# M2Det Introduction

--M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network

----y

# Background

<https://arxiv.org/abs/1811.04533v1>

<https://github.com/qijiezhao/M2Det>

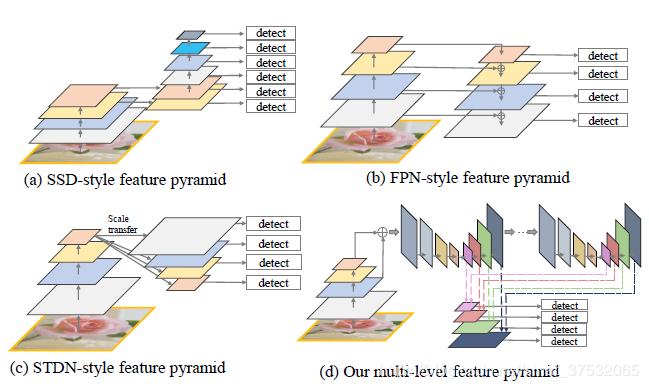
From Alibaba/Peking University

Nov.2018

AAAI 2019会议

# Features

## MLFPN



1. SSD型：使用了主干网络的最后两层，再加上4个使用stride=2卷积的下采样层构成；
2. FPN型：也称为U型网络，经过上采样操作，然后对应融合相同的scale；
3. STDN型：基于DenseNet的最后一个dense block，通过池化和scale-transfer操作来构建；
4. MLFPN型：本文新提出，整体思想是Multi-level&Multi-scale；感觉和FPN相比，就是在最后多了一个相同分辨率的SE network融合prediction。越高层级，分辨率低，但是所包含的高级语义越多。因此经过前面的融合以后，U-shape后面的高分辨率同样也包含了高级语义。

\*：Yolo感觉属于FPN的结构。

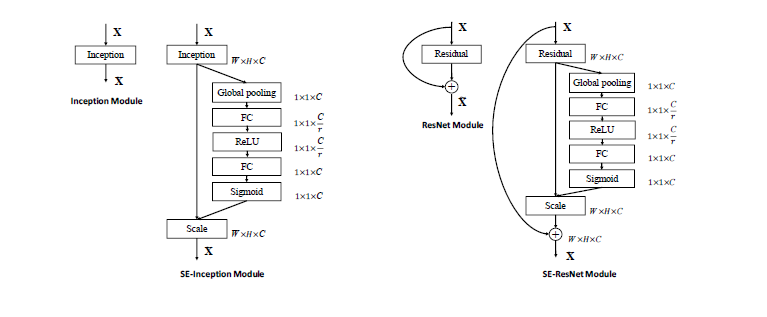
## one stage shot

1. 一是在测试阶段使用图像金字塔（如Cascade RCNN），即将原始图像进行一系列缩放，毫无疑问这会大幅增加内存和计算开销；
2. 二是在从输入图像提取出的特征金字塔上进行检测，该方法可以同时用于训练和测试阶段中，相对开销较小，易于集成，适合end-to-end。

