课程论文题目

陈岱杰

摘要

尽管基于深度学习的单目行人检测方法已经取得了很大进展，但它们仍然容易受到严重遮挡。在拥挤场景下的物体遮挡一直是影响行人检测和识别效果的重要因素，结合多个摄像机视图进行检测可以减轻拥挤场景中遮挡的影响。在多视图检测系统中，如何聚合来自多个视图的特征信息、如何从空间上相邻的位置聚集信息是提高行人检测和识别效果的重要问题。2020 年 ECCV 中提出的新型多视图检测器 MVDet 能很好的解决上述问题。在多视图聚合过程中，对于地面上的每个位置，使用多视图锚框特征作为表示的方法由于预定义锚框的不确定性会限制检测的性能。相比之下，通过特征图透视变换，MVDet 采用无锚框的表示方法，其特征向量能直接在多个视图中的对应像素之间进行采样。对于空间聚合，与需要在神经网络之外进行设计和操作的先前方法不同，MVDet 采用全卷积的方法，在多视图聚合特征图上使用大卷积核进行信息聚合。多视图检测器 MVDet 在 Wildtrack 数据集上有了很好的识别检测效果，为进一步了解该检测器的识别检测能力，我们将该模型放置于两个更大型的数据集 CityStreet 和 CVCS 中进行测试。由于数据集中人群数量的增多、场景范围的扩大、背景环境更加复杂，MVDet 检测器对行人的识别和检测效果有了大幅的下降。

关键词：多视图检测；无锚；透视变换；全卷积；合成数据

1 引言

跨场景跨视觉人群检测是近年来的一个研究热点。一直以来物体遮挡对行人的识别、检测等方面造成了严重的影响，通过融合不同视角下的行人特征信息能有效地提高在遮挡情况下对于行人的识别、检测效果。但在近年来的研究当中，大多数论文提出的模型算法都只能在场景范围较小、人数有限的数据集中表现出较好的效果，这些模型在不同的、规模更大的数据集上的测试效果值得进行进一步的研究。通过对比相同模型在不同数据集上的测试效果，能为今后模型的优化方向提供很好的参考。于是，我选取了 2020 年 ECCV 中的一篇具有代表性的文章 Multiview Detection with Feature Perspective Transformation 进行验证。

2 相关工作

2.1 多视图检测

多视图检测最具挑战性的部分是从多个视图融合关于对象或行人的特征信息。在一个多相机系统中，包含多个同步、有公共视野、标定好的相机。在多相机检测系统中，由于相机参数已知，可以通过假设行人 3D 包围框（3D bounding box）的直径和高度，计算得到每个相机中的 2D 包围框（2D bounding box）。因此，多相机检测一般在同一平面（俯瞰）上评估行人的检测效果，如：地面。之前工作一般利用相机参数和行人 3D 形状，计算每个相机内对应每个位置的锚框（anchor box）。之后，再利用锚框的特征向量（以及 ROI pooling）表示该位置的行人信息。但是，这一类方法中使用的锚框形状和位置不一定准确，会导致最终聚合的信息不准确。本文中，作者使用了一套无锚框（anchor-free）

的信息表示方法来避免锚框对实验结果的影响。对于地面上的每个位置，直接选取该位置的特征向量（feature vector）作为代表，同时该特征向量也能表示站在该处的行人的特征信息。利用从相应点的特征图中采样的特征向量来表示地平面位置，避免了不精确锚盒的影响。给定摄像机校准，可以准确地检索相应的点。利用可学习卷积核，这些特征向量可以表示来自其接受域中的自适应区域的信息。结果，通过无锚特征表示构建的地平面特征图避免了来自不准确锚盒的汇集，并且仍然包含来自 2D 图像的足够信息用于检测。

2.2 视角转换

仿射变换和透视变换等几何变换可以模拟计算机视觉中的许多现象，并且可以用一组固定的参数显式计算。为了检索无锚框的特征信息，我们使用透视变换把特征图投影于同一平面，在本文中，选取了地平面作为投影平面，即 z = 0。3D 位置（x，y，z）和 2D 图像像素坐标（u，v）之间的转换公式如下：

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P_\theta \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = A\,[R|t] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1}$$

其中 s 是实值缩放因子，P_θ 是 3×4 透视变换矩阵，A 是 3×3 相机内参矩阵。[R|t] 是 3×4 旋转平移矩阵和相机外参矩阵拼接而成的矩阵，其中 R 指定旋转，t 指定平移。为了在神经网络中实现这一点，我们将地平面位置量化为形状网格 $[H_g, W_g]$。对于摄像机 $n \in \{1, \ldots, N\}$ 通过校准后，我们可以基于公式（1）将图像投影到 z=0 的地平面上。这样采样网格就能够在地平面上生成对应投影特征图，其中剩余（不可见）位置用零填充（投影效果见图 1）。

