Cao *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:14
https://doi.org/10.1186/s13673-020-00219-9

Human-centric Computing and Information Sciences

RESEARCH Open Access

An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks

Danyang Cao[1,2*], Zhixin Chen[1] and Lei Gao[1]

*Correspondence: ufocdy@163.com
[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China
Full list of author information is available at the end of the article

Abstract

Object detection methods aim to identify all target objects in the target image and determine the categories and position information in order to achieve machine vision understanding. Numerous approaches have been proposed to solve this problem, mainly inspired by methods of computer vision and deep learning. However, existing approaches always perform poorly for the detection of small, dense objects, and even fail to detect objects with random geometric transformations. In this study, we compare and analyse mainstream object detection algorithms and propose a multi-scaled deformable convolutional object detection network to deal with the challenges faced by current methods. Our analysis demonstrates a strong performance on par, or even better, than state of the art methods. We use deep convolutional networks to obtain multi-scaled features, and add deformable convolutional structures to overcome geometric transformations. We then fuse the multi-scaled features by up sampling, in order to implement the final object recognition and region regress. Experiments prove that our suggested framework improves the accuracy of detecting small target objects with geometric deformation, showing significant improvements in the trade-off between accuracy and speed.

Keywords: Object detection, Machine learning, AI, Deformable convolution, Computer vision

Introduction

The main purpose of object detection is to identify and locate one or more effective targets from still image or video data. It comprehensively includes a variety of important techniques, such as image processing, pattern recognition, artificial intelligence and machine learning. It has broad application prospects in such areas such as road traffic accident prevention [1], warnings of dangerous goods in factories, military restricted area monitoring and advanced human–computer interaction [2, 3]. Since the application scenarios of multi-target detection in the real world are usually complex and variable, balancing the relationship between accuracy and computing costs is a difficult task.

The object detection process is traditionally established by manually extracting feature models, where the common features are represented by HOG (histogram of oriented gradient), SIFT (scale-invariant feature transform), Haar (Haar-like features) and other classic algorithms based on grayscale. Following feature extraction, the SVM (support vector machine) or Adaboost algorithms are used for classification in order to obtain target information. These traditional extracting feature models are only able to determine low-level feature information, such as contour information and texture information, and have limitations in detecting multiple targets under complex scenes due to their poor generalization performance. However, object detection models, such as the R-CNN (region-based convolutional neural networks) series and the YOLO (you only look once) or SSD (single shot multiBox detection) models based on the deep learning CNN (convolutional neural network) features are more well-known. Deep learning CNN models not only extract the detail texture features from pre-level convolution networks, but are also able to obtain higher-level information from the post-level convolution layer. Following the traditional CNN process, the R-CNN series uses an enumeration method to presuppose the target candidate region in the feature map, gradually fine-tuning the position information and optimizing the object position for classification and recognition. In contrast, other object detection models will simultaneously predict the bounding box and classification directly in the feature map by applying different convolution sets. The R-CNN model has two operation stages (candidate region proposal and further detection) that allow for higher detection accuracy, while SSD and YOLO are able to directly detect the classification and position information, improving the detection speed.

We propose a novel multi-scaled deformable convolution network model to deal with the trade-off between accuracy and speed in object detection. The multi-scaled deformable convolutional neural network uses a new convolution method that has two offsets for image feature generation that are more sensitive to object deformation information. Additionally, the ability to detect objects that have geometrical deformations is improved. Secondly, feature fusion operations are performed on the multiple scale feature maps in the final detection. The image information of different scaled feature maps is simultaneously used to predict the classification and position information. This modification ensures the detection speed, enhances the target information of small objects, and also improves the accuracy of object detection.

The key contributions of our work are as follows:

1. The novel deformable convolution structure replaces the ordinary normal convolution operation for object detection. It effectively lets the CNN improve the generalization ability of extracting image features under different geometric deformations. Also, the new network automatically trains the offset of the convolution without wasting a large amount of computation time and cache space. Thus, significant performance gains on computer vision tasks, such as object detection and semantic segmentation, are observed.
2. An up-sample is applied to the feature pyramid to merge the multi-scaled feature information. This increases the accuracy of small target object detection by avoiding the loss of information of small target objects after multiple convolution and pooling operations. It also provides an important scheme for the detection of dense objects with overlapping occlusion in complex scenes.

An evaluation on the large-scale PASCAL VOC dataset shows that our approach is able to achieve a better single-model performance than well-known object detection model based on learned deep CNN, including Faster R-CNN, YOLO and SSD. For example, we achieve a 52.55% MAP (mean average precision) score at a speed of 27 frames/s. Compared with other object detection algorithms, our FPS (frames per second) increases approximately 3 times compared to the R-CNN series, with the MAP approximately 7% higher compared to the SSD and YOLO series. In addition, we perform random geometric deformation operations on the same test dataset, and find that our network structure has the better object detection accuracy on these image data.

Related work

Deep convolutional neural network

The CNN framework is an important model for deep learning theory, with a wide range of applications in image recognition and classification [4, 5]. It is developed from artificial neural networks. The previous layer is used as the input of the subsequent layer, and the back-propagation algorithm is used to update the parameters. The CNN model contains many network layers, can take the original image as the input, and may subsequently introduce many practical strategies, such as convolution, pooling and dropout, in order to improve the fault tolerance of the model [6]. Among these, convolution and pooling are necessary strategies in existing CNN models.

Convolutional neural networks have become ubiquitous in computer vision, ever since AlexNet popularized deep convolutional neural networks by winning the ImageNet Challenge: ILSVRC 2012 [7, 8]. More complex deep networks, such as VGGNet, further improved the superiority and high accuracy of classification and recognition [9], although this brought about more than one hundred million parameters and additional model calculations. In 2015, Deep ResNet, with its residual operation appeared, made it possible for deeper network structure to have hundreds of layers [6]. MobileNet used separable convolution in order to reduce the computational costs, and sought a balance between accuracy and speed [10]. Obviously, convolutional kernel and layer vulnerability factor to evaluate object detection and other computer vision tasks reliability [11]. Recently, the convolutional neural networks have incorporated classic machine learning algorithms, such as SVM, LR and so on [12]. It have achieved very good results in the classification and recognition tasks, and also achieved effective fusion with traditional algorithms [13]. In this study, we select a residual operation and separable convolution to construct the feature extraction network. The unit (1×1) convolution reduces the computational complexity, and the residual structure avoids the gradient disappearing due to the deepening of the network layer. In addition, the adaptability of the existing network model to the geometric deformation of the object is almost entirely due to the diversity of the data itself, and the internal structure of the model does not have a mechanism to adapt to the geometric deformation. This is because the convolution operation itself has a fixed geometry, and the geometry of the convolutional network constructed by the stacking is also fixed, such that it does not have the ability to model geometric deformations. Thus, we introduce the deformable convolutional network structure into model and increase the learning ability of the CNN-based object detection network for geometric deformation.

Object detection

State-of-the-art methods for detecting objects of general classes are mainly based on deep CNNs. Girshick et al. [14] proposed a multi-stage pipeline, denoted as regions with convolutional neural networks (R-CNN), for training deep CNNs in order to classify region proposals for object detection. It decomposes the detection problem into several stages, including bounding-box proposal, CNN pre-training, CNN fine-tuning, SVM training, and bounding box regression. Such a framework resulted in a high performance and was widely adopted in other work. To accelerate the training of the R-CNN pipeline, Fast R-CNN [15] was proposed, whereby each image patch is no longer wrapped to a fixed size before being fed into the CNN. Instead, the corresponding features are cropped from the output feature maps of the last convolutional layer. In the Faster R-CNN pipeline [16], the region proposals are generated by a region proposal network (RPN), and the overall framework can thus be trained in an end-to-end manner. The unique candidate frame proposal provides a high degree of precision, yet it also imposes a burden on the speed of detection. Also [17] proposed that the category and location information can be used to segment objects and image redundancy well by the combination of the region proposal method based on the convolutional neural network and super pixel method. Although subsequent improved models, such as Fast RCNN and Faster RCNN, continue their aim in creating breakthroughs in accelerating object detection, the process of generating candidate frame regions still inevitably contributes to a high level of running time.

