze the influence of adjacent feature compensation, we conduct exploratory experiments on the ratio λ (defined in Section 3.2) of the feature compensation. We set λ with different values for searching the best parameter to explore which feature compensation ratio can make our dual detection branch framework achieve the best performance. Since WFC-D does not carry out feature compensation, the compensation ratio has no impact on WFC-D, and the mAP of WFC-D has no change. Thus, the WFC-D is considered as baseline. From Table 2, we can observe that our framework achieves the best performance when λ is 1/2, while too much or too little compensation of features is less conducive to the improvement of accuracy. Therefore, 1/2 is set as the default value of λ in our experiments.

Dual detection branches are effective. In the case of a single detection branch (**-S**), it is not possible to perform which branch is selected for loss learning, so the models with a single detection branch have no values when $\varrho < 1$. From Table 1, by comparing traditional single detection branch (**-S**) and our dual detection branches (**-D**) on different features under the same training conditions ($\varrho = 1$), we find that dual detection branches are indeed conducive to the improvement of the final prediction accuracy (WFC-S vs WFC-D, ADC-S vs ADC-D, ACC-S vs ACC-D). This can be easily understood as that the dual plausible predictions are made for the same RoI, and then a better prediction is proposed as the final prediction, which obviously exceeds the situation with only one prediction. This is the charm of our dual predictions.

Diversity enhancement strategy is promising. In training the dual detection branch models (WFC-D, ADC-D, ACC-D, AFC-D), we perform different loss regression methods, including traditional regression method (ϱ=1, directly back-propagate the losses of dual predictions to corresponding detection branches), selective regression method (ϱ=0, choose a detection branch for the loss back-propagation) and our customized regression method ($\varrho \in (0, 1)$, *i.e.*, diversity enhancement strategy). The comparison among different training methods of AFC-D in Table 1 indicates that our customized regression method achieves the best detection accuracy (ϱ=0.1 : 75.8% is better than ϱ=1 : 73.1% and ϱ=0 : 74.7%), which means that our training method (DES) is more beneficial to improve the detection performance of our model. Moreover, by observing different feature compensation methods with different training methods, we find that when ϱ changes from 1 to 0.1, the accuracy of WFC-D improves by less than 1%, but for our AFC-D method, it improves over 2%. This shows that the diversity enhancement strategy specially designed for training our DDBN can better take advantage of different semantic interpretations from our DDBN.

Through the above analysis, we already know that our customized training method can indeed bring accuracy improvement to our DDBN. Now, we visualize the iterative process of the loss to observe the difference between our customized regression method and the traditional regression method. From Fig. 5, we can clearly observe that 1) the process of our customized regression is more stable when the learning rate changes in the 480Kth iteration, and 2) the fluctuation range of loss is also smaller than traditional regression. These mean that our training strategy can make our network parameters more stable, which is beneficial to the convergence of the network in training.

Therefore, our diversity enhancement strategy not only effectively improves the detection performance of our model but also makes our detector to perform more stable parameter learning.

4.2. Further analysis

To further explore how different objects benefit from different semantic predictions, that is, which type of object is suitable for which detection branch, we provide the following experiments. Using a WFC detector as a baseline, a single detection branch is applied to ADC and ACC respectively. By comparison, we can know which category is benefited from which branch. Then, we observe whether AFC based dual detection branches can get better detection results from different detection branches. From Fig. 6, we can draw the following conclusions:

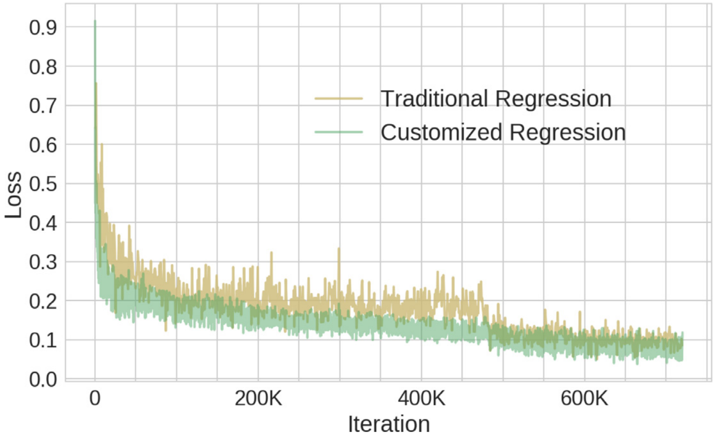


Fig. 5. The comparison of different loss regression methods, Traditional Regression (ϱ=1) and Customized Regression (ϱ=0.1), on our DDBN.