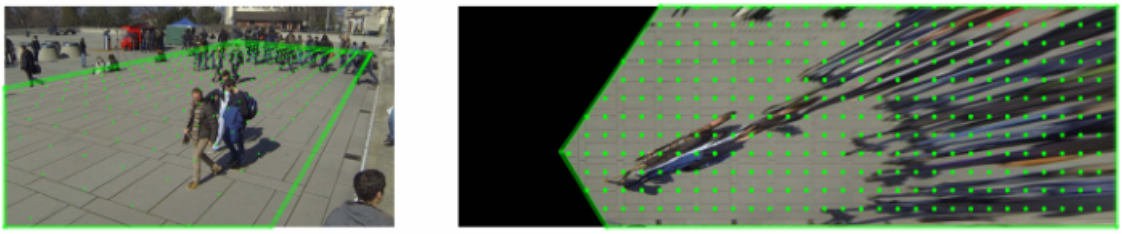

图 1: 投影效果示意图

3 数据集对比

为更好的体现 CVCS、CityStreet 数据集与 Wildtrack、MultiviewX 数据集之间的区别，接下来将对不同数据集中的量化指标进行详细的讨论。

Wildtrack 数据集的数据采集是在瑞士天气条件良好的条件下进行拍摄。拍摄场景大小为 12×16 米，其中包含 7 个不同的摄像机视角，并且这 7 个视角将以每秒 60 帧的速率对场景内的人群进行连续拍摄。拍摄生成的图像分辨率为 1920×1680。最终的数据集将从每秒 60 帧图像中以等时间间隔的方式选取其中的 2 张图片，总共选取 400 帧图片，时间间隔为 200s。数据集中每张图片中的人群数量平均为 20 人。此外，整个场景的地平面被量化为 480×1440 网格，其中每个网格单元是 2.5 厘米的正方形。场景中的每个位置都有 3.74 个摄像头覆盖。

MultiviewX 数据集是利用 Unity 引擎以及 PersonX 数据集中提供的 3D 行人模型生成。使用 Unity

引擎生成背景环境，使用 PersonX 数据库生成行人。PersonX 数据集包含 6 个背景，包括 3 个纯色背景和 3 个场景背景。有 1266 个手工制作的身份 (547 个女性和 719 个男性)。拍摄场景大小为 16×25 米，其中包含 6 个不同的摄像机视角。摄像机同样以每秒 60 帧的速率生成虚拟人群图像，最终从每秒生成的图像中以等时间间隔的方式选取其中的 2 张图片，总共选取 400 帧图片，时间间隔为 200s。生成的图像分辨率为 1920×1680。然后，使用相同的 2.5 厘米正方形网格单元将地平面量化为 640×1000 网格。数据集中每张图片中的人群数量平均为 40 人。场景中的每个位置都有 4.41 个摄像头覆盖。

CityStreet 数据集使用 5 台同步摄像机收集了一条繁忙城市街道的多视图视频。拍摄场景大小为 58×72 米，视频长约 1 小时，图片分辨率为 2704×1520，每秒 30 帧。最终从 5 个摄像机视角中选取了 3 个视角，并从视频帧中均匀地采样 500 个多视图图像用于数据集的制作。数据集中还包括了由于车辆和固定建筑结构而导致的更多人群规模变化和行人遮挡。数据集中每张图片中的人群数量在 70-150 人的范围之内。此外，整个场景的地平面被量化为 768×640 网格，其中每个网格单元是 11.25 厘米的正方形。场景中的每个位置都有 3 个摄像头覆盖。

CVCS 数据集使用游戏“侠盗猎车手 V”的插件，以游戏中的场景为背景随机生成人群图片。行人的生成采用了游戏中的 265 个人物模型：不同的人物模型具有不同的肤色、性别、形状等。此外，对于每个人物模型，它在外观上有六种变化，例如服装、发型等。为了提高人物模型的多样性，每个模型都被命令在稀疏的人群场景中进行随机动作。为了更好的拟合现实生活中可能出现的环境，数据集还随机在 7 种不同天气类型下进行拍摄，如：晴朗、多云、下雨、大雾、雷电、阴天和格外晴朗。最终，数据集包含 31 个不同场景，每个场景下约有 100 个不同视角，在每个场景下会拍摄 100 张人群多视角图像。数据集中每张图片中的人群数量在 90-180 人的范围之内。生成的图像分辨率为 1920×1680。由于拍摄的场景大小随机选择所以无法给出准确的数值，但是会从中截取大小为 40×40 米的场景用于训练和测试。并且将截取场景的地平面量化为 200×200 网格，其中每个网格单元是 0.2 米的正方形。由于场景大小不唯一所以对于每个位置摄像头覆盖的数值也无法准确得出。综上所诉，CVCS 和 CityStreet 数据集在人数、场景大小、环境复杂度等方面都远大于 Wildtrack、MultiviewX 数据集（具体见表 1）。通过使用相同的模型在 CVCS 和 CityStreet 数据集上进行测试，能很好地反映模型的泛化性和鲁棒性。

