TPAMI 2021： 基于 event stream 的步态识别，准确率高达90%！

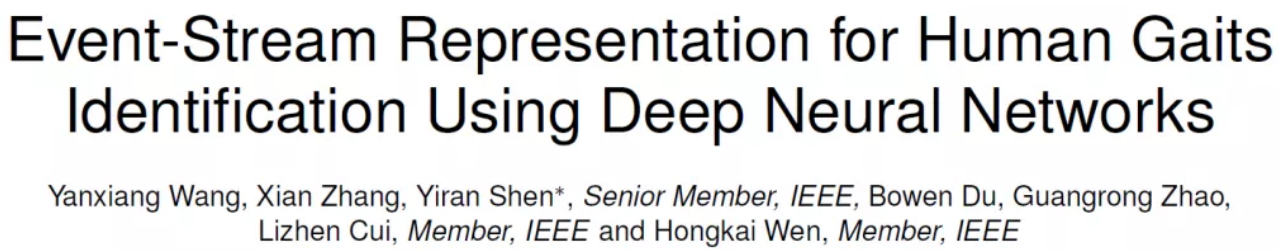
原创 CV开发者都爱看的 极市平台 2021-05-11 22:00:00 手机阅读

作者丨张贤同学
审稿丨邓富城
编辑丨极市平台

极市导读

本文介绍了作者被TPAMI接收的工作，基于 event stream 的两种不同表示形式，提出了一种新的基于 event stream 的步态识别方法。>>加入极市CV技术交流群，走在计算机视觉的最前沿

这是我们发表在 TPAMI 2021 上的一篇论文，论文题目：**Event-Stream Representation for Human Gaits Identification Using Deep Neural Networks**。这篇论文主要是基于 event stream 的步态识别。



本文基于 event stream 的两种不同表示形式，即 image-like representation 和 graph representation，提出了一种新的基于 event stream 的步态识别方法，并分别利用基于 image-like representation 的 CNN ，和基于 graph representation 的 GCN 对 event stream 的步态进行识别。这两种方法分别称为 EV-Gait-IMG 和 EV-Gait-3DGraph，分别取得了 87.3% 和 94.9% 的准确率。在两个基于 event stream 的步态数据集上进行了实验：一个来自我们采集的大型步态识别数据集，另一个把 RGB 步态识别数据集 CASIA-B 转换为 event stream 数据集，称为 EV-CASIA-B。

> 开源地址：https://github.com/zhangxiann/TPAMI_Gait_Identification

首先来看看 event stream是什么。

event stream

event stream 翻译过来是事件流数据，是使用 event camera 拍摄出来的数据，这是最近兴起的一种生物启发的新型视觉传感器，可实时高效地捕捉场景的变化。与之对应的是普通相机拍摄出来的基于帧的视频流数据。普通的手机、相机拍出来的视频都是由多帧图片构成的，一帧图片是基本构成单位。

而 event stream 数据的构成单位不是帧，而是 event。一个 event 由四元组构成：(t, x, y, p)，其中 (x, y) 表示位置，t 表示时间，p 表示 event 的极性。当光线强度增加超过一定阈值时，p=+1，当光线强度减少超过一定阈值时，p=-1。当某个像素的光线强度发生变化的对数超过一定阈值，才会产生 event：

$$\left|\log\left(I_{\text{now}}^{x,y}\right)-\log\left(I_{\text{previous}}^{x,y}\right)\right|>\theta$$

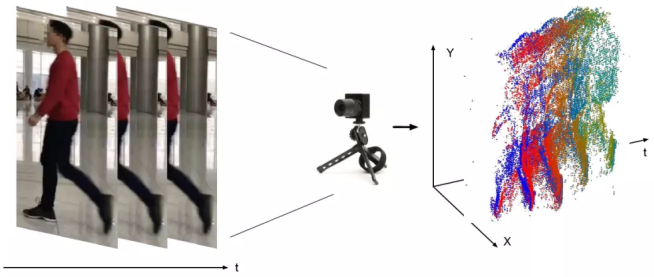
其中 $I_{\text{previous}}^{x,y}$ 表示 (x, y) 位置上一次采样的光线强度，$I_{\text{now}}^{x,y}$ 表示 (x, y) 位置这一次采样的光线强度。

event stream 的数据形式如下所示：

```
0 6 18 1
66 42 35 0
89 38 19 0
94 27 24 1
...
```

每行表示一个 event，对应（t, x, y, p）。

下面来看看看实际的效果。在下图中，左边是普通的相机拍摄出来的视频帧，而使用 event camera，拍摄出来的 event stream 经过三维可视化，得到右边的图。



