关键点检测的总结

blog.csdn.net/qq_36338754/article/details/90475722

一、三款模型

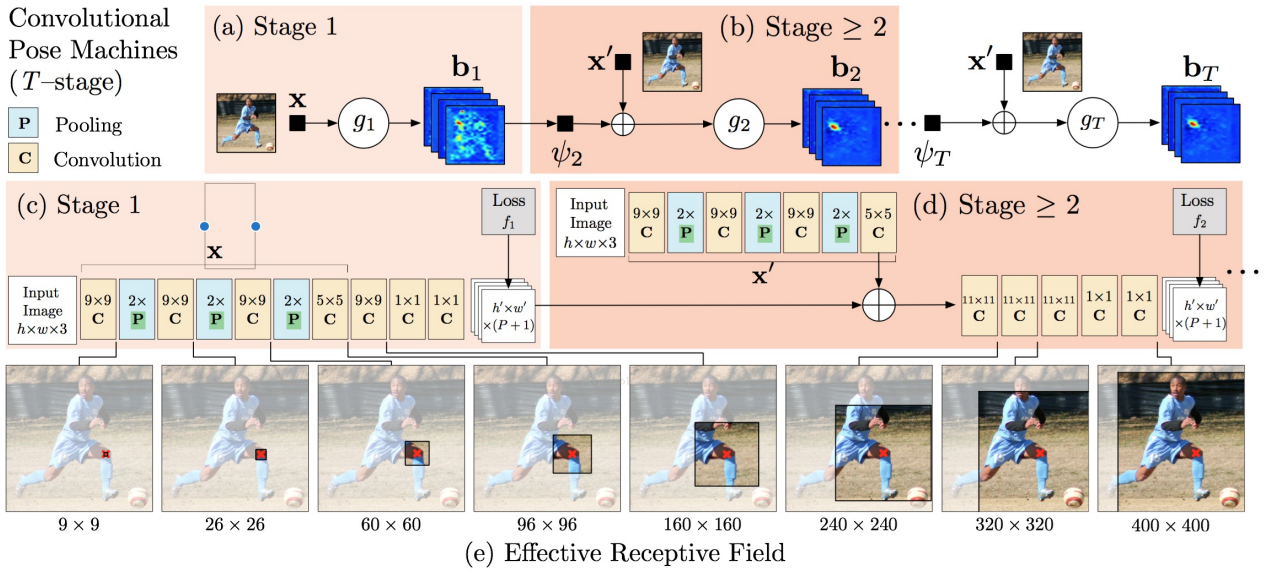
几款模型目前来看的精度：CPM < DeeperCut < CMU OpenPose < AlphaPose

模型	大体框架	多人估计的应对措施
CPM	CPM是一个cascaded网络，多个stage反复去定位、修正响应图	center map，为一个高斯响应，因为cpm处理的是单人pose的问题，如果图片中有多人，那么center map可以告诉网络，目前要处理的那个人的位置。
DeeperCut	采用了自顶向下的方法针对多人进行姿态估计，所谓自顶向下的方法就是先使用CNN检测人体，即body part candidates，再判断这些关节点属于哪一个人。最后使用ILP优化模型进行姿态估计	DeepCut是利用自适应的Fast R-CNN来进行人体部分的检测
CMU OpenPose	两条线，经过卷积网络提取特征，得到一组特征图，然后分成两个岔路，分别使用 CNN网络提取Part Confidence Maps 和 Part Affinity Fields ，得到这两个信息后，我们使用图论中的Bipartite Matching 将同一个人的关节点连接起来得到最终的结果	
AlphaPose	跟deeperCut比较像：对称空间变换网络（Symmetric STN）找人体位置； detection proposal用于训练姿态估计网络，从而获得大量符合真实测试场景数据分布的训练数据；第三个组成部件为参数化的姿态非极大值抑制器（Parametric Pose NMS），通过使用新的姿态距离度量来比较姿态相似性，来消除冗余姿态	