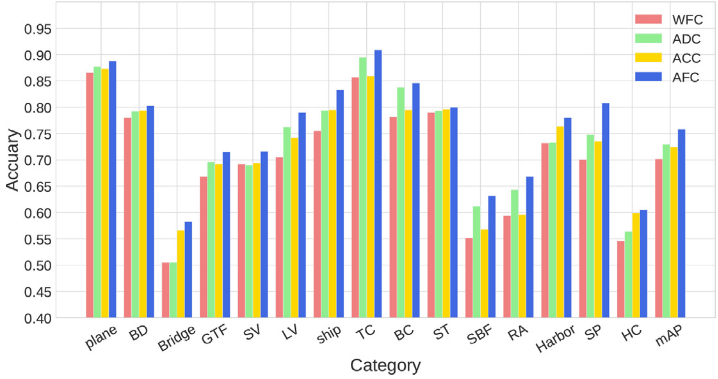


Fig. 6. The comparison of different detection results from detectors with different detection branches.

1) Compared with the WFC detector, there is a large performance improvement using ADC for the categories with ample detail information, *e.g.* TC and BC.
2) Detector based on ACC has better detection ability when the detected categories depend on rich context information, *e.g.* Bridge and Harbor.
3) Since the category of ST has a simple appearance, without rich detail information and context information, using different detection branches will not significantly improve performance.
4) Our AFC based detector has the best results in all categories, which means our dual detection branch model can achieve better predictions from different detection branches via our VDS (described in Section 3.4).

4.3. Improvement based on the baseline

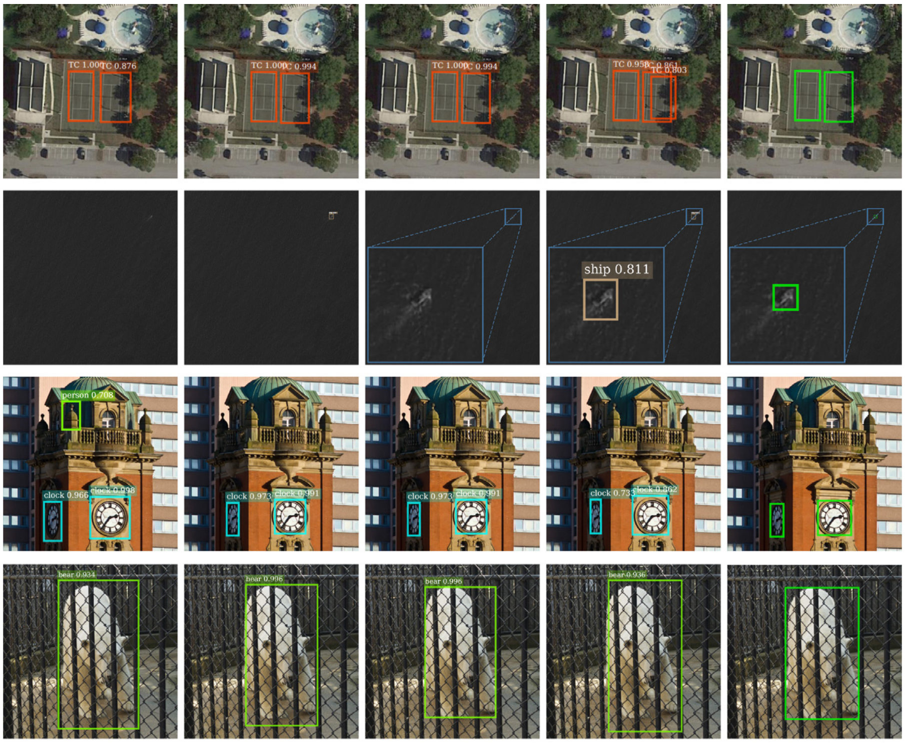
As described in Section 3.5, our DDBN method is developed on ResNet-101-FPN, so the FPN method [11] can be seen as a baseline for a comprehensive comparison in Fig. 7 and Table 3.