看上去，这种数据是非常不直观的。然而，这种数据却有很多优点：采样率非常高，一秒可以产生几千个 event（传统相机每秒只能产生上百帧），从而可以捕捉快速运动的物体而不产生失焦，动态范围也更高，可以适应复杂的光线环境。由于每次只生成一个 event，而不是一帧图片，产生的数据量更少、对应相机的能耗也非常低。

利用 event camera 来研究研究计算机视觉的问题，近年来得到了不少关注，针对 event stream 各方面研究的论文数量也有稳步上升的趋势，包括 events to imge、events to video、video to events、object classification、object detection、semantic segmentation、visual-inertial odometry 等。

本文就是针对这种数据来做步态识别。

event stream 虽然有一些优势，但这种数据不是帧，因而无法直接使用 CNN 等方法来处理这种数据，而是需要一定的转换。最直接的思路就是把 event stream 转换为（image-like）帧，再使用成熟的 CNN 等方法来处理，下面会详细阐述。

虽有上述有些方法试图表达时间上的信息，但把 event 转换为帧，多少有一点手工构造特征的意思；而且势必会丢失部分信息，因为一个长长的 event stream 被压缩为了帧。

而我们进一步思考：CNN 不能直接处理 event stream 的原因，是它的输入必须是有固定宽高的帧（结构化数据），而 event stream 是分布在时间和空间上的不连续数据（非结构化数据）。再来仔细思考每个 event 的表示方式 `(t, x, y, p)`，结合上面的图，我们有了一个的想法：把 `(t, x, y)` 看作是 event 的三维坐标，`p` 看作是每个 event 的属性，类似于 point cloud，而最近大火的图神经网络恰好可以用来对这种非结构化数据进行建模。

我们的论文探索了图表示和帧表示两种表示方法，并分别使用了 GCN 和 CNN 两种方法进行建模，分别对应 **EV-Gait-3DGraph** 和 **EV-Gait-IMG**。

图神经网络

目前常见的语音、图像、文本都是很简单的序列或者网格数据，是结构化的数据，可以很方便地使用 CNN 或者 RNN 来处理。

而现实世界中存在这很多非结构化的数据，例如社交网络、交通线路、知识图谱等。

相比于简单的文本和图像，这种网络类型的非结构化的数据非常复杂，处理它的难点包括：

1. 图的大小是任意的，图的拓扑结构复杂，没有像图像一样的空间局部性
2. 图没有固定的节点顺序，或者说没有一个参考节点
3. 图经常是动态图，而且包含多模态的特征

这些问题促使了图神经网络的出现与发展。

这里简单介绍一下图神经网络的发展：

- GCN：它首次将图像处理中的卷积操作简单的用到图结构数据处理中来，并且给出了具体的推导，这里面涉及到复杂的谱图理论。但最后的结论非常简单：$h_v^{k+1}=f\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)}W^k h_u^k+b^k\right),\quad \forall v\in\mathcal{V}$，这就是直接地聚合邻居节点的特征然后做一个线性变换。但是有两个缺点：GCN需要将整个图放到内存和显存，这将非常耗内存和显存，处理不了大图；GCN在训练时需要知道整个图的结构信息，Transductive learning的方式，需要把所有节点都参与训练。
- GraphSAGE：为了解决 GCN 的问题，提出了 GraphSAGE，包含 sample 和 aggregate 两大步骤，以 Inductive learning 的方式，可以很方便地得到新节点的表示。
- GAT：为了解决 GNN 聚合邻居节点的时候没有考虑到不同的邻居节点重要性不同的问题，GAT借鉴了 Transformer 的 idea，引入 masked self-attention 机制，在计算图中的每个节点的表示的时候，会根据邻居节点特征的不同来为其分配不同的权值。

感兴趣的同学可以阅读这篇入门文章--万字长文带你入门 GCN

https://zhuanlan.zhihu.com/p/120311352

图神经网络从大体上可以分为频域卷积和空域卷积。频域卷积是先把图上的信号，通过图傅立叶变换，映射到频域后再做卷积操作。但是频域卷积需要的图节点数量是固定的，而 event stream 产生的节点数量不是固定的，而且每个 event 的坐标也是非常重要的。而空域卷积可以利用每个节点的位置信息，并且没有限制图的结构。因此我们使用了空域卷积。