1、CPM

timctho/convolutional-pose-machines-tensorflow

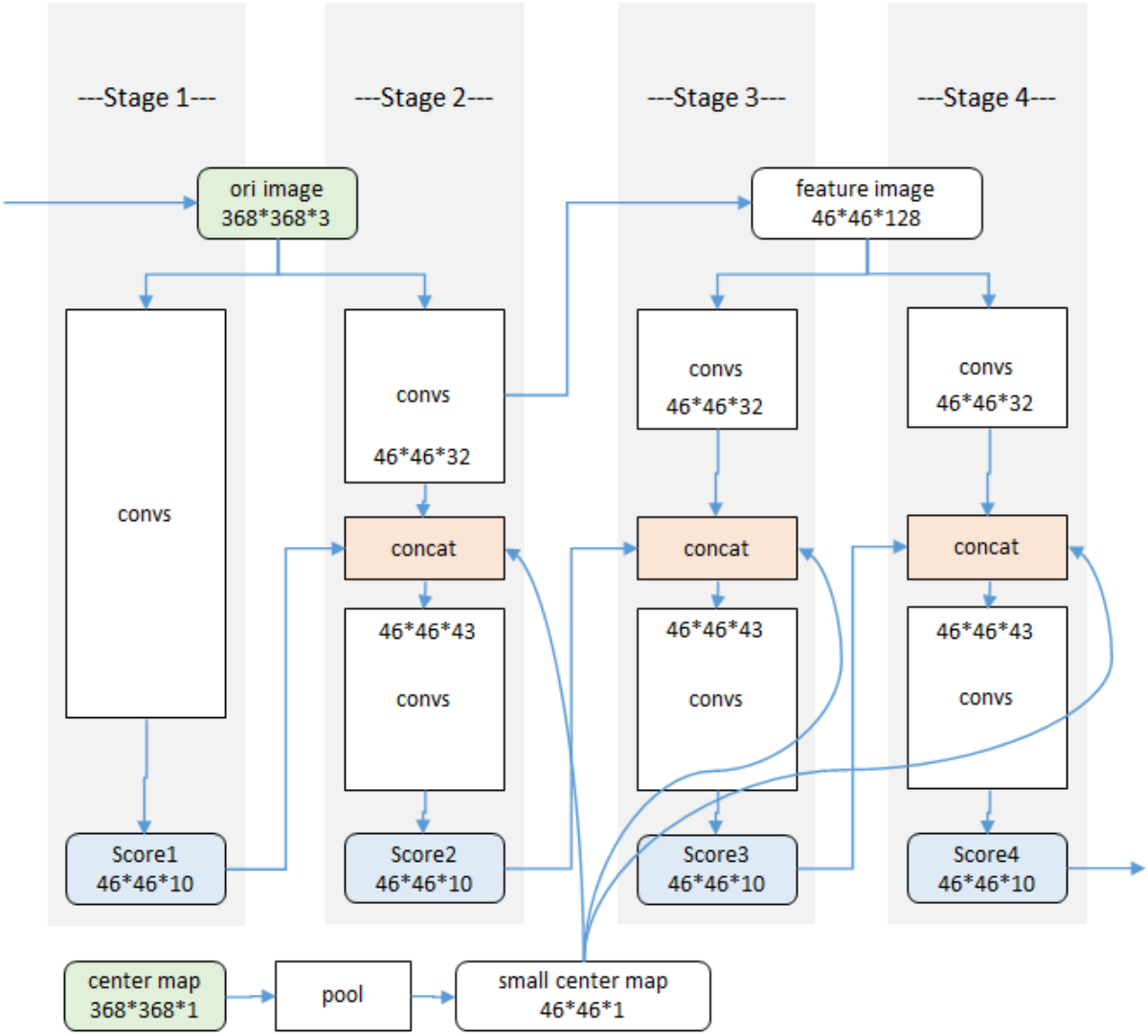
cpm是CMU开源项目OpenPose的前身，整个framework还是比较清晰的，Pose estimation任务属于FCN的一种，输入是一张人体姿势图，输出n张热力图，代表n个关节的响应。