Besides frameworks that include region proposal, methods that directly perform position regression and classification have also been proposed for object detection. YOLO [18] divides the image into even grids and simultaneously predicts the bounding boxes and classification scores for every grid. SSD [19] generates multiple fixed size anchor boxes (fixed at 6 or 9) on every location in order to predict classification scores for further regression. Omitting the process of candidate region proposals greatly improves the detection speed, yet the simple estimation of the position of the object ignores the information of many small objects and dense objects, reducing the overall detection accuracy.

Our model directly transforms the positions of target objects into the regression problem using one step, as with the strategy of YOLO and SSD. However, we combine the method of FPN (feature pyramid networks) which creates the frame regression and prediction on the multi-scaled and fused feature map [20]. Also, we try to use Soft-NMS (soft non-maximum suppression) to provide a dynamic regression to improve the detection accuracy of small target objects and dense objects [21]. In addition, the flexible use of multi-level convolution feature fusion [22], the addition of fine-grained feature classification [23], and a more comprehensive evaluation method for multi-object detection [24], all make multi-object detection become more efficient and accurate.

Methods

In this section, we first describe the overall structure of our multi-scaled deformable convolutional object detection network based on YOLO v3 [25]. We then describe the deformable convolutional network and the multi-scaled feature fusion by up-sampling. Finally, we introduce the training loss of the overall framework

Overall structure of the object detection network

Our image object detector adopts YOLO's backbone network and adds the new trick in convolution operation and feature information fusion. The overall framework is shown as Fig. 1.

The first backbone network is the Darknet53 network [26]. As a new network for performing feature extraction, it is a hybrid approach combining the network used in YOLO v2, Darknet-19, and the newer residual network tactics. The network, which is larger, uses successive 3 × 3 and 1 × 1 convolutional layers, with shortcut connections. In addition, we add three deformable convolution layers before the convolutional layers with a size of 52 × 52, 26 × 26 and 13 × 13 to modify the feature extraction (see the yellow section in Fig. 1).

The second element is the detection network section. The YOLO detection network divides the input image into 7 × 7 grids [27]. If the centre position of the ground truth falls within a certain grid, three bounding boxes and their confidences, as well as 20 class probabilities, are predicted for each grid. We also use the convolutional set, which is made up of 3 × 3 and 1 × 1 convolutional layers, in order to control the output, which includes 20 types of classification information, three frames positions and the IOU position. The new trick mentioned above refers to the detection network performing the above operations on three different feature map scales (13 × 13, 26 × 26 and 52 × 26, respectively). The upper-level feature maps will be up-sampled and merged with the low-level layer features by the channel (see the red section in Fig. 1).

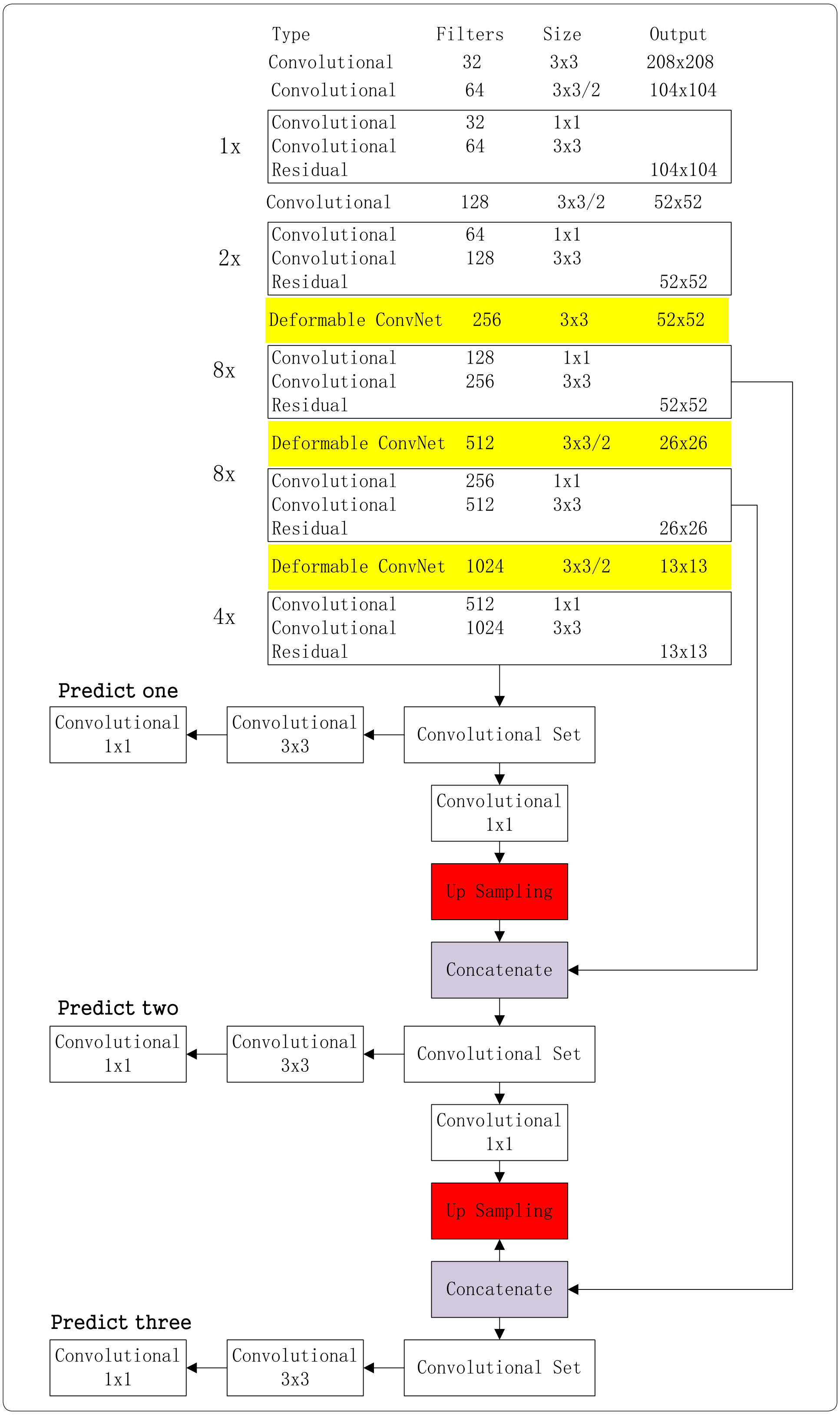
Deformable convolutional network

For object recognition of real scenes, an inevitable challenge arises from the recognition errors caused by changes in the shape, size, and posture of objects caused by the motion or different observation angles. Generally speaking, there are two common methods dealing with this question. The first is the data argument, which artificially changing the size, shape and rotation angle of the object in advance, and enhancing the diversity of the data in order to simulate the deformation of the object. However, this method increases the cost of data preprocessing, and the data will never cover all real application scenarios. Thus, the generalization ability of the model will be reduced to some extent. Another method is the use of a transform invariant feature algorithm, such as SIFT. Yet this handcrafted design of invariant features and algorithms can be difficult, or perhaps infeasible, for overly complex transformations, even when they are known [28, 29].

To solve the above problems, this study proposes the idea of applying a deformable convolution network to the one-step object detection network, and changes the fixed geometry of the convolution kernel in the traditional convolutional network, in order to enhance the modeling ability for the geometric transformation of detected objects.

(See figure on next page.)

Fig. 1 Architecture of the multi-scaled deformable convolutional neural network framework. The framework mainly consists of two components. (1) The backbone network based on Darknet53, which includes residual structures and deformable convolution. (2) The object detection network, based on multi-scaled detection and feature fusion



	Type	Filters	Size	Output
	Convolutional	32	3x3	208x208
	Convolutional	64	3x3/2	104x104
1x	Convolutional	32	1x1	
	Convolutional	64	3x3	
	Residual			104x104
	Convolutional	128	3x3/2	52x52
2x	Convolutional	64	1x1	
	Convolutional	128	3x3	
	Residual			52x52
	Deformable ConvNet	256	3x3	52x52
8x	Convolutional	128	1x1	
	Convolutional	256	3x3	
	Residual			52x52
	Deformable ConvNet	512	3x3/2	26x26
8x	Convolutional	256	1x1	
	Convolutional	512	3x3	
	Residual			26x26
	Deformable ConvNet	1024	3x3/2	13x13
4x	Convolutional	512	1x1	
	Convolutional	1024	3x3	
	Residual			13x13

The main concept is that sampling with the offset replaces the original sampling in a fixed position, which can be learned without additional supervision.