## Squeeze Excitation Network

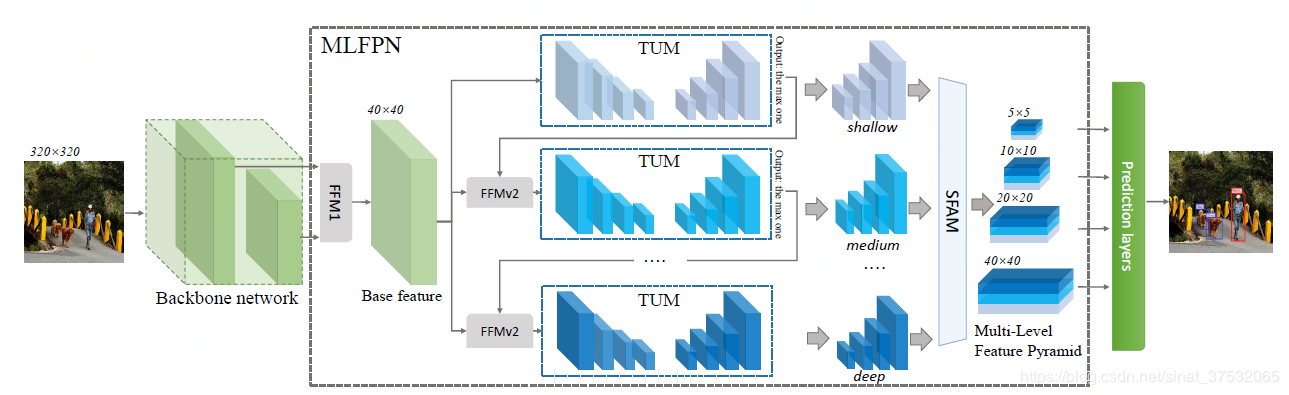
最后一届ILSVR的冠军，源自自动驾驶的初创公司。

为了利用channel之间的相关性，这种结构的原理是想通过控制scale的大小，把重要的特征增强，不重要的特征减弱。



# Structure

总体流程如下：



主要的创新是MLFPN，包括：

1）特征融合模块FFM；2种FFMv1和FFMv2

2）细化U型模块TUM；1种

3）尺度特征聚合模块SFAM；1个

## backbone网络

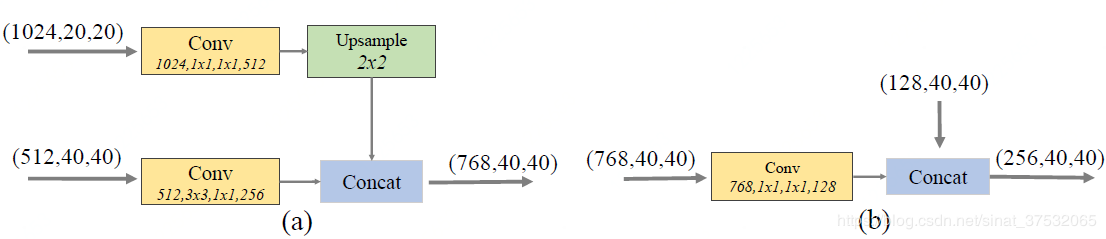
VGG conv4.3和conv5.3

或者ResNet-101

## MLFPN

### FFMv

FFMv1和FFMv2的结构如下图。



FFMv1融合从卷积网络出来的2个分辨率，(1024,20x20)和(512,40x40)，到一个cubic(768,40x40)，如上图a。

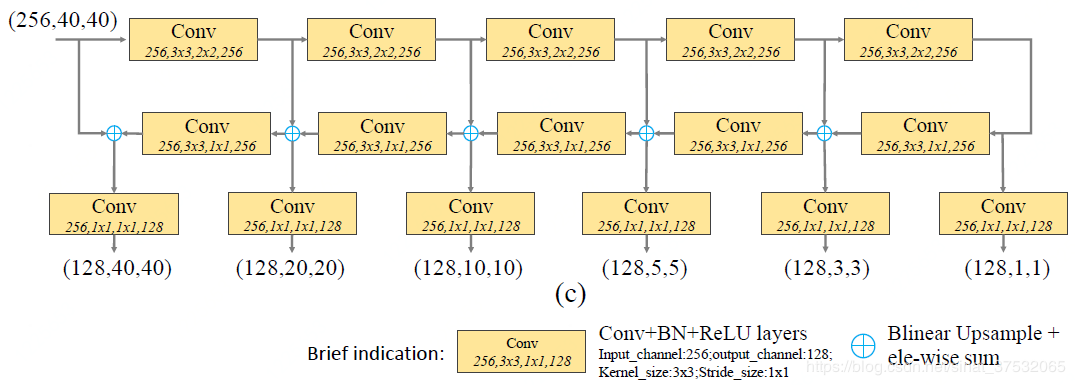
FFMv2 直接进行矩阵拼接。

### TUM

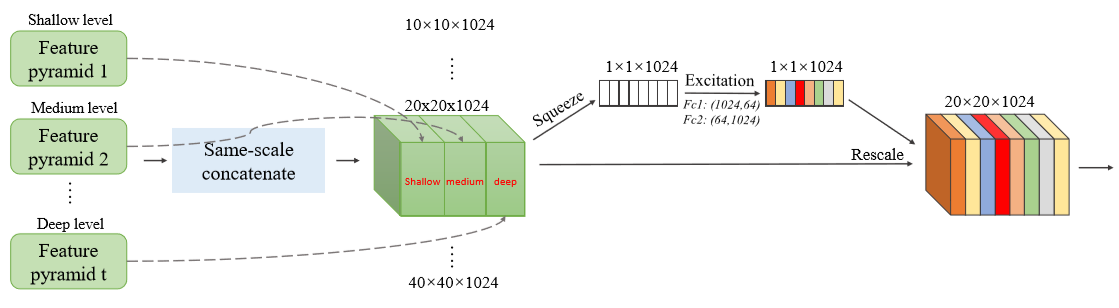
TUM使用了比FPN和RetinaNet更薄的U型网络。

MLFPN包含8个TUM。每个TUM结构如下，包含5种分辨率的卷积，输出6种分辨率的特征尺寸，因此SFAM的输入是6级金字塔。

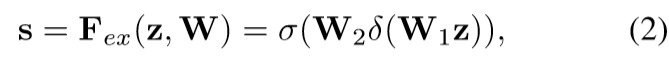
1. --convolution+RN+Relu，但是深度固定256.
2. --Upsample+ele-wise-sum
3. --有延迟，但是8个TUM可不可以pipe-line？？减小延迟提速，但是要增加内存。
4. –级数可调



### SFAM



**公式如下：**



*where σ refers to the ReLU function, δ refers to the sigmoid function, W1 ∈ RC/r ×C , W2∈RC×C /r , r is the reduction ratio (r = 16 in our experiments). The ﬁnal output is obtained by reweighting the input X with activation s:*

**

*where ˜Xi = [˜ X1i, ˜ X2i,..., ˜ XCi ], each of the features is enhanced or weakened by the rescaling operation.*