(e) Effective Receptive Field

Figure 2: Architecture and receptive fields of CPMs. We show a convolutional architecture and receptive fields across layers for a CPM with any T stages. The pose machine [29] is shown in insets (a) and (b), and the corresponding convolutional networks are shown in insets (c) and (d). Insets (a) and (c) show the architecture that operates only on image evidence in the first stage. Insets (b) and (d) shows the architecture for subsequent stages, which operate both on image evidence as well as belief maps from preceding stages. The architectures in (b) and (d) are repeated for all subsequent stages (2 to T). The network is locally supervised after each stage using an intermediate loss layer that prevents vanishing gradients during training. Below in inset (e) we show the effective receptive field on an image (centered at left knee) of the architecture, where the large receptive field enables the model to capture long-range spatial dependencies such as those between head and knees. (Best viewed in color.)

（1）级联网络，CPM是一个cascaded网络，但是CPM是一个交互的sequence framework，即上一个FCN的上下文（contextual）会作为下一个FCN的输入。
（2）分阶段，每个阶段都能输出各个部件的响应图（蓝色score），使用时以最后一个阶段的响应图输出为准。



（3）为了多人预测的center map，为一个高斯响应，因为cpm处理的是单人pose的问题，如果图片中有多人，那么center map可以告诉网络，目前要处理的那个人的位置。因为这样的设置，cpm也可以自底向上地处理多人pose的问题。
参考：Convolutional Pose Machines、convolutional pose machines, CVPR 2016、【人体姿态】Convolutional Pose Machines

2、 CMU OpenPose (Part Affinity Fields)
github： ZheC/Realtime_Multi-Person_Pose_Estimation
ildoonet/tf-pose-estimation 一些调整：mattzheng/tf-pose-estimation-applied

用Part Affinity Fields来学习如何将身体关键和个人匹配起来。主要思想是利用贪心算法自下而上的解析步骤从而达到高准确率和实时性。身体部位定位和关联是在两个分支上同时进行的。该方法获得了COCO2016 keypoints challenge中夺得第一名。
文章代码总共分为两条线，经过卷积网络提取特征，得到一组特征图，然后分成两个岔路，分别使用 CNN网络提取Part Confidence Maps 和 Part Affinity Fields ，得到这两个