A common convolution operation performs sampling in the input feature map *X* with a regular grid *R*, and sums the sample values under the weights, *w*. The grid *R* defines the size and expansion of the receptive field [30, 31]. For example, a 3 × 3 convolution kernel with an expansion size of 1 can be defined as follows:

$$R = \{(-1, -1), (-1, 0), \ldots, (0, 1), (1, 1)\}, \tag{1}$$

For the every output $y(P_o)$, the sampling must be performed with nine positions from *X*. These nine positions are in the shape of a grid diffused around a center position $X(P_o)$. The coordinates (− 1, − 1) represent the upper left corner of $X(P_o)$, while (1,1) represent the lower right corner, with the remaining follow the same representation. Under traditional convolution, for each position P_o on the output feature map *Y*, we output the feature map formula *Y* as:

$$y(Po) = \sum_{Pn \in R} w(P_n) \cdot X(P_o + P_n), \tag{2}$$

where P_n enumerates all positions in *R*.

Under the deformation convolution, for each output $y(P_o)$, nine positions are sampled from *X*, in the shape of the grid that is diffused around center position $X(P_o)$. Subsequently, a new parameter is added. The parameter ΔP_n allows the points of sampling to be diffused into an irregular shape as follows:

$$y(P_o) = \sum_{Pn \in R} w(P_n) \cdot X(P_o + P_n + \Delta P_n). \tag{3}$$

The new sampling is located at an irregular position with offset $P_n + \Delta P_n$. The offset ΔP_n is usually a decimal, while the $X(P_o)$ on the feature map is always a whole number. Formula (3) can be realized by bilinear interpolation:

$$X(P) = \sum_{q} G(q, p) \cdot X(q), \tag{4}$$

where *P* represents an arbitrary (decimal) position $(P = P_o + P_n + \Delta P_n$ in Eq. (2)), *q* enumerates all global spatial positions in the feature map *X*, and *G* is a bilinear interpolated kernel. Note that, *G* is two-dimensional and is divided into two one-dimensional cores:

$$G(q, p) = g\left(q_x, p_y\right) \cdot g\left(q_y, p_y\right), \tag{5}$$

where $g(a, b) = \max\left(0, 1 - \left|a - b\right|\right)$. Formula (3) can be calculated quickly, as $G(q,p)$ is non-zero only for some *q*. Finally, the calculation positions of the convolution kernel on the image will be changed from the original 3 × 3 squared position, as shown in Fig. 2a. Additional calculation positions are shown in Fig. 2.

As the low-level convolution feature is not sensitive enough to the position information of the image, we add the deformable convolution layers on the network under the layers of res9, res17 and res21, which are relatively backward under the network structure. Also, we prove that putting the deformable convolution layers in res9, res17

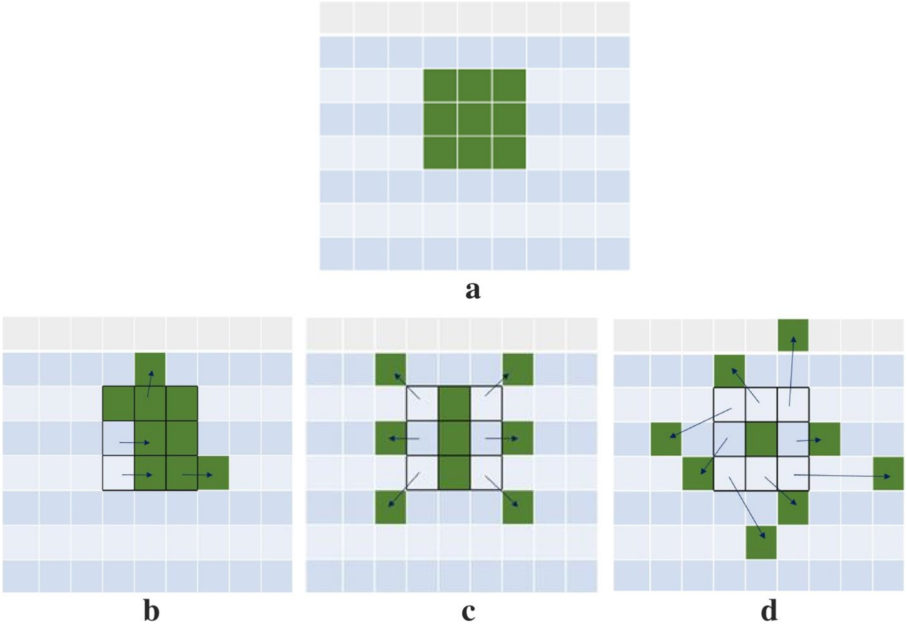


Fig. 2 Different calculation positions under traditional convolution or deformable convolution. **a** Original calculation position of convolution. **b**–**d** Different convolution positions changes after defromable convolutional networks are applied

and res21 will have the highest MAP. In addition to the contribution of the traditional convolution structure for location information, by using deformable convolution, we aim to change the position of the sampling point and automatically learn its offset. In this study, the application of the deformable convolution network improves the accuracy of object detection, and promotes the model to automatically learn the geometric transformation, particularly under images where geometric transformation exists, such as scaling and rotation [32].

The fusion of multi-scaled features

For object detection of real scenes, the accurate detection of small target objects will determine whether detection information is lost. Although a sampling operation based on convolutional networks already includes robustness to the changes of object size, it is often not sensitive enough for finer-grained small object detection. When object detection is performed on the feature map based on CNN, the feature information of the lower layer is less abundant, but the position information is more accurate. The semantic information of the upper layer is observed at a greater amount, yet the position information is often given at lower amount, as the feature map becomes smaller after pooling layers. Therefore, using the different sizes of the feature maps in the CNN network to better detect objects under different sizes, particularly small target objects that are prone to miss detection, is important for object detection performance [33].

Many studies have investigated how to overcome this challenge, with the easiest way being the application of a data argument, which changes the image size in many different scales. This involves resizing the images to different scales and training them in the convolutional network to adapt to different scales of object detection. As larger images increase memory storage and calculations, most experiments resize the images during the testing stage. Despite this, time and memory consumption cannot be avoided. Another method, similar to SPP (spatial pyramid pooling) net, has also been applied to the R-CNN series [34]. For example, Fast R-CNN and Faster R-CNN use the spatial pyramid pooling layer to pool the regions of images of any size using different pooling grids. They then generate fixed-length feature vectors for further classification and regression. The most widely used and popular method is similar to the SSD network, which introduces a regional detection mechanism on different scales of feature maps. It is reasonable to say that the detection in all different scales will obtain the most comprehensive scale information. Yet considering that low-level feature maps contain poor semantic information, the corresponding detection calculations will slow down the speed. Thus, the SSD network drops the previous low-level features and begins detection with conv4_3. In addition, some convolutional layers are added behind conv4_3 to generate additional feature maps for higher-level semantics. However, the results of the final experiment show that this is limited for the detection of small objects. The detection accuracy is poor, and far less than that of YOLO v3. Also, it is proposed recently that convolution kernels of different sizes could be used to predict classes and bounding boxes of multi-scale objects directly in the last feature map of a deep CNN for rapid object detection with acceptable precision loss is achieved [35].

