首都师范大学
CAPITAL NORMAL UNIVERSITY

人体动作识别算法阶段

汇报人：刘艾林

目录
DIRECTORY

1

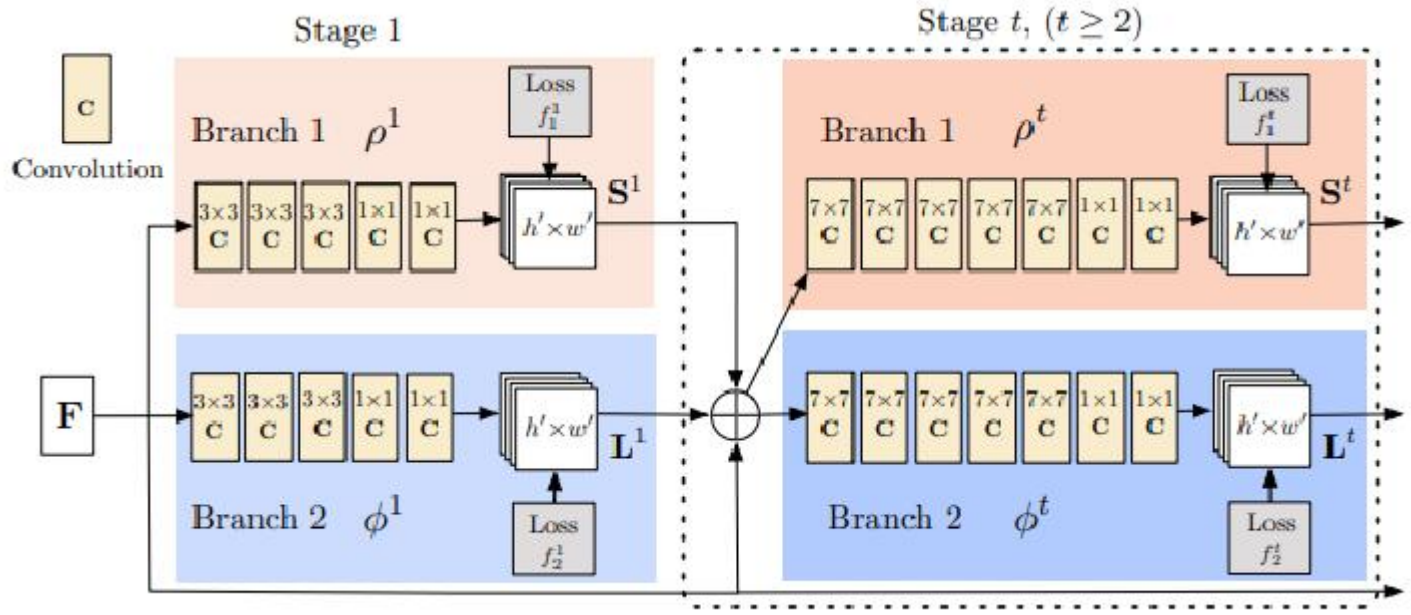
基于 OpenPose 的人体动作识别

基于 OpenPose 的人体动作识别

1.1 OpenPose

基于卡耐基梅隆大学于 2016 年提出的 **OpenPose 骨骼关节点提取算法**，详细说明该算法的执行过程与方法。先是**采用自底向上的方法回归出视频帧中所有的关节点**，然后创造性的提出了关节点亲和场这一概念和方法，获取关节点配对的置信度，最后再利用解决**二分图匹配**的思路完成人体骨架的拼接。

1.2 OpenPose网络结构



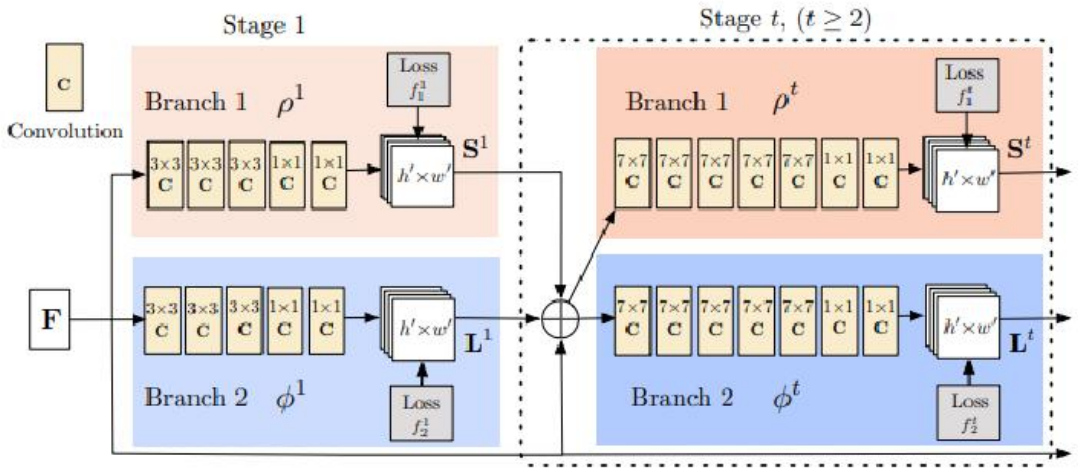
$$S^t = \rho^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \qquad (2.1)$$

$$L^t = \varphi^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \qquad (2.2)$$