信息后，我们使用图论中的 Bipartite Matching 将同一个人的关节点连接起来得到最终的结果。

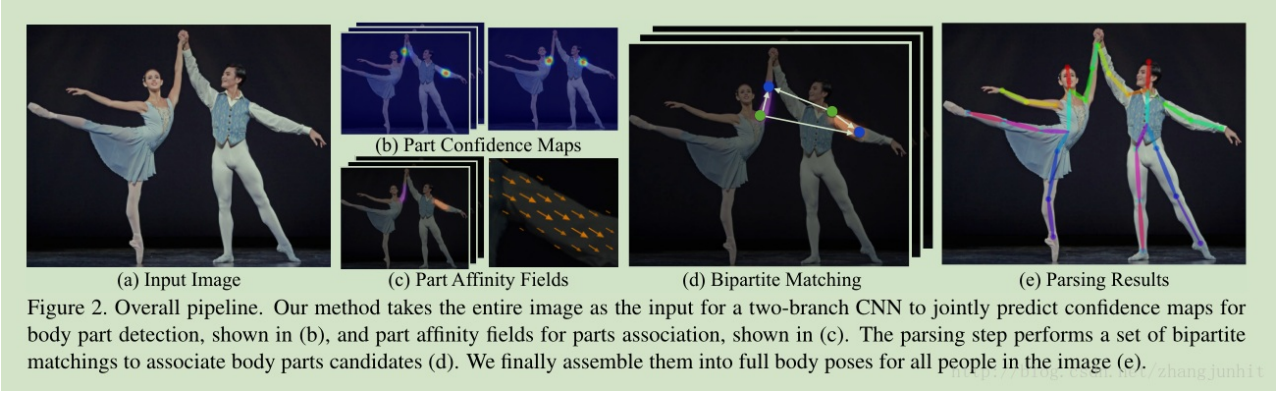


Figure 2. Overall pipeline. Our method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, shown in (b), and part affinity fields for parts association, shown in (c). The parsing step performs a set of bipartite matchings to associate body parts candidates (d). We finally assemble them into full body poses for all people in the image (e).

第一条线：求所有的关键点（头，肩膀，手肘，手腕 ... ）

- 1）一共两个cnn，第一个cnn的输入是原图，输出是热图（每一个热图包含某一种关键点）
- 2）第二个cnn输入是上一个cnn得到的所有热图，和原图。输出还是热图。

循环直至收敛

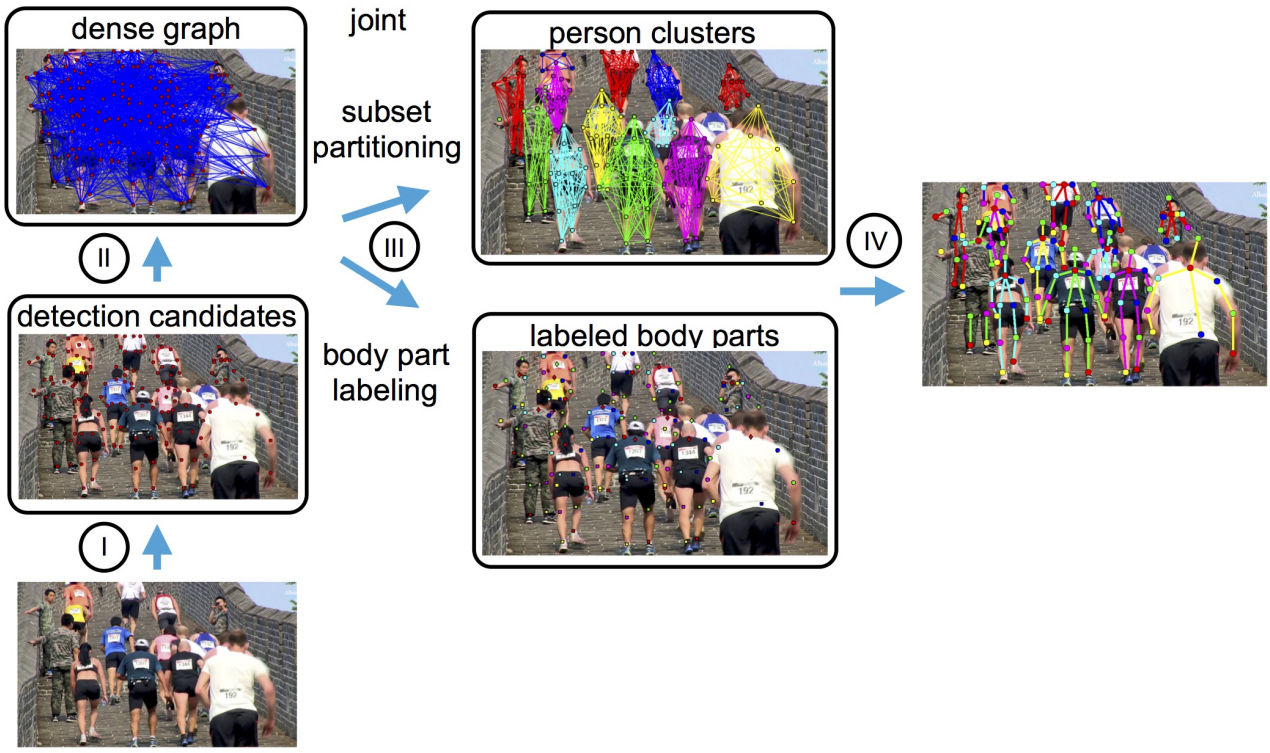
第二条线：求所有关节区域
1）一共两个cnn，第一个cnn的输入是原图，输出是热图（每一个热图包含某一种连接（可以简单理解为骨头）区域），其实它们是一整片区域，不过每个地方的概率大小不同。
2）第二个cnn输入是上一个cnn得到的所有热图，和原图。输出还是热图。
循环直至收敛