数据集	视角数目	图片数目	场景范围	分辨率	人数	视角覆盖程度	背景环境
Wildtrack	7	400	12×16	1920×1680	20 人/帧	3.75 个	1
MultiviewX	6	400	16×25	1920×1680	40 人/帧	4.41 个	6
CityStreet	3	500	58×72	2704×1520	110 人/帧	3 个	1
CVCS	100	280000	-	1920×1680	135 人/帧	-	7

表 1: 4 个不同数据集的统计数据

4 本文方法

4.1 本文方法概述

为了验证论文[1]中提出的 MVDet 检测器能否在不同数据集上具有较好的泛化性和鲁棒性，我选择了两个不同类型的数据集：CVCS 和 CityStreet 数据集。在上述两个更大规模的数据集中，我将沿用论文中提出的模型框架、使用相同的训练方法，通过最终的可视化结果和相同的评测指标对模型进行验证。MVDet 的架构如图 2所示。首先，给定来自 N 个摄像机生成的输入图像（大小为 $[3, H_i, W_i]$）。

接下来，使用 CNN 来提取每个输入图像的通道特征图。这里的 CNN 特征提取器会在 N 个输入之间共享权重。然后，我们将通道特征图重塑为 $[N, C, H_f, W_f]$ 的大小，并通过检测头-脚对来运行单视图检测。对于多视图聚合（圆圈 1），我们采用无锚方法，并基于其相机校准后对 N 个特征图进行透视变换，输出的投影特征图大小为 $[C, H_g, W_g]$。对于每个地平面位置，我们将其 X-Y 坐标存储于 2 通道坐标图中。通过将 N 个投影特征图与坐标图拼接，我们聚合了整个场景的地平面特征图（大小为 $[N \times C + 2, H_g, W_g]$）。最后，我们在地平面特征图上应用大卷积核聚合空间领域信息（圆圈 2），最终得行人密度图（大小为 $[H_g, W_g]$）。

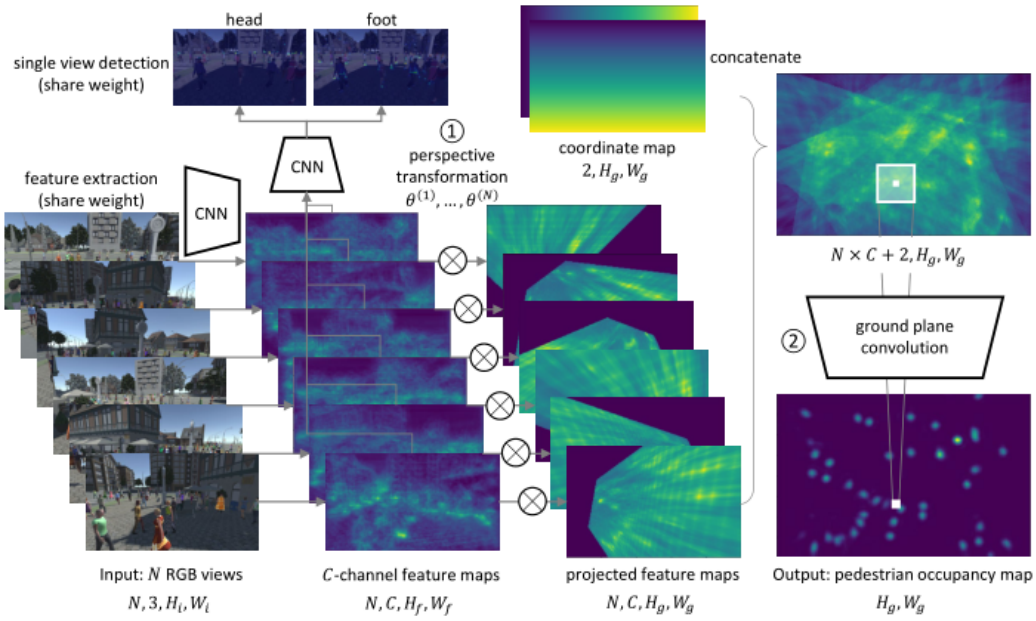

图 2: 方法示意图

4.2 损失函数定义

我们将 MVDet 训练当作回归问题。给定真实行人坐标图 g，类似于地标检测，我们使用高斯模糊来生成一个平滑的地面真实目标 $f(g)$。为了训练整个网络，我们使用网络输出的行人坐标图 $\tilde{g}$ 和高斯模糊过后的真实行人坐标图 $f(g)$ 之间的欧几里德距离 $\|.\|_2$ 作为损失函数