数据集

由于目前只有传统相机录制的步态数据集，而没有公开的针对步态识别的 event stream 数据集。

我们首先把现有的一个比较有代表性的步态数据集进行转换，得到 event stream 的数据集。CASIA-B 数据是传统相机录制的步态数据集，包括124 个人，每个人分别从 6 个不同的角度录制了步态数据，每个角度 11 个数据。我们在电脑屏幕上播放这个数据集，并且使用了 event cameras 录制该数据集，得到了 EV-CASIA-B。

这个数据集不是直接录制真实世界的步态，而是在电脑屏幕上重放进行录制的，而我们使用的屏幕只有 60 Hz 的刷新率，这可能会引入噪声，而且没有充分利用到 event camera 的高采样率的特性。因此，我们还采集了真实世界的基于 event stream 的步态数据集。

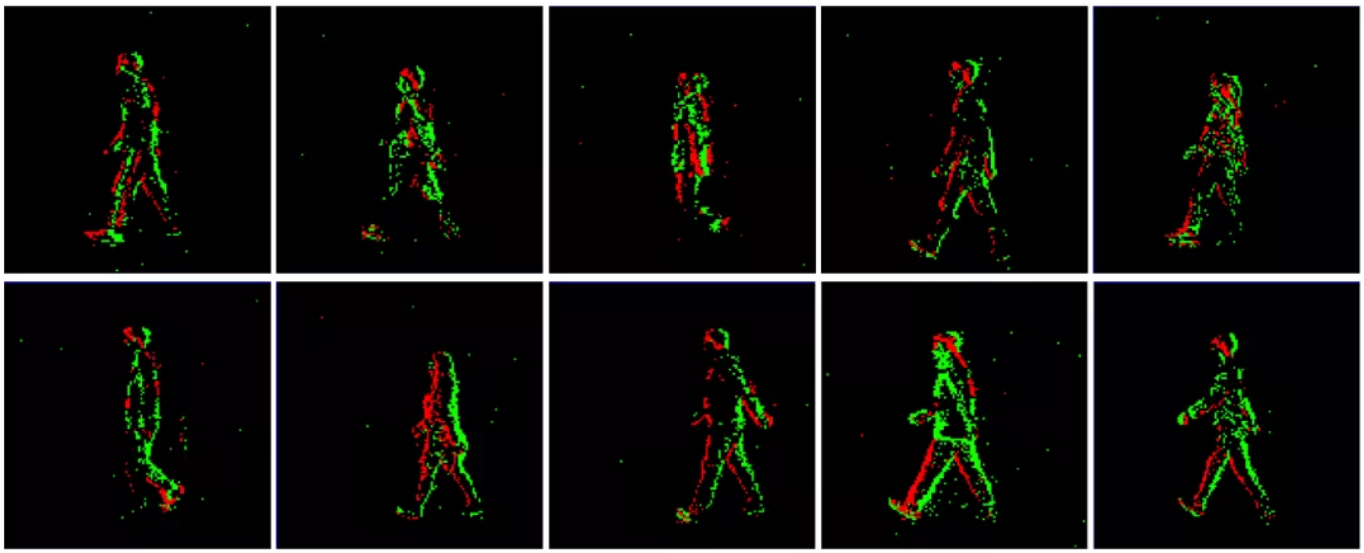
我们使用 iniLabs 的 DVS128 event camera（分辨率是 128 X 128），招募了 20 位志愿者，采集了两个数据集，分别是 DVS128-Gait-Day 和 DVS128-Gait-Night。

DVS128 event camera：https://inivation.github.io/inivation-docs/Hardware%20user%20guides/User_guide_-_DVS128.html

DVS128-Gait-Day 是在白天采集的，包括两组数据。每位志愿者采集了 100 次步态数据，每次步态行走数据时间约 3 到 4 秒，作为第一组数据：训练集。一周之后，我们对这 20 位志愿者进行了第二组数据，作为测试集。总共包括 4000 个样本。

DVS128-Gait-Day 是在夜晚采集的，也包括相隔一周的两组数据，总共包括 4000 个样本。这是为了验证 event camera 的高动态范围特性。在光线暗的环境下，传统的相机表现是很差的，而 event camera 依然能够较好地捕捉步态的数据。