其中，S^t 为 t 阶段输出的关节点置信图，L^t 为 t 阶段输出的两两关节点的亲和度，也就是权重系数。

损失函数是保证网络能收敛的最重要的关节点，因此作者对两分支的损失函数均采用 L2 损失。训练时，每个阶段都会产生损失，为了避免梯度消失，预测时只使用最后一层的输出。阶段损失值公式如 2.3、2.4 所示。

1.2 OpenPose网络结构



$$S^t = \rho^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \tag{2.1}$$

$$L^t = \varphi^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \tag{2.2}$$

$$f_S^t = \sum_{j=1}^{J}\sum_{p} W(p) \cdot \| S_j^t(p) - S_j^*(p) \|_2^2 \tag{2.3}$$

$$f_L^t = \sum_{c=1}^{C}\sum_{p} W(p) \cdot \| L_c^t(p) - L_c^*(p) \|_2^2 \tag{2.4}$$

$$f = \sum_{t=1}^{T}\left(f_S^t + f_L^t\right) \tag{2.5}$$

其中，带上标*的表示真值，带上标t的是不同阶段的预测值，p是每一个像素点，$W(p)$代表示该点缺失标记，只有0和1两个值。若为0，则损失值不予计算。总体的损失值公式如公式2.5所示，为各个阶段的损失值之和。

1.3 人体模型构建

1.3.1骨架数据规整化

格式说明如下：0 号位置是鼻子, 1 号位置是颈部，2 号位置是左肩膀，5 号位置是右肩膀，3 号位置是左胳膊肘，6 号位置是右肘， 4 号位置是左手腕，7 号位置是右手手腕，8 号位置是左髋关节，11 号位置是右髋关节，9 号位置是左腿膝盖，12 号位置是右腿膝盖，10 号位置是左脚踝，13 号位置是右脚踝，14 号位置是左眼，15 号位置是右眼，16 号位置是左耳，17 号位置是右耳和 18 号背景信息点。通过 OpenPose 预先训练好的模型，获取这 19 个骨骼关节点的 x 轴信息，y 轴信息，还有每一个点的置信度 c，代表该关节点的识别准确程度。本系统在特征提取方面只需要 x 轴信息，y 轴信息，没有加入置信度。在 RBG 图像中人体姿态关节点示例如图 3.2 所示。

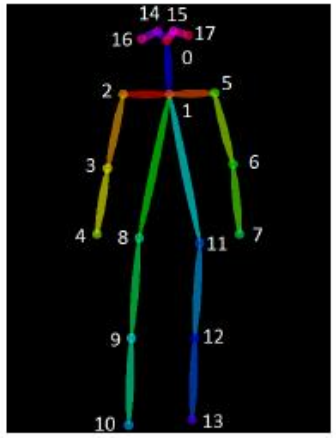


图 3.1 COCO 人体姿态骨骼模型

$$c_0 = (c_x, c_y) \tag{3.1}$$

则第 j 个关节点的平面坐标如公式 3.2 所示：

$$c_j = (c_x^j - c_x, c_y^j - c_y) \tag{3.2}$$

那么，人体骨架信息可以用 S 来表示，如公式 3.3 所示：

$$S = (C, E),\ C = \{c_0, c_1, \ldots, c_{17}\}, E = \{e_0, e_1, \ldots, e_{17}\} \tag{3.3}$$

其中， C 表示人体所有关节点位置的集合，E 表示肢体向量的集合。在对关节点位置信息规整化处理之后，在第 t 时刻，也就是第 t 帧视频中，第 j 个关节点的位置可以定义为 $c_j(t) = (x_{ij}, y_{ij})$，其中， $j \in \{0,1,\ldots 17\}$。

1.3 人体模型构建

1.3.2 全量肢体夹角特征设计

全量肢体夹角共有九组。其中包括：左手肘关节：2-3-4；右手肘关节：5-6-7；左腿膝盖关节：8-9-10；右腿膝盖关节 11-12-13；左肩膀关节：3-2-1；右肩膀关节：1-5-6；左颈部：0-1-2；右颈部：0-1-5；中间躯干：1-8-11。肢体的数学表示就是点与点连接而成的向量，肢体夹角的具体表示关系如表 3.1，骨架图如 3.3 所示：

