**地铁车站监控视频内容分析与**

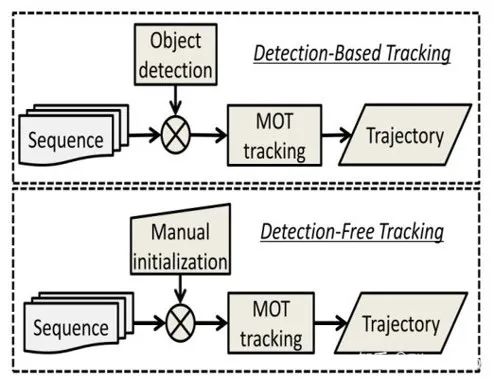
**技术要求(Ver 0.1)**

## 1.MOT多目标跟踪

## 1.1 MOT多目标跟踪简介

目标跟踪问题分为单目标跟踪问题和多目标跟踪问题。单目标跟踪是指在视频的初始帧上框出单个目标，然后预测后续帧中该目标的大小和位置，单目标跟踪典型算法有:Mean Shift、TLD(基于在线学习的跟踪)、KCF(基于相关滤波性)。多目标跟踪不像单目标跟踪一样先在初始帧上框出单个目标，而是追踪多个目标的大小和位置，且每一帧中目标的数量和位置都会有变化，所以多目标跟踪会产生额外的冗余，常用的多目标跟踪算法有：SORT、DeepSORT等。

多目标跟踪MOT(Multi-Object-Tracking)：主要任务是给定一个图像序列，找到图像序列中运动的物体，并将不同帧的运动物体进行识别，也就是给定一个确定准确的ID，这些物体可以是任意的，物体可以是人、车辆、各种动物等。



如上图所示，多目标跟踪算法可分为基于检测的多目标跟踪DBT（Detection-Based Tracking）和基于初始框无需检测器的多目标跟踪DFT(Detection-Free Tracking)。在单目标跟踪中，我们往往会使用给定的初始框，在后续视频帧中对初始框内的物体进行位置预测。而多目标跟踪算法，大部分都是不考虑初始框的，原因是目标的频繁消失与产生问题。

在多目标跟踪领域常用的跟踪策略是TBD（Tracking-by-Detecton），又或者也可叫DBT（Detection-Based-Tracking）。即在每一帧进行目标检测，再利用目标检测的结果来进行目标跟踪，这一步也称为数据关联(帧间关联性问题)。而DFT(Deteciton-Free Tracking),常常用于需要人工标定视频的第一帧图像中的目标，之后边检测边跟踪，常用于单目标跟踪，因为DFT目标需要人工标定，对于非第一帧出现的目标或者中间帧消失的目标没办法处理。

多目标跟踪又可以分为在线跟踪(Online Tracking)和离线跟踪（Offline Tracking）。在线跟踪（Online）是指当前帧的预测只能使用当前帧与之前的帧的信息来进行跟踪，也就是说视频中的第3帧只能使用视频之前的第2帧和第1帧的目标的信息来进行跟踪，Online跟踪是不允许修改以往的跟踪结果。离线跟踪(Offline) : 没有在线跟踪只能使用之前帧的局限，对于每一帧的预测，离线跟踪都能使用整个视频的信息，更容易获得一个全局最优解，Offline跟踪是可以修改以往的跟踪结果。

多目标追踪算法主要解决的问题是对视频中每一帧画面里我们标定或者想要 追踪的目标进行检测并获取在图像中的位置，对每个目标分配一个 id，在目标运动 过程中，维持每个目标的 id 保持不变。在多目标追踪问题中，目标追踪算法需要对视频中每帧图像里的所有目标进行检测，将检测到的新目标与已经分配轨迹的目标进行匹配，如果匹配成功，将归类于已有轨迹的目标中，对没有匹配成功的目标，将其归类为新出现的目标，需要分配一个新的 id，对于离开视频区域的目标， 将不再对其进行轨迹追踪，并在已有轨迹的集合中将其删除。

