

Pose Estimation

✓ 什么是姿态估计呢?

✎ 得到人体各个关键点位置

✎ 将它们按顺序进行拼接

✎ 这其中的难点是什么?

✎ OpenPose如何做的呢?



Pose Estimation

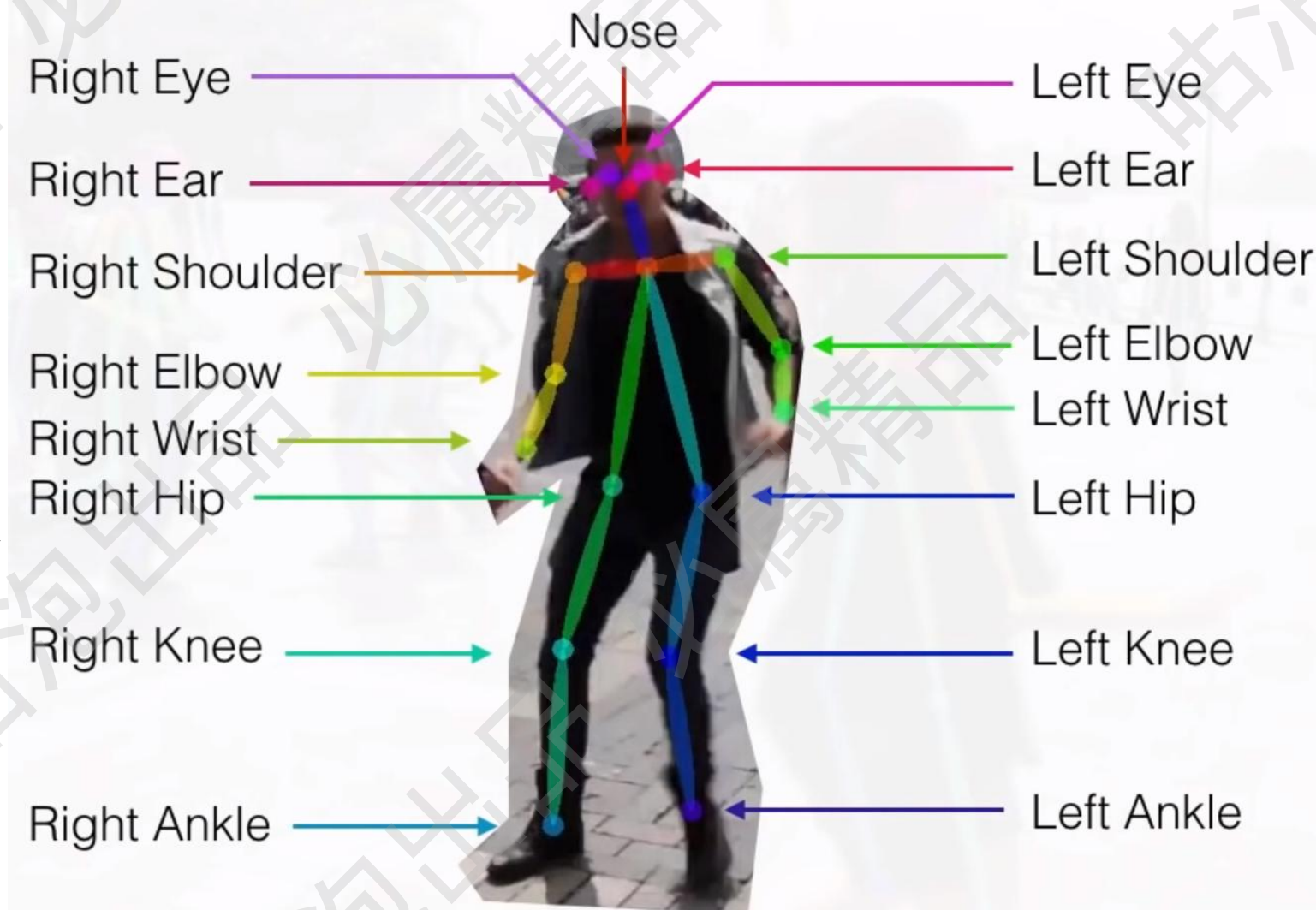
✓ 标注信息

✎ 17个关键点

✎ 脖子可以用肩膀平均值

✎ COCO数据集提供17个

✎ 咱们任务 $17+1=18$ 个



Pose Estimation

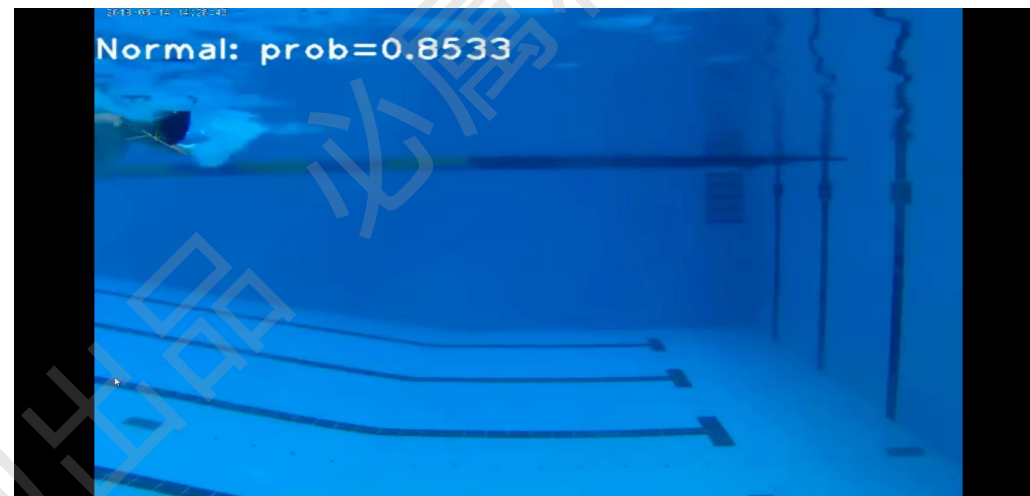
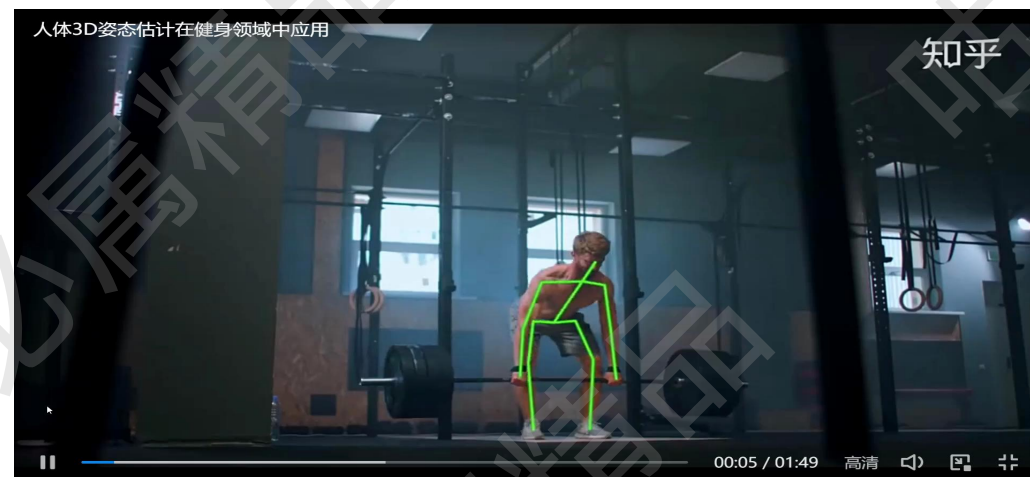
✓ 应用领域