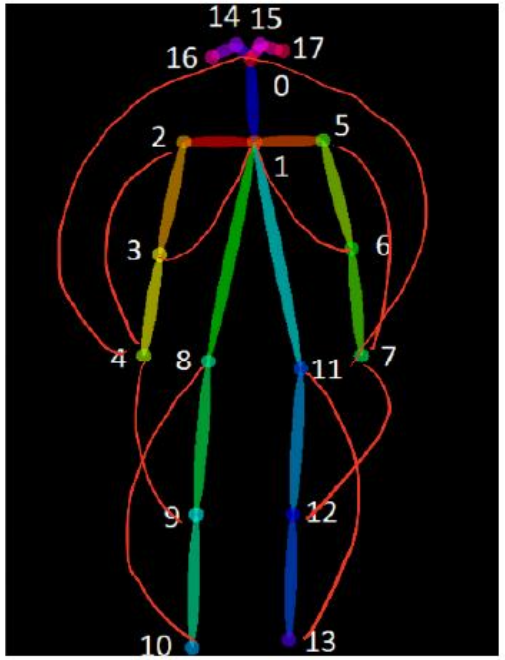


图 3.3 特征设计图

表 3.1 肢体夹角关系

肢体夹角	名称	向量
θ_1	左手肘关节	$r_{3,4}, r_{3,2}$
θ_2	右手肘关节	$r_{6,5}, r_{6,7}$
θ_3	左腿膝盖关节	$r_{9,10}, r_{9,8}$
θ_4	右腿膝盖关节	$r_{12,13}, r_{12,11}$
θ_5	左肩膀关节	$r_{2,3}, r_{2,1}$
θ_6	右肩膀关节	$r_{5,6}, r_{5,1}$
θ_7	左颈部	$r_{1,0}, r_{1,2}$
θ_8	右颈部	$r_{1,0}, r_{1,5}$
θ_9	中间躯干	$r_{1,8}, r_{1,11}$

表中 $r_{i,j}$ 是指肢干连接的向量，从关节 i 指向关节 j。其中，$i, j \in (0,1,\ldots,17)$。对于第 n 个肢干的大小为 θ_n，设两肢向量分别为 $r_{i,j}$，$r_{i,k}$，则夹角公式如 3.4 所示：

$$\theta_n = arccos \frac{r_{i,j}^1 * r_{i,k}^1 + r_{i,j}^2 * r_{i,k}^2}{\sqrt{r_{i,j}^{1\,2} + r_{i,j}^{2\,2}} + \sqrt{r_{i,k}^{1\,2} + r_{i,k}^{2\,2}}} \tag{3.4}$$

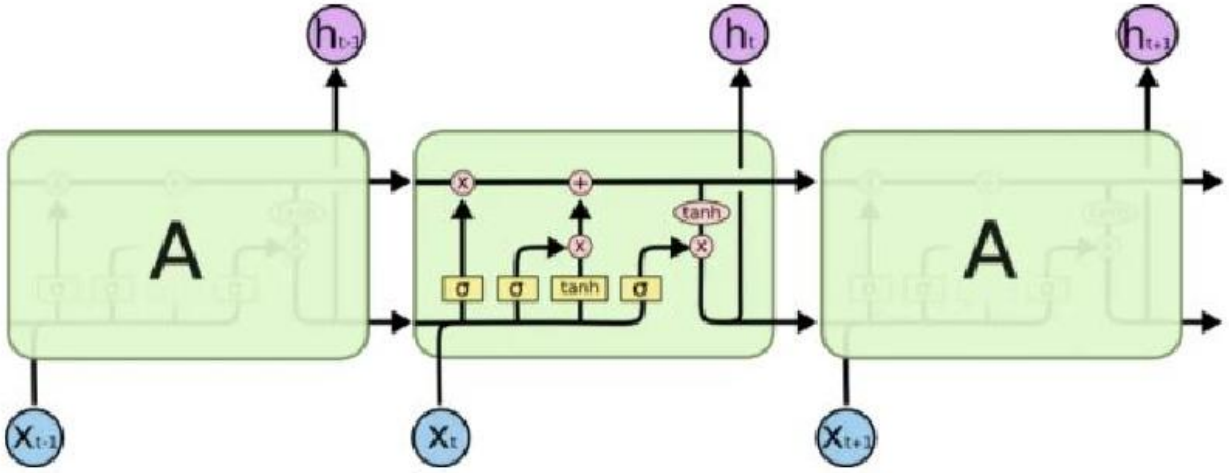
2

人体动作分类算法

2. 1 LSTM网络

在处理**无时序数据**的时候，传统的神经网络能够很好的解决问题，但是当数据是诸如语音音阶、视频帧这样的**时序数据**的时候，效果往往不是很好。

循环神经网络的结构设计类似于**自动化领域内的反馈机制**，即：上一时刻的输出变成下一时刻输入的一部分，以此充分利用获得的信息。



LSTM 结构图

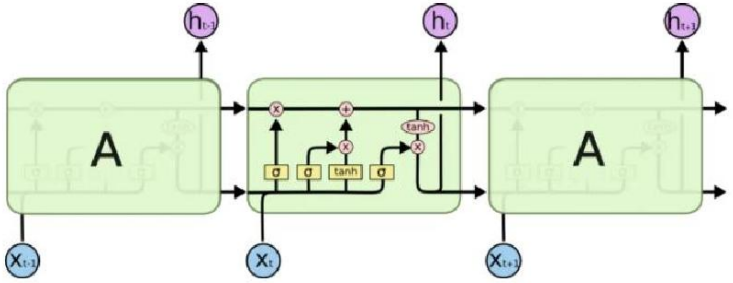
2. 2 基于 LSTM 的人体动作识别

LSTM 网络的搭建与执行过程。

首先，通过划分数据集，使用训练集训练获得网路的各项参数的最优值。

同时，为了优化网络的预测能力与效果，增加注意力机制，使得网络能够对输入的特征数据进行加权运算，进一步提高最终的识别精度。

LSTM 神经网络模型的训练过程同大多数深度学习的网络模型一致，主要包含了三部分内容，前向传播、反向传播和梯度更新。接下来主要借助该网络的前向传播来介绍网络的计算过程