$$L_{ground} = \|\tilde{g} - f(g)\|_2 \tag{2}$$

我们还使用来自 N 个摄像机输入的单视图行人头-脚检测作为另一种监督。单视图头-脚检测的训练同样当作回归问题。对于单视图头-脚检测，我们同样使用网络的预测结果 $\tilde{s}^{(n)}_{head}$、$\tilde{s}^{(n)}_{foot}$ 和相应的真实值 $s^{(n)}_{head}$、$s^{(n)}_{foot}$ 之间的欧几里德距离作为损失函数进行训练，公式如下：

$$L^{(n)}_{single} = \left\|\tilde{s}^{(n)}_{head} - f(s^{(n)}_{head})\right\|_2 + \left\|\tilde{s}^{(n)}_{foot} - f(s^{(n)}_{foot})\right\|_2 \tag{3}$$

最终，结合地平面损失 L_{ground} 和 N 个单视图损失 $L^{(n)}_{single}$，我们得到了训练 MVDet 的总损失函数

$$L_{combine} = L_{ground} + \alpha \times \frac{1}{N}\sum_{n=1}^{N} L^{(n)}_{single} \tag{4}$$

其中 α 是单视图损失权重的超参数。

5 复现细节

5.1 与已有开源代码对比

原文中使用的高斯模糊方式是通过定义一个合适大小的高斯核对模型输出的特征图进行卷积。通过卷积可以将像素周围邻域的特征信息汇聚于中心位置, 并将该中心位置及卷积核覆盖的范围作为一个行人存在的位置，最后通过损失函数的反向传播让中心位置的特征信息更加明显，以此达到模型训练的效果。但是由于 CVCS、CityStreet 数据集的场景范围过大、人群的聚集密度更大，如仍沿用原论文中设计的高斯模糊方式将会导致把多个行人的特征信息融合成一团，无法分辨其中的人群个数和具体的位置，这将会对模型的训练结果造成严重的影响，所以我使用了一个对于单点高斯模糊的方式来替换原论文中使用高斯核进行卷积模糊的方式。单点高斯模糊方式顾名思义，就是对图片中的每一个行人位置都单独进行高斯模糊。通过设计合适的高斯模糊范围，对该范围内的所有像素点进行模糊计算，使得该范围内的所有像素点值呈现中心值最高，四周的值逐渐下降的分布。效果如图 3所示，具体高斯模糊的伪代码如下：

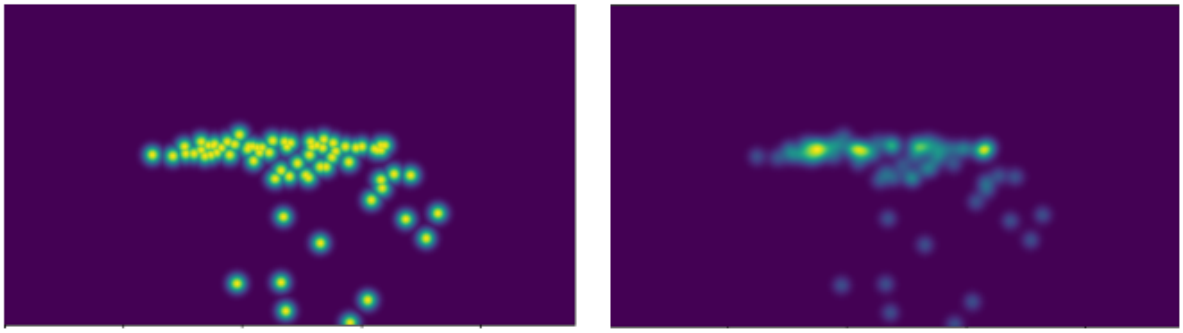

图 3: 效果对比图

Procedure 1 Gaussian Blur

Input: map *Map,* scope *S*

Output: density map *M*

H, W =Map.shape

for *X, Y* ***in*** *reference frame point* **do**

$left, right = \min(X, S),\ \min(W - X, S + 1)$

$top, bottom = \min(Y, S),\ \min(H - X, S + 1)$

$M[y - top : y + bottom][x - left : x + right] = \text{gaussien}(Map[y - top : y + bottom][x - left : x + right])$

end

另外，由于不用的数据集提供的数据参数不一致，所以我们需要对图像投影过程的数据处理部分进行改进，图片投影的方式仍然延用原论文中提出的透视变换方法。CVCS 和 CityStreet 数据集中图像的投影结果如图 4和 5所示：

