一、问题背景

人们出行时经常乘坐地铁、火车、高铁、飞机等交通工具。在这种人流量大的地方，为了保证乘客的安全，需要对乘客携带的行李使用安检机进行检查。经验丰富的安检员可以从安检机生成的X光图像中识别并找出危险品的存在。但在大量的检查中难免会出现漏检的情况下。因此辅助安检员判图的危险品检测算法开发就成为必要的事情。

二、目标任务（遮挡问题）

安检图片都存在不同等级的遮挡问题，遮挡会严重影响检测器识别危险品的准确率。如何解决严重遮挡条件下模型检测危险品问题是一个热点。

安检机返回的x光图像为RGB彩色图像。我们使用训练集中的危险品包括带电芯充电宝和不带电芯充电宝两个类别。训练集中共有6000张图片（带电芯充电宝和不带电芯充电宝各3000张，且测试集根据遮挡等级分为了1,2,3种不同的遮挡等级，不同遮挡等级的示意图见图1）。每张图片都拥有一个危险品所在的位置标注文件，标注文件里的每行表示（危险品的名称，危险品位置的左上坐标，危险品位置的右下坐标）。

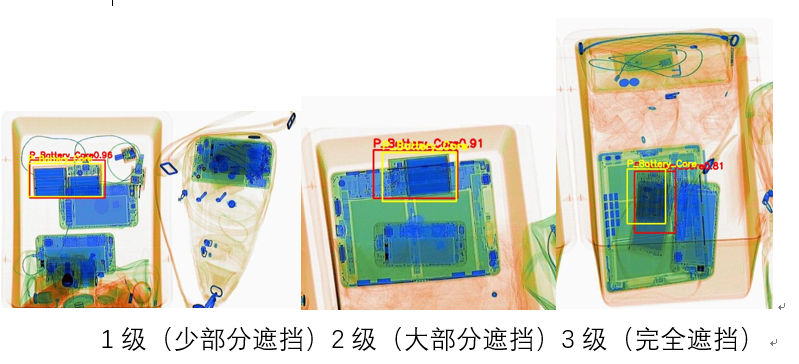


图1 目标检测模型的结果示意图

黄色框是GroundTruth，红色框是预测结果，数字为置信度。

三、测试说明

运⾏测试命令

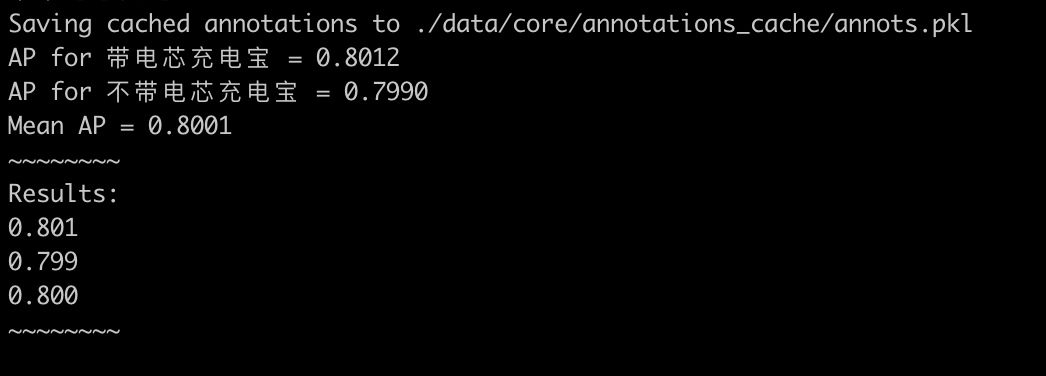
准备好数据以后，在项目目录下，执行pip install -r requirements.txt安装依赖。

之后运⾏以下命令进⾏测试：

**python eval.py --dataset\_root 数据集路径(./data/core)**

**例如：python eval.py --dataset\_root ./data/core**

执⾏完后，会输出以下信息：



测试完成之后，如果选择其他测试数据，需要先删除缓存，即项目目录下的ssd300\_120000文件夹和数据路径下的annotations\_cache文件夹和result文件夹。

