i-VisionGroup@Tsinghua

文献阅读

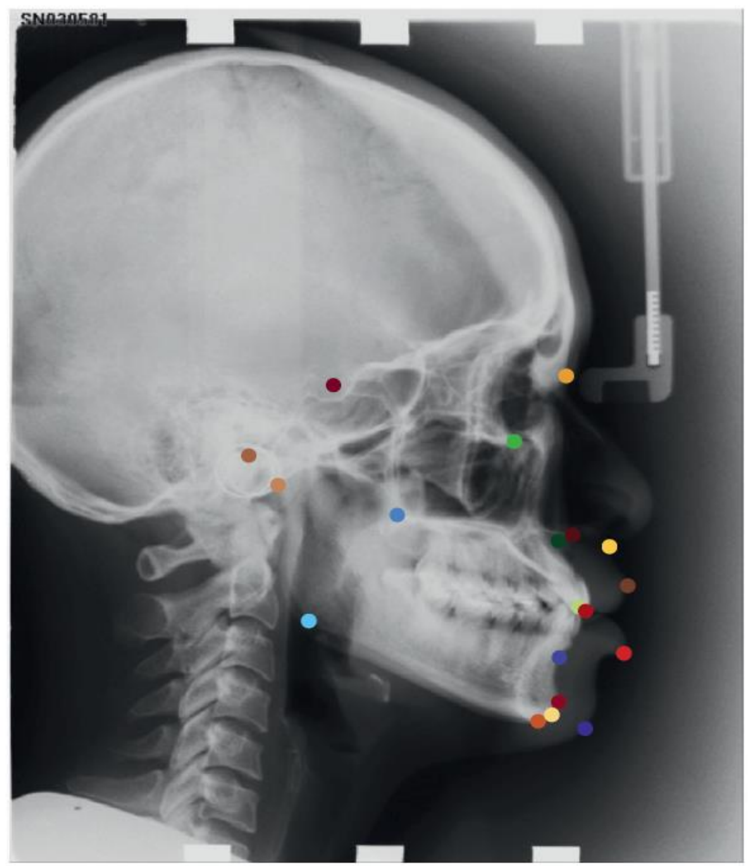
谭子萌

Dataset

2015 ISBI Challenge

400张颅骨X光数据（其中150train 250test）

1935×2400，0.1×0.1mm^2，19个关键点标注



(c) Lateral Cephalograms

Locating Cephalometric X-Ray Landmarks with Foveated Pyramid Attention

Motivation

1. 人只关心视网膜中心凹处的小部分区域，远离焦点的像素呈指数减少。CNN均匀采样的策略：高分辨率图像计算量大、内存占用大；低分辨率图像会显著影响检测精度。

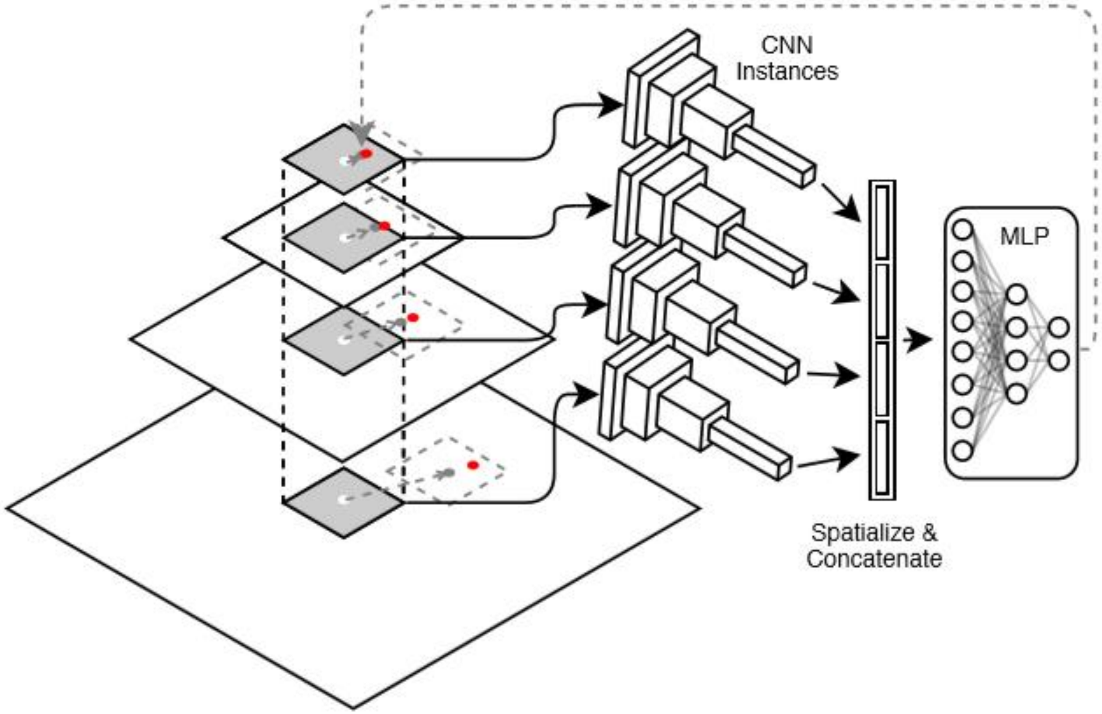
Contribution

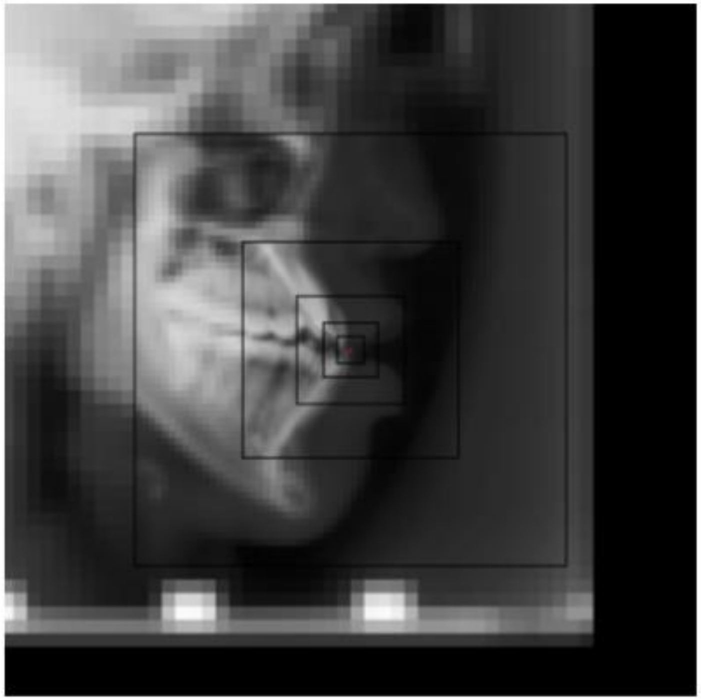
1. 围绕焦点的非均匀采样：图像金字塔
2. 错误反馈：迭代估计关键点坐标

Foveated Pyramid Attention

1. Input：对原图进行下采样2^i。以焦点为中心采样相同大小的patch N×64×64
2. 初值选取：train：在训练集拟合的高斯分布上采样；

   test：直接使用训练集的均值位置。





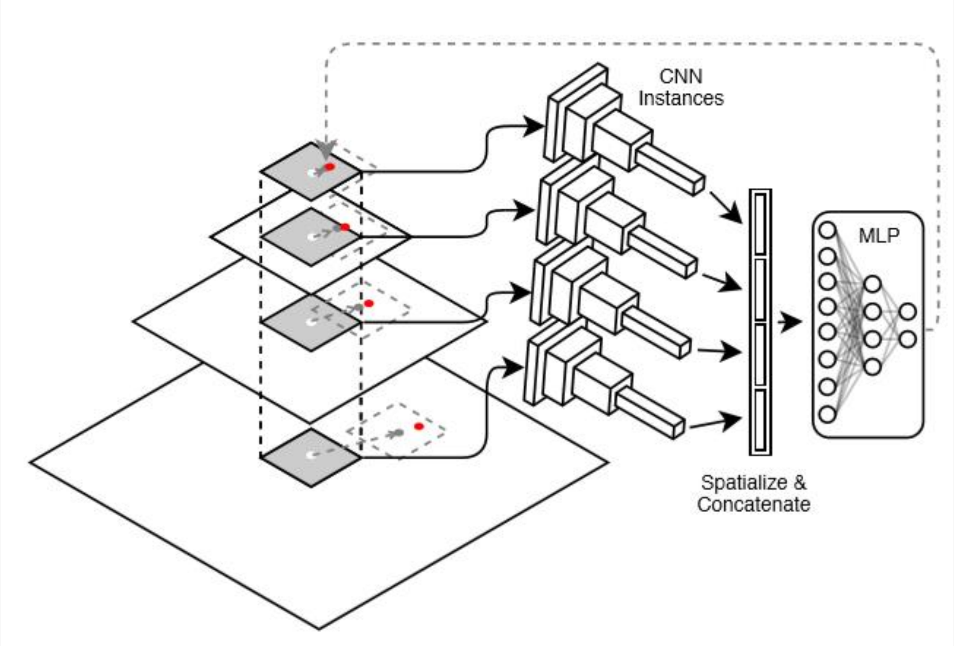
Model

1. CNN：以图像金字塔N×64×64为输入，输出低分辨率热图A：N×256×8×8

2. A->softmax->P. 以可导求期望的方式由热图计算坐标∈[-1,1]，F：N×256×3