根据前边两个阶段得到的两个热图，计算哪两个点连接比较好。这就要根据关节区域和点的位置来计算每个像素的小法向。生成一个法向图。

参考：
行人姿态估计–Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields、
姿态估计论文思路整理 – Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields、
基于部分亲和字段PAF(Part Affinity Field)的2D图像姿态估计、
Paper reading: Realtime Multi-person 2D Pose estimation using Part Affinity Fields(1)

3、DeeperCut

github：eldar/pose-tensorflow 一些调整：mattzheng/pose-tensorflow-detailed



首先使用CNN提取body part candidates，每一个候选区域对应的是一个关节点， 每一个关节点作为图中的一个节点，所有的这些候选关节点组成代表的节点组成一副完整的图，正如上图dense graph所示。节点之间的关联性作为图中的节点之间的权重。这时，可以将其看作是一个优化问题，将属于同一个人的关节点（图中的节点）归为一类，每一个人作为一个单独的类。同时，另一条分支，需要对检测出来的节点进行标记，确定他们属于人体的哪一个部分。最后，使用分类的人结合标记的部分构成最终的每个人的姿态估计。
前一个版本DeepCut：
具有以下几个优势：

1）可以解决未知个数人的图像，通过归类得到有多少个人
2）通过图论节点的聚类，有效的进行了非极大值抑制
3）优化问题表示为 Integer Linear Program (ILP)，可以有效求解

不过同样地，因为使用了自适应的Fast R-CNN来进行人体部分的检测，又使用ILP来进行多人的人体姿态估计，导致DeepCut的计算复杂度比较高。

DeeperCut 改进了：

（1）采用了Resnet来提高body part的检测，更加的有效，精度更高；
（2）使用了image-conditioned pairwise
terms可以将得到足够丰富的候选区域节点压缩到一定数量的节点，而这也是整个论文的核心部分，也是stronger &
faster的主要原因。如下图所示，即通过候选区域节点之间的距离来判断是否为不同的重要关节点。

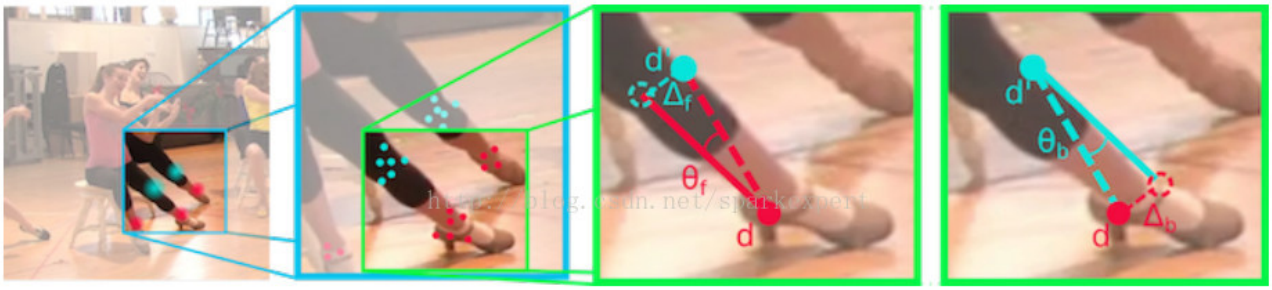


Fig. 3. Visualization of features extracted to score the pairwise. See text for details.