**实质就是一个concatenate + SE block。**

## Prediction

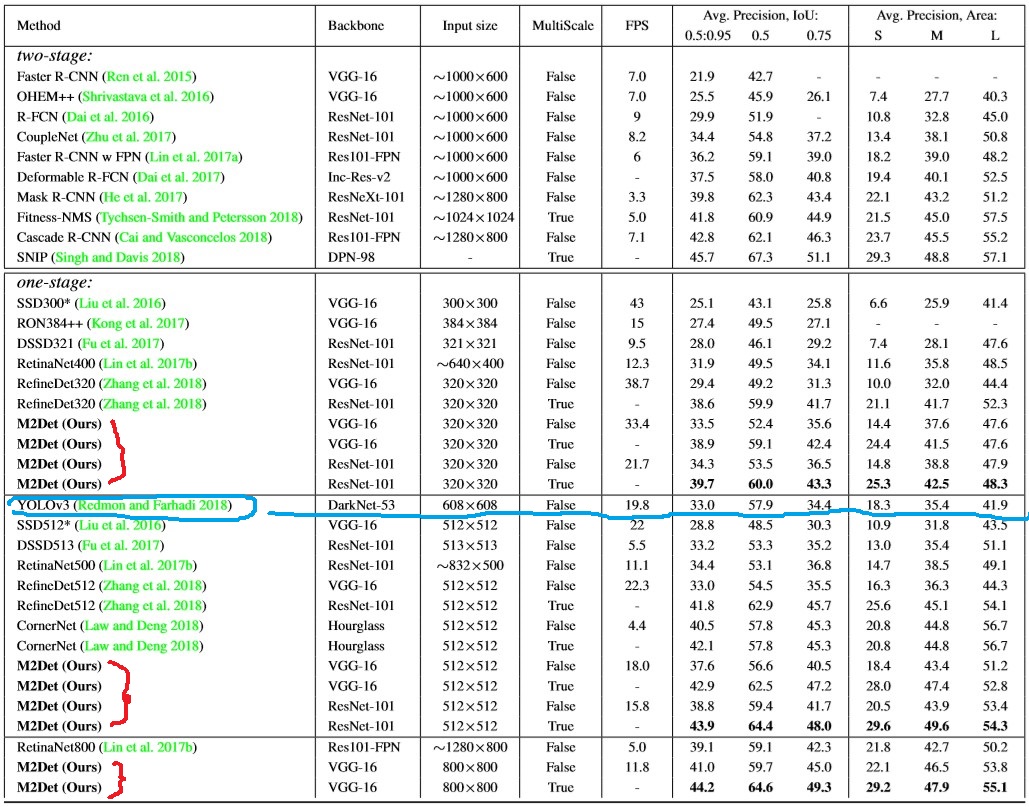
1. --6 anchor boxes，YOLOv3是3个。

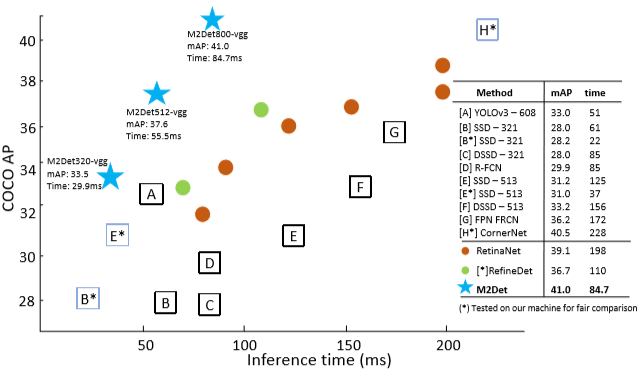
是根据分辨率来的？M2Det6级分辨率金字塔，Yolo是3个分辨率。

1. --soft-NMS/IOU

# Performance

## mAP和FPS





## 参数大小

Yolo-v3的416x416是250+M

M2Det的800x800是147M

--作者的意思，性能的提升，不是由于网络深度参数增加带来的。

--值得商榷。作者512x512的权重文件是506M，但是是pytroch的。

## 对比Yolo v3

NMS参数：IoU=0.45，threshold=0.1



M2Det结果



YoloV3 结果

# About

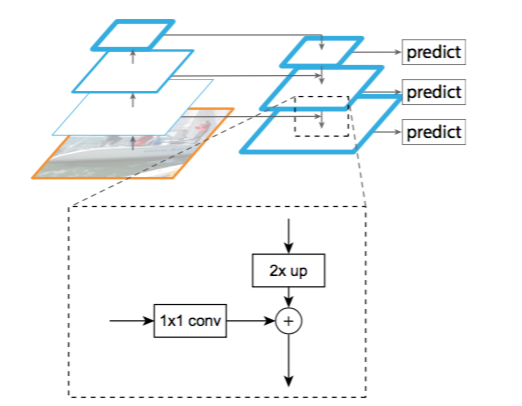
1. USDPU是否支持如下的操作：

* 矩阵拼接
* 计算是否支持U-shape，x2的Upsample这个目前看比较普遍。

1. 按理ResNet-101比Yolo-v3的DarkNet-53深，速度确没有太大影响，需要研究
2. 目前看不同的网络，支持的图片分辨率不尽一致，ISP怎么设计的。
3. U-shape怎么维度扩张的，还没看。🡪从下面FPN论文的链接看，应该是upsample+element\_add。

<https://blog.csdn.net/quincuntial/article/details/80152314>

图3显示了建造我们的自顶向下特征映射的构建块。使用较粗糙分辨率的特征映射，我们将空间分辨率上采样为2倍（为了简单起见，使用最近邻上采样）。然后通过按元素相加，将上采样映射与相应的自下而上映射（其经过1×1卷积层来减少通道维度）合并。



**----END**