四、数据预处理

1、重要性分析

在进行目标检测等相关任务时，图片预处理有非常重要的作用。好的预处理可以消除无关噪音，增强目标特征，使得任务难度大大降低。

2、数据集分析

根据x射线安检图的原理可知，X光图片原本是灰度图，为了使得图像更加清晰可辨认，会对原始图像进行彩色增强，生成我们在安检机上常见的伪彩色。这样会使得不同的物质展现出不同的色彩。通常，有机物品被定义为橙色(或黄色)，无机物质被定义为蓝色，混合物被定义为绿色。具体而言：火药、药品、塑胶、纸、布、木料、 液体、和纸等有机化合物，被显示的色调为橘色或淡黄色。大部分的金属材料、铁、铜、锌、镍、钢等显示信息为深蓝色，相对密度小的无机化合物呈浅。相对密度很大的无机化合物呈暗蓝色。绿色表示有机物质和无机物质重叠。如果混合物的主要成分是有机的，图像是浅绿色的，否则是绿色的。

3、预处理具体步骤

①去除无关色彩

通过观察数据集，我们发现充电宝的外壳呈现出绿色或浅蓝，内核呈现出深蓝色。因此，我们可以通过过滤特定色彩的方式去除无关物质，降低其他形状类似的物品的干扰。

具体而言，我们首先将色彩空间从RGB变换到HSV，在去除无关颜色之后，我们再将颜色空间变回RGB。HSV是色调（H），饱和度（S），明度（V）。在这种色彩空间可以更加容易地分离颜色。通过实验，我们发现保留蓝、绿、青可以在保持充电宝完整形态的同时尽可能地去除其他无关物品。

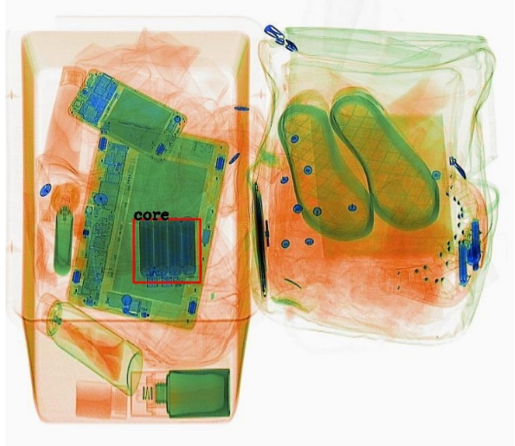
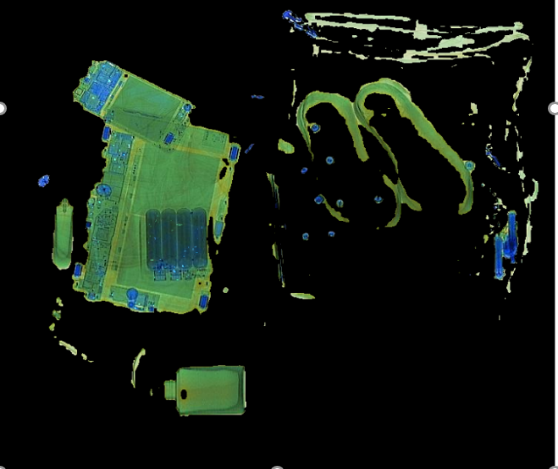
②增加颜色对比度

我们组抽到的任务是遮挡问题。通过观察样本图片，我们发现被遮挡的物体的重叠部分颜色会加深，就算完全被遮挡，可以通过颜色深浅辨认出来。为了加强这种色差，我们进行了颜色对比度的增强。

具体而言：我们首先将色彩空间从RGB转换为YUV，在增强对比度之后再将其转换回RGB。这样做的原因是，直方图均衡化等技术大多是应用于灰度图，而无法应用于彩色图片。我们将颜色空间转换成YUV之后，就可以将灰度（Y）和色度（UV）分离。这样我们就可以对Y通道进行直方图均衡化，以加强灰度对比度了。灰度其实在反映在彩色图片中就是颜色的亮度，也就是颜色深浅，这样处理之后就可以加强颜色深浅了。

