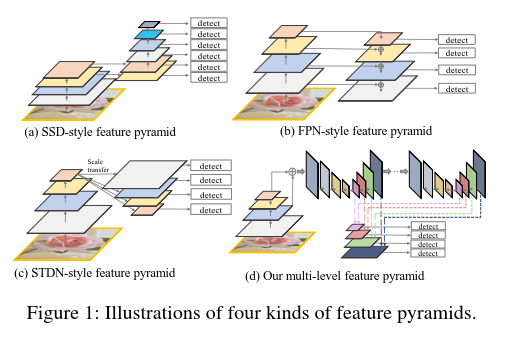
1.以往使用Feature Pyramid的网络结构，达到了卓越的效果，但是依旧存在不足：

Although these object detectors with feature pyramids achieve encouraging results,they have some limitations due to that they only simply construct the feature pyramid according to the inherent multiscale, pyramidal architecture of the backbones which are originally designed for object classiﬁcation task。

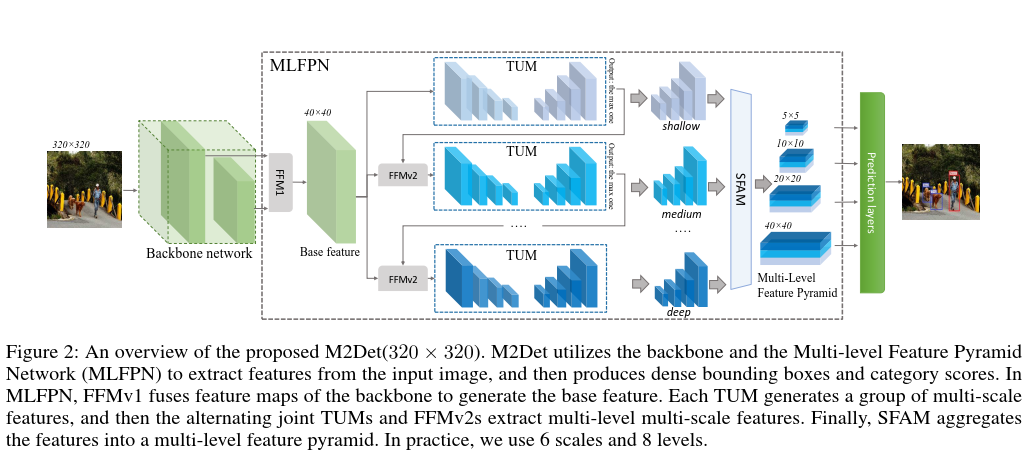
1. feature maps in the pyramid are not representative enough for the object detection task, since they are simply constructed from the layers (features) of the backbone designed **for** object classiﬁcation task。
2. each feature map in the pyramid (used for detecting objects in a speciﬁc range of size) is mainly or even solely constructed from single-level layers of the backbone, that is, it mainly or only contains single-level information。

2.网络结构对比图



Multi-Level （and Multi Scale）Feature Pyramid Network (MLFPN)

2. The overall architecture of M2Det is shown in Fig. 2



MLFPN consists of **three modules**:

Feature Fusion Module (FFM)

FFMv1 enriches semantic information into base features by fusing feature maps of the backbone

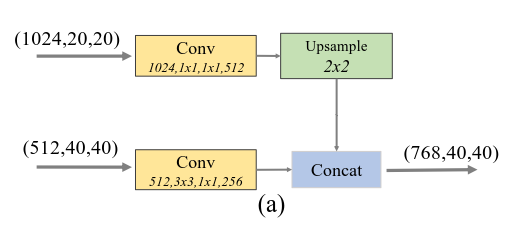
Thinned U-shape Module (TUM)

Each TUM generates a group of multi-scale features, and then the alternating joint TUMs and FFMv2s extract multi-level multiscale features.

