ICME2018

MOTDT：《REAL-TIME MULTIPLE PEOPLE TRACKING WITH DEEPLY LEARNED CANDIDATE SELECTION AND PERSON RE-IDENTIFICATION》

ABSTRACT

流行的按检测跟踪框架中的**主要挑战**是如何将不可靠的检测结果与现有跟踪相关联。在本文中，我们建议通过从检测和跟踪输出中选择候选对象来处理不可靠的检测。产生冗余候选者的直觉是，在不同情况下，检测和跟踪可以相互补充。高置信度的检测结果可长期防止跟踪漂移，并且跟踪的预测可以处理由遮挡引起的噪声检测。为了从大量候选对象中实时应用最佳选择，我们提出了一种基于全卷积神经网络的新颖评分函数，该函数在整个图像上共享大多数计算。此外，我们采用了深度学习的外观表示法，该表述在大规模人员重新识别数据集上进行了训练，以提高跟踪器的识别能力。大量的实验表明，我们的跟踪器可以在广泛使用的人员跟踪基准上实现实时和最先进的性能。

1.INTRODUCTION

在许多视频分析和多媒体应用中，例如视觉监视，运动分析和自动驾驶，在复杂场景中跟踪多个对象是一个具有挑战性的问题。多对象跟踪的目的是估计特定类别中对象的轨迹。在这里，我们通过利用人员识别来解决人员跟踪的问题。

在过去的十年中，多目标跟踪从对象检测的进步中受益匪浅。流行的按检测跟踪方法将检测器应用于每个帧，并跨帧进行关联检测以生成对象轨迹。在这种跟踪框架中，类别内的遮挡和不可靠的检测都是巨大的挑战[1、2]。 类别遮挡和对象的相似外观可能导致数据关联不明确。 融合了多个线索，包括运动，形状和对象外观，以缓解此问题[3，4]。 另一方面，检测结果并不总是可靠的。拥挤场景中的姿势变化和遮挡通常会导致检测失败，例如误报，检测丢失和边界不准确。 一些研究提出以批处理方式处理不可靠的检测[2、5、6]。 这些方法通过引入来自未来帧的信息来解决检测噪声。通过解决全局优化问题，可以采用整个视频帧或一个时间窗口的检测结果并将其链接到轨迹。批处理模式下的跟踪是非因果的，不适合对时间要求严格的应用程序。与这些作品相比，我们仅使用当前帧和过去帧来关注在线多人跟踪问题。

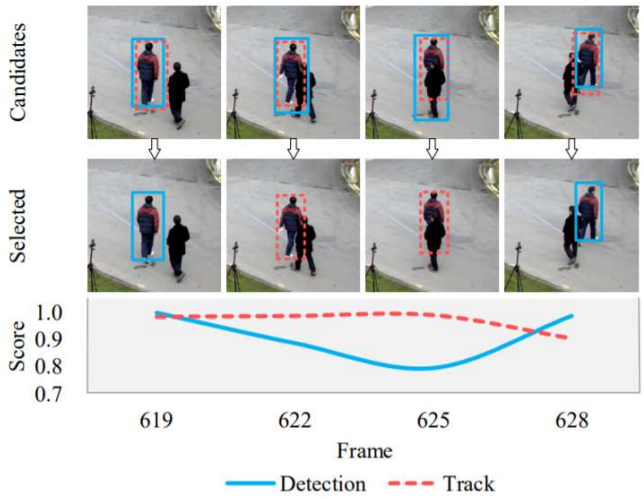

Fig. 1: Candidate selection based on unified scores. Candidates from detection and tracks are visualized as blue solid rectangles and red dotted rectangles, respectively. Detection and tracks can complement each other for data association.

为了处理在线模式下不可靠的检测，我们的跟踪框架会从每个帧的检测和跟踪输出中最佳选择候选对象（如图 1 所示）。 在大多数现有的检测跟踪方法中，当谈到数据关联时，与现有轨道相关联的候选仅由检测结果组成。Yan 等人[4]提出将跟踪器和目标检测器视为两个独立的身份，并将它们的结果作为候选。他们根据手工制作的特征选择候选对象，例如颜色直方图、光流和运动特征。生成冗余候选项背后的直觉是，在不同的场景中，检测和跟踪可以相互补充。一方面，在缺少检测或边界不准确的情况下，来自跟踪器的可靠性预测可用于短期关联。另一方面，自信的检测结果对于防止轨迹长期漂移到背景是至关重要的。如何以统一的方式对检测和跟踪的输出进行评分仍然是一个悬而未决的问题。

