Scale-Transferrable Object Detection

Xuewen Yang

August 17 2018

Abstract

Scale problem lies in the heart of object detection. In this work, the author develop a novel Scale-Transferrable Detection Network (STDN) for detecting multi-scale objects in images. In contrast to previous methods that simply combine object predictions from multiple feature maps from different network depths, the proposed network is equipped with embedded super-resolution layers (named as scale-transfer layer/module in this work) to explicitly explore the interscale consistency nature across multiple detection scales. Scale-transfer module naturally fits the base network with little computational cost. This module is further integrated with a dense convolutional network (DenseNet) to yield a one-stage object detector. They evaluate this proposed architecture on PASCAL VOC 2007 and MS COCO benchmark tasks and STDN obtains significant improvements over the comparable state-of-the-art detection models.

1. Introduction

Scale problem lies in the heart of object detection. In order to detect objects of different scales, a basic strategy is to use image pyramids to obtain features at different scales. However, this will greatly increase memory and computational complexity, which will reduce the real-time performance of object detectors.

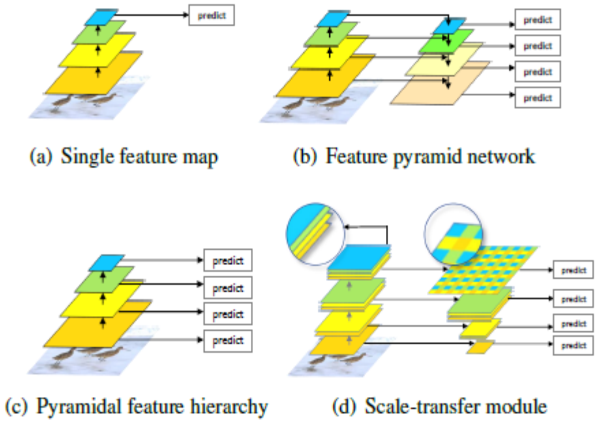
Scale problem lies in the heart of object detection. In order to detect objects of different scales, a basic strategy is to use image pyramids to obtain features at different scales. However, this will greatly increase memory and computational complexity, which will reduce the real-time performance of object detectors.

In recent years, convolutional neural networks(CNN) have achieved great success in computer vision tasks, such as image classification [2], semantic segmentation [3], and object detection. The hand-engineered features are replaced with features computed by convolutional neural networks, which greatly improves the performance of object detectors. Faster R-CNN uses convolutional feature maps computed by one layer to predict candidate region proposals with different scales and aspect ratios. Because the receptive field of each layer in CNN is fixed, there exists inconsistency between the fixed receptive field and the objects at different scales in natural images. This may compromise object detection performance. SSD and MS-CNN use feature maps from different layers within CNN to predict objects at different scales. Shallow feature maps have small receptive fields that are used to detect small objects, and deep feature maps have large receptive fields that are used to detect large objects. Nevertheless, shallow features have less semantic information, which may impair the performance of small object detection. FPN, ZIP and DSSD integrate semantic information on feature maps at all scales. As shown in Figure 1, a top-down architecture combines high-level semantic feature maps with lowlevel feature maps to yield more semantic feature maps at all scales. However, in order to improve detection performance, feature pyramids must be carefully constructed, and adding extra layers to build the feature pyramids brings additional computational cost.

In order to obtain high-level semantic multi-scale feature maps, and also without impairing the speed of the detector, we develop a Scale-Transfer Module (STM) and embed this module directly into a DenseNet [1]. The role of DenseNet is to integrate low-level and high-level features within a CNN to get more powerful features. Because of the densely connected network structure, the features of DenseNet are naturally more powerful than the ordinary convolutional features. STM consists of pooling and scale-transfer layers. Pooling layer is used to obtain small scale feature maps, and scale-transfer layer is used to obtain large scale feature maps. Scale-transfer layer is first proposed to do image super-resolution because of its simplicity and efficiency, and some people also use it to do semantic segmentation. We use this layer to efficiently expand the resolution of the feature map for object detection.

2. Related Work

State-of-the-art methods of object detection are based on convolutional neural networks. For example, SPPnet, Fast R-CNN, Faster R-CNN, R-FCN, and YOLO use fea-



(a) Single feature map (b) Feature pyramid network

(c) Pyramidal feature hierarchy (d) Scale-transfer module

Figure 1. Importance of context for object recognition. Without the context (face), it is hard to recognize the black curve in the middle area as a nose.

tures from the top layer of the convolutional neural network to detect objects of different scales. However, since each layer of the convolutional neural network has a fixed receptive field, it is not optimal to predict objects of different scales with only features of one layer.

There are generally three main types of methods to further improve the accuracy of multi-scale object detection. One is to detect objects using the combinations of multilayer features. The other is to use different layer features to predict objects at different scales. The last is a combination of the above two methods.

For the first type of method, ION uses skip pooling to extract information at multiple layers, and then the object is detected by using the combined features. HyperNet incorporates deep, intermediate and shallow features of the image for generating proposals and detecting objects. Y-OLO concatenates the higher resolution features with the low-resolution features by passthrough layer and runs detection on top of this expanded feature map. The basic idea of these methods is to enhance the power of features by combining low-level and high-level features.

References

[1] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1