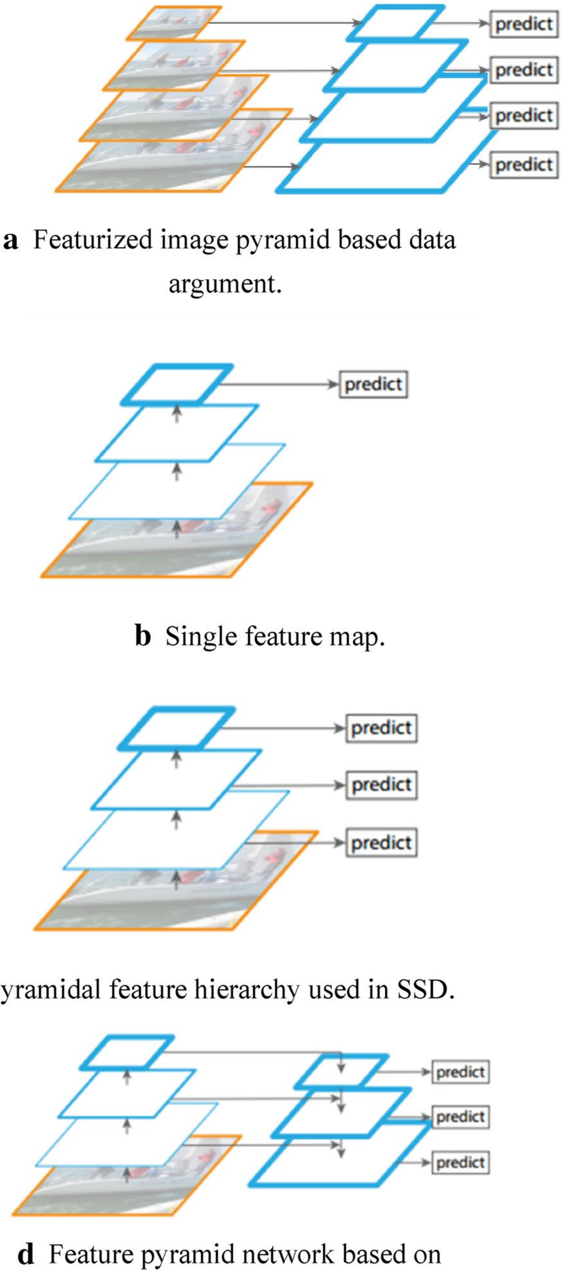
Based on the research of the FPN network, this study combines the top-level features with the low-level features using upper sampling. In addition, we use the concatenate method instead of the direct addition between feature pixels, and we achieve the fusion between high-level features and low-level features by extending the dimension of the feature map. The new model independently performs the prediction in multiple layers and controls the amount of computation, in order to better utilize the multi-scaled feature map information and further refine the object detection results.

Figure 3 presents the research concepts described in this section.

The multi-scaled feature fusion adopted in this paper mainly comes from Fig. 3d, and includes the following two steps.

The bottom-up path

The feed-forward calculation of the CNN is denoted as the bottom-up path. The feature map is calculated using a convolution kernel, and generally becomes smaller and smaller. This study will take the output of some features as the same as the original size, known as the "Same Network Stage". The above process involves defining a pyramid level for each of the same network phases, and then selecting the output of the last layer of each phase as a reference set for the feature map. In particular, for the residual depth network, we select the activation output of the last residual structure of the "Same Network Stage" as the reference. These residual module's outputs are denoted as {C3, C4, C5}, corresponding to the outputs of conv3, conv4, and conv5. It is important that their output scales have different pixel sizes of $\{52 \times 52, 26 \times 26, 13 \times 13\}$, and the previous pixel size is twice that of the following. Considering the memory usage problem and cross-semantic information in the underlying feature map, conv1 and conv2 are not included in the pyramid.



a Featurized image pyramid based data argument.

b Single feature map.

c Pyramidal feature hierarchy used in SSD.

d Feature pyramid network based on

Fig. 3 **a** Using an image pyramid to build a feature pyramid. **b** Recent detection systems have opted to use only single scale features for faster detection. **c** An alternative is to reuse the pyramidal feature hierarchy computed by ConvNet, as if it were a featurized image pyramid. **d** The feature pyramid network (FPN) is fast, as in **b** and **c**, but more accurate due to the fusion and multiple detection under different scaled features. The feature maps are indicated by blue outlines and thicker outlines denote semantically stronger features [20]

Top-down path and horizontal connection

The purpose of this step is to up-sample a more semantic high-level feature map, such that the features are laterally connected to the features of the previous. Thus, the high-level features are enhanced. It is worth noting that the two-layer features from the lateral connection must have an equal spatial size. This up-sampling can be performed using, for example, nearest neighbor up-sampling or bilinear interpolation. Once this is carried out, the layer features are combined with the corresponding previous layer feature. Note that the previous layer has to undergo a 1 × 1 convolution kernel, in order to change the number of channels to that of the last layer in the FPN. This study also uses the convolution feature concatenate method to perform feature fusion after the up-sampling operation. This expands and supplements the low-level feature information by increasing the number of channels rather than directly performing the addition between pixels, as in the FPN. In fact, the concatenate operation is the combination of information channels, which means that the feature dimension of the image itself is increased. We hope to find more information about the location characteristics of the object in the added features. Performing the addition directly between pixels does not change the number of features; it only adds more information to each original dimension. We found that the concatenate operation can skip the process (note that the FPN will use a 1 × 1 convolution kernel to change the number of channels in order to prepare for addition between feature maps), yet addition will require less computational work in subsequent convolution operations. Finally, testing proves that the concatenate operation causes just a slight improvement compared to addition, increasing the MAP by 0.02. However, it makes the network structure simpler and easier to understand, and this study thus uses the concatenate operation for the object detection network.

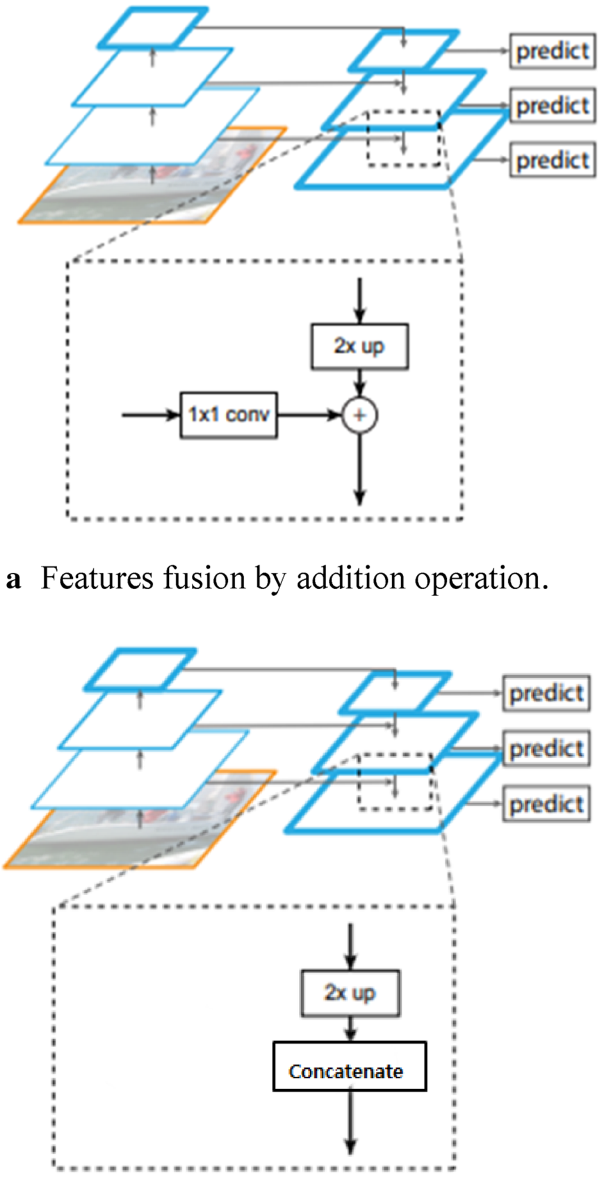
For the residual depth network structure, we first use addition for feature fusion by adding a 1 × 1 convolution kernel to generate the double-channel feature map for the pixel-to-pixel fusion operation with the previous layer in the last of C4 and C5 layers, just as in the original FPN. The specific network structure is shown in the Fig. 4a. We then aim to use the concatenate between previous layers and the last C4 or C5 layers following the up-sampling operation to realize the feature fusion. The specific network structure is shown in the Fig. 4b.

Finally, the merged feature map is processed with a 3 × 3 convolution kernel to generate the final required feature map (in order to eliminate the aliasing effect of the up-sampling). The fused feature layers corresponding to the {C3, C4, C5} layer is {P3, P4, P5}, and the corresponding layer space sizes are the same.

The training loss function of the object detection network

The training loss function measures the difference between the predicted value and the true value [36]. Designing a reasonable and effective loss function is very important for the training of the target model. Especially for the one-stage object detectors in computer vision, the dynamic loss is proposed as an improved version of the focal loss in the literature [37]. Our multi-scaled deformable convolutional object detection network model uses the mean square error as the loss function [38]. It consists of three parts; the coordinate error of boxing, the IOU error between predicted boxing and the groundtruth, and the category error, and can be expressed as Formula (6):

$$Loss = \sum_{i=0}^{S^2} (coordErr + iouErr + clsErr). \tag{6}$$



a Features fusion by addition operation.

b Features fusion by concatenate operation.