近年来，深度神经网络，特别是卷积神经网络（CNN）在计算机视觉和多媒体领域取得了长足的进步。在本文中，我们充分利用了深度神经网络来解决不可靠的检测和类别内遮挡的问题。我们的贡献有三个方面。首先，我们通过组合检测和跟踪结果作为候选对象，并基于深度神经网络选择最佳候选者，来处理在线跟踪中的不可靠检测。其次，我们提出一种分层数据关联策略，该策略利用空间信息和深度学习的人员重新识别（ReID）特征。第三，在广泛使用的人员跟踪基准上演示我们的跟踪器的实时性和最新性能。

2.RELATED WORK

通过检测进行跟踪已成为多对象跟踪的最流行策略。Bae 等[1]根据其置信度值，以不同的方式将轨迹与检测相关联。Sanchez-Matilla 等[7]利用多检测器提高跟踪性能。他们在所谓的过度检测过程中收集了多个检测器的输出。合并多个检测器扫描的结果可改善跟踪性能，但对于实时应用而言效率不高。相比之下，我们的跟踪框架只需要一个检测器，并从现有轨道中生成候选对象。Chu 等[8]使用二进制分类器和单对象跟踪器进行在线多对象跟踪。他们共享特征图进行分类，但计算复杂度仍然很高。

批处理方法将跟踪公式化为全局优化问题[4、5、6、9]。这些方法利用来自未来帧的信息来处理噪声检测并减少数据关联中的歧义。刘等[10]提出了风向追踪策略，以生成包含未来信息的后向小径，以获得更稳定的相似度度量以进行关联。在[6、9、11]中还对人员重新识别进行了探索，以进行全局优化。我们的框架在在线模式下利用了深度学习的 ReID 特征，从而在解决类别内遮挡问题时提高了识别能力。

3.PROPOSED METHOD

3.1.架构综述

在这项工作中，我们通过从检测和跟踪的输出中收集候选项来扩展传统的按检测跟踪。我们的框架包括两个连续的任务，即候选人选择和数据关联。

我们首先使用统一的评分功能来衡量所有候选人。如第 3.2 节和第 3.3 节所述，将经过区分训练的对象分类器和设计良好的轨迹置信度融合在一起，以制定评分功能。随后使用估计分数执行非最大抑制（NMS）。在获得没有冗余的候选者之后，我们同时使用外观表示和空间信息将现有轨道与所选候选者分层关联。我们根据 3.4 节中所述的人员重新识别来深刻了解我们的外观表示。层次数据关联在第 3.5 节中详细介绍。

3.2.实时对象分类

将检测和跟踪的输出结合在一起将导致过多的候选对象。我们的分类器通过使用基于区域的卷积神经网络（R-FCN）[12]在整个图像上共享大多数计算。因此，与从高度重叠的候选区域裁剪出来的图像补丁分类相比，它的效率要高得多。 这两种方法的时间消耗的比较可以在图中找到。

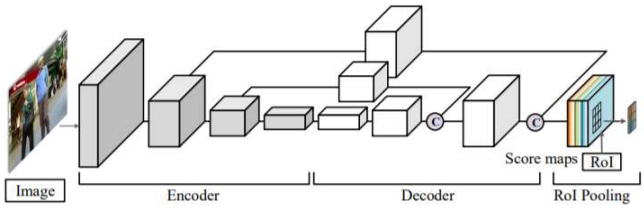

Fig. 2: R-FCN architecture for efficient classification. Features from the encoder network are concatenated with upsampled features in the decoder part, to capture both the semantic and low-level information. Each color in the last block represents a specific score map.

图 2:R-FCN 体系结构可实现有效分类。 来自编码器网络的功能与解码器部分中的上采样功能相结合，以捕获语义信息和底层信息。 最后一块中的每种颜色代表一个特定的分数图。

我们的有效分类器如图 2 所示。在给定图像帧的情况下，使用具有编码器-解码器体系结构的全卷积神经网络预测整个图像的得分图。编码器部分是实现实时性能的轻量级卷积 backbone，我们为解码器部分引入了上采样，以增加输出得分图的空间分辨率，以供以后分类。每个要分类的候选对象定义为感兴趣区域（RoI），其中 x=(x0,y0,w,h)x=(x0,y0,w,h)，其中 x(x0,y0)x(x0,y0)表示左上角点，w,h 表示该区域的宽度和高度。为了提高效率，我们希望每个 RoI 的分类概率由共享分数图直接投票。一种直接的投票方法是构造图像上所有点的前景概率，然后计算 RoI 内点的平均概率。但是，这种简单的策略会丢失对象的空间信息。例如，即使 RoI 仅覆盖对象的一部分，仍然可以获得较高的置信度得分。