下图是对我们采集的步态数据 event stream 的一个可视化：



使用图神经网络处理 event stream（EV-Gait-3DGraph）

把 event stream 转换为图，是在这篇论文里首次提出的：`[Graph-based Feature Learning for Neuromorphic Vision Sensing](javascript:void(0))` 。作者首先把 event stream 平均切分为 8 个 slice，将每个 slice 表示为一个图，通过图卷积神经网络（Graph2Grid 模块）将每个 slice 转换为一张帧，得到 8 张帧，每张帧看作是一个 channel，输入到 3DCNN 来学习时空特征。

不同于该论文将 event stream 切分为多个 slice 进行处理，我们这里直接对整个 event stream 构造图，进行处理。

我们都知道，图数据的主要元素有两个：节点和边。对于 event stream，每个 event 就是图中的节点，因此我们要做的首先是构造节点之间的边。

降采样

这里有一个需要注意的点是，考虑到 event camera 的采样率非常高，达到 10000Hz 以上，而我们每次步态数据持续 3 到 4 秒，产生几万个 event，这个数据规模是非常大的，而如果直接在所有节点上面构造边，边的数量要比节点数量再大一个数量级，这是很消耗计算资源的。因此为了减少计算量，我们首先对 event stream 进行了降采样。这里使用了 nonuniform OctreeGrid filtering 算法来减少 event 的数量，同时保留原始 event stream 的结构。前面我们讲过，每个 event 表示为：(t, x, y, p)，我们把 event 的 (t, x, y) 当作是 event 的三维坐标，这里 x 和 y 的取值范围是 [0, 128]，而 t 的取值单位是纳秒，我们把 t 也缩放到 [0, 128] 之间，构成 [128 X 128 X 128] 的 grid box，然后把这个 box 划分为多个小 box，每个小 box 里的 event 数量不超过 MaxNumEvents（MaxNumEvents 的取值是 [20, 80]，这是一个超参数），每个小 box 里随机保留一个 event，其余的 event 数据丢弃。这样，我们可以大大减少 event 数量。

构造边

降采样后，剩下的节点之间需要构造边。我们使用了一种很直观的方法，对于两个节点 $v_i = (x_i, y_i, t_i, p_i)$ 和 $v_j = (x_j, y_j, t_j, p_j)$ 来说，如果它们之间的距离小于一定阈值（R），那么就认为这两个节点之间有连接的边。距离计算共识如下：

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha(t_i - t_j)^2} < R$$

其中 α 是用于调节时域和空域之间的比例。<r$< p=""></r$<>

图的连通表示为 G=(V, E, P)，其中 V 表示顶点，E 表示边，P表示 event 的光线强度变化，也是节点的特征。最后得到图的邻接矩阵 A，如果点 v_i 和 v_j 之间有边，那么矩阵中 $A_{i,j}$ 的值为1，否则为0，对角线上的值全为 1，表示节点自己有边。

GMM 图卷积

本文使用的是 Gaussian Mixture Model (GMM)-based graph convolution。公式如下：

$$f'_p = \sum_{k=1}^{K} \sum_{y \in \mathcal{N}(x)} g_k w_k^p(u(x,y)) f(y) \quad p = 1, 2, 3, \ldots, P$$

其中 f'_p 是 P 维输出特征向量的其中一个元素，g_k 是 第 k 个 Gaussian kernel 的权重，$f(y)$ 是节点 $v(y)$ 的特征向量，$\mathcal{N}(x)$ 是节点 x 的邻居节点，$w_k^p(u(x,y))$ 是节点 x 和 y 之间的权重参数，用于聚合相邻节点的特征向量。在 Gaussian Mixture Model (GMM)-based kernel 中，权重参数 $w_k(u) = \exp\left(-\frac{1}{2}(u - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(u - \boldsymbol{\mu}_k)\right)$。

其中 $\sum_k^{-1}$ 是第 k 个高斯模型的协方差矩阵，μ_k 是第 k 个高斯模型的均值向量。

此外，我们还使用了图神经网络的 Graph-ResNet，这与普通的残差连接类似。

Graph MaxPooling

最后，我们使用了 Graph MaxPooling，可以降低网络的复杂度，缓解网络深度过拟合的问题。由于我们上面把 event 的坐标限定为 [128 X 128 X 128]，这里的 MaxPooling 与 CNN 里的 MaxPooling 类似，把坐标每个维度按照 $poolingsize = d$ 切分，经过 MaxPooling 的图在每个坐标维度的最大 event 数量是 [$\frac{128}{d} X \frac{128}{d} X \frac{128}{d}$]。