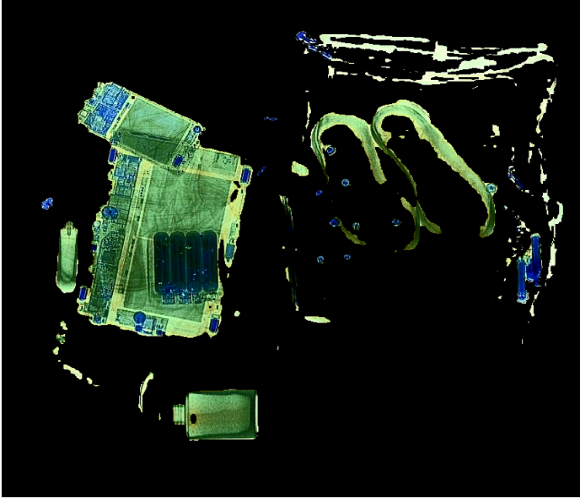
③加强边界线

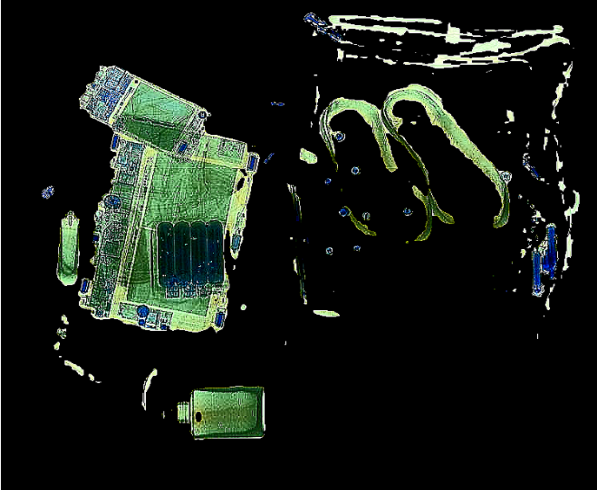
为了更好地识别遮挡物体，我们希望加强物体的边界线，以使得物体更加容易辨认。因此我们使用简单的锐化操作，突出物体的轮廓。效果展示：



去除无关颜色

原图





锐化后

增加对比度

五、模型

(一)模型选择

我们选用的是SSD（ Single Shot MultiBox Detector）模型，原因如下：

1、SSD 是one-stage方法，简单，快速。

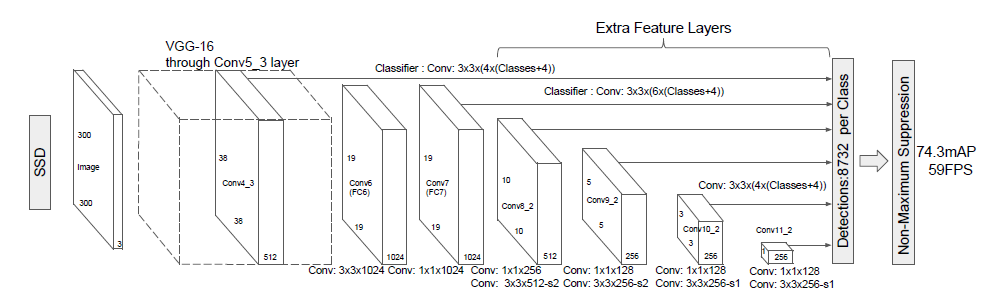
2、SSD提取了不同尺度的特征图来做检测，大尺度特征图（较靠前的特征图）可以用来检测小物体，而小尺度特征图（较靠后的特征图）用来检测大物体；

3、SSD采用了不同尺度和长宽比的先验框（Prior boxes, Default boxes在Faster R-CNN中叫做锚，Anchors）。解决了输入图像目标大小尺寸不同的问题，同时提高了精度，可以理解为一种特征金字塔；相比同是one-stage的Yolo算法，在一定程度上克服难以检测小目标，而且定位不准的缺点。