## 1.2 MOT多目标跟踪存在问题

MOT多目标跟踪，需要处理以下几个问题：

1. 处理新目标的出现和老目标的消失(跨镜头跟踪)；

**B**. 跟踪目标的运动预测和相似度判别，即上一帧与下一帧目标的匹配(帧间相关性)；

**C**. 跟踪目标之间的重叠和遮挡处理(消除冗余和错检漏检)；

**D**. 跟踪目标丢失一段时间后再重新出现的再识别(ReID)。

## 1.3 MOT多目标跟踪算法工作流程(检测，跟踪，重识别)

第一步：给定视频的原始帧；

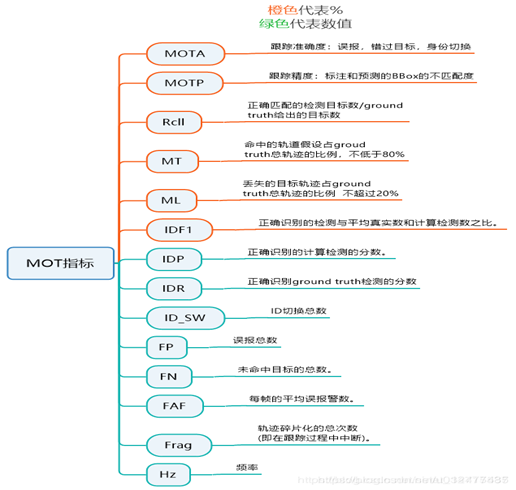
第二步：运行对象检测器，如Faster Rcnn、YOLO等目标检测算法，检测以获得对象的边界框；

第三步：将所有目标检测框中对应的目标抠出来，进行特征提取(包括表观特征或者运动特征)；

第四步: 进行相似度计算，计算前后两帧目标之间的匹配程度（前后属于同一个目标的之间的距离比较小，不同目标之间的距离比较大）；

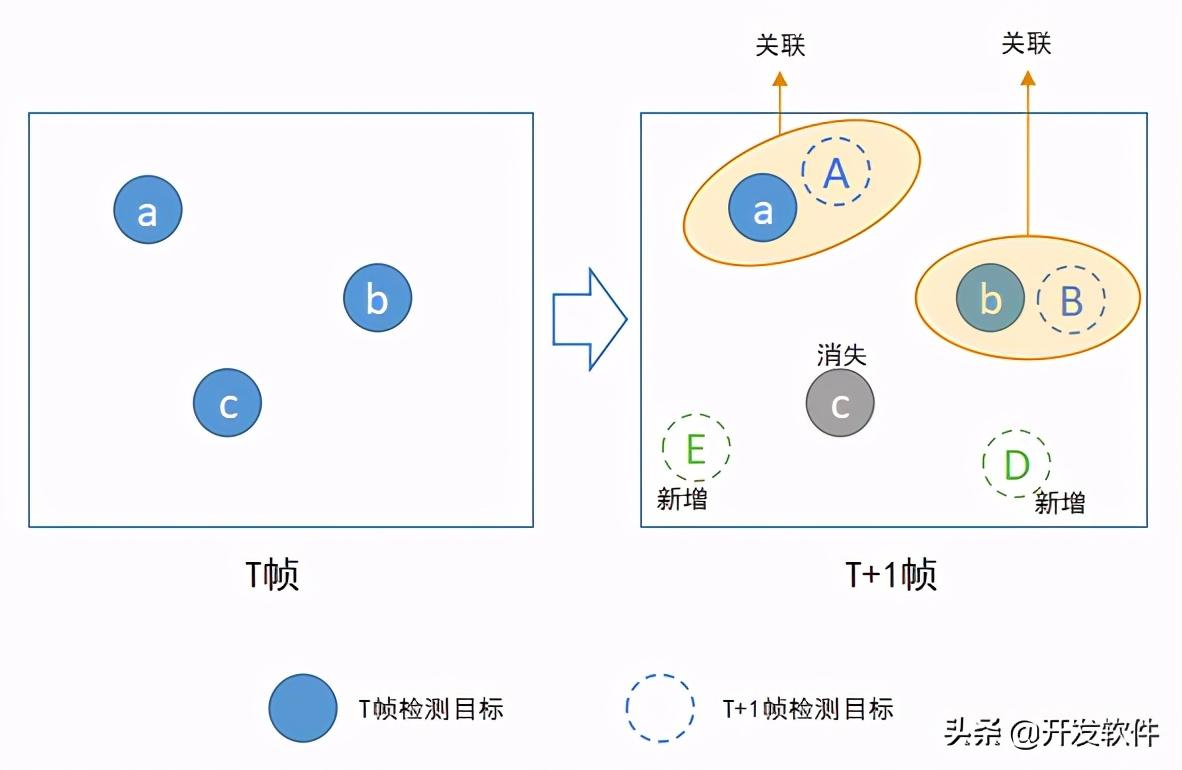
第五步：数据关联，为每个对象分配目标的ID，常用匈牙利算法。

## 1.4 MOT多目标跟踪算法评价指标

对于多目标跟踪，最主要的评价指标就是MOTA，这个指标综合了三点因素：FP、FN、IDsw。FP即FalsePostive，为误检测的目标数量；FN即FalseNegetive，为未检出的真实目标数量；IDsw.即同一目标发生ID切换的次数。MOTA越高，代表一个Tracker综合性能越好，上限为100，下限负无穷。