$$\boldsymbol{f}_k = \begin{bmatrix} f_{kx} \\ f_{ky} \\ f_{ka} \end{bmatrix} = \sum_{y=1}^{H=8} \sum_{x=1}^{W=8} p_k(x,y) \begin{bmatrix} (x-4.5)/4 \\ (y-4.5)/4 \\ A_k(x,y) \end{bmatrix}$$

3. F->flatten->S：N×768->3层MLP->此阶段的偏移量$\bar{x}$，$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \bar{\mathbf{x}}_t$



L1 loss：scale-free 梯度大小只与方向有关。所有尺度以大致相同的速率学习。

Algorithm 1: Procedure for a Single Image

Input: Image $\mathbf{X}$

Output: Estimated Landmark Location $\hat{\mathbf{x}}$

$\boldsymbol{\mu}$ = mean landmark position and $\boldsymbol{\sigma}$ = standard deviation from the training set

if *training* **then**

  Initialize initial location estimate: $\hat{\mathbf{x}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$

else

  Initialize initial location estimate: $\hat{\mathbf{x}} \leftarrow \boldsymbol{\mu}$

end

Initialize Gaussian Pyramid $\mathbf{I}$ with N levels from image $\mathbf{X}$

for $t \leftarrow 1$ **to** 10 **do**

  Initialize an empty vector of spatialized features $\mathfrak{s}$

  for $i \leftarrow 1$ **to** N **do**

    Crop a zero padded 64×64 glimpse patch g_i from pyramid level I_i centered on $\hat{\mathbf{x}}$

    Process g_i with the CNN to produce a $C \times H \times W$ activation volume $\mathbf{A}$

    Spatialize the channels of $\mathbf{A}$ into a flat $3 \times C$ vector $\boldsymbol{s}_i$ of C spatialized features

    Append $\boldsymbol{s}_i$ to $\mathfrak{s}$

  end

  Process $\mathfrak{s}$ with the MLP to produce an offset estimate $\bar{\mathbf{x}}$

  Update the current location estimate: $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \bar{\mathbf{x}}$

  if *training* **then**

    Backpropogate the ℓ_1 error of the label $\mathbf{x}$ and the current estimate: $||\mathbf{x} - \hat{\mathbf{x}}||_1$