✎ 各种特效肯定离不开关键点信息

✎ 行为特征分析，需各位置信息

✎ 异常检测，突变现象

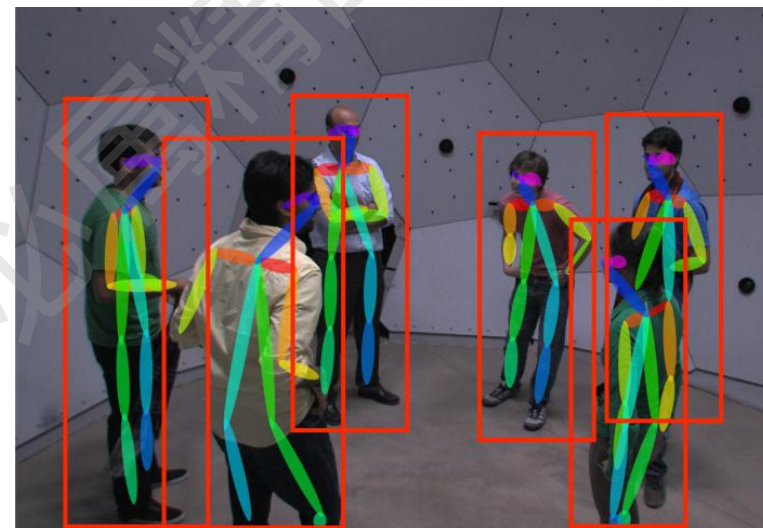
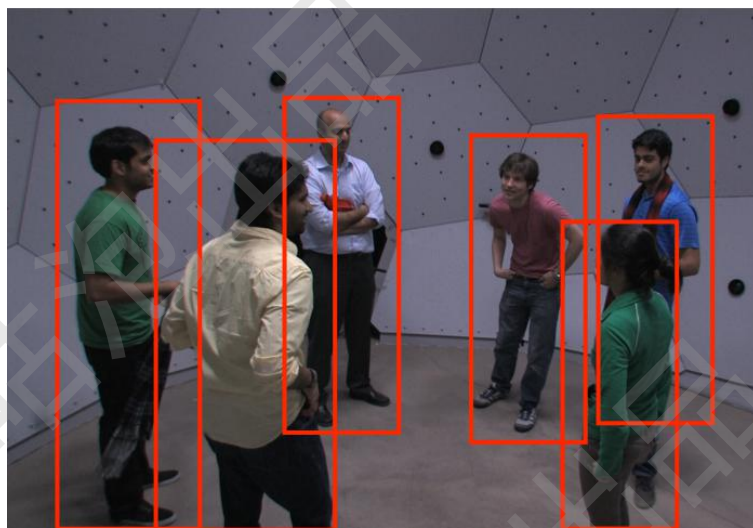
✎ 各类配准分析任务都需要它作为基础



Pose Estimation

✓ Top-down方法两步走

✎ 1.检测得到所有人的框; 2.对每一个框进行姿态估计输出结果



Pose Estimation

✓ Top-down方法的问题

- ✎ 姿态估计做成啥样主要由人体检测所决定，能检测到效果估计也没问题
- ✎ 但是如果俩人出现重叠，只检测到一个人，那肯定会丢失一个目标
- ✎ 计算效率有点低，如果一张图像中存在很多人，那姿态估计得相当慢了
- ✎ 能不能设计一种方法不依赖于人体框而是直接进行预测呢？

Pose Estimation

✓ 一个恐怖的例子

✎ 如何得到姿态估计结果呢?分几步走?