Fig. 4 The building block illustrating the lateral connection and the top-down pathway, merged by an addition operation (**a**) and a concatenate operation (**b**)

Formula (6) should also consider the contribution rate of each loss when performing simple addition. Thus, we set the weight of *coordErr* λ_{coord} as 5.0 during training. When calculating *iouErr* for the grid containing the object and the grid containing no object, their contributions of the *iouErr* to the network's training loss are different. If the same weights are set, the confidence value of the lattice containing no object is approximately 0, and the influence of the confidence error of the lattice containing the object in calculating the gradient of the network parameter will be magnified. To solve this problem, we use the modified *iouErr* and set the weight λ_{noobj} as 0.5. Note that 'containing' here means that there is an object whose center coordinates fall into the grid. For equal error values, the effect of the errors of a large object on detection should be less than the effect of the errors of a small object. This is because the same positional deviation in the large object is much smaller than that in the small object [18]. We follow the YOLO method and try to modify this problem by applying the square to the information items (width, w and height, h) of the object size. In summary, the training loss can be divided into three parts, which can be calculated as follows:

$$\begin{aligned} coordErr = \lambda_{coord} \sum_{i}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} \left[\left(x_i - \hat{x}_i \right)^2 + \left(y_i - \hat{y}_i \right)^2 \right] \\ + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} \left[\left(\sqrt{\omega_i} - \sqrt{\hat{\omega}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \end{aligned} \tag{7}$$

$$iouErr = \lambda_{noobj} \left(\sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{obj} \left[\left(C_i - \hat{C}_i \right)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B} l_{ij}^{noobj} \left[\left(C_i - \hat{C}_i \right)^2 \right] \right) \tag{8}$$

$$clsErr = \sum_{i=0}^{S^2} l_i^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c) \right)^2 \tag{9}$$

It is indicated that the feature map is divided into S*S grids, and B candidate frames are generated in each grid. The $\hat{x}, \hat{y}, \hat{\omega}, \hat{h}$ r represent the predicated coordinate and the width and height of the region box, respectively. And the $\hat{C}$ represents the confidence values and the $\hat{p}$ represents the category probabilities, respectively. The remaining parameters are the label values. The variable $\mathrm{l}_{\mathrm{ij}}^{\mathrm{obj}}$ indicates that the object falls in the jth region box of the ith grid. If there is no target object in a certain grid, the classification error is not back-propagated. Among all region boxes that with the highest IOU (intersection over union: the overlap between the candidate bound and the ground-truth bound) with ground-truth performs the back propagation of the coordinate error [39, 40].

Experiments

The experiments are divided into four parts. First, we introduce the dataset and hardware environment. We then design three sets of experiments for the two contributions in our study and prove the significant improvements over other networks. Finally, we make a comprehensive comparison on the evaluation indicators of the object detection task, and use experimental data to prove that we have an improved performance.

Experiments on the PASCAL VOC2007 and VOC2012 datasets

Based on common practice, our backbone network is pre-trained on the ImageNet classification dataset [40], and we continue to perform fine-tuning training on the object detection dataset, PASCAL VOC [41]. PASCAL VOC provides a set of standardized and excellent data sets for the training and verification of image recognition and classification. PASCAL VOC07+12 includes more than 27,000 images, accumulating more than 100,000 object tags, including 20 kinds of objects, such as humans, animals, vehicles and common indoor objects for training and verification. Specific objects include airplanes, bicycles, birds, boats, bottles, buses, cars, chairs, cows, dining tables, dogs, cats, horses, motorbikes, people, potted plants, sheep, sofas, trains and tv monitors. All image data contains tags that represent image information in the annotation file. This mainly includes category information of all objects which are identifiable in the image and location information [42]. The category information includes the 20 categories mentioned above, and the position information is usually expressed in the form of four coordinate points (xmin, xmax, ymin, ymax).

In addition, we performed our object detection model on the standard test set, which contains 4900 images, as well as their correlative tag data, and calculated the evaluation indicators in the test set. Our code, based on a multi-scaled deformable convolutional object detection network, was implemented on the TensorFlow framework. Our model was trained and implemented in the laboratory environment. The machine has a 64-bit operating system, 64G of RAM, an Intel$^{@}$ Xeon(R) CPU and a TITAN xp GPU.

The model requires an input size of 416 × 416 for the image, 100 training iterations on the VOC07+12 dataset, an initial learning rate of 0.001, and a momentum factor of 0.9. In addition, this study also draws on the size and number of anchors obtained by k-means clustering in Yolo v3, given as (116 × 90) (156 × 198) (373 × 326) (30 × 61) (62 × 45) and (59 × 119), respectively. The threshold of the predetermined IOU is 0.5.

Experiments in deformable convolutional neural networks

In order to verify the influence of the deformable convolution on the object position detection results, the IOU is calculated on the model with and without deformable convolution. Among these, the IOU can be understood as the degree of coincidence between the frame predicted by the system and the frame marked in the original image [43]. The calculation method is the intersection of the detection result and the ground-truth, which can also be represented as the accuracy of the detection [7]. The test results of the two models and the average IOU on 20 types of objects are shown in Table 1.

As can be seen from Table 1, most categories (approximately 16) have higher IOU values under the deformable convolutional network. The experimental results show that the deformable convolution can predict region boxing closer to the ground-truth. And compared with other techniques like SSD and Yolo, deformable convolution can further approach the real position information of the object especially when the objects are susceptible to deformation and improves the detection accuracy of the object position information.

In addition, geometric transformations (mainly 45° and 180° rotations) were performed on the test image set to verify the learning ability of deformable convolution for geometric transformations. The experimental results are shown in Fig. 5.

In Fig. 5, the first line presents the results of object detection on the original image, and the second line and the third line are the result of the object detection after 45° and 180° rotations are applied, respectively. The results show that the deformable convolution network can learn the geometric transformation of the object, and can correctly identify the deformed object. Although it is not sensitive enough to identify all kinds of geometric transformations (for example, the detection of the potted plant is lost under a

Table 1 IOU calculation for the classification and recognition of 20 different types of objects under the models which include, and do not include, the deformable convolution structure

	IOU (%)	
	Without deformation	**With deformation**
Airplane	68.56	*75.23*
Bicycle	78.01	*80.75*
Bird	65.45	*65.63*
Boat	*38.90*	35.70
Bottle	40.45	*49.76*
Bus	*61.90*	60.39
Car	68.98	*77.94*
Chair	43.59	*59.25*
Cow	68.60	*70.50*
Dining table	40.71	*45.12*
Dog	68.58	*74.24*
Horse	56.68	*69.50*
Motorbike	73.68	*79.01*
Person	*78.08*	75.07
Potted plant	38.02	*46.89*
Sheep	65.98	*68.71*
Sofa	45.64	*56.09*
Train	*81.95*	80.14
Tv monitor	68.32	*75.89*
Cat	43.26	*60.12*

Italic values refer to the higher IOU values for classification and recognition

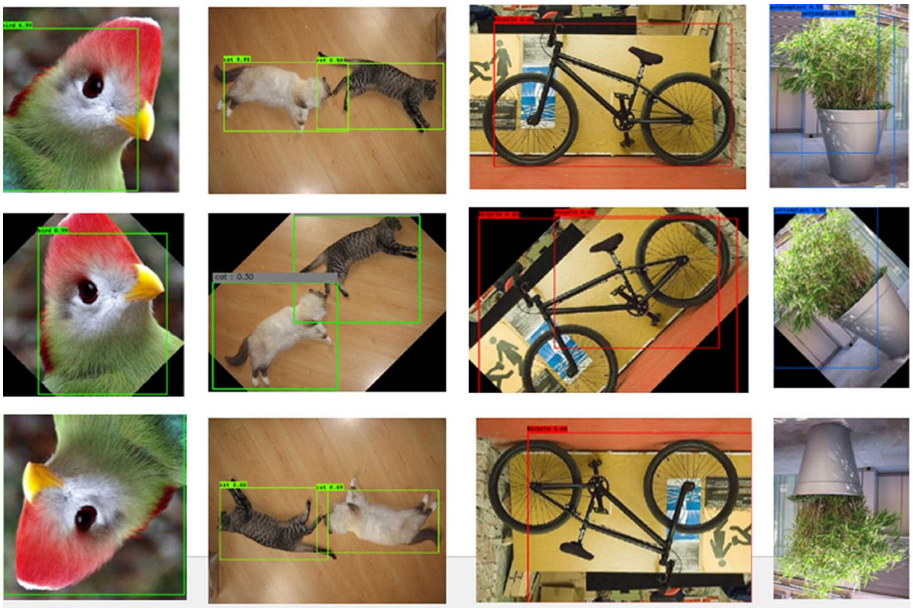


Fig. 5 Our model's object detection results under different geometric transformations. These mainly includes the detection in the original images, with images that have undergone a 45° and 180° rotation transformation

180° rotation in the fourth column of Fig. 5), in general, positive results are observed for the network with deformable convolution in the geometric transformations.

