重点学习了解了一种基于深度学习的人流统计模型——YOLOv3-MP-DeepSORT,具体来说，首先基于YOLOv3提出了鲁棒性和泛化性能更加优越的YOLOv3-MP行人检测模型，然后通过与基于检测的多目标跟踪(tracking-by-detection)算法DeepSORT相结合实现行人的实时跟踪，最后设计跨线计数规则实现了双向人流统计。

## YOLOv3-MP 行人检测模型

基于CNN的行人检测方法具有较好的检测效果，但目前仍需要解决两类问题:一种是网络结构参数量过大使得所需储存空间很大，在检测时严重影响实时性;另一种是道路上的行人背景复杂、目标尺寸变化大并且存在不同程度的遮挡，容易漏检。

YOLOv3借鉴了YOLOv1和YOLOv2，虽然没有太多的创新点，但在保持YOLO家族速度的优势的同时，提升了检测精度，尤其对于小物体的检测能力。YOLOv3算法使用一个单独神经网络作用在图像上，将图像划分多个区域并且预测边界框和每个区域的概率。

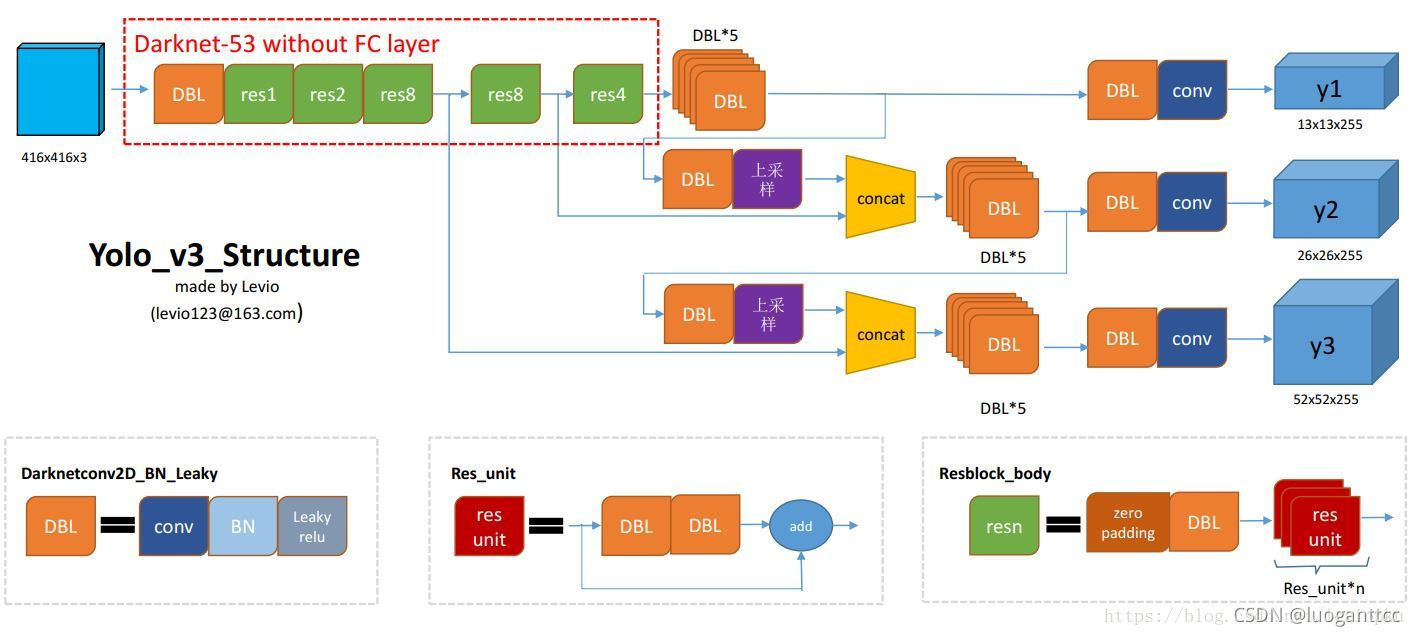


图1 YOLOv3结构

DBL: 如图1左下角所示，也就是代码中的Darknetconv2d\_BN\_Leaky，是yolo\_v3的基本组件。就是卷积+BN+Leaky relu。对于v3来说，BN和leaky relu已经是和卷积层不可分离的部分了(最后一层卷积除外)，共同构成了最小组件。

resn：n代表数字，有res1，res2, … ,res8等等，表示这个res\_block里含有多少个res\_unit。这是yolo\_v3的大组件，yolo\_v3开始借鉴了ResNet的残差结构，使用这种结构可以让网络结构更深(从v2的darknet-19上升到v3的darknet-53，前者没有残差结构)。对于res\_block的解释，可以在图1的右下角直观看到，其基本组件也是DBL。

concat：张量拼接。将darknet中间层和后面的某一层的上采样进行拼接。拼接的操作和残差层add的操作是不一样的，拼接会扩充张量的维度，而add只是直接相加不会导致张量维度的改变。