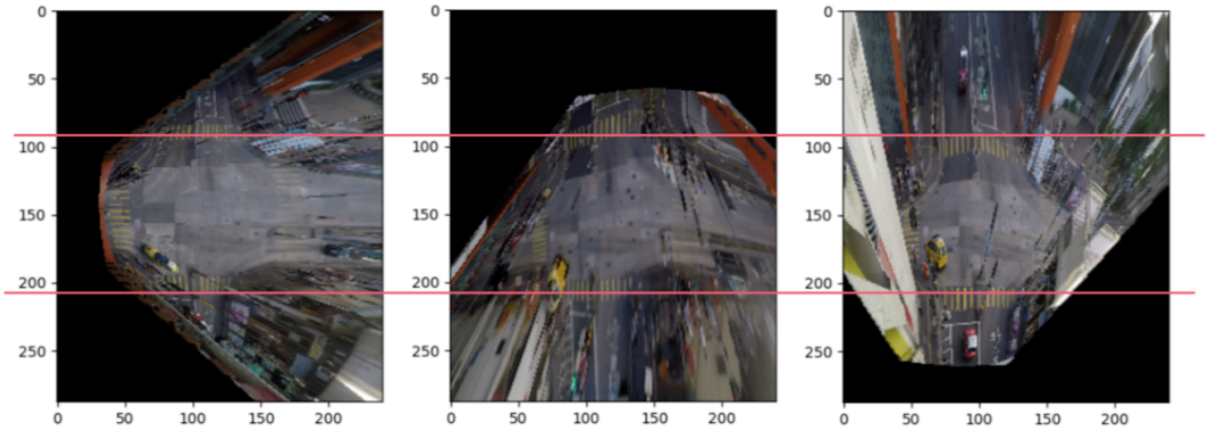

图 4: CityStreet 数据集的图像投影结果图

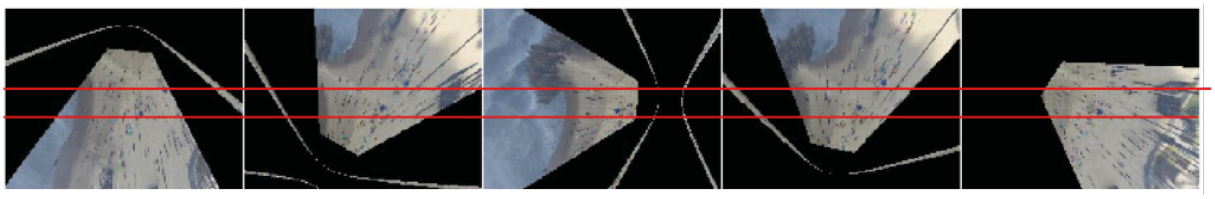

图 5: CVCS 数据集的图像投影结果图

通过图 4中的斑马线和图 5中的两个深蓝色的点可以证明，我们修改后的投影模块可以实现不同视角的图像在同一平面上的投影。

5.2 实验环境搭建

我们在两个 RTX-3090 GPU 上完成了所有实验。实验代码使用了以下的库：

- python 3.7+
- pytorch 1.4+
- torchvision
- numpy
- matplotlib
- opencv
- pillow
- kornia
- matlab & matlabengine