Experiments in small object detection

In order to verify the effectiveness of the FPN trick, which uses the top-level feature to combine the low-level feature by up-sampling, we chose the common Yolo network without multi-scaled feature fusion for an experimental comparison under the same network framework. Also, in order to further verify the gap between the SSD network and our method, we compare the effects on the unified test data. The experimental comparison of the three network models is shown in Fig. 6.

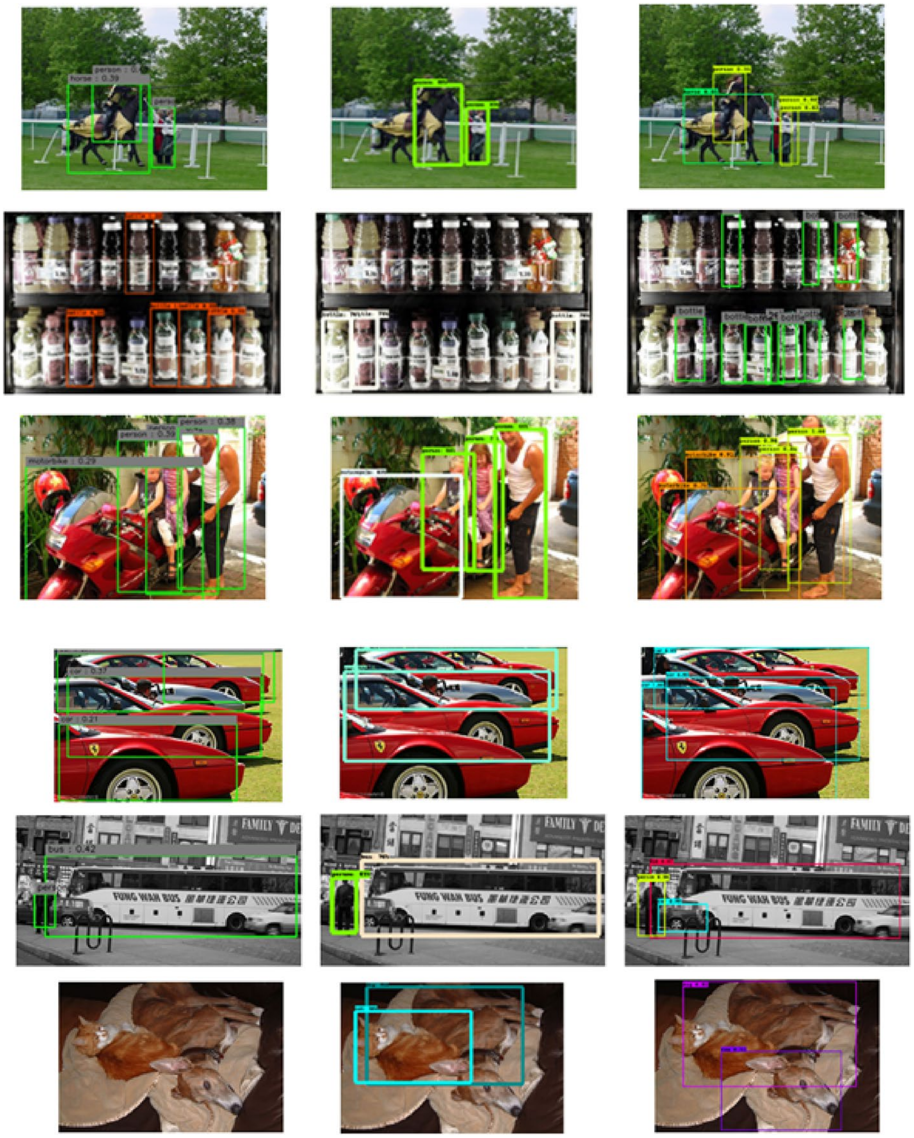
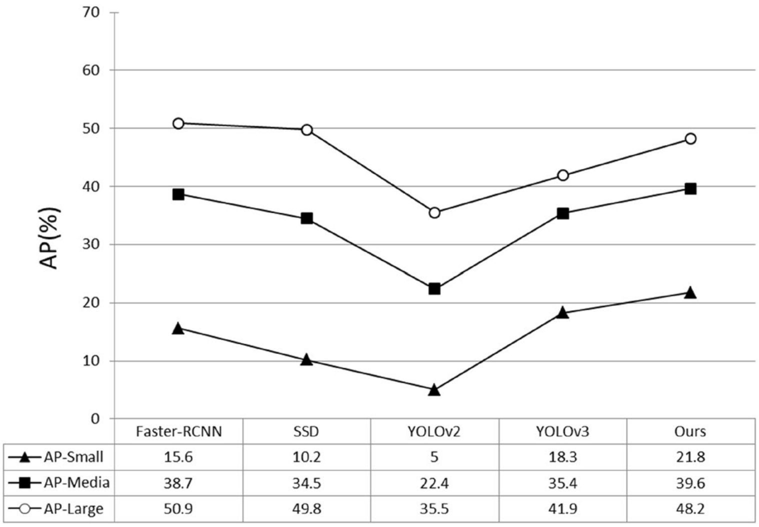


Fig. 6 Selected examples of object detection results on the PASCAL VOC 2007 test set using the YOLO, SDD and our multi-scaled deformable object detection systems, respectively

In Fig. 6, the first and second columns present the results of the object detection on the common YOLO network model and the SSD network model. The third column shows the result on the multi-scaled deformable object detection network proposed in this study. The comparison results show that the multi-scaled feature fusion can effectively reduce the missing detected features and increase the object detection accuracy of multiple objects in complex environments. In particular, the YOLO v1 model, which is in the first column in Fig. 6, presents zero detection in the last photo. In addition, our method is more sensitive to the object detection of small objects compared to the SSD network. Under the same image, the detection ability is stronger in the detection of small and dense objects. Of course, our model also has shortcomings in the detection of small target objects (for example, we cannot detect all the bottles on the second photo). This means that there is more room for improvement in dealing with small target objects and dense object detection.

In order to further analyze the object detection ability of our multi-scaled deformable convolutional neural networks for objects which have different sizes in the image, we divide the objects which exist in the origin image into three parts; small objects, medium objects and large target objects, according to the different area proportions of all existed objects. We calculate the three parts of MAP (AP-small, AP-media, AP-large) separately, and quantitatively observe the object detection ability of small, medium, and large objects.

From Fig. 7, we can see that, our model produces better object detection results among the relatively small target objects; particularly for small target objects (with the highest AP-Small, 21.8%). Compared to other best performing models, our model has a 3.5% greater rate for small object detection, but just 0.9% in medium-sized object detection. We also found that our model did not achieve the best results in the detection of large targets, yet in comparison, the detection was not too low.



	Faster-RCNN	SSD	YOLOv2	YOLOv3	Ours
▲ AP-Small	15.6	10.2	5	18.3	21.8
■ AP-Media	38.7	34.5	22.4	35.4	39.6
○ AP-Large	50.9	49.8	35.5	41.9	48.2

Fig. 7 AP-small, AP-media and AP-large under different object detection models on the PASCAL VOC 07 dataset set

On the whole, small object detection is more difficult compared to the detection of large objects during the object detection task. Small objects are also more likely to lose feature information during multiple convolutions and pooling operations, thus affecting the overall object detection accuracy. At the same time, this also means that there is more room for improvement in the detection accuracy of small objects.