✎ 1.首先得到所有关键点的位置

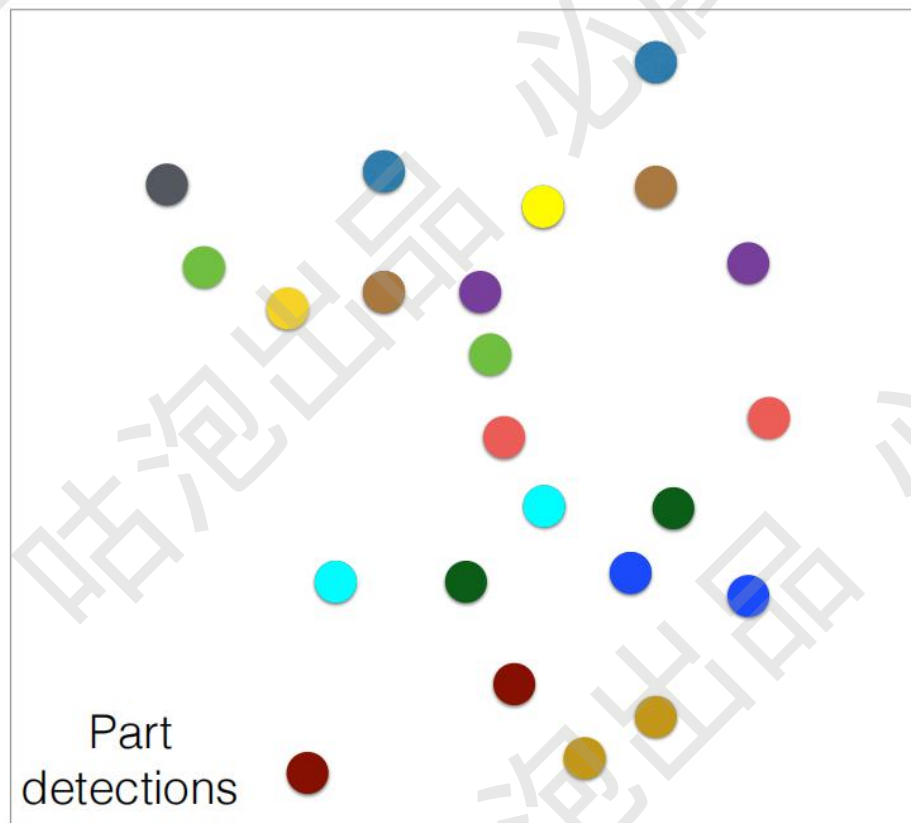
✎ 2.图中有多个人, 我们需要把属于同一个人的拼接到一起



Pose Estimation

✓ bottom-up挑战任务（首先得到所有关键点再拼接）

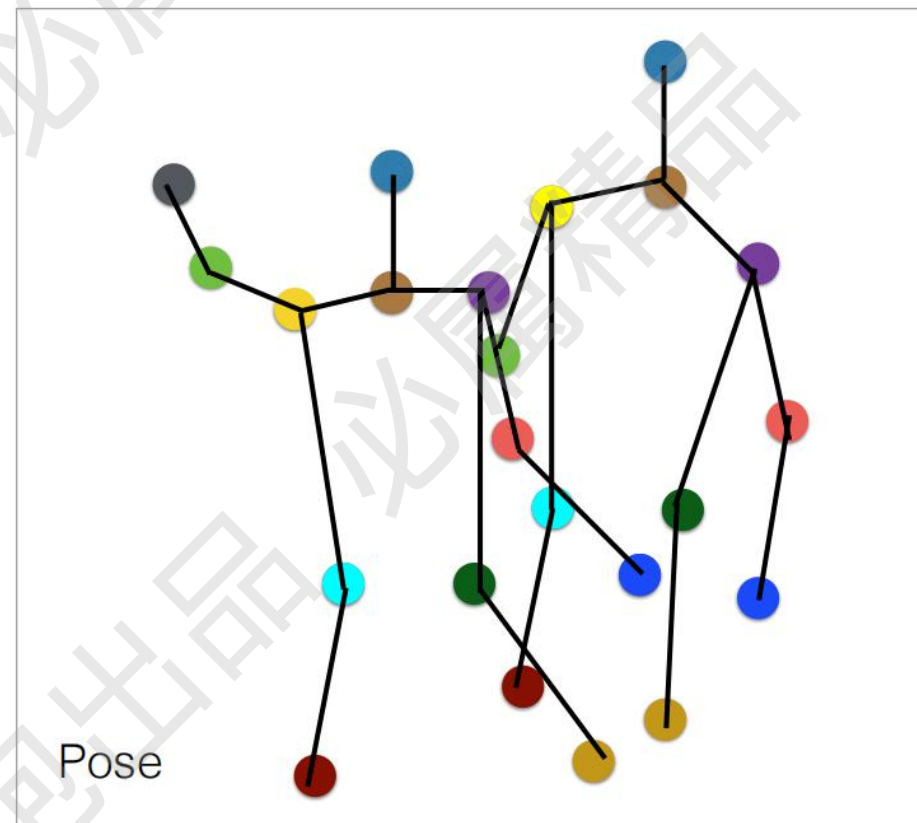
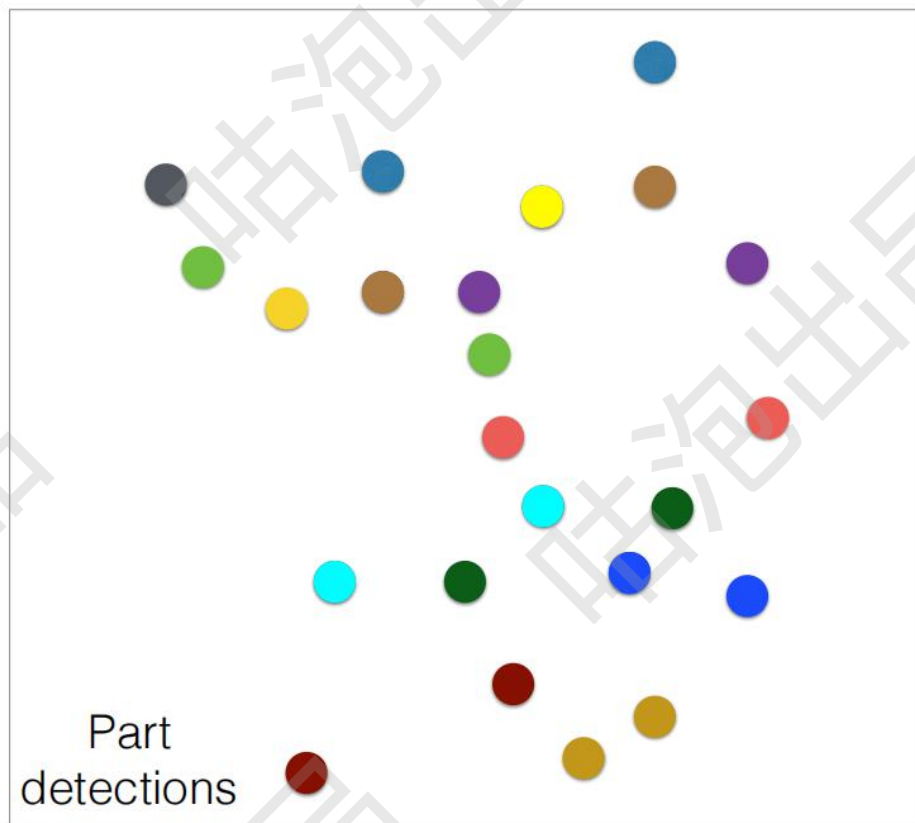
✎ 现在只检测到了所有关键点位置，你能把他们拼起来嘛？



Pose Estimation

✓ 挑战任务

✎ 现在只检测到了所有关键点位置，你能把他们拼起来嘛？



Pose Estimation

✓ Multi-Person Pose Estimation using Part Affinity Fields

📎 如果得到关键点位置：通过热度图（高斯）得到每一个关键点的预测结果



Pose Estimation

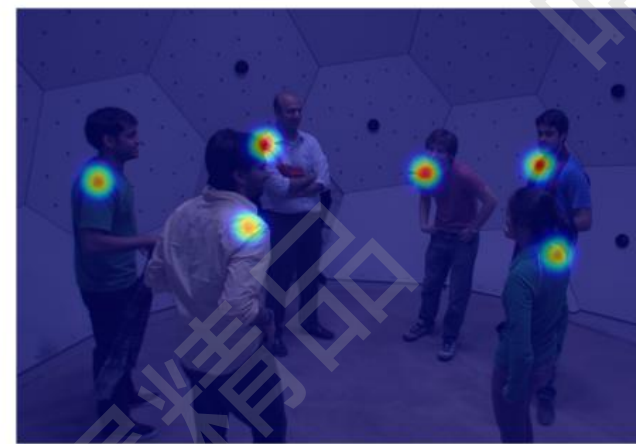
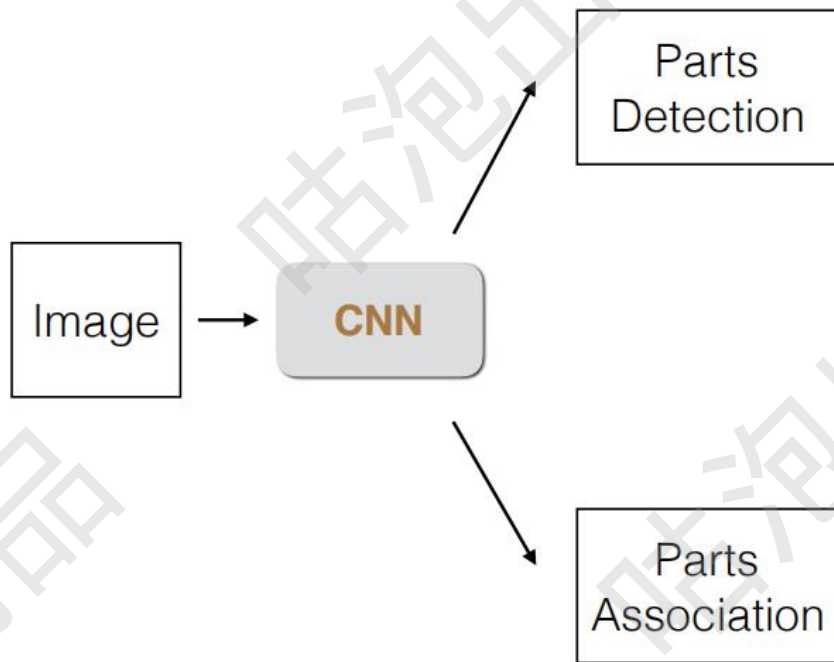
✓ Multi-Person Pose Estimation using Part Affinity Fields