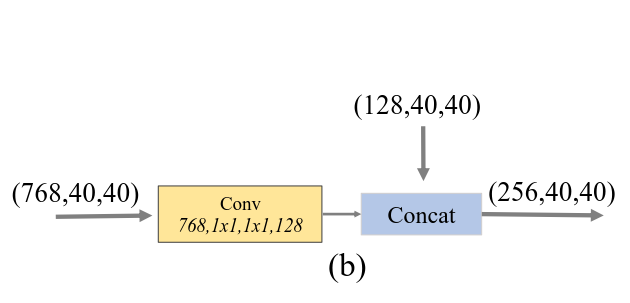
Scale-wise Feature Aggregation Module(SFAM)

SFAM aggregates the features into the multi-level feature pyramid through a scale-wise feature concatenation operation and an adaptive attention mechanism

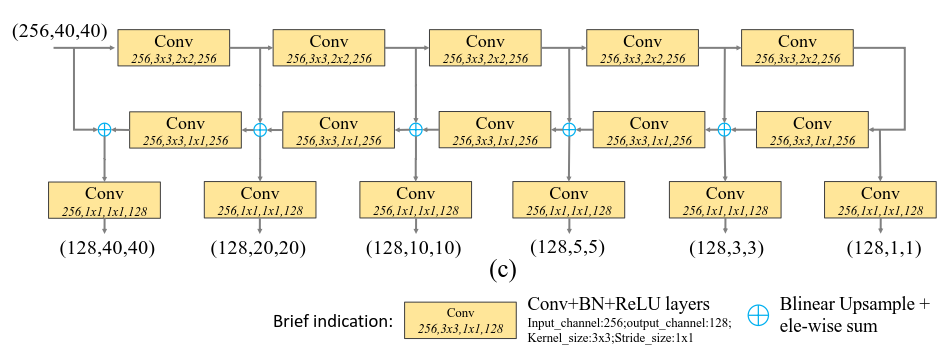
**Deteail（****FFMv1、****FFMv2、TUM、****SFAM）：如下图**