Metrics in object detection algorithms

The accuracy and speed of the model are calculated on the test set of PASCAL VOC07. Tests and comparisons are performed between common object detection networks without multi-scaled deformable convolutional networks and the new object detection networks with multi-scaled deformable convolutional networks. Among them, the PASCAL VOC07 test set includes a total of 4500 test pictures, including 20 kinds of objects, and each of which has the corresponding category label and the ground-truth of each object.

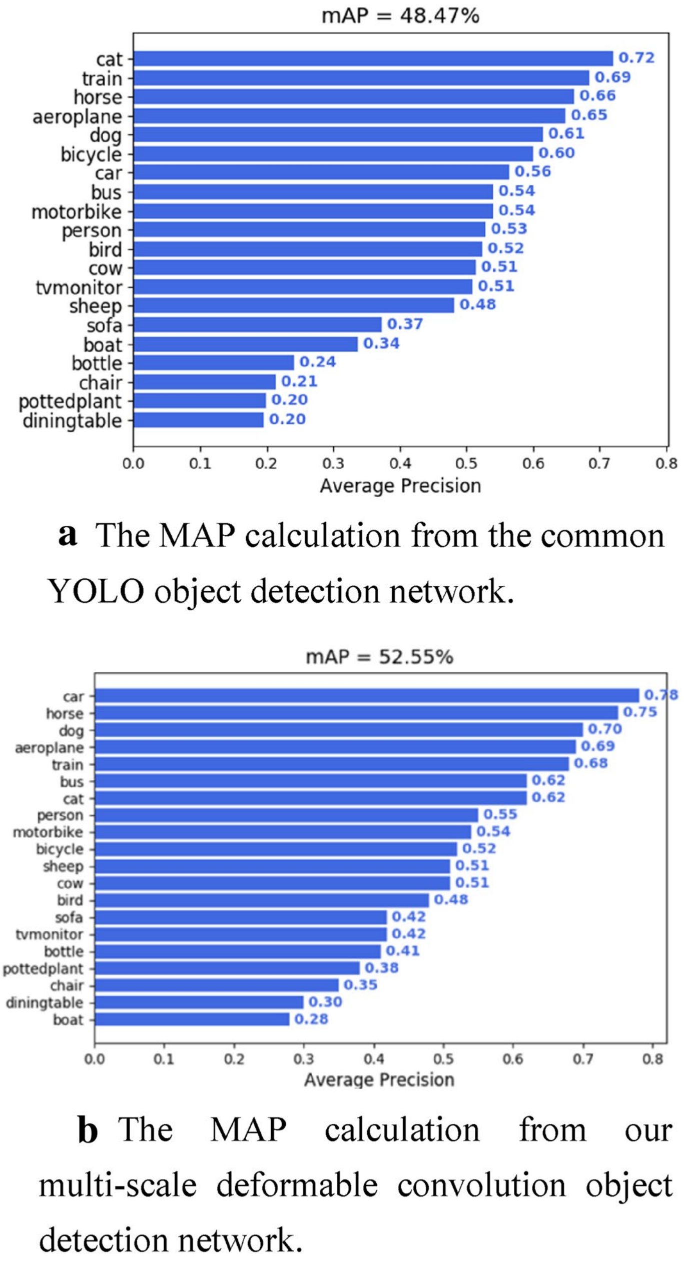
This paper selects MAP as the main metrics in object detection, representing the accuracy of the object detection algorithm. It undertakes two concepts: precision and recall. For the object detection task, the precision (P) and recall (R) for each object can be calculated. Each class can obtain the P-R curve after multiple calculations and tests. The area under the P-R curve is the value of AP. This "mean" in the MAP denotes the average of the AP of each class. Finally, the MAP is calculated, and should be within the interval [0, 1].

The MAP for on the multi-scaled deformable convolutional model in this study is shown in Fig. 8a. The MAP in Fig. 8b is derived from the YOLOv3 object detection network without the multi-scaled deformable convolutional structure.

Figure 8 shows that the object detection network using both the new method and the trick improves both AP and the total detection accuracy, MAP, with the total MAP increasing by approximately 4%.

The FPS is the definition in the image field, referring to the number of frames per second transmitted by the screen [44]. In image classification and object detection tasks, the FPS can represent the number of images that the model recognizes or classifies per second. It can be used to measure the average speed of images processed in the model [45]. Table 2 shows the experimental results of Faster-RCNN, YOLOv1, SSD, YOLOv3 and our novel multi-scaled deformable convolution object detection network model under the unified VOC07 test set. The MAP value and FPS are compared to analyze the trade-off between accuracy and speed.

As we can also see from the Table 2, the multi-scaled deformable convolutional object detection network has the highest MAP for input images of a size of 600×600 when compare with Faster R-CNN. In this experiment, the FPS with multi-scale deformable convolution is 27 frames/s when the input images have a size of 416×416. As the new network adds a certain amount of computation, it is slower than the YOLOv3-416, which does not include deformable convolution. Overall, the new network is still much faster than the Faster R-CNN, YOLOv1 and SSD series.



a The MAP calculation from the common YOLO object detection network.

b The MAP calculation from our multi-scale deformable convolution object detection network.

Fig. 8 The MAP calculation in the classifications and recognition of 20 different kinds of objects under the common YOLO system and our multi-scaled deformable object detection system on the whole PASCAL VOC 07 dataset

Conclusions and future work

Based on the tricks of both the FPN and deformable convolutional networks, this study proposes a new multi-scaled deformable convolutional object detection network structure. This network uses a deformable convolution structure instead of an ordinary convolution operation in order to increase the learning ability of the model with respect to object geometric deformation, as well as increasing the accuracy of object detection. This study also uses multi-scaled feature maps that combine

Table 2 Performance comparisons between the well-known object detection networks and that presented in this paper

Network model-image size	Train	Test	MAP@.50IOU	FPS
Faster R-CNN-600	VOC07+12	07 test	52.40	5
YOLOv1-416	VOC12	–	39.60	*45*
SSD-500	VOC12	–	41.02	18
YOLOv3-416	VOC07+12	–	48.47	30
Ours-416	VOC07+12	–	52.55	27
Ours-600	VOC07+12	–	*55.30*	21

Italic values refer to respectively the highest MAP value and FPS value

We control the sizes of the input images under different models to design multiple sets of comparison experiments, and compare the object detection accuracy and speed

low-level features by up-sampling to extract target object position information. This increases the ability of the model to detect small target objects and dense objects, and also greatly makes up for the defect in missing detections, which is always present in other object detection models. As the deformable convolution structure and the multi-scale fusion techniques in this study do not greatly increase computational costs, the effect on the calculation rate is also optimistic enough, while guaranteeing accuracy. Comprehensive experiments show that our multi-scaled deformable convolutional object detection network steadily pushes forward the performance (speed-accuracy trade-off) for object detection in images. Compared with other object detection algorithms, the FPS of our network is approximately four times greater than the R-CNN series. In addition, the MAP is approximately 7% and 12% higher than the YOLO v1 and SSD models, respectively. Also, the MAP is increased by approximately 4% under the same backbone compared to the original backbone network without the multi-scaled deformable convolutional operation.

The deformable convolution and multi-scale feature fusion are still new and sustainable research ideas for future object detection tasks. We will continue to explore how to set-up and use the deformable convolution structure to further minimize structural changes of the feature extraction backbone network. We hope that the application of the deformable convolution can avoid any further incremental training of the backbone network, which may mitigate the burden of the overall training task under the pre-training model. In addition, we will further explore the application of multi-scale deformable convolution networks to the field of video object detection. Our method can provide important ideas for the real-time detection of deformed objects after motion in the videos.

Abbreviations

HOG: Histogram of oriented gradient; SIFT: Scale-invariant feature transform; Haar: Haar-like feature; SVM: Support vector machine; R-CNN: Region-based convolutional neural networks; YOLO: You only look once; SSD: Single shot multiBox detection; CNN: Convolutional neural network; MAP: Mean average precision; FPS: Frames per second; Soft-NMS: Soft non-maximum suppression; SPP: Spatial pyramid pooling; FPN: Feature pyramid network; IOU: Intersection over union.