✎ 将属于同一个人的不同关键点按顺序拼接



Pose Estimation

✓ 模型要完成的任务



Pose Estimation

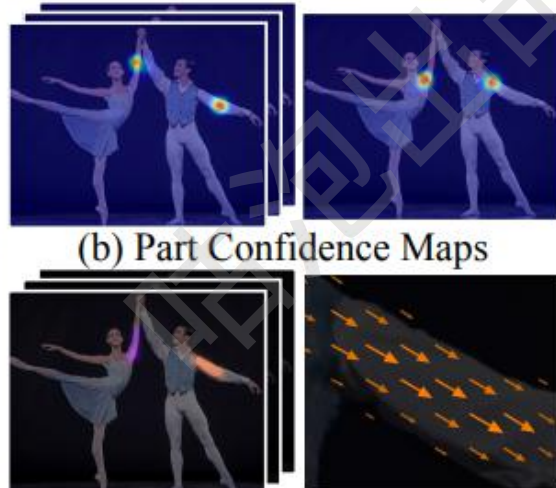
✓ 任务流程

✎ 分别经过两个分支得到各自预测结果

✎ 基于各分支结果组合成骨架



(a) Input Image



(b) Part Confidence Maps

(c) Part Affinity Fields



(d) Bipartite Matching



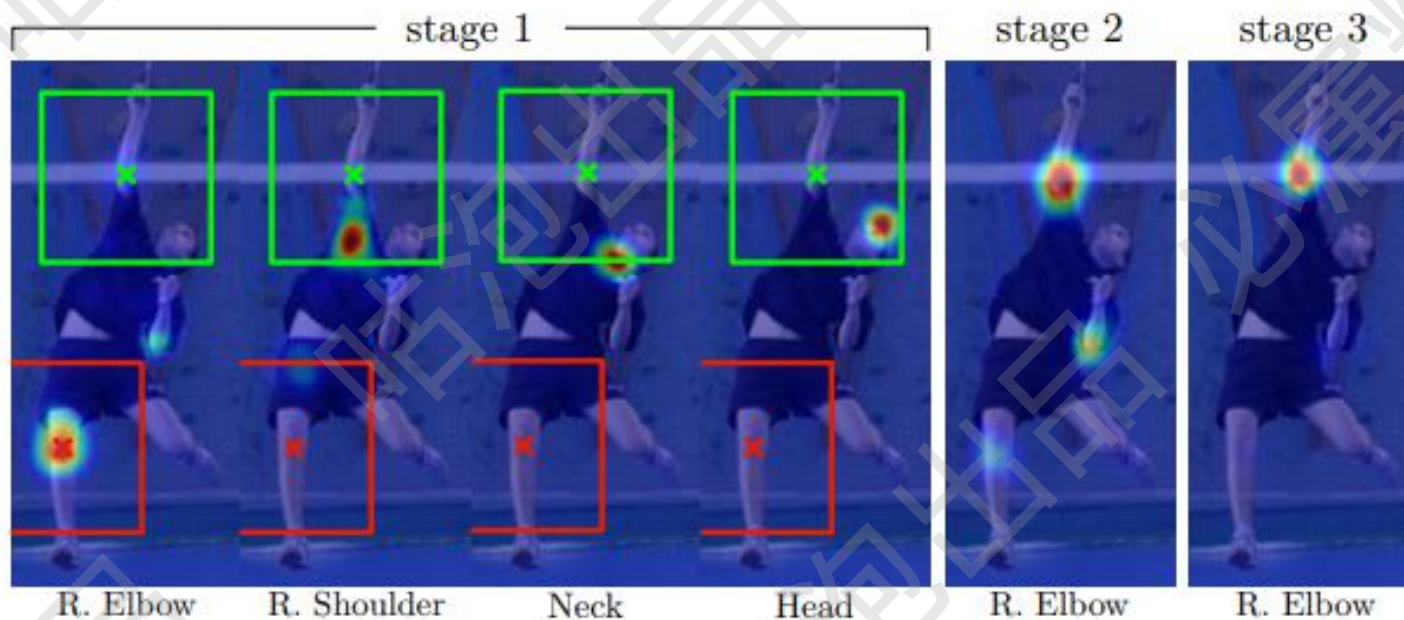
(e) Parsing Results

Pose Estimation

✓ Convolutional Pose Machines

✎ 为OpenPose后面的工作奠定了基础，也可以当作基础框架