LSTM 结构图

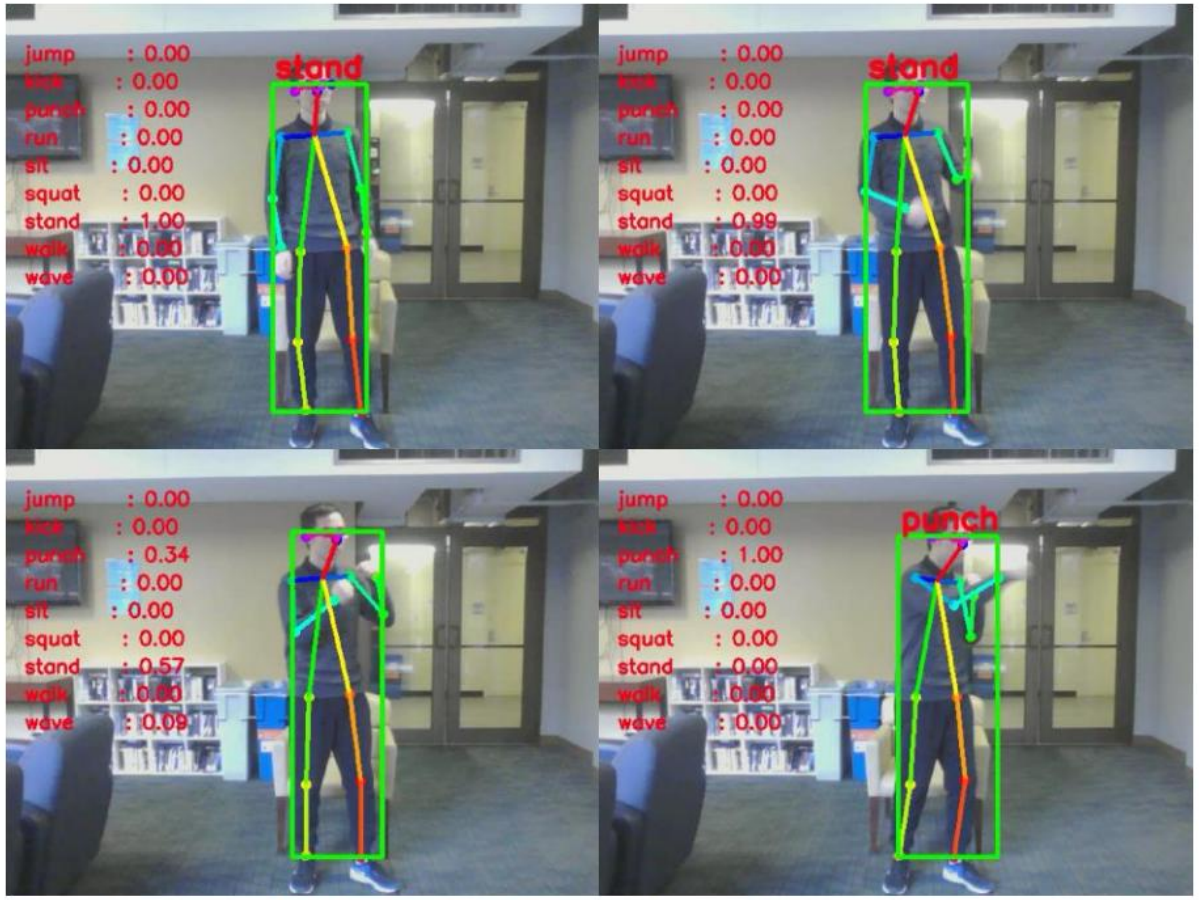
2. 3 基于 OpenPose与LSTM 的人体动作识别系统架构

基于 OpenPose 与LSTM 的人体行为识别系统，主要由数据获取、动作分割、动作模型构建、Opencv 绘图、神经网络构建识别共五个模块组成。

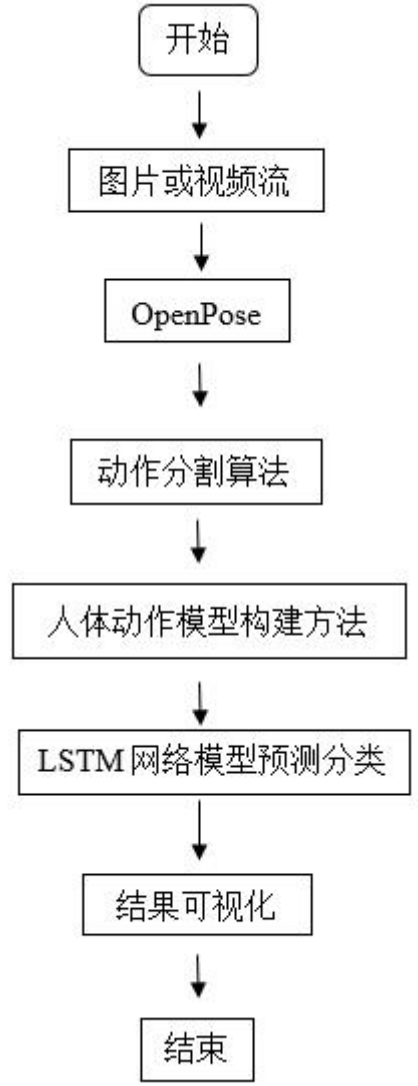
在数据获取模块中，通过 OpenPose对视频帧数据处理拿到关节点位置信息，输出关节点时空信息矩阵。动作分割模块用于对关节点时空矩阵的裁剪过滤操作。动作模型构造模块主要是对关节点信息进行处理，获得网络所需要的特征信息。