As shown in the first row of Fig. 7, since the Tennis Court (TC) has more internal texture information, our detail detection branch (DDBN-P$_d$) has better detection results, which benefits from the better detail representation ability of our ADC feature. Moreover, through careful observation, we find that in the detection of the right Tennis Court affected by tree shadows, our detail detection branch (DDBN-P$_d$) can more accurately locate the boundaries of the Tennis Court than FPN. Therefore, the detail detection branch has a stronger detection ability to the object with ample detail information.

In addition to VHR (very high resolution) remote sensing images, there are also low-resolution blurred SAR images. In the second row of the Fig. 7, there is a less obvious small ship, which can be detected by the context detection branch (DDBN-P$_c$), but neither the detail detection branch (DDBN-P$_d$) nor the baseline method (FPN) can do this. This is mainly because the water waves generated by the ship's running is also a kind of context information, which increases the probability of it being recognized by our DDBN-P$_c$.

From the improvement of the detection results of the above two categories, we can easily get a conclusion that the design of our dual detection branch model based on context and detail is reasonable. Furthermore, the two clues (detail information and context information) grasped by our DDBN can not only enhance the prediction ability of each object, but also suppress the generation of false-positive predictions. As shown in the third row of Fig. 7, since the green region includes an object similar to a person's head, FPN predicts the region as a person, nevertheless, our DDBN does not make a wrong prediction for this area. Meanwhile, we also find an interesting phenomenon, the bounding box of the detail detection branch is very closer to the target object or even slightly smaller than the object (like the DDBN-P$_d$ of the clock). However, the bounding box from the context detection branch is looser, which can be easily observed in the detection results of the Bear and the ship.

Based on the VDS (described in Section 3.4), our DDBN always obtains a better prediction from the different predictions of the dual detection branches. For the right TC in the first row, the predictions of DDBN-P$_c$ have lower prediction scores, thus the prediction of DDBN-P$_d$ is chosen as a final prediction of DDBN. And for the case where both detection branches have good detection capabilities, our VDS can produce a better prediction that has a more suitable bounding box than FPN, which can be clearly seen from



(a) FPN (b) DDBN (c) DDBN-P_d (d) DDBN-P_c (e) Ground Truth

Fig. 7. Comparison of detection results from FPN and our DDBN method on the DOTA dataset (row 1&2) and MS-COCO dataset (row 3&4); (a) detection results of FPN method; (b) detection results of DDBN generated by VDS based on the results of (c) and (d); (c) detection results of the detail detection branch in our DDBN (DDBN-P_d); (d) detection results of the context detection branch in our DDBN (DDBN-P_c); (e) the corresponding ground truth. The predictions are displayed with scores greater than 0.7.

the Bear in the fourth row. In short, whether our VDS chooses a better prediction from DDBN-P_d and DDBN-P_c (like the cases of the TC, the ship, and the Clock), or generates a better prediction based on these two predictions as the DDBN detection result (like the cases of the Bear), it enables our DDBN to obtain better detection results than FPN.

Through the comparison of the detection results mentioned above, we know that our DDBN model has a better detection ability. To further compare our model with the baseline method, we quantify the comparison in three different scenarios (*i.e.*, three different datasets: DOTA [12], MS-COCO [13], Pascal-VOC [14]). As shown in Table 3, our DDBN achieves consistent accuracy improvements, which demonstrates the advanced detection ability of our detector and its generality in different scenarios. On the other hand, we also find that the accuracy improvements of our model are more obvious in the DOTA dataset and COCO dataset. We infer that (1) For the DOTA dataset, remote sensing images are not as clear as natural images due to the imaging environment. This means that remote sensing images are more dependent on various clues (detail or context information) to enhance the recognition ability of objects. And this is what our model is good at, so our DDBN can bring more obvious accuracy improvement on remote sensing images. (2) For the COCO dataset, because it has more categories (80 categories), the traditional single detection branch must learn the semantics of 80 categories at the same time, which makes the parameters of this single detection branch have to learn a tradeoff among 80 categories. However, the dual detection branches of our model can effectively alleviate this dilemma.

4.4. Comparison with mainstream detectors

Comparison on DOTA. To verify the detection performance of our DDBN, we first conduct a comparison of the detection accuracy with mainstream detectors on DOTA-v1.0 dataset. These mainstream detectors include popular detectors (Deformable FR-H [24], RetinaNet [38], IoU-AD RCNN [44], SNIPER [45]), also include some current SOTA methods (CSL [46], BBAVectors+rh [47]). As shown in Table 4, it is easy to find that our DDBN achieves the best detection performance on mAP, and our model has the highest accuracy in 9 geological categories we find that in 15 classes. This is mainly due to the better feature representation ability of our model and the dual-prediction ability based on semantic diversity.