## 1.5 MOT多目标跟踪目标关联原理

目标跟踪的本质是关联视频前后帧中的同一物体（目标），第T帧中有M个检测目标，第T+1帧中有N个检测目标，将前一帧中M个目标和后一帧中N个目标一一关联起来，并赋予唯一标识TrackID，这个过程就是Tracking By Detecting跟踪算法的宏观流程。



上图描述目标关联的具体流程，在实际目标关联过程中，我们需要考虑的有：

1、如何处理中途出现的新目标；

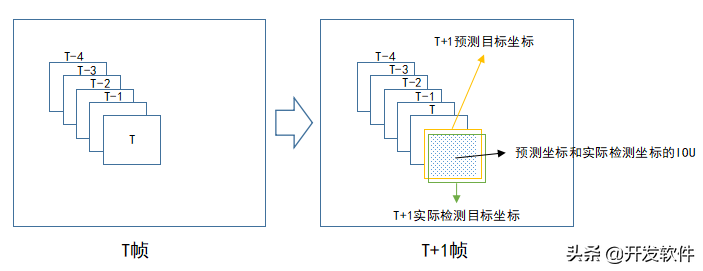
2、如何处理中途消失的目标；

3、正确目标关联。

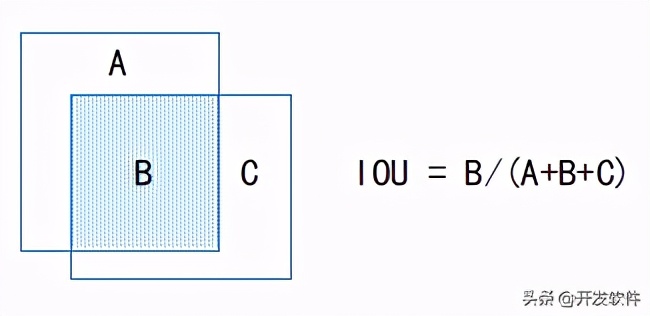
理想情况下，同一个物体（目标）在视频画面中从出现到消失，跟踪算法能赋予它唯一一个标识（TrackID），不管目标是否被遮挡、目标是否发生严重形变、是否和其他目标相距太近（相互干扰），只要这个目标被正确检测出来，跟踪算法都应该能够正确关联上。但实际上，物体遮挡是跟踪算法最难解决的难题之一，物体被频繁遮挡是TrackID变化的主要原因。原因很简单，物体被遮挡后（或其他原因），检测算法检测不到，跟踪算法无法连续关联到每帧的数据，等该物体再出现时，物体在画面中的位置、物体的外观形状与消失之前相比都发生了很大变化，而跟踪算法恰恰主要是根据物体的位置、外观来进行数据关联的。下面主要介绍目标跟踪中两种方式，一种容易实现、速度快，算法纯粹基于目标在画面中的位置来进行数据关联；另一种相对复杂，速度慢，算法需要提取前后帧中每个目标的图像特征（features），然后根据特征匹配去做数据关联。

1. **基于坐标的目标关联**

基于坐标（目标中心点+长宽）的目标关联是相对简单的一种目标跟踪方式，算法认为前后帧中挨得近的物体为同一个目标，因为物体移动是平滑缓慢的，具体可以通过IOU（交并比，前后两帧中目标检测方框的重叠程度）来计算，这种算法速度快、实现容易，在前面检测算法相对稳定的前提下，这种跟踪方式能够取得还不错的效果，由于速度快，这种方式一般可以用于对实时性（realtime）要求比较高的场合。缺点也很明显，因为它仅仅是以目标的坐标（检测算法的输出）为依据进行跟踪的，所以受检测算法影响非常大，如果检测算法不稳定，对于一个视频帧序列中的目标，检测算法经常漏检，那么通过这种方式去跟踪效果就非常差。另外如果场景比较复杂，目标比较密集，这种跟踪方式的效果也不会太好，因为目标密集，相邻目标的坐标（left、top、width、height）重合度比较高，这给基于坐标的目标关联带来困难。



如上图，在T+1帧中，我们根据目标前面若干帧的坐标预测它在本帧中的坐标（预测坐标），然后再将该预测坐标与本帧实际检测的目标坐标进行数据关联。之所以需要先进行预测再关联，是因为为了减少关联过程的误差，常见预测算法可以使用卡尔曼滤波，根据目标前面若干坐标值预测下一坐标值，并且不断地进行自我修正，卡尔曼滤波算法网上有开源代码。IOU（交并比）是衡量两个矩形方框的重叠程度，IOU值越大代表矩形框重叠面积越大，它是目标检测中常见的概念。在这里，我们认为IOU越大，两个目标为同一物体的可能性越大。