参考：
DeepCut及DeeperCut：基于Tensorflow的人体姿态估计
论文阅读：Deepcut&Deepercut:Joint Subset Partition and Labeling for Multi Person Pose Estimation

4、AlphaPose
——来源：姿态估计相比Mask-RCNN提高8.2%，上海交大卢策吾团队开源AlphaPose
上海交通大学卢策吾团队，今日开源AlphaPose系统。该系统在姿态估计（pose estimation）的标准测试集COCO上较现有最好姿态估计开源系统Mask-RCNN相对提高8.2%，较另一个常用开源系统OpenPose（CMU）相对提高17%。同时，卢策吾团队也开源了两个基于AlphaPose的工作：

（1）一个高效率的视频姿态跟踪器（pose tracker），目前姿态跟踪准确率第一。
（2）一个新的应用“视觉副词识别“（Visual Adverb Recognition）。
开源github:https://github.com/MVIG-SJTU/AlphaPose
训练框架：torch
预测框架：tf+torch

交大MVIG组提出RMPE的两步法框架（ICCV 2017论文），并基于此开发了AlphaPose这一人体关键点检测系统。

RMPE框架采用自顶向下的方法，先检测人，再去做姿态估计。该框架有三个主要组成部分：

首先是对称空间变换网络（Symmetric STN），用于解决传统两步法中的主要问题，即imperfect proposal的问题。对于质量较差的人体检测结果，symmetric STN能够自动调整proposal的位置，将refine过后的结果输入单人姿态估计网络，并将输出映射回原空间，从而使得在人体检测框不准确的情况下，姿态估计网络依然能够有良好的效果。

第二个组成部件为由姿态引导的样本生成器（Pose-guided Proposals Generator），该部件能够根据不同人体姿态生成额外的detection proposal用于训练姿态估计网络，从而获得大量符合真实测试场景数据分布的训练数据。

第三个组成部件为参数化的姿态非极大值抑制器（Parametric Pose NMS）。传统的两步法中，人体定位框会有较多的冗余检测。作者通过使用新的姿态距离度量来比较姿态相