神经网络模块完成动作分类识别的任务,Opencv 绘图模块完成骨架图的画面和识别结果的展示

2. 3 基于 OpenPose与LSTM 的人体动作识别系统架构



基于 OpenPose 的行为识别系统工作流程



3

实验仿真与分析

3.2实验仿真与分析

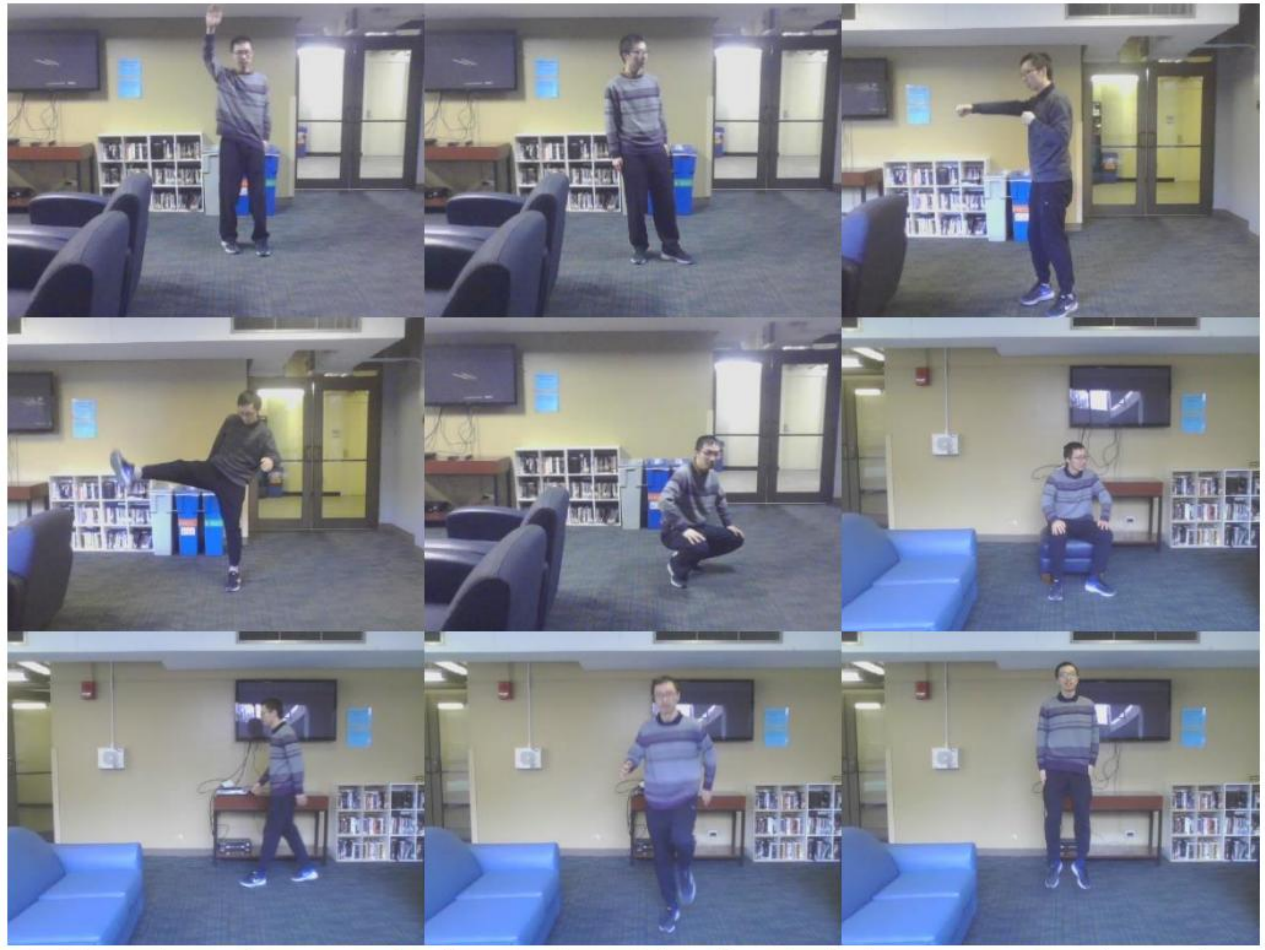
实验环境配置如下表所示，基于 Tensorflow 搭建的网络模型，编程语言 Python 3.7 实现。

表 4.1 训练与测试的环境配置

软件硬件实验平台	具体参数型号
操作系统	Windows 10 专业版
机器学习框架	Tensorflow
实验软件	Pycharm
CPU	Intel Core i7-7700HQ @ 2.80GHz 四核
GPU	NVIDIA GEFORCE GTX1060