Comparison on COCO. To further evaluate the overall performance of our DDBN, we compare it with several mainstream methods on the common object detection datasets (MS-COCO) in Table 5, including classical detection models (SSD [32], FPN [11], RetinaNet [38], and Mask R-CNN [40]), also including the advanced methods (FCOS [48], RefineDet [35], Libra R-CNN [49], SWN [50], DETR [51], and CBNet [52]). At the same time, we also list some other methods based on information compensation, such as DP-FCN [2], MPNet [23], CoupleNet [5], ION [26], EfficientDet [28] and R-FCN [53]. In training, we use the same training hyperparameters of FPN (implementation details described in Section 3.5), like, 720K iterations, 0.9 momentum, and the weight decay of 0.0005. And for the extra hyperparameters about our DDBN, we set the optimal compensation ratio (λ) as 1/2, and ϱ as 0.1 for our DES and VDS.

Accuracy. From Table 5, our DDBN surpasses other object detection methods in detection accuracy (*i.e.*, AP). Even compared with the state-of-the-art methods (SWN, DETR, CBNet), our model also has better detection accuracy. Meanwhile, because our model is based on detail and context information compensation, we also compare with other related information compensation methods, such as details-based DP-FCN and CoupleNet, MPNet, ION and EfficientDet based on comprehensive feature fusion, and context-

Table 4
Detection accuracy comparison with the mainstream detectors on the dataset of DOTA-v1.0.

Detectors	backbone	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
Deformable FR-H [24]	-	86.5	77.5	42.7	64.4	67.6	63.6	77.9	90.3	77.8	75.4	52.1	56.8	68.9	62.0	54.9	67.9
RetinaNet [38]	ResNet101	87.7	79.2	50.5	74.1	66.0	76.2	83.4	89.5	79.8	83.3	62.2	64.3	78.3	74.8	49.4	73.3
IoU-AD RCNN [44]	-	88.6	80.2	53.2	66.9	76.3	72.6	84.0	90.7	81.0	76.2	57.1	66.7	74.1	55.4	56.9	72.7
SNIPER [45]	ResNet101	88.2	79.7	51.0	74.6	66.5	76.7	83.9	90.4	80.3	83.8	62.7	64.8	78.8	**75.3**	49.9	73.8
CSL [46]	ResNet152	**90.2**	**85.5**	54.6	75.3	70.4	73.5	77.6	90.8	86.2	86.7	69.6	68.0	73.8	71.1	**68.9**	76.2
BBAVectors+rh [47]	ResNet101	88.6	84.1	52.1	69.6	78.3	80.4	88.1	**90.9**	**87.2**	86.4	56.1	65.6	67.1	72.1	64.0	75.4
DDBN(our)	ResNet101	89.8	85.3	**61.2**	**78.5**	**79.7**	**84.0**	**88.4**	**90.9**	84.3	**87.1**	65.2	**68.8**	**84.9**	81.8	59.5	**79.3**

Table 5
Detection accuracy comparison with the mainstream detectors on COCO test-dev. Note that "ResNet-101-FPN*" here indicates that we use ResNet-101-FPN as basic framework, but our DDBN replaced feature pyramid with our AFC-based dual detection branches. TTA: test-time augmentation, which includes multi-scale testing, horizontal flipping, etc.

