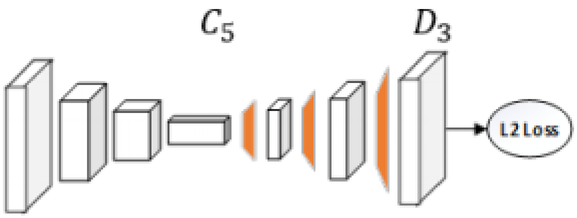
1.Introduction

aims to ease the problem of the model complexity。
反卷积层+ResNet

2. Deconvolution Head Network

Adds a few deconvolutional layers over the last convolution stage in the ResNet, called C_5
The network structure is illustrated below:



> Three deconvolutional layers with batch normalization and Relu activation are used, each layers has 256 filters with 4x4 kernel,stride is 2, a 1x1 convolutional layers is added at last to generate predicted headmaps.

obtaining high resolution feature maps is crucial.

3. Pose Tracking Based on Optial Flow

For Multi-Person Pose tracking in videos:

1. estimates human poses in frames
2. track these human pose by assigneing a unique identification number

greedy matching algorithm :

- assign the id of P_i^{k-1} in fram I^{k-1} to P_j^k in frame I^k if the similarity between P_i^{k-1} and P_j^k is the highest
- remove these two instances and repeat id assigning porcess with the highest similarity.

there are two difference between out implementation and above.

1. two kinds of human boxes, one is from a human detector, the other is generated from previous frames.
2. use a flow-based pose similarity metric.

3.1 Joint Propagation using Optical Flow

> 光流： 通过检测图像像素点的强度随时间变化进而推断出物体移动速度及方向的方法

Due to Motion blur or occlusoin, the single detector in optical flow may lead to missing detections and false detections. so **Add the processng frame from nearby frames using temporal information expressed in optical flow**.

3.2 Flow_based Pose Similarity

之前是使用IoU(Intersection-over-Union)作为相似性度量，这在实例运动很快没有重叠或者人群中的实例没什么关系时会有问题。对此有一个更精准的：

Pose similarity: calculates the body joints distacne between two intstances using Object Keypoint Similarity(OKS)

但是Pose similarity在不同帧上人的姿态改变时也会有问题，为此我们使用flow-based pose similarity.

$$S_{Flow}(J_i^k, J_j^l) = OKS(\hat{J}_i^l, J_i^l) \qquad (1)$$

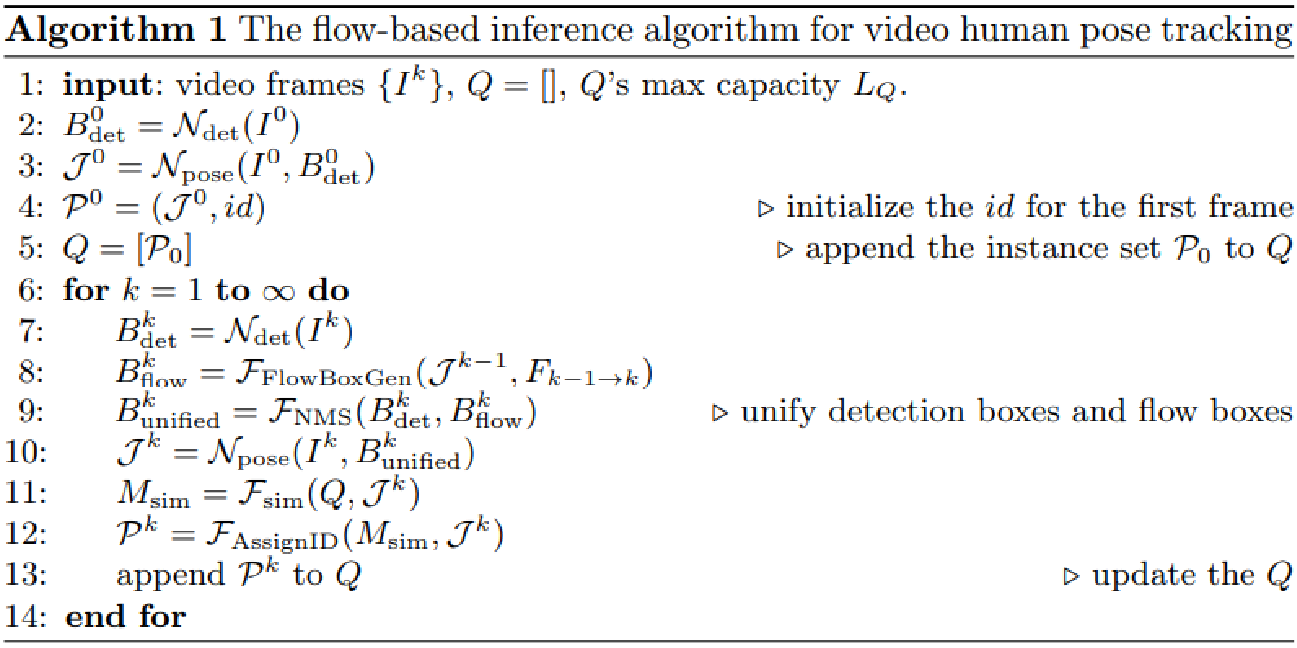
其中的关键在于$\hat{J}_i^l$, 代表着上一节中的propagated joints for J_i^k from frame I^k to I^l using optical flow feild $F_{k\to l}$.

在视频中，有时人还可能消失和重新出现，为此两个frame（帧）之间的flow不够用，我们可以利用之前的多个帧。

3.3 Flow-based Pose Tracking Algorithm

算法流程：

1. 解决姿态估计问题，在处理每一帧时，将之前用propagating joints 产生的box与human detector产生的box用Non-Maximum Suppression合起来，基于此做姿态估计。
2. 解决姿态追踪的问题，在第k帧时，我们计算未被追踪的身体关节实例J^k和之前帧存储的实例Q之间的flow-based pose similarity matrix M_{sim}, 然后对每个关节分配id， 得到实例P^k，将其加入Q中。



Algorithm 1 The flow-based inference algorithm for video human pose tracking

```
input: video frames $\{I^k\}$, $Q = []$, $Q$'s max capacity $L_Q$.
$B^0_{\text{det}} = \mathcal{N}_{\text{det}}(I^0)$
$\mathcal{J}^0 = \mathcal{N}_{\text{pose}}(I^0, B^0_{\text{det}})$
$\mathcal{P}^0 = (\mathcal{J}^0, id)$                          ▷ initialize the $id$ for the first frame
$Q = [\mathcal{P}_0]$                                 ▷ append the instance set $\mathcal{P}_0$ to $Q$
for $k = 1$ to $\infty$ do
    $B^k_{\text{det}} = \mathcal{N}_{\text{det}}(I^k)$
    $B^k_{\text{flow}} = \mathcal{F}_{\text{FlowBoxGen}}(\mathcal{J}^{k-1}, F_{k-1\to k})$
    $B^k_{\text{unified}} = \mathcal{F}_{\text{NMS}}(B^k_{\text{det}}, B^k_{\text{flow}})$    ▷ unify detection boxes and flow boxes
    $\mathcal{J}^k = \mathcal{N}_{\text{pose}}(I^k, B^k_{\text{unified}})$
    $M_{\text{sim}} = \mathcal{F}_{\text{sim}}(Q, \mathcal{J}^k)$
    $\mathcal{P}^k = \mathcal{F}_{\text{AssignID}}(M_{\text{sim}}, \mathcal{J}^k)$
    append $\mathcal{P}^k$ to $Q$                                 ▷ update the $Q$
end for
```