An Algorithm for Target Detection Based On Aggregation Multi-Scale Feature

Yishun Chen*, Rongsheng Dong

Guilin University of Electronic Technology

18357913106@163.com

Abstract: In order to alleviate the multi-scale problem caused by the scale change between object instances, pyramids are widely used in target detection. Although these target detectors with characteristic pyramid structure have achieved good results, they have some limitations because they simply construct characteristic pyramids according to the skeleton multi-scale pyramid structure originally used for target classification tasks. In this study, based on M2Det, a multi-scale feature with richer multilevel information is proposed to construct a more effective feature pyramid to detect targets of different scales. First, the basic features with multi-level features will be fused and extracted from the backbone. Then, the basic features are input into the M module, and the features generated by each P module are used as the features of the detection object in the form of dense connection. Finally, the attention mechanism is introduced into the L module to assemble the features with the same scale and construct a feature pyramid for target detection. On the COCO dataset, MLPNet implements the results of AP37.8.

Key words: multi-scale problem, multi-scale characteristics, attention mechanism

1. Introduction

Multi-scale problems have always been one of the difficulties of target detection. In convolutional neural networks, deeper features have larger receptive fields and richer semantic information. It is robust to changes in object pose, occlusion, local deformation, etc. of the deep features, but the reduction of resolution leads to the loss of geometric details. On the contrary, shallow features have smaller receptive fields and richer geometric details, but the problem is higher resolution and lack of semantic information. In convolutional neural networks, the semantic information of objects can appear in different layers. Generally speaking, shallow features of small objects include some of its detailed information. As the number of layers deepens, the geometric details of the extracted features
may disappear completely due to the large receptive field. It becomes very difficult to detect small objects through deep features. And the semantic information of large objects will appear in the deeper features.

In order to solve the multi-scale problem, the early methods, such as SSD, used shallow feature maps and deep feature maps for detection at different levels of feature maps, which effectively alleviated the multi-scale problem, but each The layers are independent of each other, and the integration of high and low characteristics has not been achieved. In the later proposed FPN, in order to make up for the lack of semantics in low-level features, FPN further enhances a top to down pathways and horizontal connections to incorporate strong semantic information in high-level features to form a feature pyramid for target detection. But the feature pyramid also has its limitations. Each feature map in the pyramid is mainly or even only composed of a single layer of backbone, which means that it mainly or only contains single-level information. Generally speaking, high-level features in the deeper layer are more distinguishing for the classification subtask, and the low-level features in the shallower layer can help the object position regression subtask. In the recently popular attention mechanism, the weight of each feature channel is automatically obtained through learning, and then according to this weight to enhance useful features and suppress features that are not useful for the current task.

Based on these issues, this article mainly does the following: 1. In the backbone stage, the high and low-level features are merged to generate the basic feature for subsequent detection. 2. In order to get more abundant feature information, a bottom to up path is added to the P module. In the M module, each P module is connected to each other in the form of dense connection, and a multi-scale feature with multi-level information is obtained. The attention mechanism is introduced in the L module, and the feature maps with the same scale are first merged, and then the importance of each feature channel is obtained by learning. The contributions of this paper are as follows: 1. In order to alleviate the multi-scale problem, the multi-scale feature with richer multi-level information is generated by integrating the high-level feature and the low-level feature. 2. Construct a more effective feature pyramid through the obtained multi-scale features with richer multi-level information to detect targets of different scales.

2. MLPNet algorithm
The overall architecture of MLPNet is shown in Figure 2. It consists of four parts: backbone network, M module, P module and L module. MLPNet uses the backbone network to extract basic features. In the M module, the basic features are connected to each P module in the dense connection form of densenet, and then all the features of the same scale are merged through the L module to obtain multi-scale features with multi-level information.
The purpose of this article is to create a feature pyramid with rich information for multi-scale target detection. For this purpose, the pyramid model is constructed through top to down, bottom to up path and horizontal connections. As shown in the figure, the bottom-up path in the P module, through the horizontal connection, uses a higher resolution feature map and a lower feature map to generate a new feature map. Each feature map first passes a $3 \times 3$ convolutional layer with a step size of 2 to reduce the space size. Then through the horizontal connection, each element in the lower feature map is added to the down-sampled feature map. The fused feature map is additionally processed by a $3 \times 3$ convolutional layer, and the new feature map generated is used in the subsequent network. This is an iterative process, and there is a ReLU behind all convolutional layers.
The P module can be formally described as a six-tuple:

$$P = \langle \text{input}, l, C, U, D, \text{output} \rangle$$

input: represents the input in the P module. The input of the first P module is the basic feature, and the input of the other P modules is the feature after the fusion of the outputs of all previous P modules and the basic feature.

l: Represents the horizontal connection operation, using 1*1 convolution for dimensionality reduction operations.