✎ 通过多个stage来不断优化关键点位置（stage1预测完全错误，2和3在纠正）

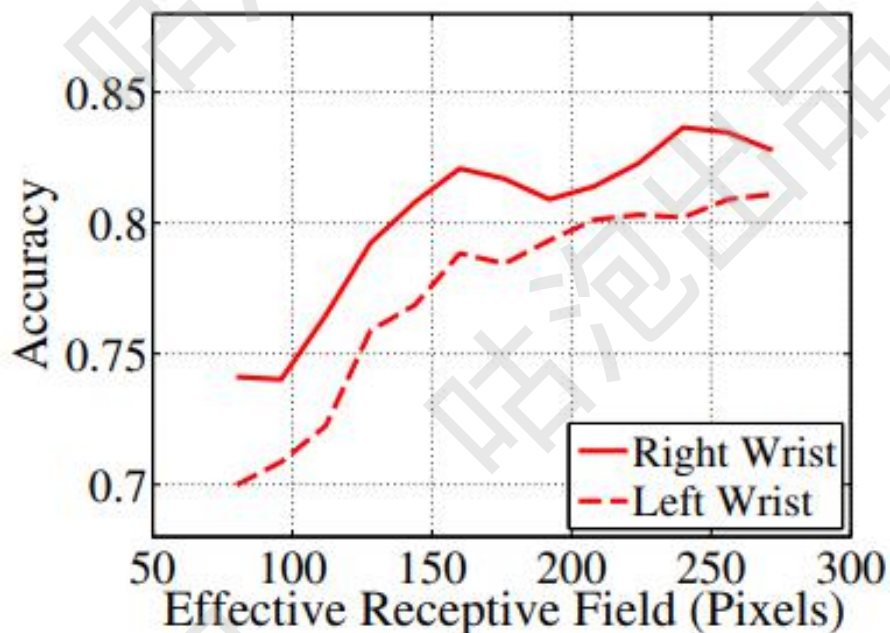


Pose Estimation

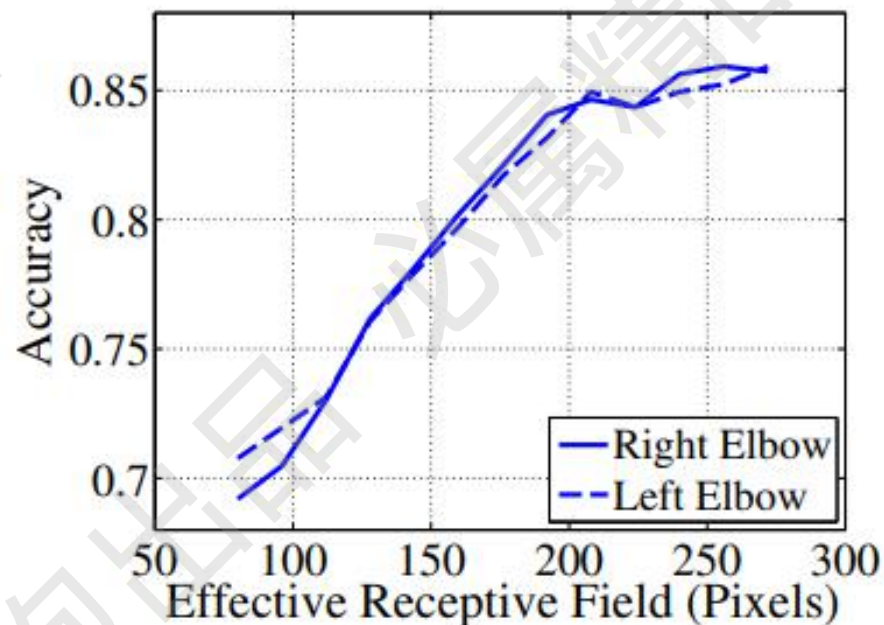
✓ Convolutional Pose Machines

✎ stage越多相当于层数越深，模型感受野越大，姿态估计需要更大的感受野

FLIC Wrists: Effect of Receptive Field



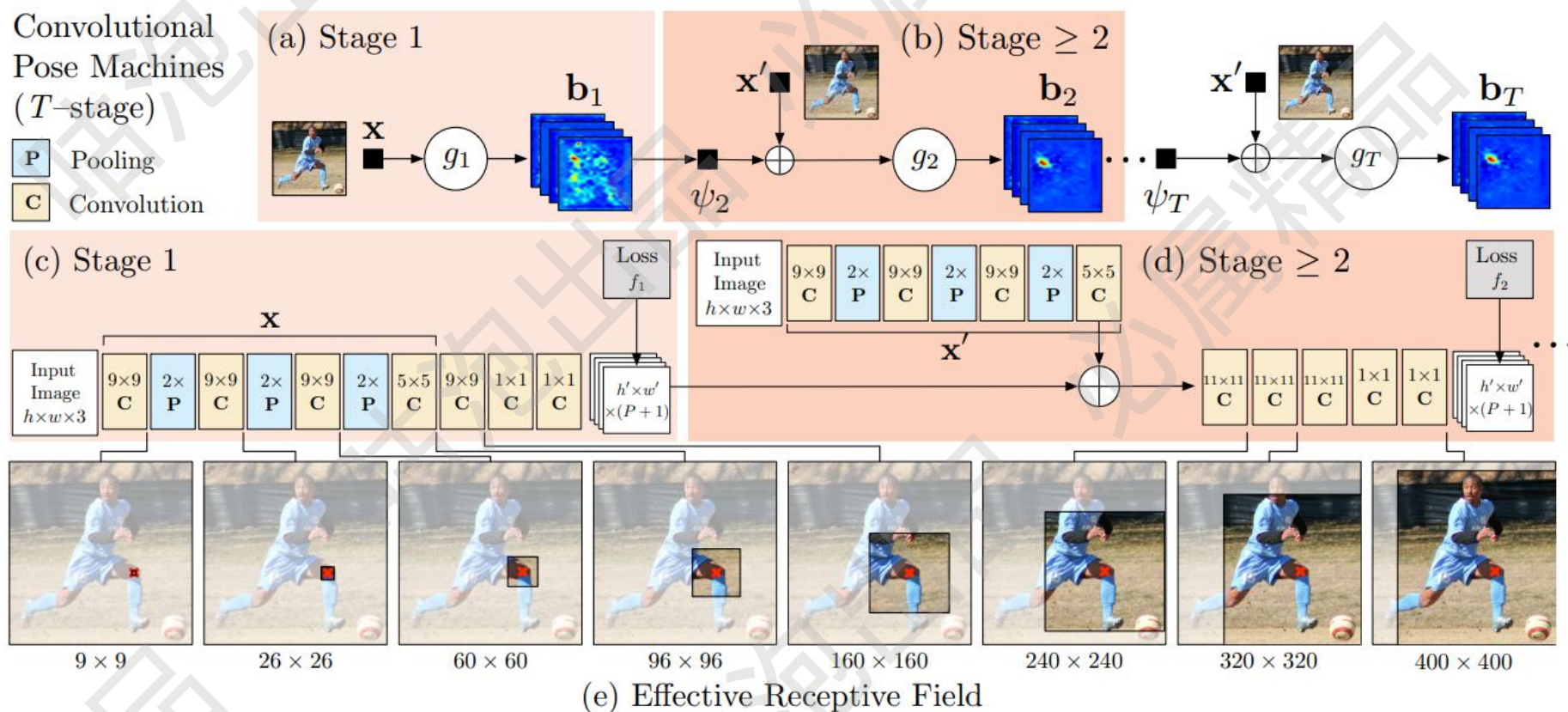
FLIC Elbows: Effect of Receptive Field



Pose Estimation