整个v3结构里面，是没有池化层和全连接层的。前向传播过程中，张量的尺寸变换是通过改变卷积核的步长来实现的，比如stride=(2, 2)，这就等于将图像边长缩小了一半(即面积缩小到原来的1/4)。在yolo\_v2中，要经历5次缩小，会将特征图缩小到原输入尺寸的1/32。输入为416x416，则输出为13x13(416/32=13)。

yolo\_v3也和v2一样，backbone都会将输出特征图缩小到输入的1/32。所以，通常都要求输入图片是32的倍数。

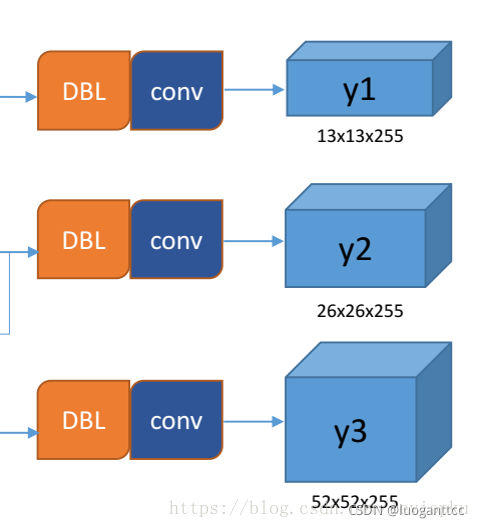


图2

yolo v3输出了3个不同尺度的feature map，如上图所示的y1,y2,y3。这也是v3论文中提到的为数不多的改进点：predictions across scales

这个借鉴了FPN(feature pyramid networks)，采用多尺度来对不同size的目标进行检测，越精细的grid cell就可以检测出越精细的物体。

y1,y2和y3的深度都是255，边长的规律是13:26:52

对于COCO类别而言，有80个种类，所以每个box应该对每个种类都输出一个概率。

yolov3设定的是每个网格单元预测3个box，所以每个box需要有(x,y,w,h,confidence)五个基本参数，然后还要有80个类别的概率。所以3\*(5+80)=255。这个255就是这么来的。（还记得yolov1的输出张量，7x7x30，只能识别20类物体，而且每个cell只能预测2个box）

v3用上采样的方法来实现这种多尺度的feature map，可以结合图1和图2右边来看，图1中concat连接的两个张量是具有一样尺度的(两处拼接分别是26x26尺度拼接和52x52尺度拼接，通过(2,2)上采样来保证concat拼接的张量尺度相同)。作者并没有像SSD那样直接采用backbone中间层的处理结果作为feature map的输出，而是和后面网络层的上采样结果进行一个拼接之后的处理结果作为feature map。

YOLOv3的改进点主要有3个:①借鉴ＲesNet残差网络的思想，保证模型即使在很深的网络结构下，仍能正常收敛;②采用多尺度融合预测方法，在3种不同尺度上进行预测，提高了对小目标检测的能力;③损失函数采用二值交叉熵损失来替换了softmax，在保证了每一个目标的预测准确率的同时使得每个边界框(bounding box)可以预测多个目标。但是，其主干网络Darknet53包括52层卷积层和1个全连接层，巨大的计算量和参数量使得模型在实际应用场景中的达不到实时性要求。而YOLOv3-tiny是一种低参数量的检测方法，只有不到900万的参数量，只需33．7MB的储存空间，有训练速度快、训练时所需显存少、检测速度快等优点。但YOLOv3-tiny对图像的深层特征的提取能力较弱，泛化能力较差，尤其是对光线变化强，遮挡严重的物体以及小物体的检测效果不理想。

考虑 到 以 上 YOLOv3 与 YOLOv3-tiny 的 优 缺 点，本文基于 YOLOv3 提出 YOLOv3-MP 模型，它的 总体架构如图 1 所示，其中输出尺寸 O 满足:

式中I为输入图片的尺寸，K表示卷积核的大小，卷积需要填充的个数为P，S表示步长。在主干网络上，使用基于深度可分离卷积构建的轻量级的深层神经网络MobileNet替换YOLOv3的骨干网络Darknet－53;在预测目标框前，采用K-means聚类重置anchor来加快模型收敛的速度。

### 轻量级网络 MobileNet

MobileNet神经网络采用深度可分离卷积(depthwise separable convolution)，它的核心思想是将原本标准的卷积操作分解成一个深度卷积(depthwise convolution，DW)和一个1×1的点卷积(pointwise convolution，PW)操作，即图3中的Convdw和Conv操作，其中PW的一个卷积核负责一个通道，一个通道只被一个卷积核卷积。与标准卷积相比，假定输出特征图的大小是DF×DF，其中DF是特征图的宽和高，使用卷积核DK×DK，其中Dk是卷积核的宽和高，标准卷积的计算量为:DK×DK×M×N×DF×DF;而深度可分离卷积的计算量为:DK×DK×M×DF×DF+M×N×DF×DF，其中加号前面为DW的计算量，后者为PW的计算量。因此，两者之比为:

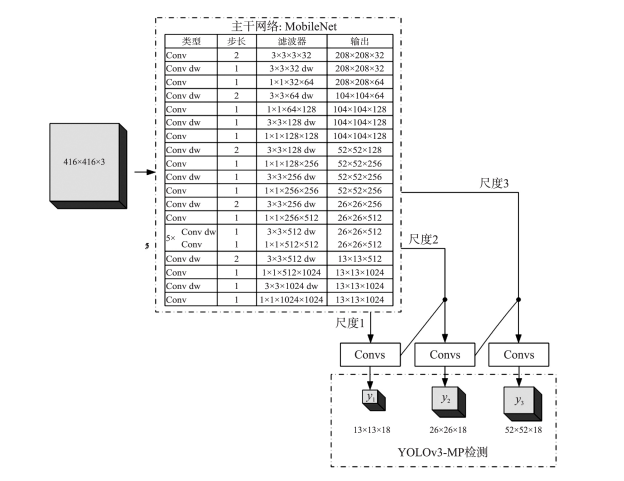


图3

### K-means 重聚

考虑到初始候选框尺寸对行人数据集未进行优化而使初始候选框大小及比例与数据集中行人实际边界框相差大，导致对小尺寸行人漏检率增大的问题，本文借鉴YOLOv3中的anchor boxes机制，即先验框的思想，通过K-means聚类算法来获取最优的anchor个数和宽高维度。其中聚类距离通过重叠度交并比(Intersection over Union，IoU)来反映候选框与真实框之间的误差，其距离公式为:

式中，center为所有簇的中心，box为样本聚类结果，IoU表示所有center与所有box的交并比。

### 交叉熵损失函数

在YOLOv3中误差的计算由边框位置损失lbox、置信度损失lobj和分类误差lclass三部分组成，本文属于单目标检测不考虑分类误差，因此将loss的计算定义为:

## DeepSORT 行人跟踪模型

DeepSORT是SORT算法的改进版本，它在图像空间中通过卡尔曼滤波预测，使用匈牙利算法逐帧关联数据，由关联度量计算边界框的重叠率，在高帧速率下有良好的性能表现。它在处理跟踪问题上的具体过程主要包括轨迹处理与状态估计、信息关联和级联匹配等步骤。

……

## YOLOv3-MP-DeepSOＲT 人流统计 模型

在行人目标计数部分采用虚拟线设置技术，在监控区域设计设置虚拟计数线，通过判断行人跟踪模块获得的行人轨迹(由跟踪窗口中心点组成)是否越线来对人流量进行统计。考虑到行人在监控区域行为的任意性，采用单虚线计数无法对行人的运动方向、在监控区域的滞留情况进行准确判断，因此模型采取如下图4所示的双线虚拟计数，当行人轨迹由A→B视为下行，当越过B线时，下行人数加1;由B→A视为上行，当行人跨过A线时，上行人数加1;上行人数与下行人数的和记为总人数.

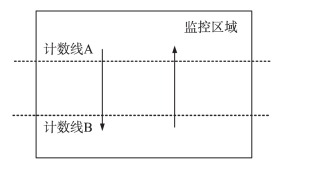


图4

YOLOv3-MP-DeepSOＲT人流统计模型，具体来说，首先在行人检测部分使用改进的行人检测模型YOLOv3-MP对行人目标定位。其中为了提高网络对于细粒度特征的检测，改善对于走远的小尺度行人和遮挡的行人的检测精确度，在数据预处理中对数据进行筛选、标注，并在输入训练网络前进行旋转和HSV(Hue Saturation Value)的随机增强，使得算法对于行人的复杂运动以及多变的光照条件具有更强的鲁棒性;行人跟踪部分采用tracking-bydetection的DeepSORT算法，将行人检测框的结果写进跟踪队列进行轨迹处理与状态估计，而后通过信息关联和级联匹配进行实时跟踪。最后，采用跨线计数方案实现双向计数，整体算法流程如图5所示。

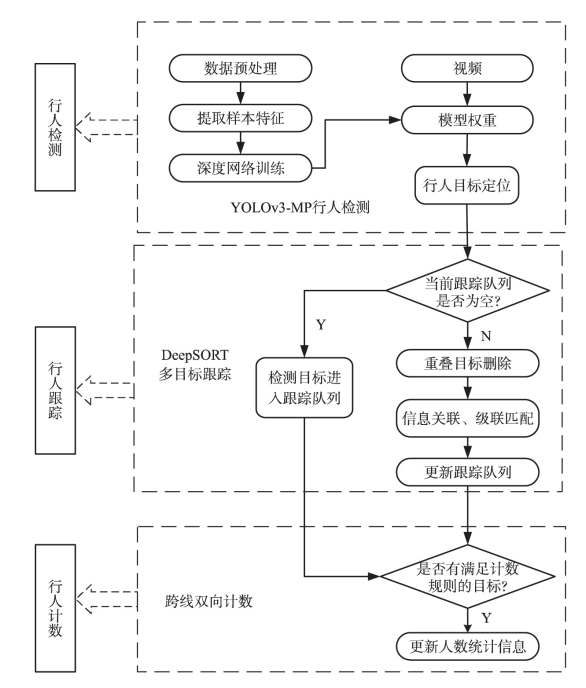


图5