为了将空间信息明确编码为得分图，我们采用了 position-sensitive RoI pooling 层，并从 k^2 个 position-sensitive score maps z估计分类概率。特别地，我们通过规则网格将 RoI 分割成 k×k 的小格子。 每个格子具有相同的大小$[\frac{w}{k} \times \frac{h}{k}]$，并表示对象的特定空间位置。我们从 k^2 个 score maps 中提取 k×k 格子的响应。每个分数图仅对应一个格子。RoI x 的最终分类概率公式为：

$$p(y|\mathbf{z},\mathbf{x}) = \sigma(\frac{1}{wh}\sum_{i=1}^{k^2}\sum_{(x,y)\in bin_i}\mathbf{z}_i(x,y)), \qquad (1)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, and $\mathbf{z}_i$ denotes the i-th score map.

在训练过程中，我们随机抽取 GT 边界框附近的 RoI 作为正例，并从背景中获得相同数量的 RoI 作为负例。 通过端到端训练网络，解码器部分顶部的输出（即 k^2 score map）将学习对目标特定空间位置的响应。例如，如果 k = 3，则有 9 个得分图响应到顶部 -分别位于对象的左，上中心，右上，…，右下。这样，RoI 池层对空间位置敏感，并且在不使用可学习参数的情况下，具有很强的区分对象的能力。请注意，提出的神经网络仅针对候选分类进行训练，而不针对边界框回归进行训练。

3.3.轨迹置信度和评分函数

给定一个新的帧，我们使用卡尔曼滤波器估计每个现有轨道的新位置。这些预测用于处理由于对象的视觉属性变化和拥挤场景中的遮挡而导致的检测失败。但是它们不适合长期跟踪。如果长时间未通过检测更新卡尔曼滤波器，则其精度可能会降低。Tracklet 置信度旨在

使用时间信息来测量滤波器的准确性。

通过从连续帧中对候选者的时间关联来生成 Tracklet。在其生命周期内，由于轨道可以中断和找回，因此我们可以将轨道拆分为一组 Tracklet。**每次从丢失状态中找回轨道时，都会重新初始化卡尔曼滤波器**。因此，仅使用最后一个轨迹的信息来形成轨迹的置信度。在这里，我们定义L_{det}为与小轨迹相关联的检测结果的数量，L_{trk}为在最后一次检测相关联后的轨道预测的数量。Tracklet 置信度定义为：

$$s_{trk} = \max(1 - log(1 + \alpha \cdot L_{trk}), 0) \cdot \mathbb{1}(L_{det} \geq 2), \quad (2)$$

其中$\mathbb{1}(.)$是指示符函数，如果输入为 true，则等于 1，否则等于 0。我们要求$L_{det} \geq 2$，以便在使用轨迹作为候选之前，使用观察到的检测来构建合理的运动模型。通过融合分类概率和 Tracklet 置信度：

$$s = p(y|\mathbf{z}, \mathbf{x}) \cdot (\mathbb{1}(\mathbf{x} \in C_{det})) + s_{trk}\mathbb{1}(\mathbf{x} \in C_{trk})). \quad (3)$$

在这里，我们使用C_{det}表示检测到的候选者，使用C_{trk}表示跟踪的候选者，而$s_{trk} \in [0,1]$用来惩罚不确定轨迹的候选者。最终，使用 NMS，基于统一分数，选择数据关联的候选人。我们通过阈值τ_{nms}定义了最大的 IoU，同时也有最小得分阈值τ_s。

3.4.用 ReID 特征表示外观

候选者之间的相似性特征是数据关联的关键要素。我们认为，通过数据驱动方法深度学习的对象外观在相似性估计任务上优于传统的手工特征。为了学习对象的外观和相似度函数，我们采用了深度神经网络从 RGB 图像中提取特征向量，并使用获得的特征之间的距离来模拟相似度。