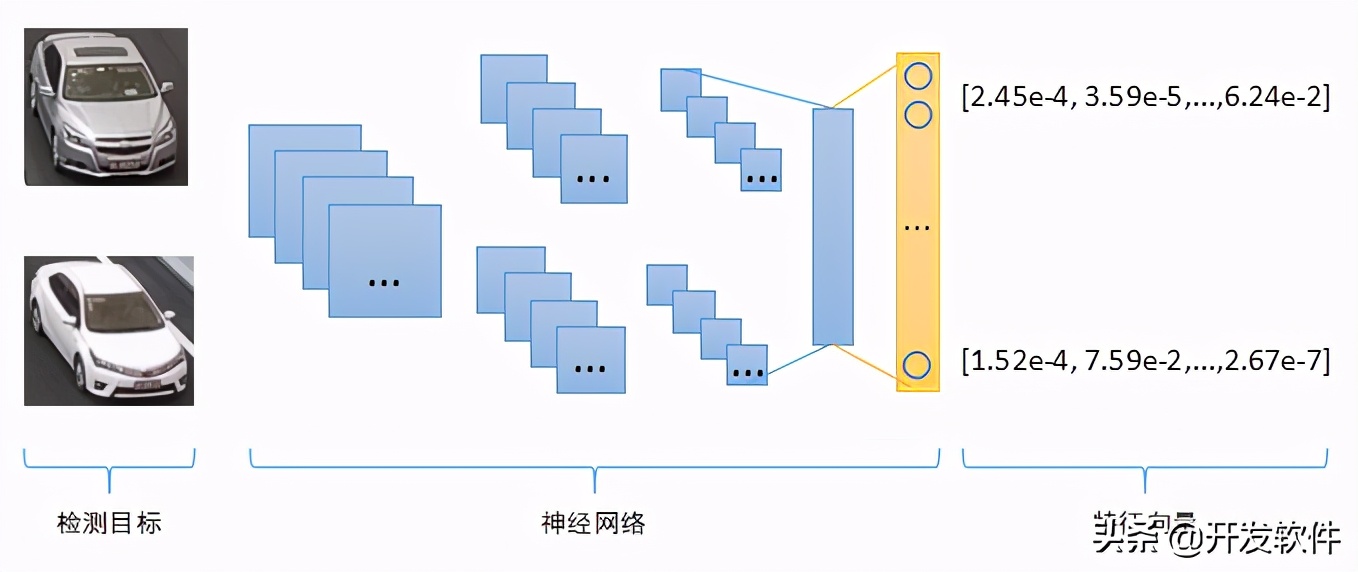


1. **基于特征的目标关联**

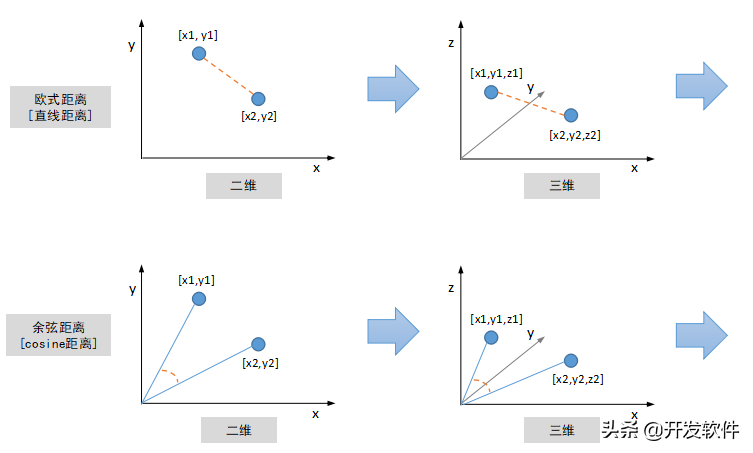
纯粹基于坐标的目标跟踪算法有一定的局限性，单靠目标坐标去关联前后帧的同一目标在有些场合下效果比较差。在此基础上，有人提出结合目标外观特征匹配做目标关联，换句话说，在做目标关联的时候，除了依赖目标坐标外，还考虑目标的外观特征，道理很简单：前后两帧中挨得近的物体且外观长得比较像的物体为同一目标。