5.3 界面分析与使用说明

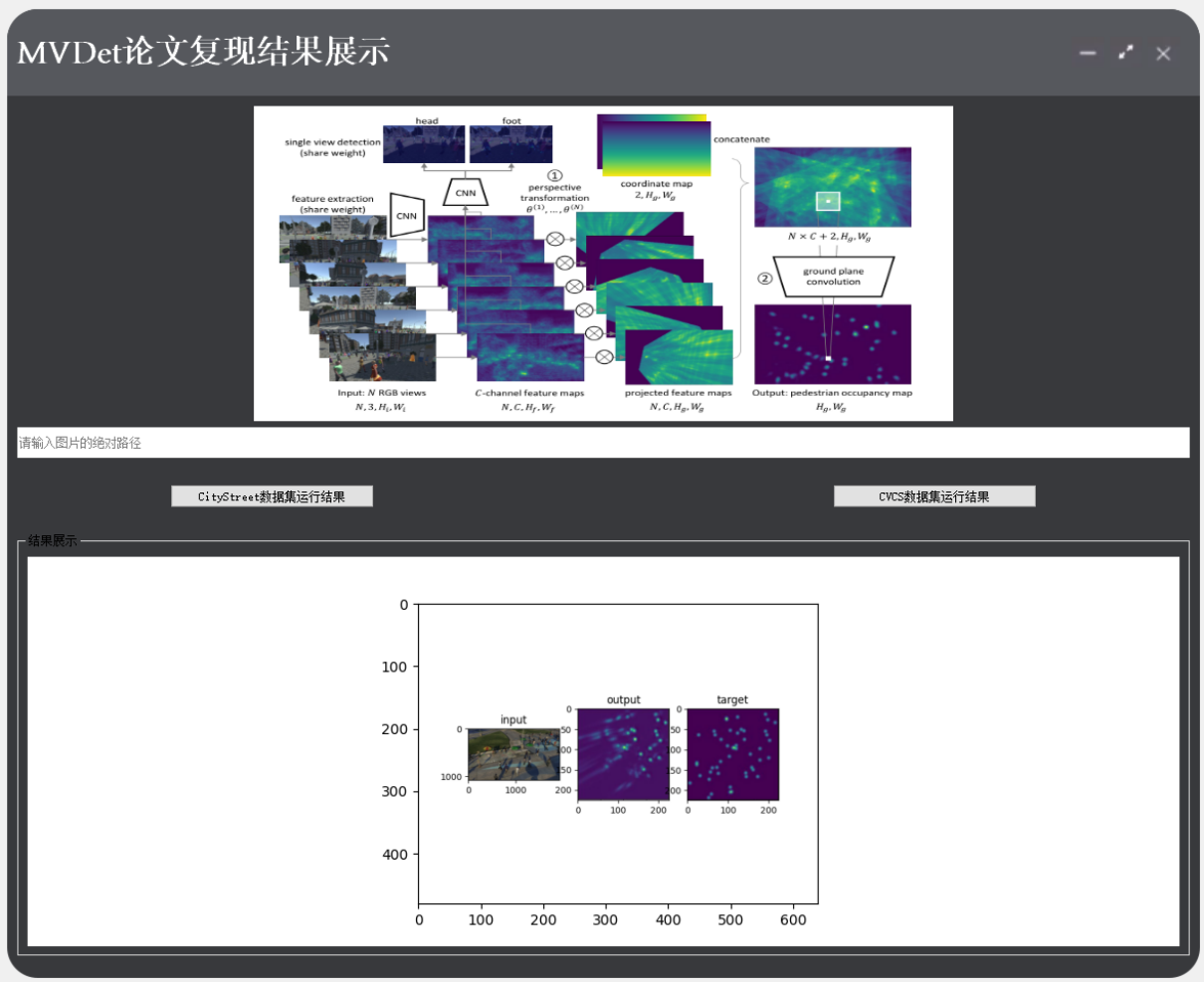

图 6: 操作界面示意

为了更好的便于用户了解实验结果，我使用 python 设计了一个简单的 GUI 界面。使用时只需要将一个对应数据集中的图片的绝对路径放入文本输入框中点击对应的按钮即可在下方显示出经过模型训练后的鸟瞰密度图。