✓ Convolutional Pose Machines

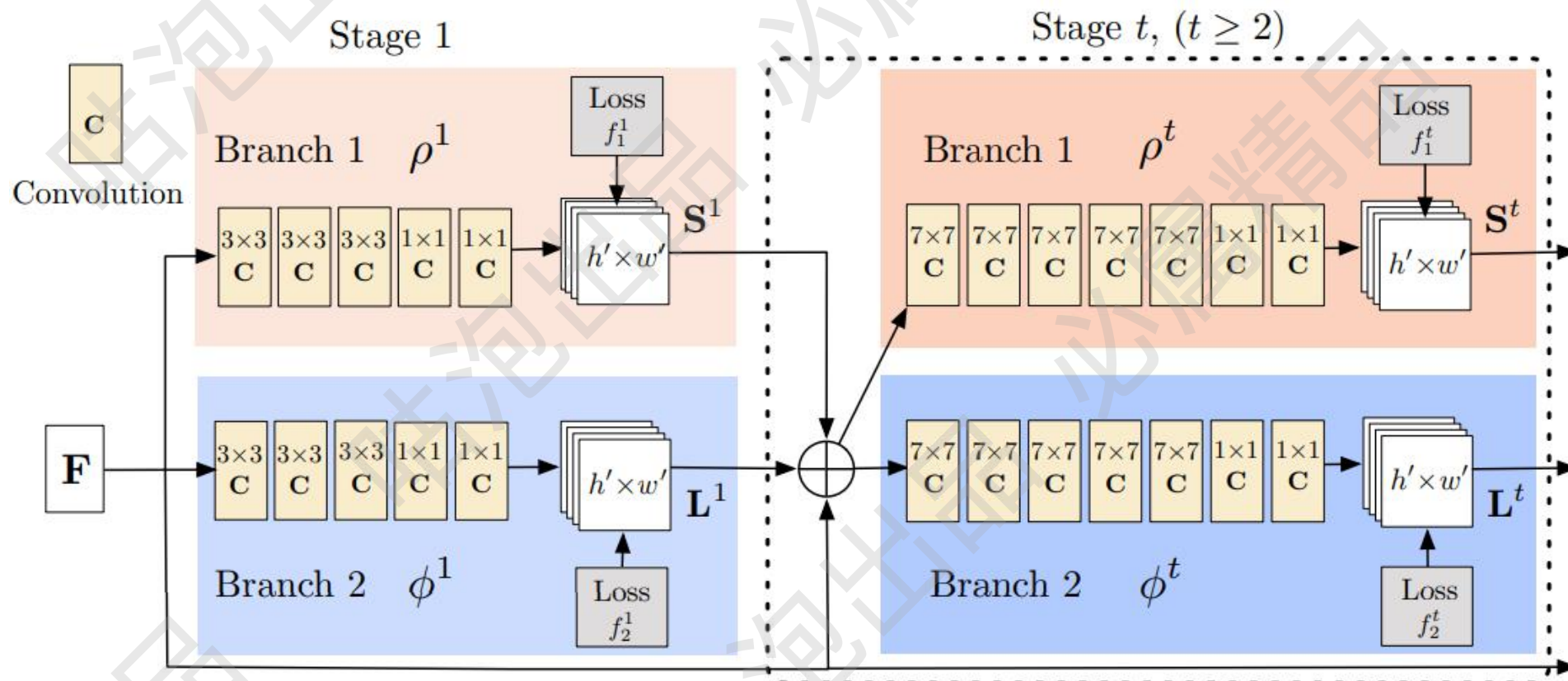
✎ 每个stage都加损失函数，也就是中间过程也得做的好才行



Pose Estimation

✓ 基本思想

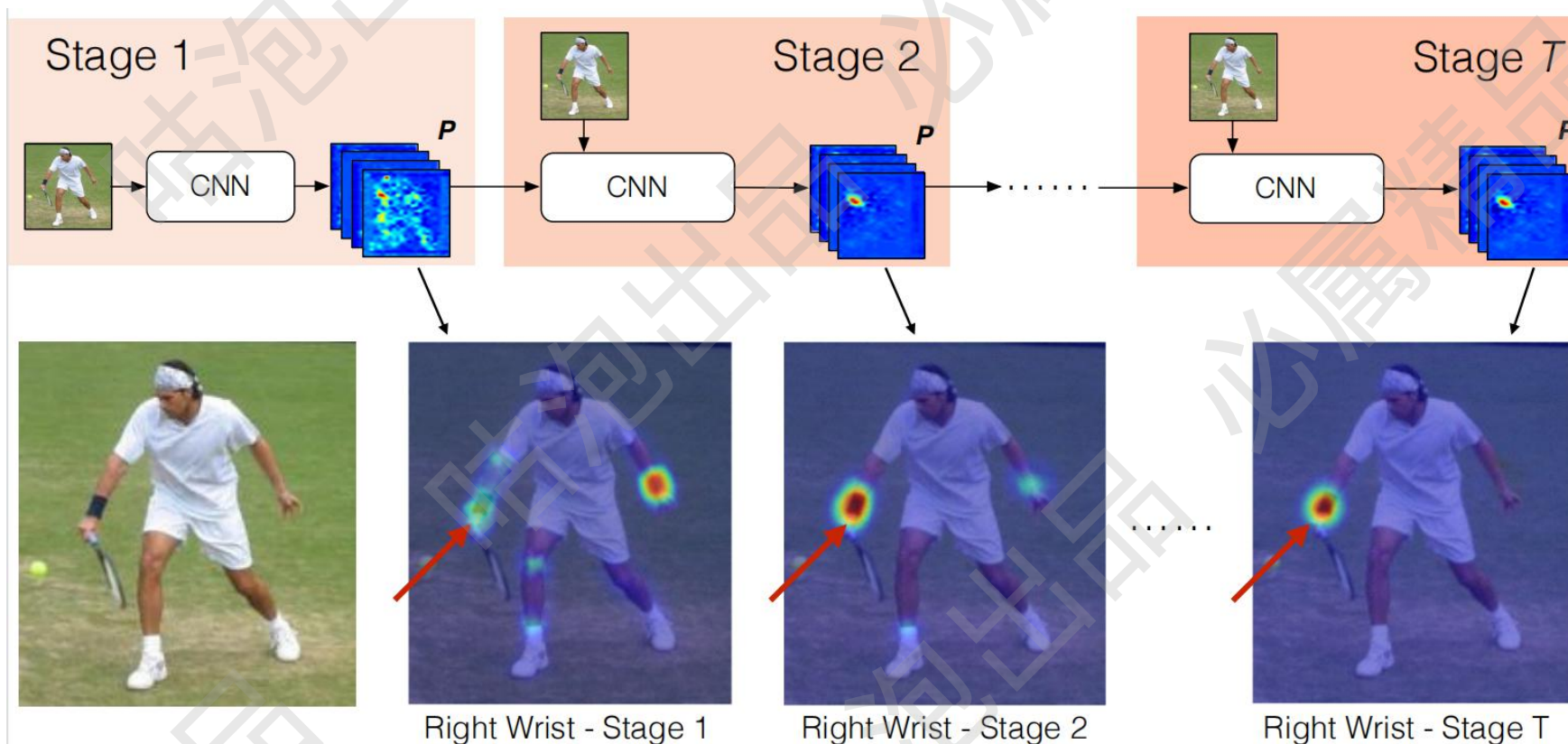
📎 两个网络结构分别搞定：1. 关键点预测；2. 姿势的 ‘亲和力’ 向量



Pose Estimation

✓ 序列的作用

✎ 多个stage, 相当于纠正的过程, 不断调整预测结果



Pose Estimation

✓ 关键点预测

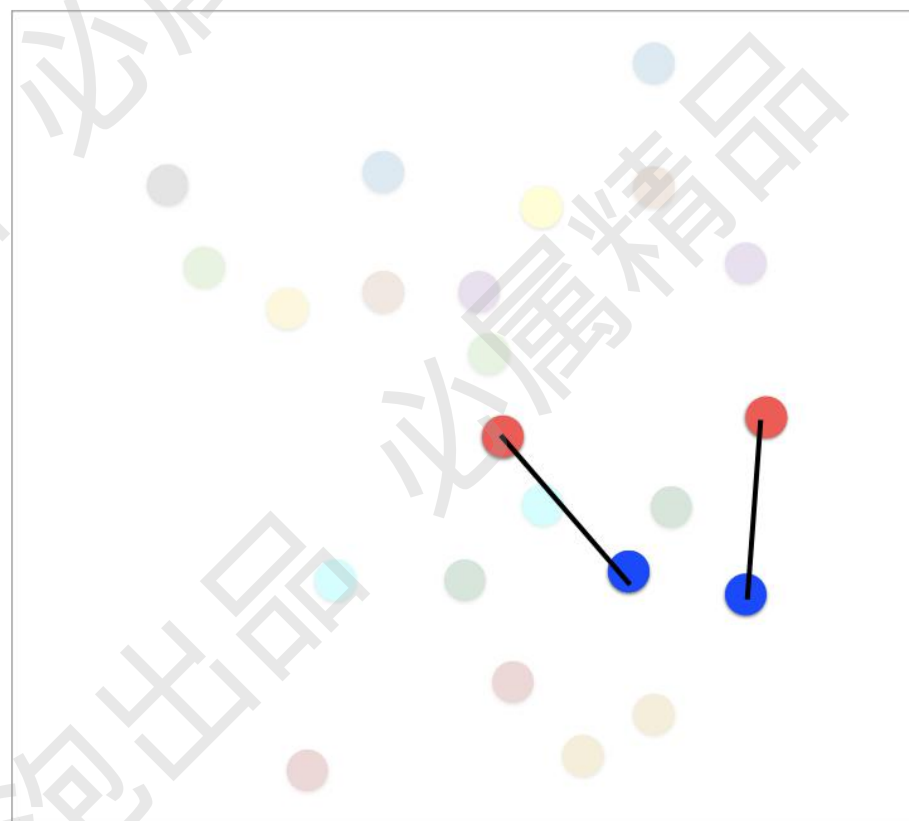
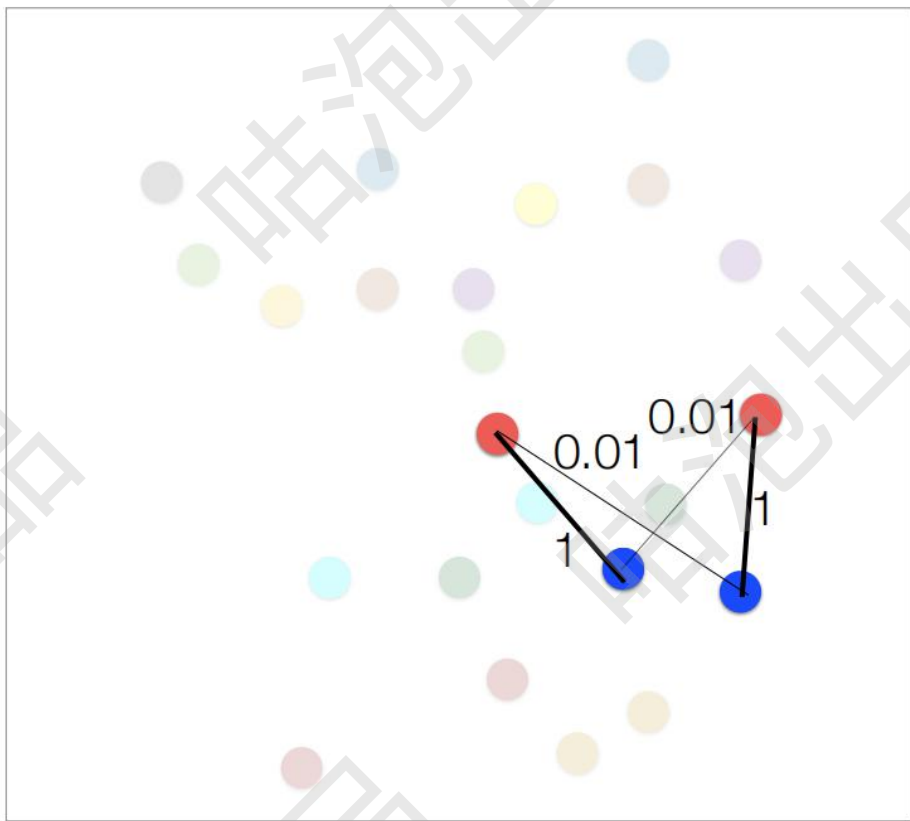
📎 一口气预测所有关键点的热度图