EV-Gait-3DGraph 网络如下：

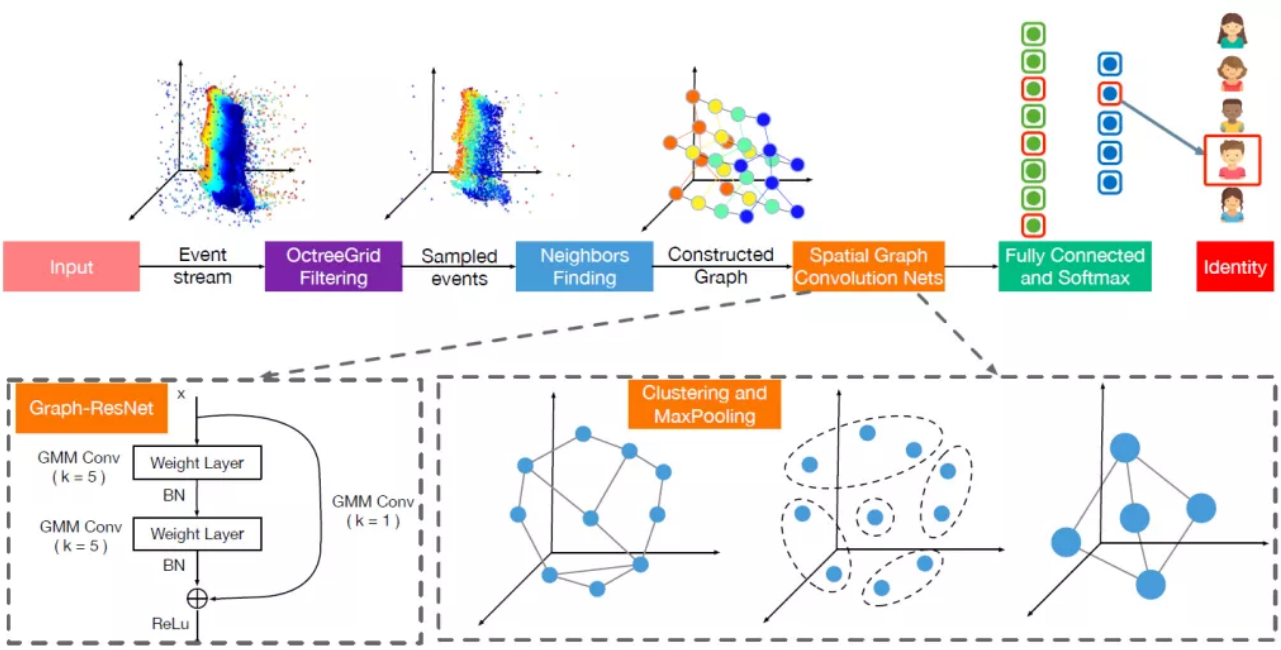


Fig. 2: Workflow of EV-Gait-Graph3D.

graph 数据首先经过 $\mathrm{GC}_0(5,64)$，表示 kernel size 是 5，输出的 feature size 是 64；接着使用 $\mathrm{MP}_0(4)$ 来聚合图中的节点，类似于 CNN 中的 polling（更详细的说明请看原论文）。接着使用 3 个 Graph-ResNet：$\mathrm{Gres}_1(5,1,128)$、$\mathrm{Gres}_1(5,1,256)$、$\mathrm{Gres}_1(5,1,512)$，输出的 feature size 分别是 128、256、512。每个 Graph-ResNet 的 kernel size 分别是 $K_1=5$，$K_2=1$。每个 Graph-ResNet 后面都会接一个 pooling layer，polling size 分别是 6、24、64。最后经过 FC 层得到输出。

综上所述，EV-Gait-3DGraph 的网络结构可以描述为：

$$\mathrm{GC}_0(5,64)-\mathrm{MP}_0(4)-\mathrm{GRes}_1(5,1,128)-\mathrm{MP}_1(6)-\mathrm{GRes}_2(5,1,256)-\mathrm{MP}_2(24)-\mathrm{GRes}_3(5,1,512)-\mathrm{MP}_3(64)-\mathrm{FC}(1024)$$