Methods	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	time(ms)
Information compensation methods									
DP-FCN2.0 [2]	ResNeXt-101		34.8	54.8	38.4	15.8	37.2	49.0	-
CoupleNet + [5]	ResNet-101		34.4	54.8	37.2	13.4	38.1	50.1	-
MPNet [23]	ResNet-101		33.2	51.9	36.3	13.6	37.2	47.8	-
ION [26]	ResNet-101		33.1	55.7	34.6	14.5	35.2	47.2	-
EfficientDet-D1 [28]	ResNet-101		39.3	58.7	42.0	19.2	45.6	57.1	-
R-FCN [53]	ResNet-101		32.1	54.3	33.8	12.8	34.9	46.1	-
Deformable R-FCN [24]	ResNet-101		35.7	56.8	38.3	15.2	38.8	51.5	-
FPN-based methods									
RetinaNet [38]	ResNet-101-FPN		37.5	57.1	40.3	20.3	42.0	50.5	159
Mask R-CNN [40]	ResNet-101-FPN		38.2	60.3	41.7	20.1	41.1	50.2	133
Libra R-CNN [49]	ResNet-101-FPN		41.1	62.1	44.7	23.4	43.7	52.5	117
FPN [11]	ResNet-101-FPN		38.8	61.1	41.9	21.3	41.8	49.8	124
DDBN (ours)	ResNet-101-FPN*		42.3	62.3	46.3	23.4	46.1	56.8	132
Classical and newest methods									
SSD513 [32]	ResNet-101		31.2	50.4	33.3	10.2	34.5	49.8	-
RetinaNet	ResNet-101-FPN		39.1	59.1	42.3	21.8	42.7	50.2	-
RefineDet512 [35]	ResNet-101		41.8	62.9	45.7	25.6	45.1	54.1	-
FCOS [48]	ResNet-101		41.5	60.7	45.0	24.4	44.8	51.6	-
SWN [50]	ResNeXt-101		40.8	63.1	43.8	23.2	44.0	51.1	-
DETR [51]	ResNet-101		42.0	62.4	44.2	20.5	45.8	61.1	-
Deformable DETR [54]	ResNeXt-101	✓	49.0	68.5	53.2	29.7	51.7	62.8	-
EfficientDet-D7 [28]	EfficientNet-B6	✓	52.2	71.4	56.3	-	-	-	-
CBNet [52]	Triple-ResNeXt152	✓	53.3	71.9	58.5	35.5	55.8	66.7	-
DDBN (ours)	ResNeXt-101-64x4d	✓	53.7	72.2	58.6	37.3	56.9	66.3	-

based R-FCN and Deformable R-FCN. Our model obviously exceeds these information compensation methods. Through the comparison of accuracy, we draw a conclusion that our DDBN can bring stronger detection performance via the semantic diversity predictions of dual detection branches, which is better than the traditional single detection branch models with single-prediction.

Inference. Since our model can be seen as a variant of FPN, for a fair comparison, we choose some FPN-based detectors for time comparison. By comparing the inference time of different models (RetinaNet, Mask R-CNN, Libra R-CNN, FPN, and our DDBN), we find that although we have two detection branches, the parallel design of these dual detection branches reduces the influence of diversity predictions on the detection speed.

All in all, our dual detection branch model can bring obvious improvement in detection accuracy via semantic diversity predictions, but it only requires similar inference consumption of the traditional single detection branch models.

5. Discussion

At first glance, our dual detection branch model with the double number of detection heads seems to be related to the multi-scale detector. In fact, there is an obvious difference. Multi-scale detectors with multiple detection heads are primarily used to detect the different-scale objects/RoIs, rather than providing different semantic interpretations for each RoI as our DDBN does.

In terms of name, it is easy to mistake our architecture as similar to the existing architectures named after two branches [55,56]. However, the key difference is that the existing dual-branch networks attempt to produce two different temporary features. Limited by the conventional single-prediction mechanism, these temporary features have to be ultimately concatenated (fused or cascaded) into one type of feature again. Therefore, the existing dual-branch networks are mainly used to generate temporary features, but our dual detection branch framework focuses on conducting different detection tasks by introducing a multi-prediction mechanism. Similarly, there are some works [8,28] that have some feature generation structures similar to our AFC, such as top-down and bottom-up, these methods also only generate one kind of feature for the detection task. However, our AFC leverages a simple and effective fashion to generate different semantic features for our model to conduct diversity predictions.

Overall, although our model only has two detection branches to make predictions currently, our studies establish the first multi-prediction (more than one prediction for each RoI) model/mechanism, referring to the network design, and the training and testing strategies of the multi-prediction model. Through a lot of experiments and analysis, it has been confirmed that multiple/dual predictions based approach does have better detection accuracy than the conventional single-prediction models. Therefore, there will be a number of works to extend this work in the future.

6. Conclusion and future work

Different from the single-prediction detectors that just operate features to improve the utilization of detail and context information, we present the first multi-prediction framework (DDBN) to use different information to conduct different predictions for improving object detection performance. In our model, we leverage adjacent feature compensation to construct two types of features. Then, we equip a detection head for each type of feature to form dual detection branches. A customized training method, the diversity enhancement strategy, is used to enable our dual detection branches to provide different semantic predictions to obtain a better detection performance. Extensive ablation experiments have fully demonstrated the effectiveness of our DDBN. Compared with the baseline detector, significant performance improvements have been achieved on all three datasets (DOTA, MS-COCO, and PASCAL VOC), which shows that our model is a general and effective detector. Finally, by comparing with the current mainstream detectors, our dual detection branch model is superior to traditional single detection branch frameworks in object detection.