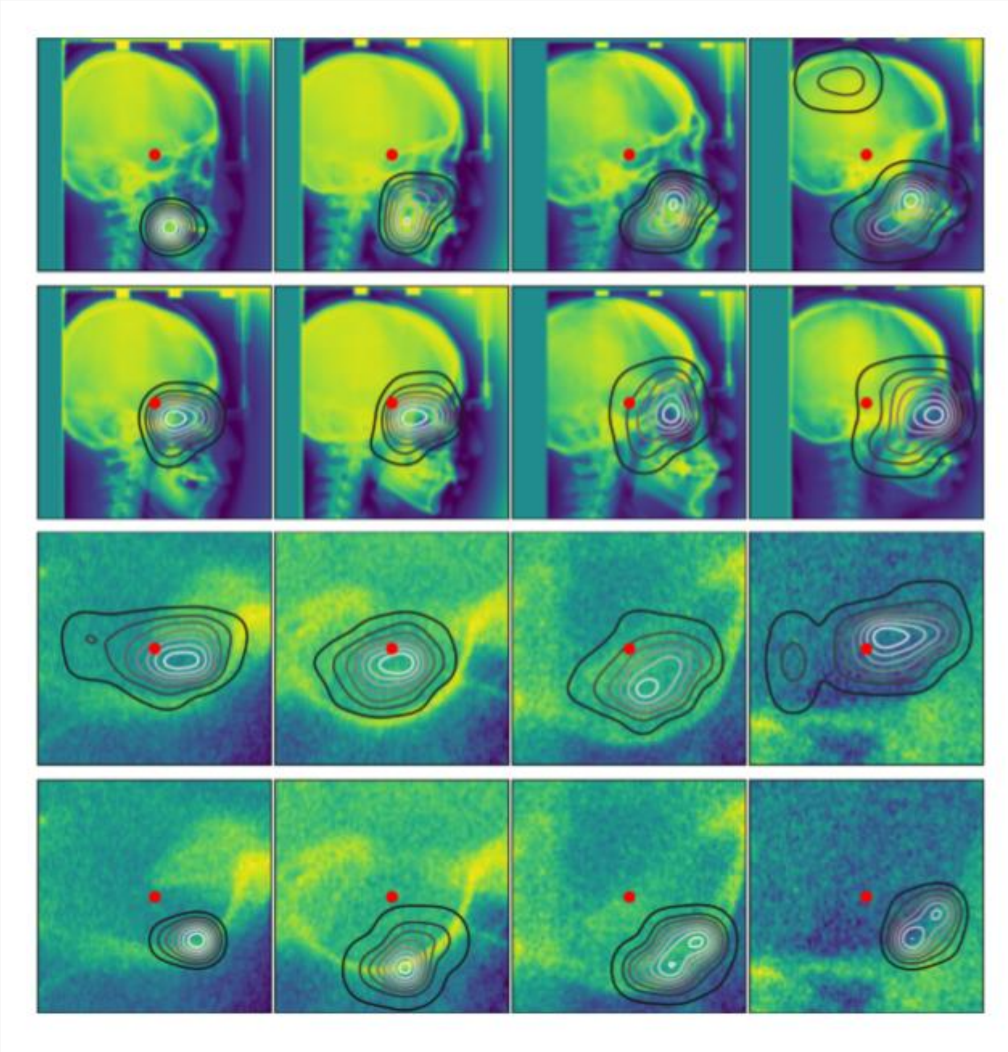
  end

end

ua

Train

CNN：ResNet结构，在ImageNet上做预训练



Structured Landmark Detection via Topology-Adapting Deep Graph Learning

Motivation

1. 热图方法的主要缺点：没有考虑形状/结构约束

   直接回归坐标的方法具备整合结构知识的潜力（使用训练集均值做初始化其实间接地注入弱结构知识）

2. 手工定义点间关系/分组会引入主观因素，导致表现次优