在计算机视觉中，有一个专门的研究领域叫Target Re-Identification（目标重识别），先通过对两个待比较目标进行特征编码（特征提取），然后再根据两个特征的相似度，来判断这两个目标是否为同一个物体，两个特征越相似代表两个目标为同一个物体的可能性越大。Target Re-Identification常用在图像搜索、轨迹生成（跨摄像机目标重识别）以及目标跟踪。

神经网络的主要作用就是对原始输入数据进行特征编码，尤其在计算机视觉中，卷积神经网络主要用于图像的特征提取（Feature Extraction），从二维图像中提取高维特征，这些特征是对原始输入图像的一种抽象表示，因此训练神经网络的过程也可以称为Representation Learning。相同或者相似的输入图片，神经网络提取到的特征应该也是相同或者相似的。我们只要计算两个特征的相似度，就可以判断原始输入图像的相似性。



计算两个图像特征的相似度：图像特征的数学表示是一串数字，组合起来就是一个Vector向量，二维向量可以看成是平面坐标系中的点，三维向量可以看成立体空间中的点，依次类推，因此图像特征也被称作为“特征向量”。有很多度量标准来衡量两个特征向量的相似程度，最常见的是“欧式距离”，即计算两点之间的直线距离，二维三维空间中两点之间的直线距离我们都非常熟悉，更高维空间中两点距离计算原理跟二三维空间保持一致。另外除了“欧式距离”之外，还有一种常见距离度量标准叫“余弦距离”，计算两个向量（点到中心原点的射线）之间的夹角，夹角越小，代表两个向量越相似。



外观特征提取是一个耗时过程，因此对实时性要求比较高或者需要同时处理视频路数比较多的场合可能不太适合。但是这种基于外观特征的跟踪方式效果相对更好，对遮挡、目标密集等问题鲁棒性更好，因为目标遮挡再出现后，只要特征提取网络训练得够好，目标尺寸、角度变化对它的外观特征影响不大，因此关联准确性也更高。

## 2.YOLOv5（特征提取&目标检测）

在官方发布的代码中，共有四个检测网络。其中，YOLOv5s具有最小深度和特征图宽度，其他三种则是在该网络上进行了拓展。实际上，YOLOv4与v5具有相似的结构，差异主要体现在如下两点。

（1）YOLOv5新增的Focus结构主要为对图片的切片。对输入图片，一部分区域，对它的宽和高，每隔两个步长，从0开始取值；而对别的区域，从1开始取值；以此类推，该图片的三个通道均使用如上的操作。最后把三个通道的切片，按照通道连接。

（2）CSP结构上的差异 YOLOv4 中，CSP的使用只体现在 Backbone，而 YOLOv5 还在 Neck 使用了 CSP2\_X。YOLOv5 将具有残差结构的 CSP1\_X应用在 Backbone中，因为该网络较深，让层间反向传播时，梯度值增强，这样防止了网络加深时导致的梯度消失。而在 Neck 中，把主干网络的输出分成了两个分支，再通过concat操作，这样让网络对特征的融合能力得到有效提升，也让更多的特征信息得到保留。

在比较 YOLO算法网络模型中的骨干网络及它们在不同数据集上的检测速度和精度之后可以发现，在使用残差网络后的 YOLOv3，精度得到很大提高，之后的版本由于要在速度上进行提升，检测精度难免下降。而实际上，最新的YOLOv5 可以在权衡精度和速度的同时，缩小模型尺寸，使其部署成本更低，让图像检测更加便捷，进一步说明使用该模型进行图像检测的可行性。

## 3.DeepSORT（目标跟踪）

在 详 细 介 绍 DeepSORT 算 法 前， 首 先 介 绍SORT 算法的核心：匈牙利算法和卡尔曼滤波。

1. **匈牙利算法**

匈牙利算法主要为分配问题的解决算法，问题的阐述如下：假设有 N 个人和 N 个任务，每个任务可以任意分配给不同的人，每个人完成每个任务花费的代价不相同，求分配方式使得总代价最小。该算法基于以下定理：如果代价矩阵的某一行或某一列同时加上或减去某个数，则这个新的代价矩阵的最优分配仍为原代价矩阵的最优分配。