我们利用[13]中提出的网络体系结构，并在几个大规模的每人重新识别数据集的组合上训练网络。网络H_{reid}由 GoogLeNet 的卷积骨干[14]组成，然后是部分对齐的全连接（FC）层的分支。有关网络体系结构的更多信息，请参阅[13]。给定一张人的 RGB 图像I，外观表示写为$f = H_{reid}(I)$。我们直接使用特征向量之间的欧式距离来测量两个图像I_i和I_j的距离d_{ij}。 在训练过程中，训练数据集中的身份图像形成为一组三元组$T =< I_i, I_j, I_k >$，其中$< I_i, I_j >$是同一个人的正对，而$< I_i, I_k >$是来自两个不同人的负对。给定 N 个三元组，将要最小化的损失函数公式为：

$$l_{triplet} = \frac{1}{N} \sum_{\langle \mathbf{I}_i, \mathbf{I}_j, \mathbf{I}_k \rangle \in \mathbf{T}} \max(d_{ij} - d_{ik} + m, 0), \quad (4)$$

其中，m> 0 是预定义的边距。我们忽略了易于处理的三元组，即$d_{ik} - d_{ij} > m$，以增强学习的特征表示的判别能力。

3.5.分层数据关联

利用轨迹的预测来处理在拥挤的场景中发生的丢失检测。受类内遮挡的影响，这些预测可能与其他对象有关。为了避免将其他不需要的对象和背景用于外观表示，我们使用不同的特征将轨道与不同的候选对象进行了层次化关联。

Algorithm 1: The proposed tracking algorithm.

Input: A video sequence v with N_v frames and object detection $\{\mathcal{D}_k\}_{k=1}^{N_v}$
Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$; appearance of tracks $\mathcal{F}_{trk} \leftarrow \emptyset$
foreach *frame* f_k *in* v **do**
  Estimate score maps $\mathbf{z}$ from f using R-FCN
  /* collect candidates */
  $C_{det} \leftarrow \mathcal{D}_k$; $C_{trk} \leftarrow \emptyset$
  foreach t *in* $\mathcal{T}$ **do**
    Predict new location $\mathbf{x}^*$ of t using Kalman filter
    $C_{trk} \leftarrow C_{trk} \cup \{\mathbf{x}^*\}$
  end
  /* select candidates */
  $C \leftarrow C_{det} \cup C_{trk}$
  $S \leftarrow$ unified scores computed from Equation 3
  $C, S \leftarrow \texttt{NMS}(C, S, \tau_{nms})$
  $C, S \leftarrow \texttt{Filter}(C, S, \tau_s)$ // filter out if $s < \tau_s$
  /* extract appearance features */
  $\mathcal{F}_{det} \leftarrow \emptyset$
  foreach $\mathbf{x}$ *in* C_{det} **do**
    $\mathbf{I}_{\mathbf{x}} \leftarrow \texttt{Crop}(f_k, \mathbf{x})$
    $\mathcal{F}_{det} \leftarrow \mathcal{F}_{det} \cup H_{reid}(\mathbf{I}_{\mathbf{x}})$
  end
  /* hierarchical data association */
  Associate $\mathcal{T}$ and C_{det} using distances of $\mathcal{F}_{trk}$ and $\mathcal{F}_{det}$
  Associate remaining tracks and candidates using IoU
  $\mathcal{F}_{trk} \leftarrow \mathcal{F}_{trk} \cup \mathcal{F}_{det}$
  /* initialize new tracks */
  $C_{remain} \leftarrow$ remaining candidates from C_{det}
  $\mathcal{F}_{remain} \leftarrow$ features of C_{remain}
  $\mathcal{T}, \mathcal{F}_{trk} \leftarrow \mathcal{T} \cup C_{remain}, \mathcal{F}_{trk} \cup \mathcal{F}_{remain}$
end

尤其是，我们第一次对检测的候选者使用不超过阈值τ_d的外观表示来进行数据关联。然后，我们基于候选者和轨迹之间的 IOU 来关联剩余的候选者和未匹配的轨迹，IOU 阈值为τ_{iou}。只当轨迹关联到检测时我们才对轨迹的外观表示进行更新。通过从关联的检测中保存 ReID 特征来进行更新。最后，根据剩余的检测结果对新轨道进行初始化。所提出的在线跟踪算法的详细信息在算法 1 中进行了说明。通过分层数据关联，我们只需要为每帧从检测中提取候选的 ReID 特征即可。将其与以前有效的评分函数和小轨迹置信度相结合，我们的框架可以实时运行。