C: Represents feature fusion operation.

U: Represents two times upsampling operation.

D: Represents an operation of Conv3x3+BN+ReLU.

output: indicates the output in the P module.

2.2 M module
First, the shallow features and deep features are merged from the backbone to form basic features. Then multiple P modules in the M module directly connect all layers to each other in order to ensure the maximum information flow between the layers in the network. In order to retain the feedforward characteristics, each P module obtains other inputs from all previous P modules and basic features, and passes its own feature map to all subsequent P modules (The first P module takes input only from the underlying characteristics). In the traditional convolutional neural network, the L layer has L connections, while in the M module of this model, the L layer has \( \frac{L(L+1)}{2} \) connections, that is, the
input of each layer comes from the fusion of the outputs of all previous layers. This strengthens the transfer and integration of features and makes more effective use of features.

\[
x_i = \begin{cases} 
  \text{base feature} & i = 1 \\
  F(\text{base feature}, y_1, y_2, ..., y_{i-1}) & i > 1 
\end{cases}
\]

\(x_i\) represents the input of the i-th P module, \(y_i\) represents the output of the i-th P module, and F represents the fusion feature operation.

2.3 L module

As shown in Figure 3, the L module combines multiple feature maps generated by multiple P modules into a new deep-level feature pyramid. The first part of the L module is to fuse all the feature maps with the same scale, and then go through the SE block in the second half of SENet, the weight of each feature channel is automatically obtained through learning, and then according to this weight to enhance useful features and suppress features that are not very useful for the current task, that is, let the network use contextual information to selectively enhance useful feature channels and suppress useless feature channels, so as to achieve adaptive calibration of feature channels.

The finally obtained brand new deep-level feature pyramid can be represented as \(y = (y^1, y^2, ..., y^j)\), where \(y^j = \text{concat}(y^j_1, y^j_2, ..., y^j_f) \in R^{W_j \times H_j \times C}\) represents the feature of the jth scale.

![Figure 3. L module](image)

The Squeeze operation in the SE block uses the global average pooling operation to compress each feature map after obtaining multiple feature maps U, so that the C feature maps finally become a 1*1*C real number sequence.

\[
z_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i, j)
\]

Excitation operation: the weights for each feature channel are generated using parameters. The parameters are learned to model the correlations between feature channels explicitly.

\[
S = F_{\alpha}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z))
\]
Among them, $\sigma$ refers to the ReLU function, and $\delta$ refers to the sigmoid function, $W_1 \in R^{C \times C}$, $W_2 \in R^{C \times C}$. In order to limit the complexity of the model and assist generalization, the paper introduces two fully connected layers (both are 1*1 conv layers), that is, the dimensionality reduction layer parameter is $W_1$, the dimensionality reduction ratio is $r$ ($r$ is set to 16), and then passes through a ReLU, and then an ascending layer with a parameter of $W_2$. Finally, the 1*1*C real number sequence is combined with U (multiple feature maps) to perform the Scale operation through the formula to obtain the final output.

3. Data sets and experiments

This experiment evaluates the performance of the MLPNet algorithm on the COCO data set. There are 82,783 training sets in the COCO data set and 40,504 verification sets. The validation set is divided into two parts, one is miniVal with 5000 images, and the remaining 35504 images and training set are called Trainval35k. This article uses trainval35k for training, and then tests on test-dev.