算法流程：

（1）减去每一行/列中的最小元素；

（2）以最少的水平或垂直线覆盖所有的0；

（3）若水平线或垂直线的数量为 N，那么找到了最优分配，否则继续；

（4）找到矩阵中未被线覆盖的最小元素，对所有未覆盖的行减去该值，所有未覆盖的列加上该值，回第二步。匈牙利算法在 DeepSORT算法中的作用，主要

是将前一帧中的跟踪框与当前帧中的检测框进行关联，通过外观信息和马氏距离，或者 IOU 匹配来计算代价矩阵。

1. **卡尔曼滤波**

卡尔曼滤波主要基于传感器的测量值对预测值进行更新，以获得更为准确的估计值。在目标跟踪时，需要估计跟踪框的两个状态：

（1）位置信息：用均值表示，由对象中心坐标、宽高比、高，以及速度变化值等构成。

（2）不确定性：用协方差表示，为8x8的对角矩阵，矩阵中的数字越大说明越具有不确定性。

卡尔曼滤波主要有两个阶段：

（1）基于前一时刻的状态预测当前时刻的状态以及跟踪框下一时刻的位置。基于以下公式：x'=F（x）：x为跟踪框在前一时刻的均值，F为状态转移矩阵，该公式的结果 x' 为预测的当前时刻的值。这里的卡尔曼滤波是一个匀速模型。P'=FPFT+Q：P为跟踪对象位置在前一时刻的协方差，Q是系统的噪声矩阵，衡量系统的可靠程度，初始化为极小值，此公式用来预测当前时刻的协方差矩阵。

（2）根据当前时刻的检测框，更新预测的位置，修正有关跟踪框的当前状态。基于以下公式：y=z-Hx'：z 为检测对象的均值向量，H为测量矩阵，把跟踪对象的均值向量 x' 映射到检测空间。用于计算检测框和跟踪框之间的均值误差。

S=HP'HT+R：R为噪声矩阵，该矩阵为4x4的对角矩阵，对角线上的值为中心点两个坐标以及宽高的噪声值。将协方差矩阵映射到检测空间，然后再加上噪声矩阵 R。k=P'HTS-1：得到卡尔曼增益，对误差的重要程度进行估计。x=x'+ky、P=（I-KH）P'：计算更新后的均值向量和协方差矩阵。

1. **整体流程**

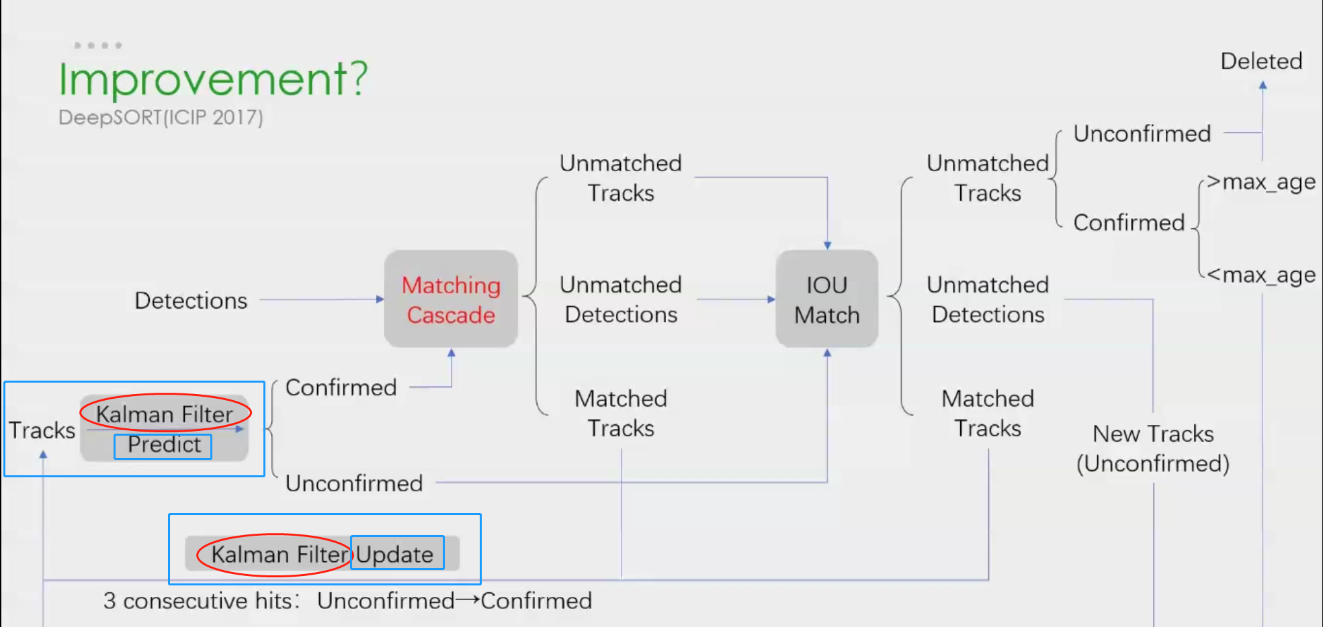
（1）预测状态：上一轮产生的跟踪目标经过卡尔曼滤波的预测，计算出本轮的均值和协方差矩阵，状态保持不变。