****

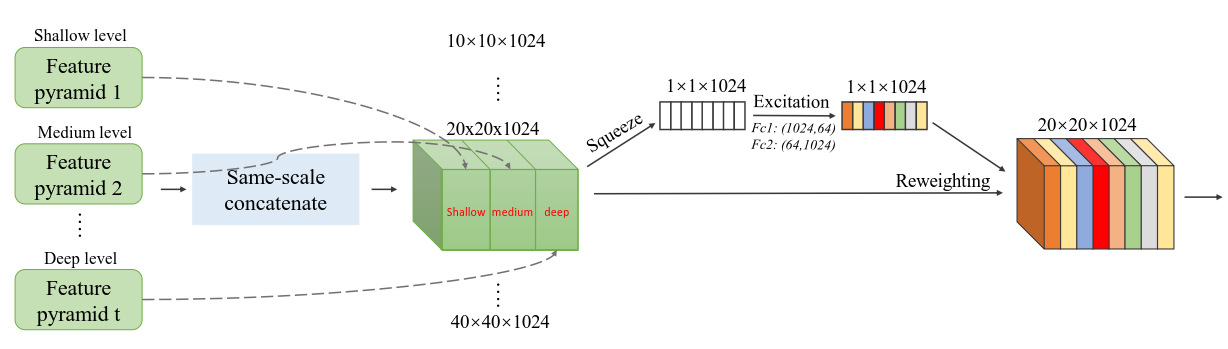
**FFMv1**

****

**FFMv2**

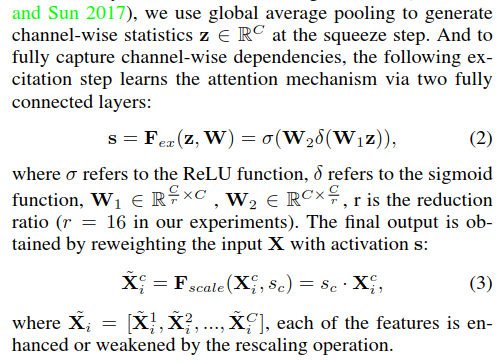
****

**TUM**

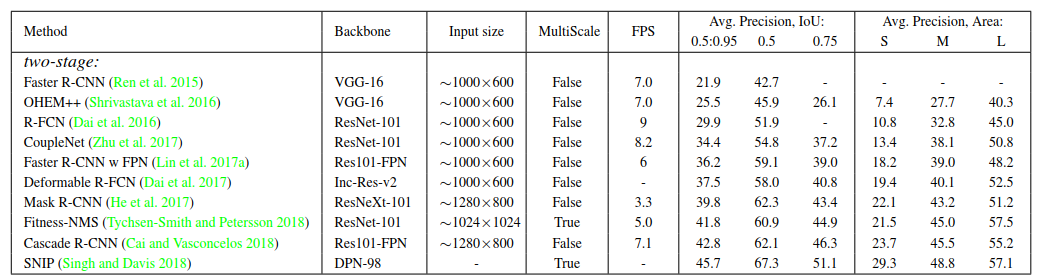
****

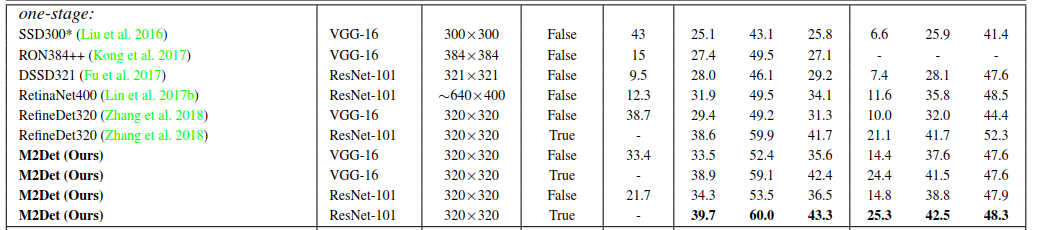
**SFAM**

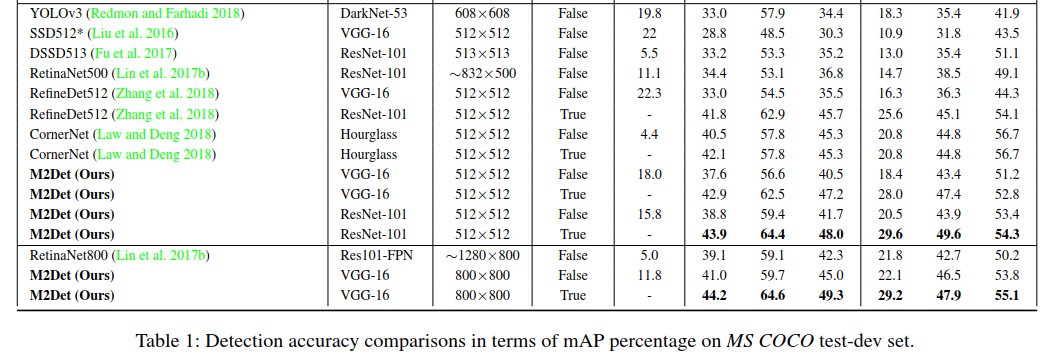
In the second stage, we introduce a channel-wise attention module to encourage features to focus on channels that they beneﬁt most

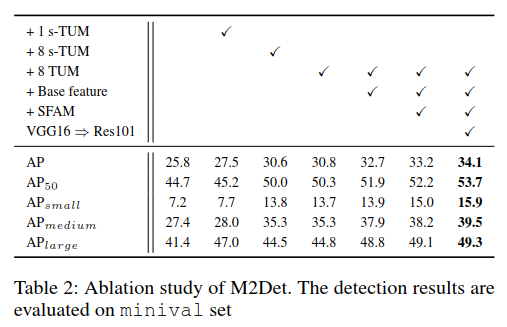
****

For all experiments based on M2Det, we start training with warm-up strategy for 5 epochs, initialize the learning rate as 2 × 10−3, and then decrease it to 2 × 10−4 and 2 × 10−5 at 90 epochs and 120 epochs, and **stop at 150 epochs**. M2Det is developed with PyTorch v0.4.0 1. When input size is 320 and 512, we conduct experiments on a machine with 4 NVIDIA Titan X GPUs, CUDA 9.2 and cuDNN 7.1.4, while for input size of 800, we train the network on NVIDIA Tesla V100 to get results faster.

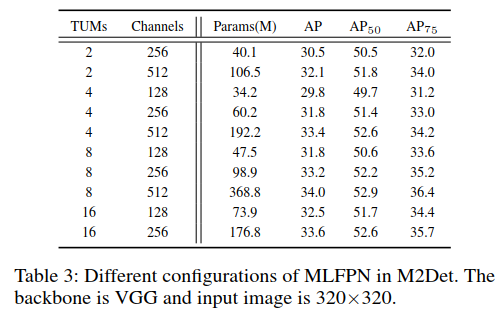








表明M2Det中各个改进点，对性能提升的作用。



表明TUM组数，对最终结果的影响