Through extensive experiments, it has been proved that our multi-prediction model with semantic diversity can bring significant performance improvement of object detection. Since each visual task usually has its own network architecture and corresponding loss function, our multi-prediction model can not be directly applied to other computer vision tasks, such as image segmentation. In the future, we will continue to explore how to apply multi-prediction mechanism into different visual tasks.

Disclosure of conflicts of 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.

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.

Acknowledgements

This work was supported in part by the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170 and in part by the National Natural Science Foundation of China under Grant No. 62073248.

References

[1] K. Tian, Y. Xu, S. Zhou, J. Guan, Versatile multiple choice learning and its application to vision computing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6349–6357.

[2] T. Mordan, N. Thome, G. Henaff, M. Cord, End-to-end learning of latent deformable part-based representations for object detection, Int J Comput Vis 127 (11–12) (2019) 1659–1679.

[3] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans Pattern Anal Mach Intell 32 (9) (2009) 1627–1645.

[4] L. Zhang, J. Wu, T. Wang, A. Borji, G. Wei, H. Lu, A multistage refinement network for salient object detection, IEEE Trans. Image Process. 29 (2020) 3534–3545.

[5] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: Coupling global structure with local parts for object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 4126–4134.

[6] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, et al., Crafting gbd-net for object detection, IEEE Trans Pattern Anal Mach Intell 40 (9) (2017) 2109–2123.

[7] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, S. Yan, Attentive contexts for object detection, IEEE Trans Multimedia 19 (5) (2017) 944–954.

[8] G. Ghiasi, T.-Y. Lin, Q.V. Le, Nas-fpn: Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7036–7045.

[9] Y. Kong, M. Feng, X. Li, H. Lu, X. Liu, B. Yin, Spatial context-aware network for salient object detection, Pattern Recognit 114 (2021) 107867.

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[11] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[12] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, Dota: A large-scale dataset for object detection in aerial images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3974–3983.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[14] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int J Comput Vis 88 (2) (2010) 303–338.

[15] A. Guzman-Rivera, D. Batra, P. Kohli, Multiple choice learning: Learning to produce multiple structured outputs, in: Advances in Neural Information Processing Systems, 2012, pp. 1799–1807.

[16] C. Zhu, Y. He, M. Savvides, Feature selective anchor-free module for single-shot object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 840–849.

[17] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int J Comput Vis 60 (2) (2004) 91–110.

[18] Z. Xiao, Q. Liu, G. Tang, X. Zhai, Elliptic fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images, Int J Remote Sens 36 (2) (2015) 618–644.

[19] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[21] T. Li, H. Song, K. Zhang, Q. Liu, Learning residual refinement network with semantic context representation for real-time saliency object detection, Pattern Recognit 105 (2020) 107372.

[22] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[23] S. Zagoruyko, A. Lerer, T.-Y. Lin, P.O. Pinheiro, S. Gross, S. Chintala, P. Dollár, A multipath network for object detection, arXiv preprint arXiv:1604.02135 (2016).

[24] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773.

[25] J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, Z.-Y. Li, Gated CNN: integrating multi-scale feature layers for object detection, Pattern Recognit 105 (2020) 107131.

[26] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.

[27] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.

[28] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.

[29] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[30] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[31] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[33] Q. Lin, J. Zhao, G. Fu, Z. Yuan, Crpn-sfnet: a high-performance object detector on large-scale remote sensing images, IEEE Trans Neural Netw Learn Syst (2020) 1–14, doi:10.1109/TNNLS.2020.3027924.

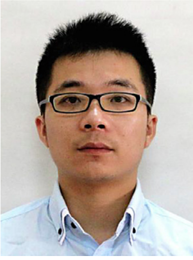
[34] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[35] K. Chen, J. Li, W. Lin, J. See, J. Wang, L. Duan, Z. Chen, C. He, J. Zou, Towards accurate one-stage object detection with ap-loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5119–5127.

[36] K. Shuang, Z. Lyu, J. Loo, W. Zhang, Scale-balanced loss for object detection, Pattern Recognit 117 (2021) 107997.