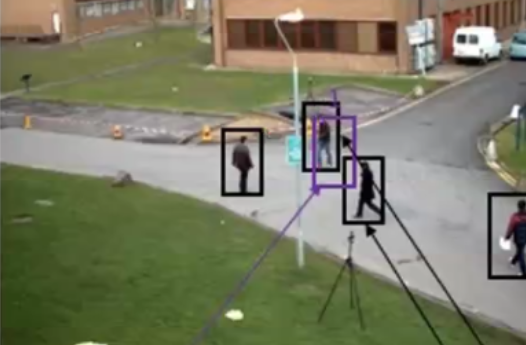
（2）第一次匹配：将（1）中跟踪的目标和本轮目标检测器的检测结果一起进行匹配，产生三种状态的结果，分别为未匹配的跟踪对象、未匹配的检测对象以及已匹配的跟踪对象。

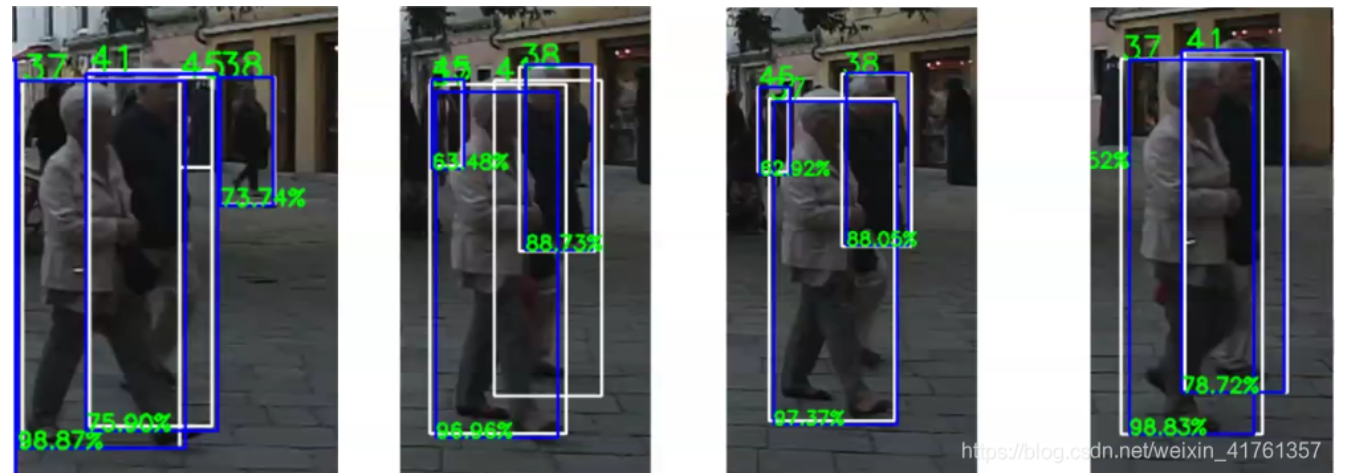
（3）第二次匹配：（2）的检测产生遗漏，将它们与（1）中未确认的目标合并，用IOU匹配再进行一次匹配。产生和上步同样的三种状态的结果。

（4）处理失效对象： 未匹配的目标中未确认的和已确认但总帧数超过阈值的状态设为删除。

（5）输出结果并为下一轮准备数据，合并下面三个来源的对象：①将（3）和（4）中的已匹配的目标合并，进行卡尔曼滤波的更新，同时将总帧数加 1，输出跟踪框；②（3）中的未匹配的检测框新建跟踪框；③（4）中已确认而且未超过阈值的跟踪框。这三种跟踪框作为本轮的输出也迭代进入下一轮输入，继续进行（1）。

对于跟踪，算法并不知道自己跟踪的目标内容是什么，它只负责跟踪，具体是什么并不需要知道，那么就会出现重识别（ReID）的问题：



如上图，紫色框表示上一帧的目标track，黑框表示当前帧检测到的四个目标，上一帧的目标（紫）跟检测出来的四个目标（黑）中哪一个是同一个目标呢？这也是多目标跟踪要做的数据关联：上一时刻的目标，和当前时刻的目标，怎么匹配关联起来。实现策略有很多，思路就是训练一个网络使它最小化类内误差，最大化类间误差。在目标跟踪中还会遇到一个问题就是身份变换（IDswitch）问题：白色框表示track，蓝色框表示detection；图中有四个ID：37,41,45,38。随着目标移动，ID41把ID38挡住（图4），detection检测不到ID38，而ID37如果刚好走到ID38原来的位置附近（从图片可以看出来，37走向和38走向是相对的，38被从右侧走到了左侧途中经过了遮挡，37从左侧走到了右侧可能也发生了遮挡），检测器可能会把37识别成38，把后继帧出现的38识别成37，从而导致身份变换。DeepSORT算法是为了解决此问题进行的多目标跟踪算法。