4、相比于Faster R-CNN，SSD提出了一个彻底的end to end的训练网络，保证了精度的同时大幅度提高了检测速度，且对低分辨率的输入图像的效果很好；

(二)模型解析

1、模型结构



将VGG16的FC6和FC7改成卷积层，又在后面添加了三个尺寸大小逐级减小的卷积层和一个平均池化层。具体用于分类回归的层有：Conv4\_3、Conv7、Conv8\_2、Conv9\_2、Conv10\_2和Pool11。最后contact后传给loss层。利用不同层次的特征图来预测offset（真实边界框相对锚框的偏移量，简称偏移量）和confidence（置信度），可以检测不同尺寸的物体。

结构特点：

①多尺度特征图（Mult-scale Feature Map For Detection）

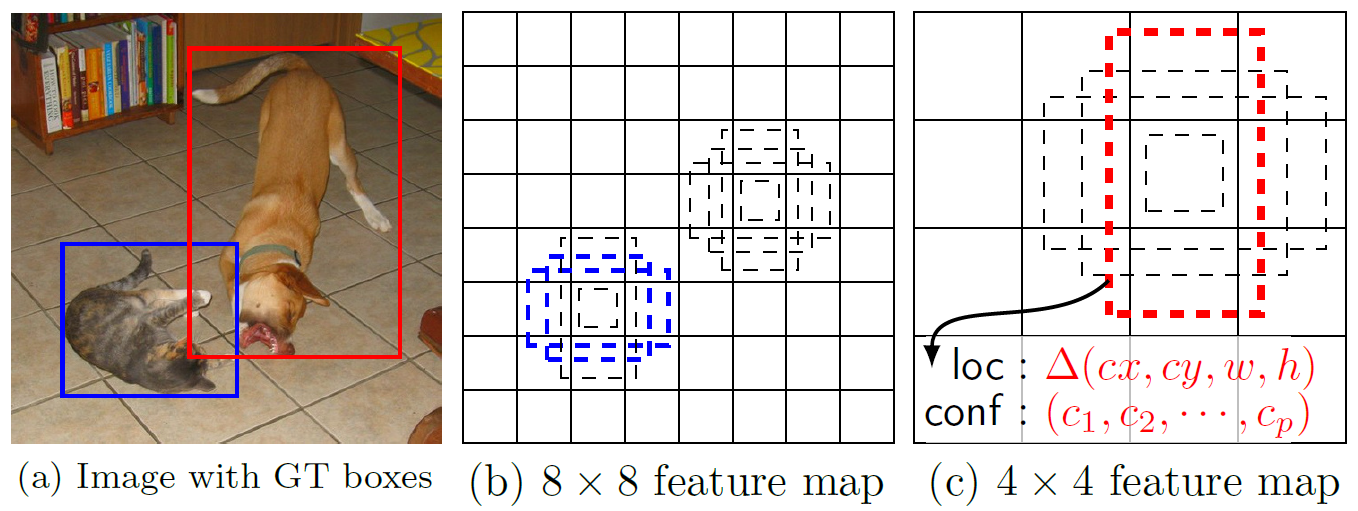
在图像Base Network基础上，将Fc6，Fc7变为了Conv6，Conv7两个卷积层，添加了一些卷积层（Conv8，Conv9，Conv10，Conv11），这些层的大小逐渐减小，可以进行多尺度预测。

②卷积预测器（Convolutional Predictors For Detection）

每个新添加的卷积层和之前的部分卷积层，使用一系列的卷积核进行预测。对于一个大小为m\*n大小，p通道的卷积层，使用3\*3的p通道卷积核作为基础预测元素进行预测，在某个位置上预测出一个值，该值可以是某一类别的得分，也可以是相对于Default Bounding Boxes的偏移量，并且在图像的每个位置都将产生一个值。