6 实验结果分析

MVDet 检测器在 CVCS 数据集的可视化结果如下图 7所示

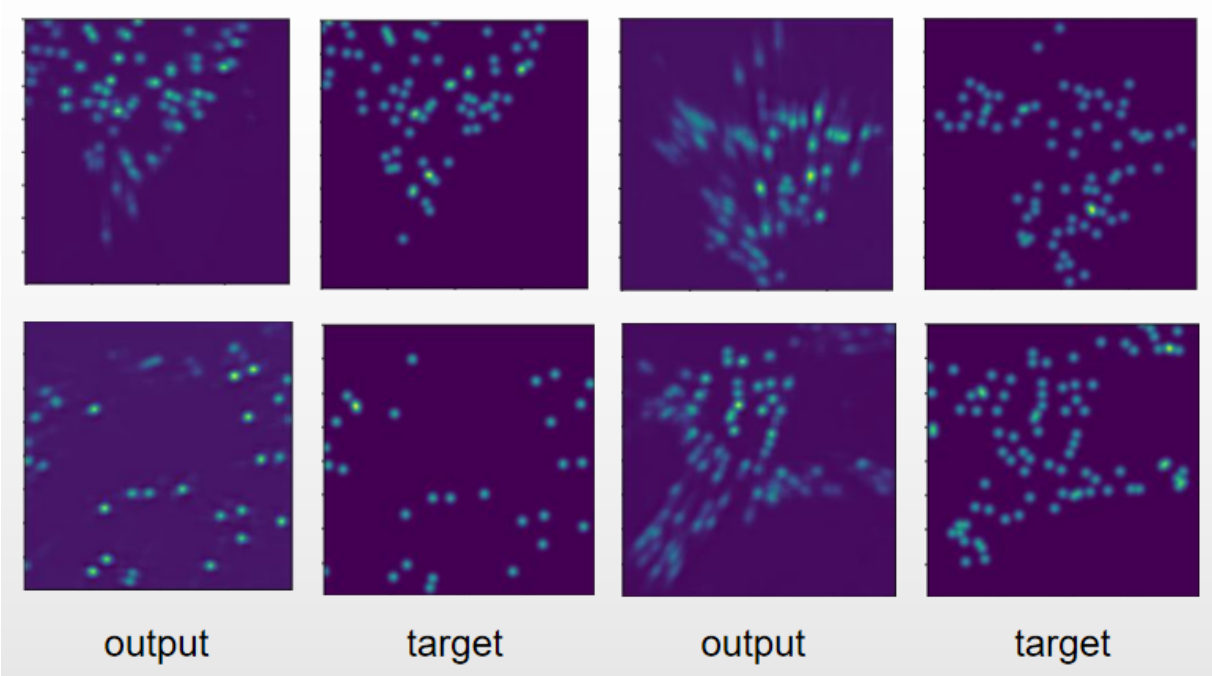

图 7: CVCS 数据集上的实验结果示意图

MVDet 检测器在 CityStreet 数据集的可视化结果如下图 8所示

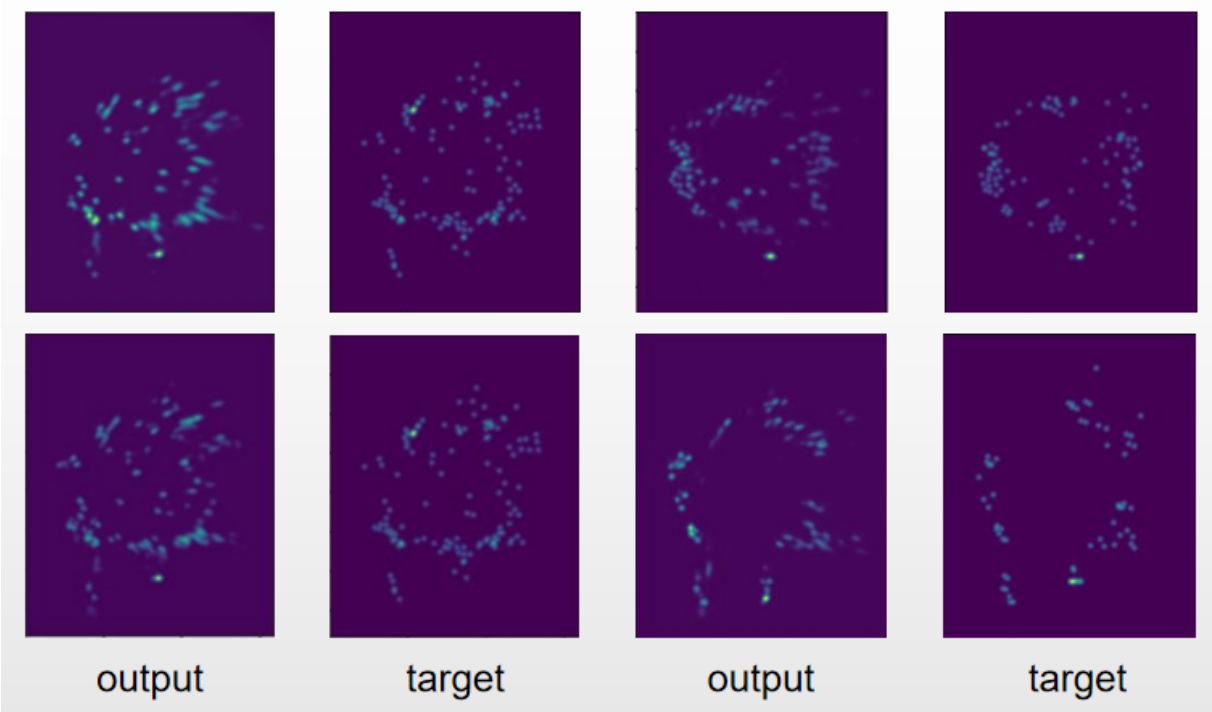

图 8: CityStreet 数据集上的实验结果示意图

为了能够定量分析论文中提出的 MVDet 检测器在不同数据集中对行人的识别检测效果，我将沿用论文中使用的 4 个评测指标对模型在新数据集上的效果进行测评。接下来，我将对论文中使用到的多目标检测精确度（MODA）、多目标检测准确度（MODP）、精确度（Precision）和召回度（Recall）这 4 个测评指标进行详细的介绍。