使用 CNN 处理 event stream（EV-Gait-IMG）

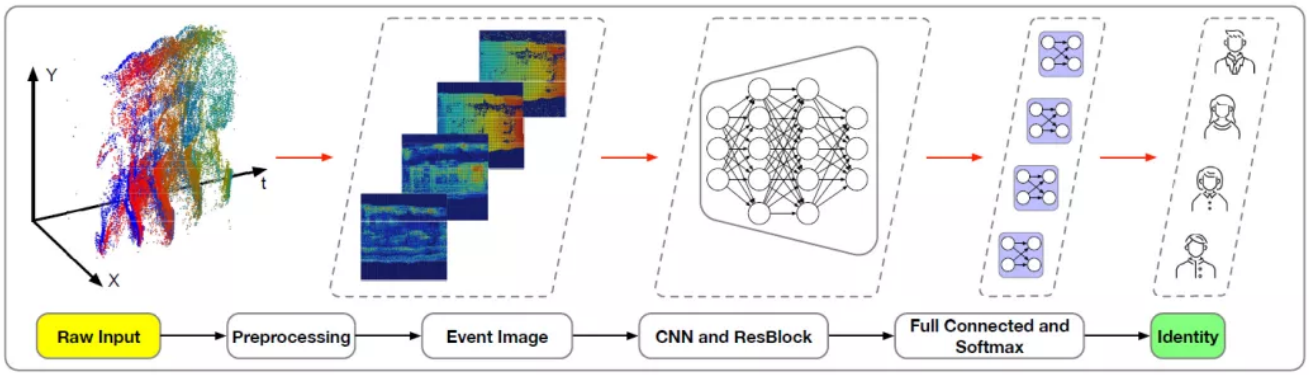
把 event stream 转换为 帧

前面说过，要使用 CNN 处理 event stream，首先需要把 event stream 转换为帧。在转换上有很多方法，例如：

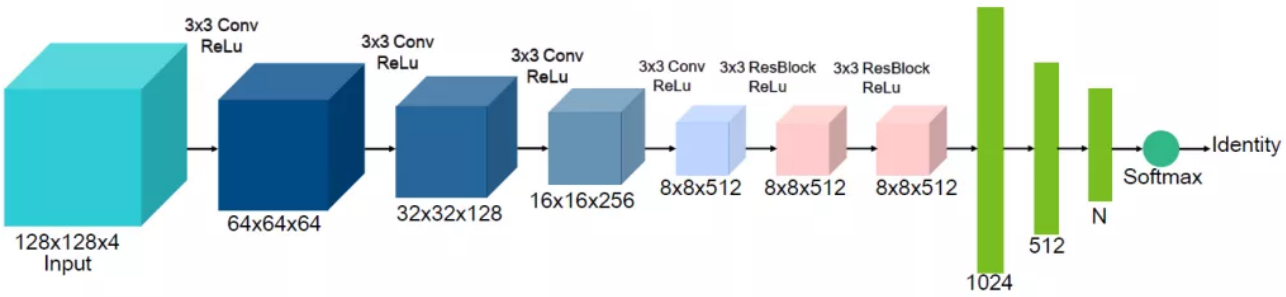
- 一帧的每个像素点 `(i, j)` 的值是每个像素点发生的 event 数量的和，根据 event stream 的 p 有 +1 和 -1 两种取值，一个帧计算 `p=+1` 的 event 数量，另一个计算 `p=-1` 的 event 数量。这种表示方法表达 event stream 的空间属性。
- 针对每个像素 `(i, j)`，计算 $r_{i,j}=\frac{t_{i,j}-t_{begin}}{t_{end}-t_{begin}}$，其中 $t_{i,j}$ 表示像素 `(i, j)` 位置发生的最后一个 event 的时间，t_{begin} 表示整个 event stream 的第一个 event 的时间，t_{end} 表示整个 event stream 的最后一个 event 的时间。同样，区分 `p=+1` 和 `p=-1`，可以得到两个帧。这种表示方法表达 event stream 的时间属性。
- 第三种就是上述两种方法的结合，得到四个帧。
- 如果不区分 `p=+1` 和 `p=-1`，则第三种方法可以得到 2 个帧。
- 更早期的一些方法，如 Histograms of Time Surfaces，以及 Histograms of Averaged Time Surfaces，不再做更多的赘述。