3. 1 数据集制作

本项目收集了9种数据格式的视频数据分别是['stand', 'walk', 'run', 'jump', 'sit', 'squat', 'kick', 'punch', 'wave']



3. 1 数据集制作

本项目收集了9种数据格式的视频数据分别是**['stand', 'walk', 'run', 'jump', 'sit', 'squat', 'kick', 'punch', 'wave']**

每个视频的长度从**0.8秒到2分钟**不等，并且每个视频仅限包含一种类型的操作。例如，在一个视频中，踢了0.8秒；在另一个视频中，不停地挥舞着手臂长达2分钟。

这些视频是以**640x480的大小**和**10帧/秒**的帧速率记录的，因此它足够快，可以捕捉动作的整个动作。的总数每个动作的帧如表2.1所示

这些视频是以**640x480的大小**和**10帧/秒**的帧速率记录的，数据集分布

Number of frames of the 9 actions as training data.

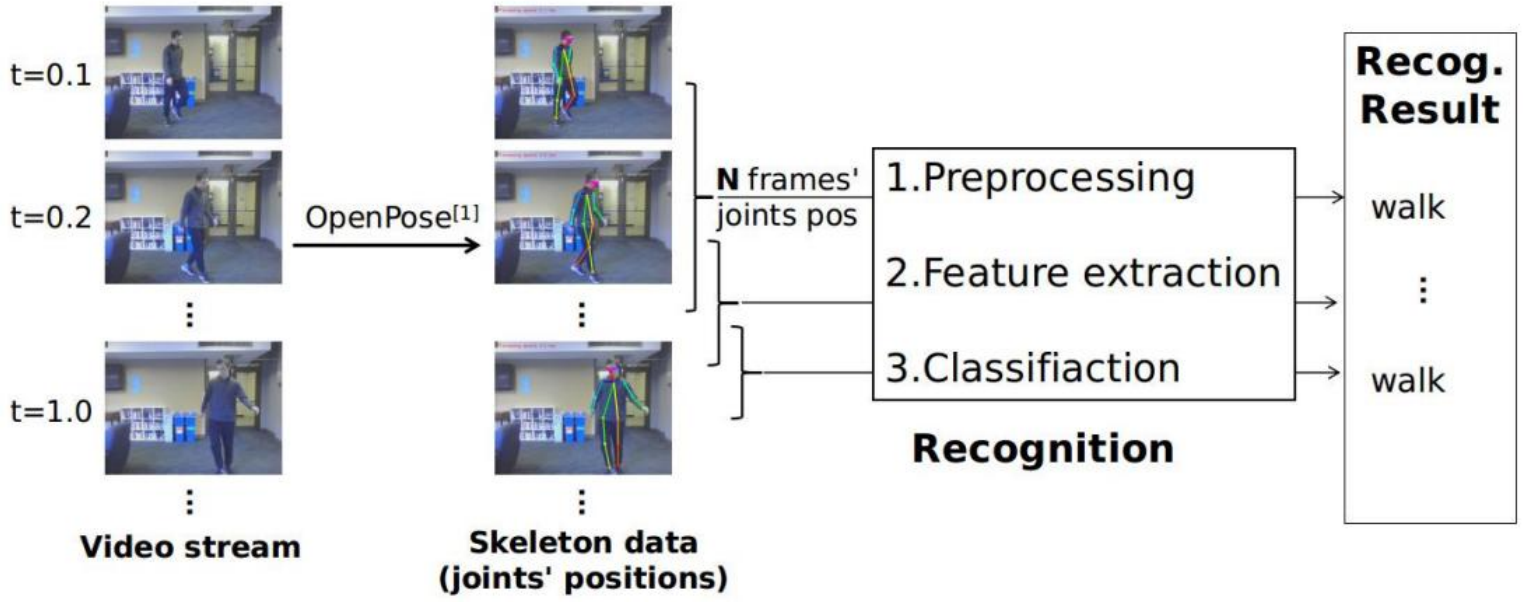
Actions	1	2	3	4	5	6	7	8	9	Total
	wave	stand	punch	kick	squat	sit	walk	run	jump	
Number of frames	1239	1703	799	1162	964	1908	1220	1033	1174	11202