Acknowledgements

The work was supported by Supported by Yuyou Talent Support Plan of North China University of Technology (107051360019XN132/017), The Fundamental Research Funds for Beijing Universities (110052971803/037), Special Research Foundation of North China University of Technology (PXM2017_014212_000014), and Beijing Natural Science Foundation (4162022). We acknowledge TopEdit LLC for the linguistic editing and proofreading during the preparation of this manuscript.

Authors' contributions

Authors contributed in various important aspects. LG conducted the experiments, ZXC analyzed the results and drafted the manuscript. WS proposed the structure design. DYC provided valuable suggestions on improving the standards of the manuscript. All authors read and approved the final manuscript.

Funding

This study was funded by Supported by Yuyou Talent Support Plan of North China University of Technology (107051360019XN132/017), The Fundamental Research Funds for Beijing Universities (110052971803/037), Special Research Foundation of North China University of Technology (PXM2017_014212_000014), and Beijing Natural Science Foundation (4162022).

Availability of data and materials

The datasets supporting the conclusions of this article are available in the Pascal VOC repository (http://host.robots.ox.ac.uk/pascal/VOC/).

Competing interests

The authors declare that they have no competing interests.

Author details

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China. [2] Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing 100144, China.

Received: 26 September 2019 Accepted: 26 March 2020
Published online: 11 April 2020

References

1. Shine L, Jiji CV (2020) Automated detection of helmet on motorcyclists from traffic surveillance videos: a comparative analysis using hand-crafted features and CNN. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-08627-w
2. Liu J, Yang Y, Lv S, Wang J, Chen H et al (2019) Attention-based BiGRU-CNN for Chinese question classification. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01344-9
3. Cao D, Zhu M, Gao L et al (2019) An image caption method based on object detection. Multimed Tools Appl 78(24):35329–35350
4. Xudong L, Mao Y, Tao L (2017) The survey of object detection based on convolutional neural networks. Appl Res Comput 34(10): 2881–2886 + 2891
5. Aamir M, Pu Y, Rahman Z, Abro WA, Naeem H, Ullah F, Badr AM (2018) A hybrid proposed framework for object detection and classification. J Inf Process Syst 14(5):1176–1194
6. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada, 26–30 June 2016, pp 770–778
7. Krizhevsky A, Sutskever I, Hinton G E (2012) ImageNet classification with deep convolutional neural networks. In: Paper presented at the twenty-sixth annual conference on neural information processing systems, Lake Tahoe, Nevada, 3–6 December 2012, pp 1097–1105
8. Szegedy C, Liu W, Jia Y, Sermanet, P, Reed S (2015) Going deeper with convolutions. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, 7–12 June 2015, pp 1–9
9. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Paper presented at the international conference on learning representations, San Diego, California, 7–9 May 2015, pp 1–14
10. Andrew G, Menglong Zhu, Bo Chen, Dmitry Kalenichenko (2017) MobileNets: efficient convolutional neural networks for mobile vision. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, 21–26 July 2017
11. dos Santos FF, Carro L, Rech P (2019) Kernel and layer vulnerability factor to evaluate object detection reliability in GPUs. IET Comput Digital Tech 13(3):178–186
12. Ghrabat MJJ, Ma G, Maolood IY et al (2019) An effective image retrieval based on optimized genetic algorithm utilized a novel SVM-based convolutional neural network classifier. Human-centric Comput Inf Sci 9:31
13. Zhang F, Wu T, Pan J et al (2019) Human motion recognition based on SVM in VR art media interaction environment. Human-centric Comput Inf Sci 9:40
14. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Columbus, Ohio, 23–28 June 2014
15. Girshick R (2015) Fast R-CNN. In: Paper presented at IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015, pp 1440–1448
16. Ren S, He K, Girshick R et al (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
17. Jinbo C, Zhiheng W, Hengyu L (2018) Real-time object segmentation based on convolutional neural network with saliency optimization for picking. J Syst Eng Electron 29(6):1300–1307
18. Redmon J, Divvala S, Girshick R, et al (2016) You only look once: unified, real-time object detection. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada, 26–30 June 2016, pp 779–788
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In: Paper presented at the 14th European conference on computer vision, Amsterdam, The Netherlands, 11–14 October 2016

20. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, 21–26 July 2017, pp 2117–2125
21. Bodla N, Singh B, Chellappa R, et al (2017) Soft-NMS—improving object detection with one line of code. In: Paper presented at IEEE international conference on computer vision, Venice, Italy, 22– October 2017
22. Sun A, Li Y, Huang Y et al (2018) Facial expression recognition using optimized active regions. Human-centric Comput Inf Sci 8:33
23. Hou Y, Luo H, Zhao W, Zhang X, Wang J, Peng J et al (2019) Multilayer feature descriptors fusion CNN models for fine-grained visual recognition. Comput Anim Virtual Worlds 30:e1897
24. Wen Longyin, Dawei Du, Cai Zhaowei et al (2020) UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking. Comput Vis Image Underst 4(193):102907
25. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. arXiv preprint, arXiv:1804.02767v1 [cs.CV], Unpublished
26. Redmon J (2013–2016) Darknet: open source neural networks in c. http://pjreddie.com/darknet/. Accessed 30 July 2018
27. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, 21–26 July 2017, pp 6517–6525
28. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Columbus, Ohio, 23–28 June 2014, pp 580–587
29. Brink H, Vadapalli HB (2017) Deformable part models with CNN features for facial landmark detection under occlusion. In: Paper presented at the South African Institute of Computer Scientists and Information Technologists, ACM, Thaba\"Nchu, South Africa, 26–28 September 2017, pp 1–9
30. Jeon Y, Kim J (2017) Active convolution: learning the shape of convolution for image classification. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, 21–26 July 2017, pp 1846–1854
31. Jifeng D, Haozhi Q, Yuwen X, Yi L, Guodong Z, Han H and Yichen W (2017) Deformable convolutional networks. In: Paper presented at IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017, pp 764–773
32. Mordan T, Thome N, Cord M, Henaff G (2017) Deformable part-based fully convolutional network for object detection. In: Paper presented at British machine vision conference (BMVC), London, United Kingdom, 4–7 Sep 2017
33. Zeng H, Liu Y, Li S, Che J, Wang X (2018) Convolutional neural network based multi-feature fusion for non-rigid 3D model retrieval. J Inf Process Syst 14(1):176–190
34. He K, Zhang X, Ren S et al (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
35. Sun S, Yin Y, Wang X, Xu D, Wu W, Gu Q (2018) Fast object detection based on binary deep convolution neural networks. CAAI Trans Intell Technol 3(4):191–197
36. Song W, Zou S, Tian Y, Fong S, Cho K (2018) Classifying 3D objects in LiDAR point clouds with a back-propagation neural network. Human-centric Comput Inf Sci 8:29
37. Zhao K, Zhu X, Jiang H et al (2018) Dynamic loss for one-stage object detectors in computer vision. Electron Lett 54(25):1433–1434
38. Krasin I, Duerig T, Alldrin N, Ferrari V, Abu-El-Haija S, Kuznetsova A, Rom H, Uijlings J, Popov S, Veit A, Belongie S, Gomes V, Gupta A, Sun C, Chechik G, Cai D, Feng Z, Narayanan D, Murphy K (2017) Openimages: a public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages. Accessed 30 July 2018
39. Uijlings JRR et al (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171
40. Deng J, Dong W, Socher R, et al (2009) ImageNet: a large-scale hierarchical image database. In: Paper presented at IEEE Conference on computer vision and pattern recognition, Miami, Florida, 20–25 June 2009, pp 248–255
41. Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
42. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. Paper presented at the IEEE conference on computer vision and pattern recognition, Boston, Massachusetts, 7–12 June 2015, pp 3431–3440
43. Gultepe E, Makrehchi M (2018) Improving clustering performance using independent component analysis and unsupervised feature learning. Human-centric Computi Inf Sci 2018(8):25
44. Wang K, Zhang D, Li Y, et al (2017) Cost-effective active learning for deep image classification. IEEE Trans Circuits Systems Video Technol (99):1–1
45. Huang J, Guadarrama S, Murphy K, et al (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: Paper presented at the IEEE conference on computer vision and pattern recognition, Honolulu, Hawaii, 21–26 July 2017, pp 3296–3297

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.