在我们的上一篇论文 `EV-Gait: Event-based Robust Gait Recognition using Dynamic Vision Sensors(CVPR 2019)` 中，也是把 event stream 转换为帧，再输入 CNN 模型。

EV-Gait-IMG 网络如下：



(a)



(b)

这部分比较简单，就是使用了简单的 CNN 和 FC 层。输入的图片首先经过 4 个 CNN，每个 CNN 的 filter size 是 3×3，stride 是 2，输出的 channel 数分别是 64、128、256、512。然后经过 2 个 ResBlock：filter size 是 3×3，stride 是 1，输出的 channel 数是 512。最后经过两个 FC 层得到输出。

实验结果

EV-Gait-3DGraph vs. EV-Gait-IMG

在 DVS128-Gait-Day 数据集上，

针对 EV-Gait-3DGraph，我们对 MaxNumEvent（降采样的超参数）、neighboring range（构造边的距离阈值）、kernel Size、pooling size、number of layers（网络层数） 等超参数进行了充分实验。

一个有趣的发现是：我们发现 MaxNumEvent 太高或者太低，准确率都会有所下降。我们认为：这是因为 MaxNumEvent 太小时，降采样不够充分，导致边太多，引入噪声；而 MaxNumEvent 太大时，降采样过度，导致原始 event stream 的信息不能够很好地保留。neighboring range 的参数选择也有类似的规律。

而对 EV-Gait-IMG，我们帧的 4 种转换方法、ResBlock 的 kernel size、number of layers（网络层数）等超参数进行了充分实验。

EV-Gait-3DGraph 达到了 94.9% 的准确率，而 EV-Gait-IMG 准确率是 87.3%。

其他方法

我们还对比了其他方法：如 2DGraph-3DCNN、LSTM-CNN、SVM-PCA。其中最好的方法是 2DGraph-3DCNN，准确率有 92.2%，比我们的方法略逊一筹。

One More Thing

此外，我们还对比了不同训练样本数量对 EV-Gait-3DGraph 和 EV-Gait-IMG 的影响。一个有趣的发现是：

- 当数据量足够大时，EV-Gait-3DGraph 的效果最好。

- 但是当数据量少时，EV-Gait-IMG 性能更好，能够更快收敛。

这表明了图神经网络虽然性能好，但需要的数据量可能更大。

最后，我们把原始持续 3 到 4 秒的 event stream 拆分为更短的 event stream 进行训练。

结果表明：训练时每个 event stream 样本越短，准确率越低；每个 event stream 样本越长，准确率越高。因为 event stream 越长，包含的步态特征越多，因此模型的准确率越高。这表明了模型主要是从 event stream 中的步态特征中进行学习，而不是仅仅从人的体型特征中学习。

EV-CASIA-B

在 EV-CASIA-B 数据集上，我们使用 EV-Gait-3DGraph、EV-Gait-IMG 和 **3D-CN**、**Ensemble-CNN** 方法进行了对比，结果差不多。这里不再赘述，感兴趣来阅读原论文吧。

最后

event camera 的确是一种有趣的设备，采样率非常高，所以基本没有运动模糊，能处理非常高速的运动。但这货不输出传统图像，所有传统 CV 算法都跑不了。最直接的方法就是把 event stream 转换为传统 CV 算法能处理的形式。另一种方法就是使用图神经网络之类的来处理种数据，当然也要经过一定的转换。今年也看到了有人把这种相机用于无人驾驶、以及无人机上，与传统相机结合，说不定能带来更多的 insight。期待这方面更多的研究。

本文亮点总结

1.本文基于 event stream 的两种不同表示形式，即 image-like representation 和 graph representation，提出了一种新的基于 event stream 的步态识别方法，并分别利用基于 image-like representation 的 CNN ，和基于 graph representation 的 GCN 对 event stream 的步态进行识别。

算法技巧（trick）： 深度神经网络模型训练中的 tricks（原理与代码汇总）｜神经网络训练trick总结｜深度学习调参tricks总结

最新CV竞赛： 2021 高通人工智能应用创新大赛｜CVPR 2021 | Short-video Face Parsing Challenge｜3D人体目标检测与行为分析竞赛开赛，奖池7万+，数据集达16671张！