多目标检测准确度（MODP）: 我们利用真实值和模型输出值之间的空间重叠度的平均值作为我们的多目标检测准确度，并使用多目标检测准确度指标评估模型对行人定位的精度。通过空间的重叠

度的交并比公式我们可以计算出对于每一帧图片的多目标检测准确度，具体计算公式如下：

$$MODP(t) = \frac{\sum_{i=1}^{N_{mapping}^{t}} \frac{|G_i^{(t)} \cap D_i^{(t)}|}{|G_i^{(t)} \cup D_i^{(t)}|}}{N_{mapping}^{t}} \tag{5}$$

其中 $G_i^{(t)}$ 表示在第 t 帧图片中第 i 个物体的真实值，$D_i^{(t)}$ 表示在第 t 帧图片中第 i 个物体的预测值，$N_{mapping}^{t}$ 表示在第 t 帧图片中的行人总数。通过公式 (1) 我们可以计算出任意帧图片中所有物体的准确性，因此我们可以用所有相关评估图片的多目标检测准确度之和 N-MODP 来规范化度量。N-MODP 能够很好反映出模型的空间检测准确度。

$$N - MODP = \frac{\sum_{t=1}^{N_{frames}} MODP(t)}{N_{frames}} \tag{6}$$

多目标检测精确度（MODA）: 为了评估模型性能的精确性，我们将对漏报情况和误报情况进行统计。在每一帧图片 t 中，错报数用 m_t 表示，误报数用 f_{p_t} 表示，我们可以计算出每一帧图像的多目标检测精度。

$$MODA(t) = 1 - \frac{c_m(m_t) + c_f(f_{p_t})}{N_G^t} \tag{7}$$

公式 (3) 中 c_m 和 c_f 表示漏检和误报情况的代价函数,N_G^t 表示第 t 帧图片中的行人总数。同理，我们可以通过累加求和取平均数的方式来对所有图片的检测精确度进行规范化度量。

$$N - MODA = 1 - \frac{\sum_{i=1}^{N_{frames}} (c_m(m_i) + c_f(f_{p_i}))}{\sum_{i=1}^{N_{frames}} N_G^i} \tag{8}$$

准确度（Precision）: 准确度用于评估检测器在检测成功基础上的正确率。针对模型判断出的所有正样本，计算出模型判断出的正样本数目占数据集中所有正样本数目的比例。

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

其中 TP 表示本来是正样本，检测结果也为正样本的数目，FP 表示本来是负样本，检测结果为正样本的数目。

召回度（Recall）: 召回度用于评估检测器对所有待检测目标的检测覆盖率。针对数据集中的所有正例，计算出模型正确判断出的正样本数目占数据集中所有正样本数目的比例。

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

其中 TP 表示本来是正样本，检测结果也为正样本的数目，FN 表示本来是正样本，检测结果为负样本的数目。

经过实验，MVDet 检测器在不同数据集中的测评数据如表 2 所示，其中表中的数据都是使用了改进后的高斯模糊方法运行生成的：

数据集	MODA	MODP	Precision	Recall
Wildtrack	87.6	75.3	91.6	92.4
MultiviewX	83.1	78.6	94.2	84.7
CityStreet	44.6	65.7	79.8	59.8
CVCS	44.0	78.3	91.9	48.3

表 2: 4 个不同数据集下的测试结果

7 总结与展望

在本文中，我们通过使用两个场景范围更大、人群数目更多、图片内容信息更加复杂的数据集对MVDet 检测器进行测试，发现效果有了大幅度的下滑，证明 MVDet 检测器在规模更大、图片信息更加复杂的数据集中无法拥有良好的鲁棒性和泛化性。其中评估值 Recall 和 MODA 的数值相较于原数据集下测试的数值下降了将近一倍。测试数值的降低代表 MVDet 检测器在数据集内容信息更复杂的情况下出现了更多错误识别的情况，把没有人的地方错误判断成有人。通过可视化的结果可以看到有一部分区域的行人坐标点生成得十分模糊，无法形成颗粒状。对比真实值的相应区域不难发现该部分行人之间的距离较为接近，遮挡情况较为严重，这种情况下 MVDet 检测器生成的投影特征图中的信息也十分模糊。能否加入更多便于区分某个像素点前后背景差异的模块或者如何能减少行人在不同视角下前后背景信息的差异将会成为模型进一步优化的关键方向。

参考文献

[1] HOU Y, ZHENG L, GOULD S. Multiview Detection with Feature Perspective Transformation[C]//ECCV. 2020.