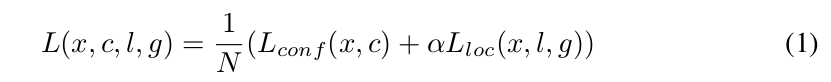
③默认框和比例（Default Boxes And Aspect Ratio）

在特征图的每个位置预测K个Box。对于每个Box，预测C个类别得分，以及相对于Default Bounding Box的4个偏移值，这样需要(C+4)\*k个预测器，在m\*n的特征图上将产生(C+4)\*k\*m\*n个预测值。这里，Default Bounding Box类似于Faster-RCNN中Anchors，如下图所示。



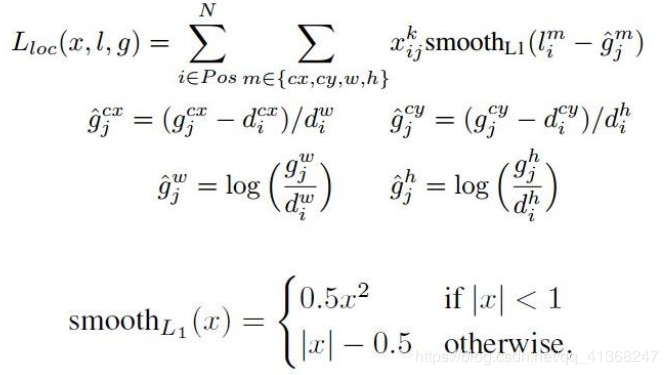
2、损失函数

损失函数定义为位置误差（locatization loss， loc）与置信度误差（confidence loss, conf）的加权和，

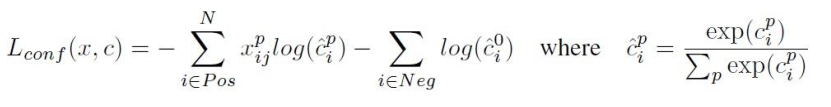


其中N是先验框的正样本数量。c为类别置信度预测值。为先验框的所对应边界框的位置预测值，而g是ground truth的位置参数。权重系数α通过交叉验证设置为1。

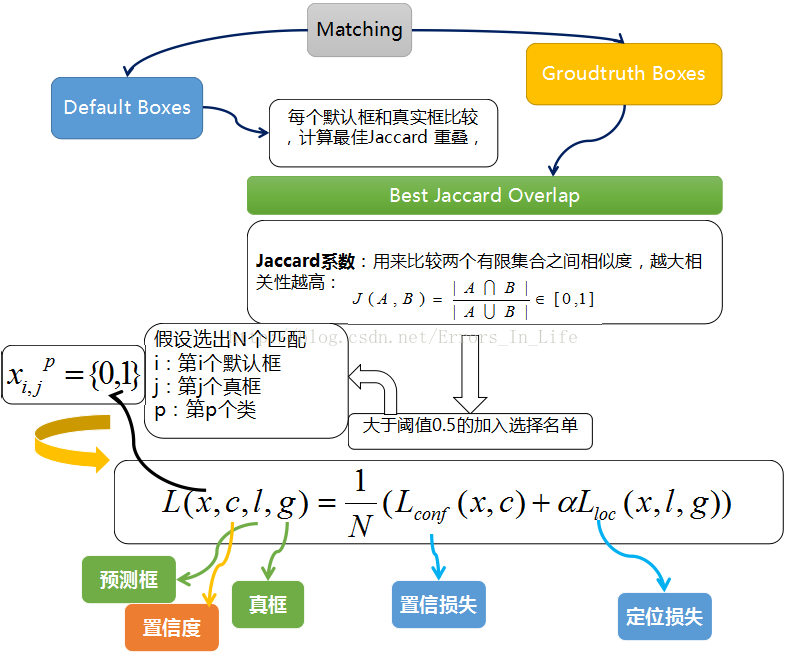
对于位置误差，其采用Smooth L1 loss，定义如下



对于置信度误差，其采用softmax loss，定义如下：



3、训练过程



训练过程中的 prior boxes 和 ground truth boxes 的匹配，基本思路是：让每一个 prior box 回归并且到 ground truth box，这个过程的调控需要损失层的帮助，它会计算真实值和预测值之间的误差，从而指导学习的走向。