3. 1 数据集制作

数据集处理流程：

系统的输入是来自摄像机或视频文件的视频流。

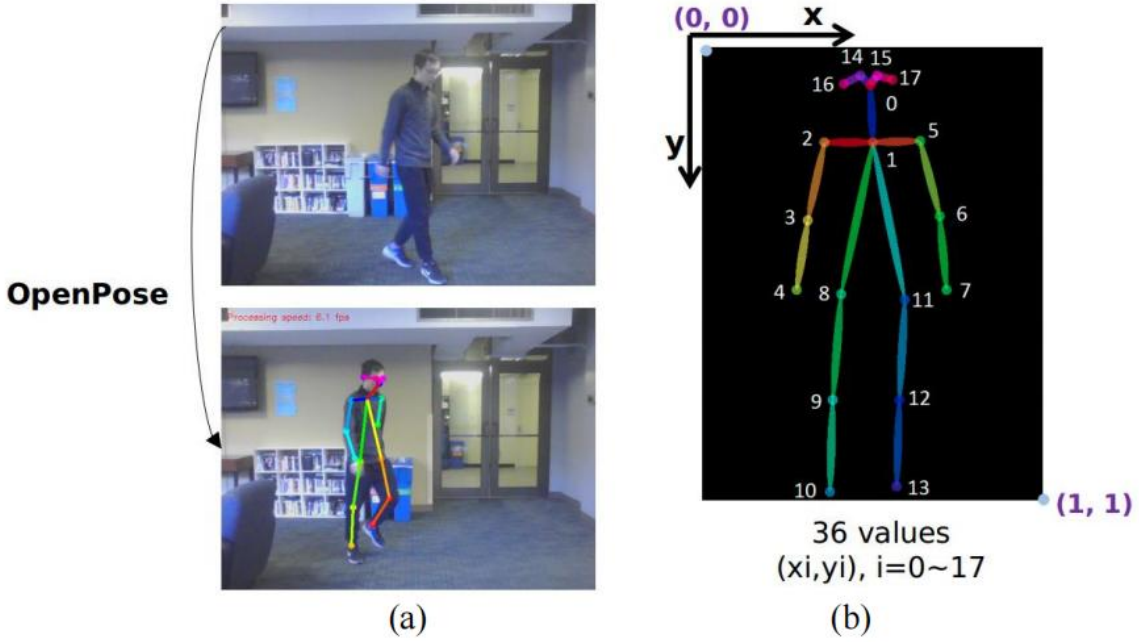
然后采用**OpenPose**算法处理每个帧。接下来，大小为N的滑动窗口聚合前**N个帧的骨骼数据**。这些骨骼数据经过预处理并用于**特征提取**，然后将其**输入分类器**以获得（该窗口的）最终识别结果。类似地，为了实现实时识别框架，窗口沿着的时间维度逐帧滑动并输出每个视频帧的标签。



3. 1 数据集制作

数据集处理流程：

采用OpenPose算法从图像中检测人体骨骼。OpenPose的输入是一个图像，输出是所有人类的骨骼算法检测。每个**骨骼有18个**关节，包括**头部、颈部、手臂和腿部**，如图所示如图3.2所示。每个关节位置在图像坐标中表示，坐标值为x和y，所以每个骨架总共有36个值。



3. 1 数据集制作

数据归一化操作:

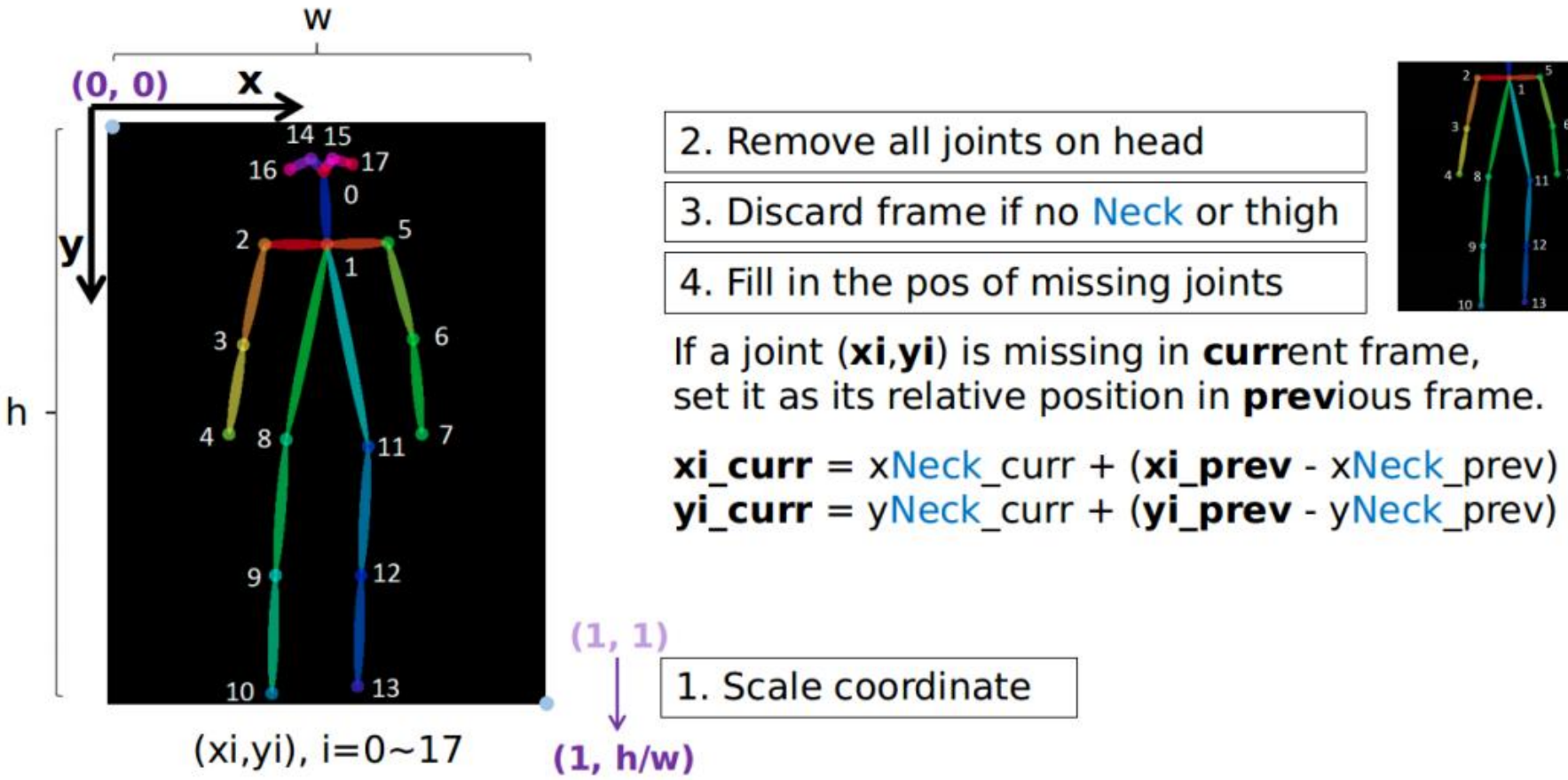
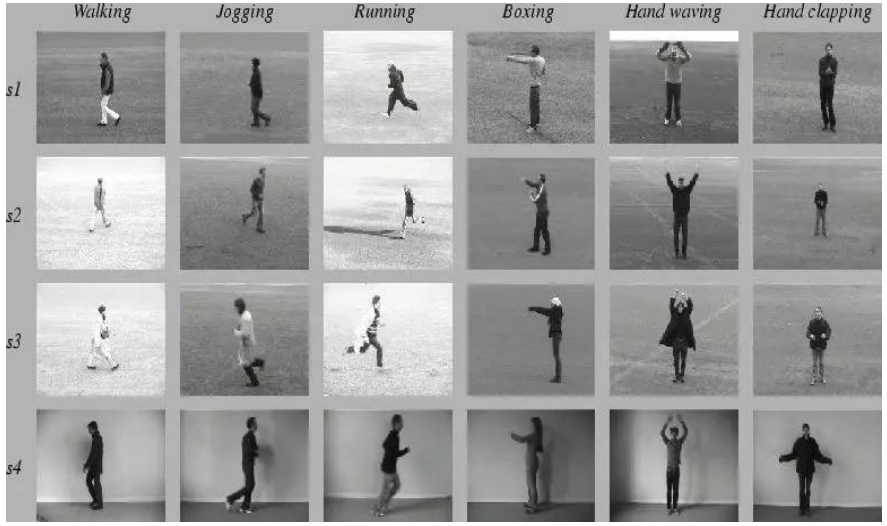