Pose Estimation

✓ 如何确定关键点的连接?

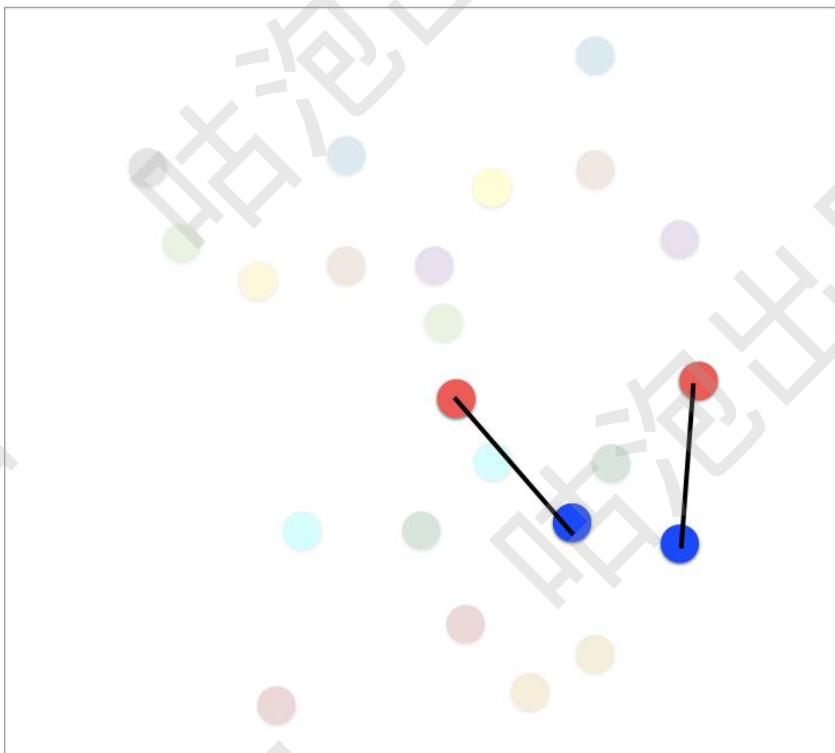
✎ 如果能给我一个权值 (分数, 代价矩阵) 这件事就能做了



Pose Estimation

✓ Part Affinity Fields

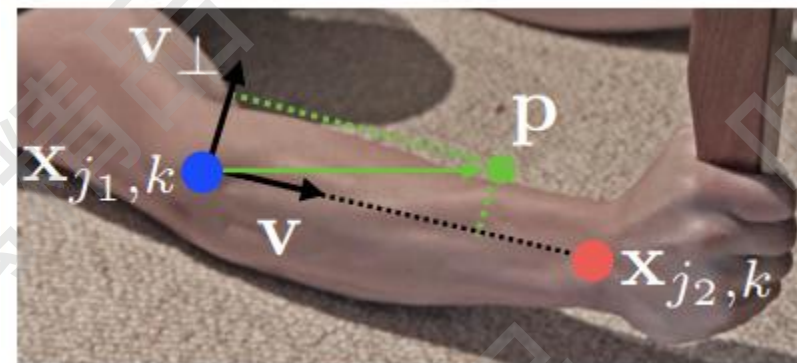
✎ 在标签中，我们还需要设计PAF来表示关键点连接向量（这是GT）



Pose Estimation

✓ PAF标签定义

✎ 蓝色和红色分别是两个关键点， V 是其向量



✎ V 要做成单位向量: $\mathbf{v} = (\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}) / \|\mathbf{x}_{j_2,k} - \mathbf{x}_{j_1,k}\|_2$

✎ 在手臂上任何一点的向量与 V 一致: $\mathbf{L}_{c,k}^*(\mathbf{p}) = \begin{cases} \mathbf{v} & \text{if } \mathbf{p} \text{ on limb } c, k \\ \mathbf{0} & \text{otherwise.} \end{cases}$

✎ P 点需满足条件（在矩形也就是胳膊上）论文中给定了距离阈值（感觉是经验值）

$$0 \leq \mathbf{v} \cdot (\mathbf{p} - \mathbf{x}_{j_1,k}) \leq l_{c,k} \quad \text{and} \quad |\mathbf{v}_\perp \cdot (\mathbf{p} - \mathbf{x}_{j_1,k})| \leq \sigma_l,$$

Pose Estimation

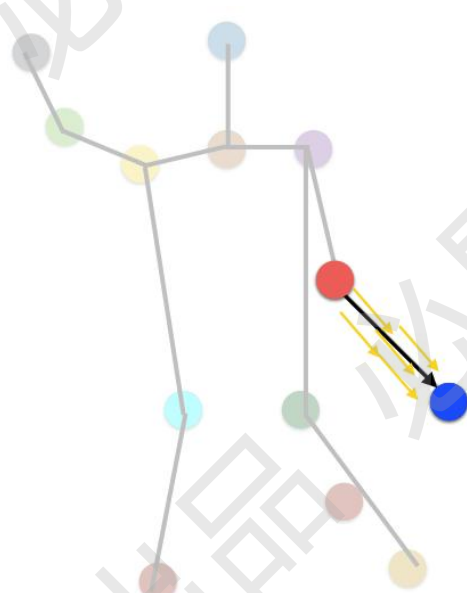
✓ PAF标签

✎ 在特征图上每一个固定连接区域都包括一组向量 (X方向, Y方向)

✎ 总共19组 ($19 \times 2 = 38$ 个特征图)

✎ 每种连接方式对应两个特征图

✎ 19组是先验 (例如右图的连接方式)