SSD 训练的目标函数具体过程是让每一个 prior box 经过Jaccard系数计算和真实框的相似度，阈值只有大于 0.5 的才可以列为候选名单；假设选择出来的是N个匹配度高于百分之五十的框，令 i 表示第 i 个默认框，j 表示第 j 个真实框，p表示第p个类。那么，总的目标损失函数（objective loss function）就由 localization loss（loc） 与 confidence loss（conf） 的加权求和。

4、模型特点

①提出了SSD目标检测方法，在速度上，比之前最快的YOLO还要快，在检测精度上，可以和Faster RCNN相媲美

② SSD的核心是在特征图上采用卷积核来预测一系列default bounding boxes的类别分数、偏移量

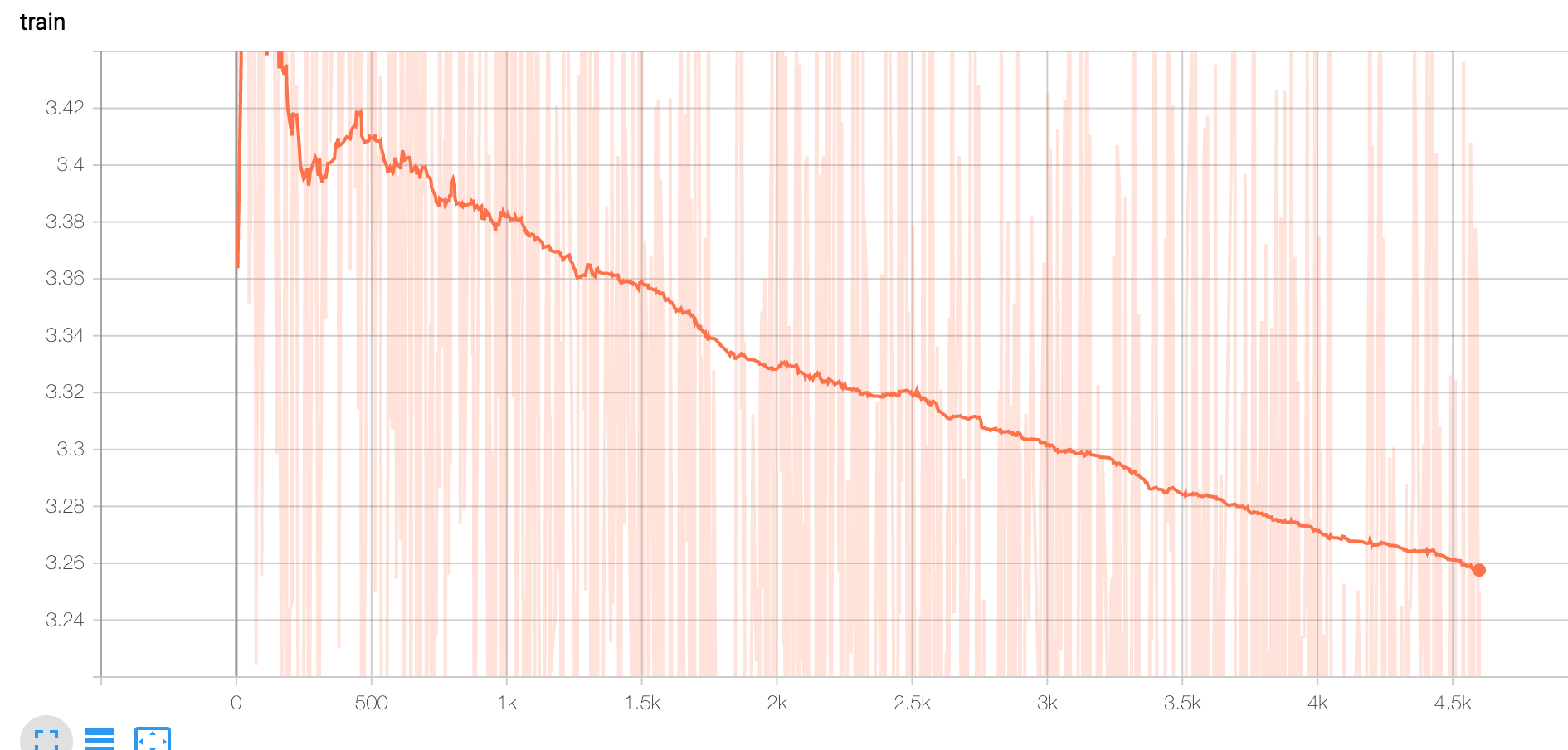
③提高检测准确率，在不同尺度的特征图上进行预测，此外，还得到具有不同aspect ratio的结果