Fig. 3.3. Preprocessing joint positions in 4 steps.

3. 1 数据集制作

数据集相关获取方式：

KTH数据集：2004年发布，**包含 6 类人体行为：行走、慢跑、奔跑、拳击、挥手和鼓掌**，每类行为由 25 个人在四种不同的场景（室外、伴有尺度变化的室外、伴有衣着变化的 室外、室内）执行多次，相机固定。该数据库总共有 **2391个视频样本**。视频帧率为 25 fps，分 辨率为 **160×120**，平均长度为 4 秒。



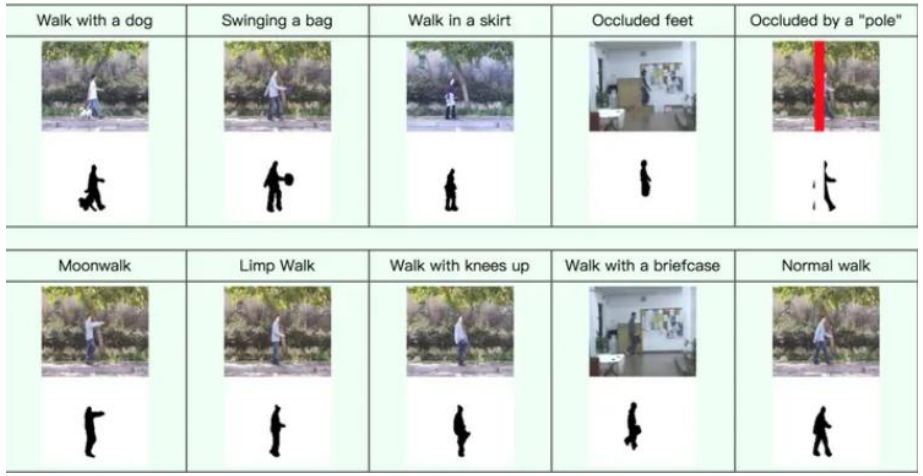
官网： **https://www.nada.kth.se/cvap/actions/**

3. 1 数据集制作

数据集相关获取方式：

数据同样是固定镜头下的**10个典型动作**的视频，同时数据集提供了一些带有其他物体的动作作为干扰，可以测试模型的鲁棒性。

官方同时提供了去除背景的程序，但是数据集的数据量比较少的90组常规数据和21组鲁棒测试数据，对于目前的模型训练来说显得有些不足，不过对于本来就需要用小数据的模型比如迁移学习或者One-short Learning来说或许是适合的数据集。



官网：

http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html

谢谢观看