## 4.人流量统计原理

4.1 人流量密度

通过人流量密度能直观反映出一片区域的拥挤程度，它是估算发生事故几率的重要指标，用式（2）表示。ρ=N/S×P， （2）其中，ρ为人流量密度；N为单位面积上的行人；S 为单位面积；P 为 YOLOv3目标检测的准确率。

4.2 人流量风险评估和预警

人流量密度的风险值不仅和行人的行进方向、速度有关，且被行人所处环境（如人行道、辅路等）因素影响。因此，进行人流量风险评估时需要根据当前监控区域划分不同场景，如楼梯、广场、会场等。根据不同的场景分配不同的权重，最后获得一个人流量风险评估值，为决策层的判断提供依据。人流量风险评估值（Prisk）用式（3）表示。Prisk=ρ×weight，（3）式（3） 中：ρ为人流量密度；weight 为不同场景下的权重值。

## 5.多相机多目标跟踪（MTMCT）

MTMCT匹配方法

轨迹到轨迹（Tracklet-to-Tracklet）匹配

第一步：局部轨迹生成阶段，该阶段在单个摄像头内跟踪每个检测到的目标，并为其生成局部轨迹；

第二步：跨摄像头轨迹匹配阶段，该阶段在所有摄像头中匹配局部轨迹来为整个跨摄像头网络中的每个目标生成其完整的轨迹，即使用轨迹到轨迹（Tracklet-to-Tracklet）匹配。

局部轨迹与目标间的匹配（Tracklet-to-Target )

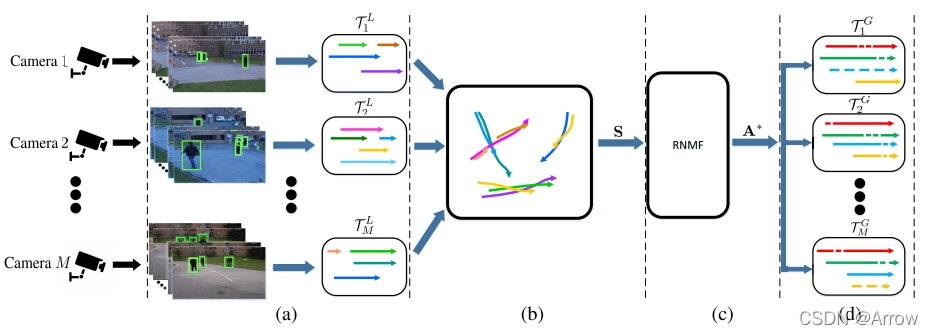
框架的输入是来自M个摄像头的M个视频剪辑

(a)：局部轨迹生成模块；

(b)：轨迹相似度测量模块；

(c)：跨摄像头轨迹匹配模块；

(d)：全局轨迹生成模块。

行人重识别（ReID）与多目标多相机跟踪（MTMCT）任务之间的区别。给定一个查询，重识别在所有相机的图库中全局搜索真实的匹配图。相比之下，多目标多相机的跟踪在单摄像机跟踪（SCT）只考虑相邻帧的匹配，在多摄像机跟踪（MCT）中只考虑相邻相机间的匹配。具体来说，多摄像头跟踪时，当目标出现在摄像头2中，就不考虑摄像头3，因为目标从未出现在这两个摄像头（摄像头可能太远了）。ＭＴＭＣＴ通常是从多个相机采集的视频中跟踪多个人（或多个目标）来确定每个人或目标的位置；Ｒｅ－ｉｄ则指从一系列图片中检索与一张被查询图片相似的图片，然后根据与被查询图片的相似度降序排列。两者的任务不同，意味着度量方式也不同。前者侧重于分类能力，后者则侧重于排序能力。一个ＭＴＭＣＴ系统通常由目标检测、相似度估计以及数据关联这３个基本部分组成。基于 ＭＴＭＣＴ的目标跟踪，其关键的因素之一仍然在目标的特征度量方面，通过一些先进的特征度量方法，可为后续的数据关联算法提供有效的数据支撑。当前对于 ＭＴＭＣＴ中的特征度量大多采用了 Ｒｅ－ｉｄ的全局特征学习方式，并将学习到的特征运用到 ＭＴＭＣＴ中。