Pose Estimation

✓ PAF标签

✎ 对于其中某个PAF特征图（19种连接方式种的1种）

✎ 这就相当于得到的PAF标签值了，包括所有人该连接处的向量



Pose Estimation

✓ PAF标签

✎ 现在已知所有关键点位置（例如头和脖子）

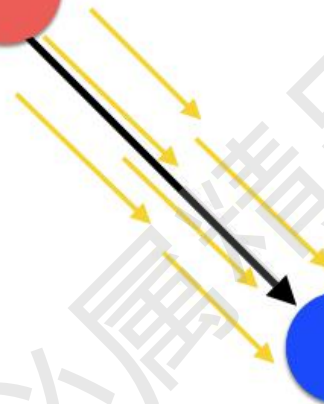
✎ 要看如何连接才能符合PAF向量趋势

✎ 可以这种符合程度当作得分

✎ 对得分结果进行匈牙利匹配

Part candidates 1

p_1



Part candidates 2

p_2



$$\begin{aligned} \text{Affinity score between } p_1 \text{ and } p_2 \\ = \text{sum}(\vec{v} \cdot p_1 \vec{p}_2) \end{aligned}$$

Pose Estimation

✓ PAF权值计算

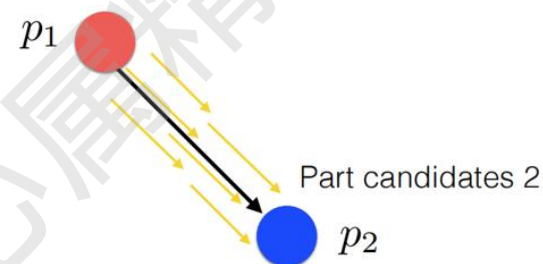
✎ 两个关键点j1与j2之间的权值计算方法

$$E = \int_{u=0}^{u=1} L_c(p(u)) \cdot \frac{d_{j_2} - d_{j_1}}{\|d_{j_2} - d_{j_1}\|_2}$$

✎ d_{j_1}, d_{j_2} 分别表示j1与j2两点的坐标

$$p(u) = (1 - u)d_{j_1} + ud_{j_2}$$

✎ 求j1和j2间各点的PAF在线段j1j2上投影的积分



✎ 其实就是线段上各点的PAF方向如果与线段的方向越接近权值就越大

Pose Estimation

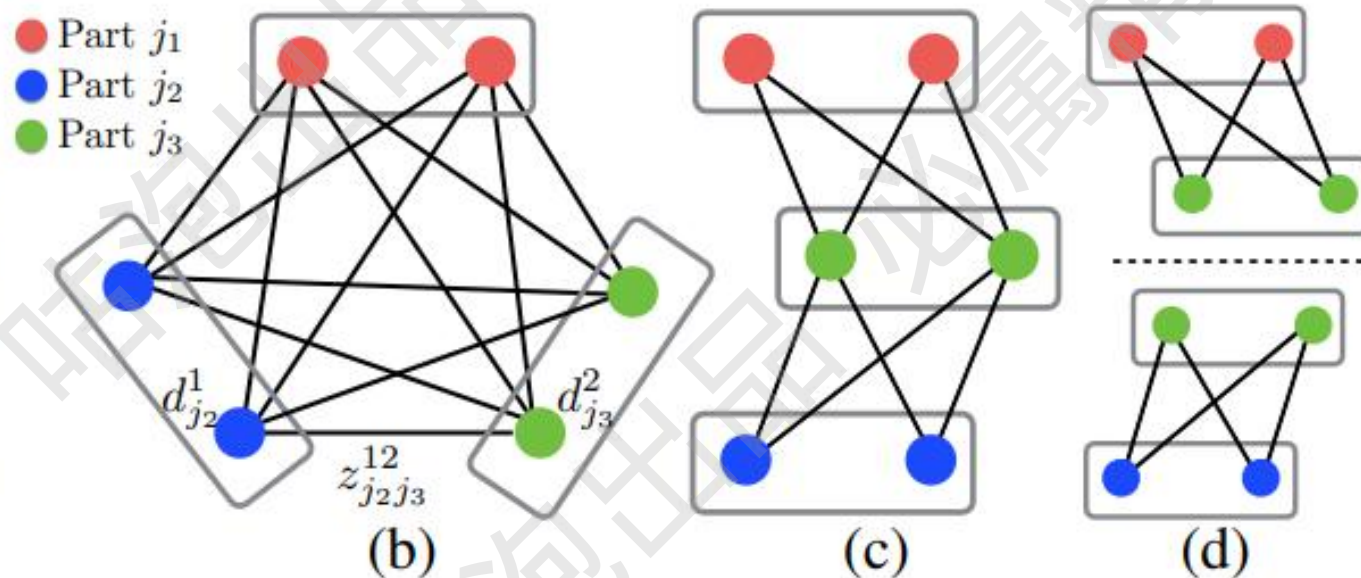
✓ 匹配方法

✎ 如果同时考虑多种匹配，那太难了

✎ 咱们固定好就是二分图，这样可以直接套匈牙利算法

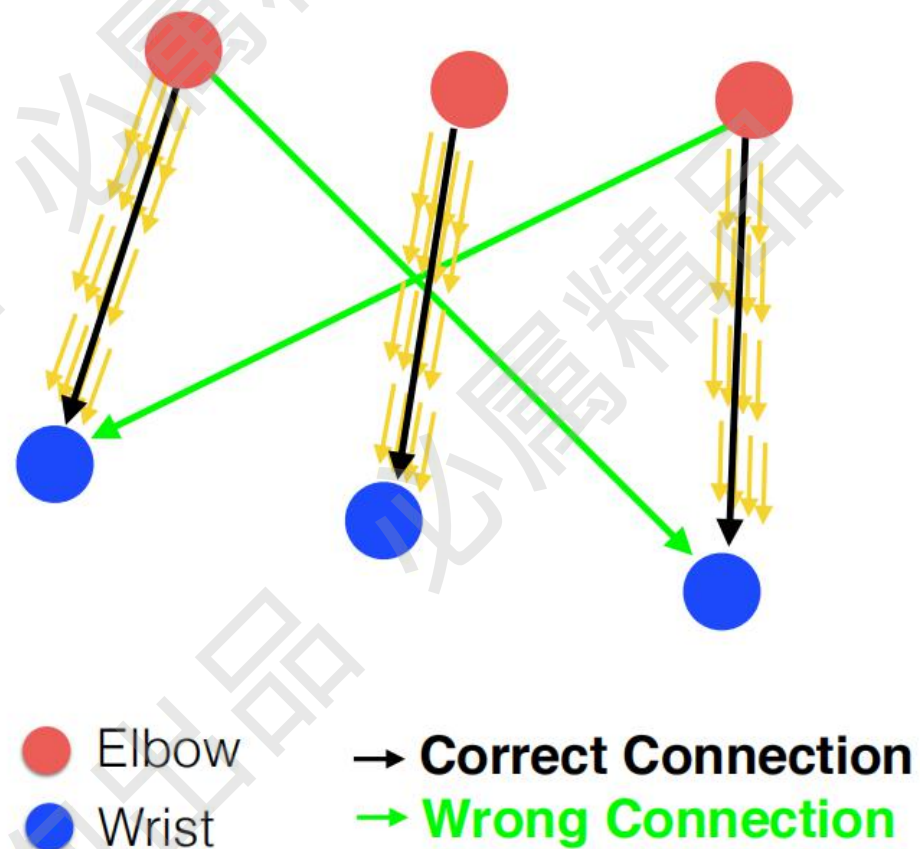


(a)



Pose Estimation

✓ 匹配结果



Pose Estimation

✓ 整体框架

📌 两个分支都要经过多个阶段，注意每个阶段后要把特征拼接一起