| method | Backbone  | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|--------|-----------|------|-----------|-----------|--------|--------|--------|
| Fast R-CNN | VGG-16 | 19.7 | 35.9 | - | - | - | - |
| Faster R-CNN | VGG-16 | 21.9 | 42.7 | - | - | - | - |
| Faster R-CNN with FPN | ResNet-101 | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| YOLOV3 | DarkNet-19 | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD512 | VGG-16 | 28.8 | 48.5 | 30.3 | 10.9 | 31.8 | 43.5 |
| STDN513 | DenseNet-169 | 31.8 | 51.0 | 33.6 | 14.4 | 36.1 | 43.4 |
| RefineDet512 | VGG-16 | 33.0 | 54.5 | 35.5 | 16.3 | 36.3 | 44.3 |
| RetinaNet500 | ResNet-101 | 34.4 | 53.1 | 36.8 | 14.7 | 38.5 | 49.1 |
| M2Det512 | VGG-16 | 37.6 | 56.6 | 40.5 | 18.4 | 43.4 | 51.2 |
| MLPNet | VGG-16 | 37.8 | 56.7 | 40.7 | 19.1 | 43.6 | 52.9 |

Tab.1 Comparison of MLPNet with other algorithms

The results of MLPNet compared with other algorithms are shown in Table 1. According to the figure, it can be seen that MLPNet is better than other target detection algorithms in the table in all indicators. The AP of MLPNet using VGG16 as the backbone reaches 37.8, which exceeds most excellent target detection algorithms. For example, compared with the earlier Fast R-CNN (R.Girshick et al-15), it has increased by 91.9%. For example, the one-stage representative algorithm YOLOV3 has an AP of 21.6, and the STDN using a scale conversion module has an AP of 31.8, based on SSD The AP of the improved RefineDet is 33.0, and the AP of the new structure RetinaNet (TY Lin et al-17) using Focal Loss is 34.4. Compared with the recent M2Det (Zhao et al-19), the AP under the same conditions is also increased.
It can be seen from the table that compared with M2Det, MLPNet has the largest increase in the detection of small targets, reaching 3.8%, and the increase in detection of large targets has also reached 3.3%. This is due to the bottom-up path added in the P module and the dense connection form in the M module. The P module integrates deep and low-level features, and the densely connected form in the M module enables the low-level features to be continuously reused, enriching geometric information, which makes the detection of small objects easier and more beneficial for the detection of large objects.

![Figure 4: Results of different numbers of P modules](image)

As shown in Figure 4, it shows the influence of different Numbers of P modules on the results. It can be found that increasing the number of P modules can effectively improve the results. The reason is that the P module combines low-resolution, strong semantic features with high-resolution, weak semantic features through two paths, top to down and bottom to up, thereby enhancing the effect. However, the effect of the improvement gradually decreases with the increase of the number. Due to the stacking of multiple P modules, a large number of parameters will be introduced and the time will be affected.

4. Conclusion
In this article, we propose a target detection algorithm based on aggregated multi-scale features—MLPNet for the multi-scale problem in target detection tasks. MLPNet uses the high-level feature fusion in the P module, the dense connection method in the M module and the attention mechanism in the L module to obtain a multi-scale feature with multi-level information to construct a more effective feature pyramid for detection. The MLPNet algorithm was tested on the COCO data set. Compared with the previous target detection algorithm, MLPNet has better results.

References
[1] R. Girshick, J. Donahue, T. Darrell and J. Malik 2014 Rich feature hierarchies for accurate object detection and semantic segmentation Proc. of the IEEE Conf. on computer vision and pattern recognition pp 580–587
[2] R. Girshick 2015 Fast r-cnn Proc. of the IEEE Int. Conf. on computer vision pp 1440–1448
[3] S. Ren, K. He, R. Girshick and J. Sun 2015 “Faster r-cnn: Towards real-time object detection with region proposal networks” in Advances in neural information processing systems pp 91–99
[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi 2016 You only look once: Unified, real-time object detection Proc. of the IEEE Conf. on computer vision and pattern recognition pp 779–788

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

[6] T.-Y. Lin, P. Doll’ar, R. B. Girshick, K. He, B. Hariharan and S. J. Belongie 2017 Feature pyramid networks for object detection in CVPR vol 1 p 4

[7] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt and Joan M Ogden 1984 Pyramid methods in image processing RCA engineer pp 33–41

[8] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi and Jiaya Jia 2018 Path aggregation network for instance segmentation in CVPR

[9] Piotr Dollar, Ron Appel, Serge Belongie and Pietro Perona 2014 Fast feature pyramids for object detection IEEE Transactions on Pattern Analysis and Machine Intelligence pp 1532–1545

[10] Zhao Q , Sheng T ,Wang Y and et al 2019 M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network Proc of the AAAI Conf. on Artificial Intelligence pp 9259-9266