④这些改进设计，实现了end-to-end训练，并且，即使图像的分辨率比较低，也能保证检测的精度

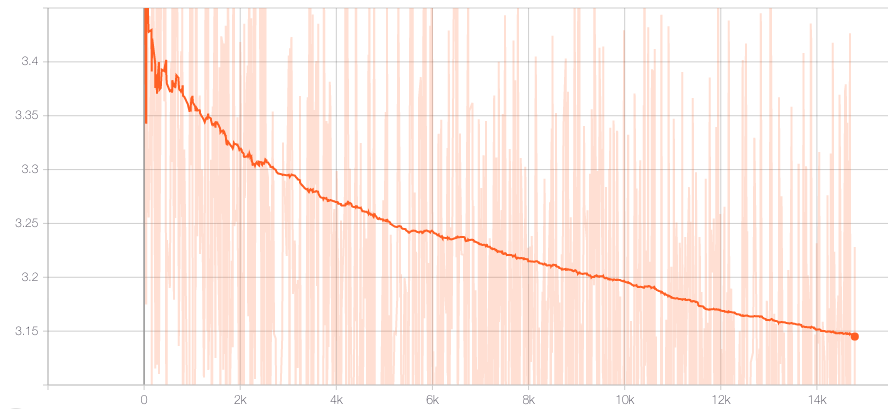
六、参数调优

1、对学习率进行调整，从原本的1e-3调整为1e-4，保证了训练过程的顺利进行，避免了模型loss出现Nan的情况。

2、最初始“总的目标损失”大小为20左右,当以学习率le-4时，从20降到3.5用时需要7天,此时Mean\_AP=0.6801。后续总体loss在想减少，速度非常缓慢。如下图



3.当总体损失为3.8，总体训练为9w次时,以学习率le-4，gamma=0.9，每1000轮进行一次对比“总体损失”，然后调整学习率，发现总体损失会在2.75-3.4来回浮动, Mean\_AP在0.7657~0.7619.虽说总体损失依旧是下降趋势如下。



4. 当总体损失为3.8，总体训练为9w次时,以学习率le-6，gamma=0.8，每1000轮进行一次对比“总体损失”，总体损失不会下降，Mean\_AP会在0.69左右徘徊。

5. 当总体损失为3.8，总体训练为9w次时,以学习率le-4，gamma=0.9，每1000轮进行一次对比“总体损失”，然后调整学习率，如果训练次数达到14w次时，Mean\_AP会达到0.81。

七、反思与展望

1、在训练模型之前，数据集的划分需要仔细的确定，训练集少了，模型得不到充分的训练，测试集少了，模型的验证也不够充分。

2、数据预处理是一个很重要的环节，在训练之前，将图片上的无关噪声去除，做适当的图像处理，可以加快收敛速度。

3、模型的参数对于模型的训练需要反复优化，尤其是学习率，学习率太大，可能会导致学习不稳定，即使loss在下降，但是在测试集上的表现也很差，学习率太小，会导致学习不够充分。

4、随着深度模型层数不断加深，我们需要不断加大样本数据，训练时间才能找到一个更好的模型，达到项目规定的指标需求。

5、目标损失函数设计的好坏，能够决定模型收敛的速度

6、学习率的设置，设置过高，过低都会影响模型的收敛速度。