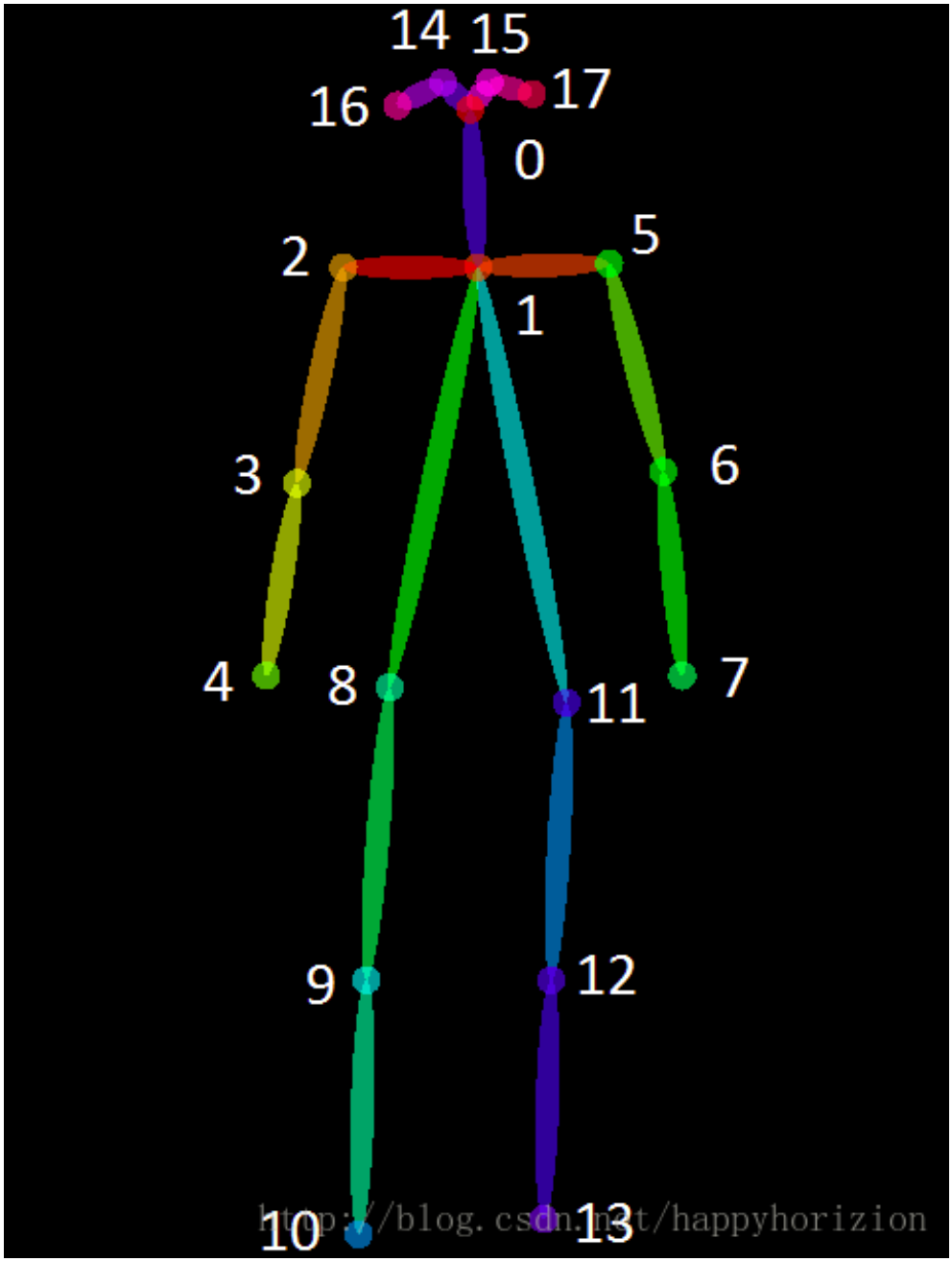
似性，来消除冗余姿态。

拓展：视觉副词识别（Visual Adverb Recognition）
利用了姿势（pose）信息的，使用表情信息， RGB 和光流信息。同时，他们构建了对应的数据集：ADHA，这一数据集标注了视频中人物的位置、动作和可以描述这一动作的副词，我们还为数据用户提供了人物的 tracking 结果。
主页（包括代码）：http://mvig.sjtu.edu.cn/research/adha.html
数据：http://mvig.sjtu.edu.cn/research/adha/adha.html

二、几款相关比赛：

Multi-person

（1）MSCOCO



（2） AI challenger

Single person

MPII Pose Dataset

三、如何训练
笔者这边用的是：mattzheng/pose-tensorflow-detailed，还算好，原作中写了用自己数据做训练的方式，
但是，mattzheng/tf-pose-estimation-applied 这个方法，原作用的是coco，封装的太好。
mattzheng/convolutional-pose-machines-tensorflow，别人用过，貌似也不错的样子。

这大兄弟在玩 AI challenger人体骨骼关节点赛题的时候，同样自己训练并开源出来。
但是，该比赛的关键点只有：14个（参考：赛题与数据），该作者在生成时候（ai2coco_art_neckhead_json.py），拼凑成17个点，与coco一致，然后就可以完全使用coco框架训练（多人模式），同时共享pairwise stat。
该作者在比赛数据上当时迭代了60W次，最终的得分为:0.36，而原来的coco数据集，多人关键点定位需要180W次。

笔者自己fork的几个项目，只玩了两个：

mattzheng/tf-pose-estimation-applied
mattzheng/AlphaPose
mattzheng/pose-tensorflow-detailed
mattzheng/Realtime_Multi-Person_Pose_Estimation
mattzheng/convolutional-pose-machines-tensorflow