[37] Q. Lin, J. Zhao, B. Du, G. Fu, Z. Yuan, Mednet: multiexpert detection network with unsupervised clustering of training samples, IEEE Trans. Geosci. Remote Sens. (2021).

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[39] 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, 2016, pp. 770–778.
[40] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[41] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[42] U. Michelucci, Training Neural Networks, Apress, Berkeley, CA, 2018, pp. 137–184.
[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int J Comput Vis 115 (3) (2015) 211–252.
[44] J. Yan, H. Wang, M. Yan, W. Diao, X. Sun, H. Li, Iou-adaptive deformable r-CNN: make full use of iou for multi-class object detection in remote sensing imagery, Remote Sens (Basel) 11 (3) (2019) 286.
[45] B. Singh, M. Najibi, L.S. Davis, Sniper: Efficient multi-scale training, in: Advances in Neural Information Processing Systems, 2018, pp. 9310–9320.
[46] X. Yang, J. Yan, Arbitrary-oriented object detection with circular smooth label, in: European Conference on Computer Vision, Springer, 2020, pp. 677–694.
[47] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, D. Metaxas, Oriented object detection in aerial images with box boundary-aware vectors, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2150–2159.
[48] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9627–9636.
[49] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-CNN: Towards balanced learning for object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[50] Q. Cai, Y. Pan, Y. Wang, J. Liu, T. Yao, T. Mei, Learning a unified sample weighting network for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14173–14182.
[51] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.
[52] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, H. Ling, Cbnet: A novel composite backbone network architecture for object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 11653–11660.
[53] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in neural information processing systems, 2016, pp. 379–387.
[54] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: deformable transformers for end-to-end object detection, arXiv preprint arXiv:2010.04159 (2020).
[55] H. Cao, H. Liu, E. Song, C.-C. Hung, G. Ma, X. Xu, R. Jin, J. Lu, Dual-branch residual network for lung nodule segmentation, Appl Soft Comput 86 (2020) 105934.
[56] V. Vaquero, I.D. Pino, F. Moreno-Noguer, J. Sola, J. Andrade-Cetto, Dual-branch CNNs for vehicle detection and tracking on liDAR data, IEEE Trans. Intell. Transp. Syst. PP (99) (2020) 1–12.



Qifeng Lin is currently pursuing the Ph.D. degree at the School of Computer Science, Wuhan University. His current research interests include object detection, instance segmentation, and remote sensing image and video processing.

Chengjiang Long is currently a Principal Scientist/Tech Leader in JD Tech R&D Center at Silicon Valley (a part of JD.COM) since June 2020. Prior to working at JD.COM, he worked as a Computer Vision Researcher/Senior R&D Engineer at Kitware from February 2016 to April 2020. He also worked as an Adjunct Professor at University at Albany, SUNY from August 2018 to May 2020, and was an Adjunct Professor at Rensselaer Polytechnic Institute (RPI) from Jan 2018 to May 2018. He received the M.S. degree in Computer Science from Wuhan University in 2011 and a B.S degree in Computer Science and Technology from Wuhan University in 2009. He got his Ph.D. degree in Computer Science from Stevens Institute of Technology in 2015. During his Ph.D. study, he worked at NEC Labs America and GE Global Research as a research intern in 2013 and 2015, respectively. To date, he has published 60 papers including top journals such as TOG, T-PAMI, IJCV, T-IP and T-MM, top international conferences such as SIGGRAPH Asia, CVPR, ICCV, AAAI and ACM MM, and owns 1 patent. He is also the reviewer for more than 20 top international journals and conferences. His research interests involve various areas of Computer Vision, Computer Graphics, Multimedia, Machine Learning, and Artificial Intelligence. He is a member of IEEE and AAAI.

Jianhui Zhao received his Ph.D. in Computer Science from Computer School of Nanyang Technological University (NTU Singapore). He is an Associate Professor working in the School of Computer Science, Wuhan University. His current research interests are artificial intelligence, image and video processing, computer graphics.

Gang Fu is working towards the PHD degree at the School of Computer Science, Wuhan University. His research interests include intrinsic image decomposition, image filtering, image enhancement.

Zhiyong Yuan is currently a Professor at the School of Computer Science, Wuhan University, China. His research interests include virtual reality, human-computer interaction, embedded system, internet of things technology, machine learning and pattern recognition.