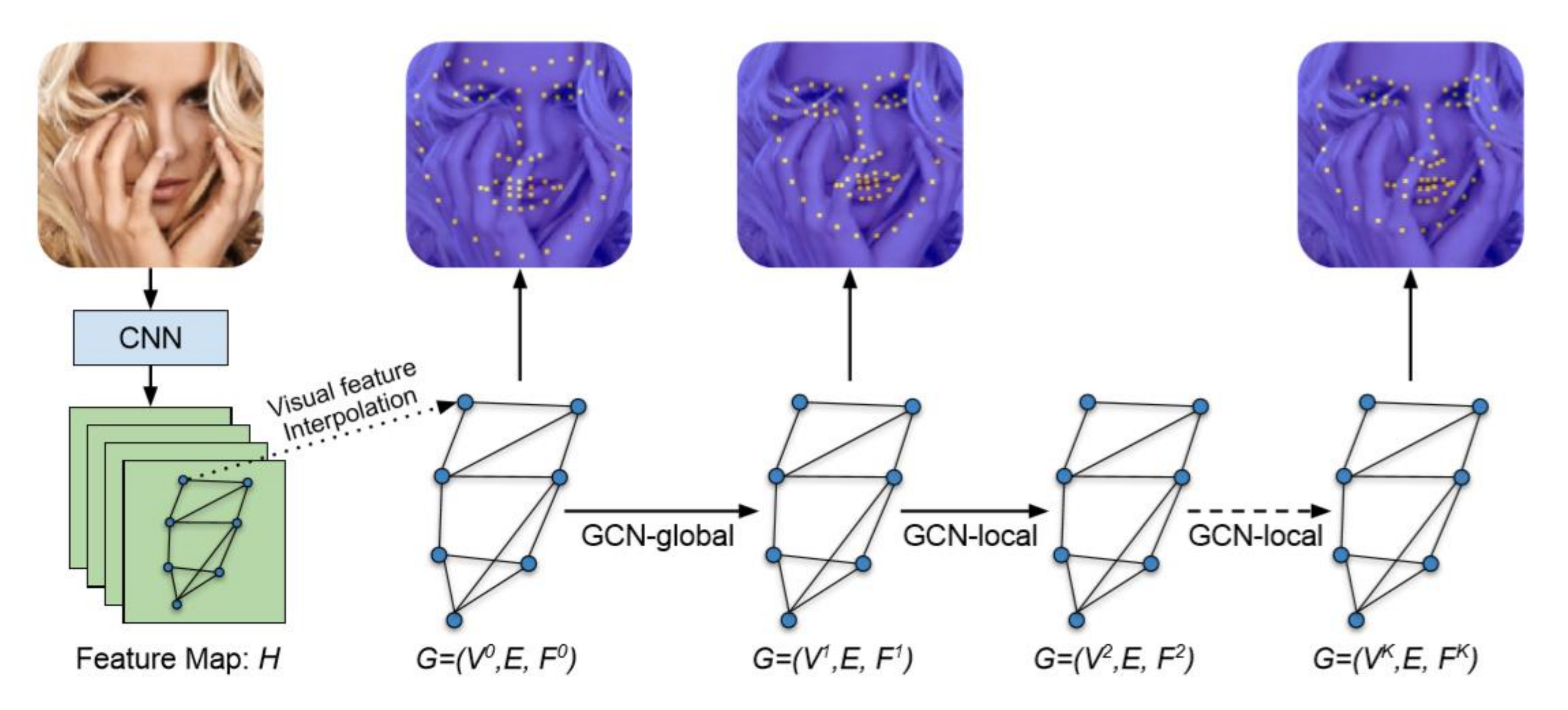
Contribution

1. 空间结构先验 + 局部图像特征
2. 减少手工定义关键点之间关系的需要，使方法用于不同任务
3. 引入形状特征，可在外观有较大变化（如遮挡）时保持高鲁棒性

Cascaded GCN

Global-to-local cascaded Graph Convolutional Networks

1. Global transformation of landmarks: 初值mean shape
2. Local refinement: 局部迭代偏移



Two-cascade GCN

无向图： G=(V, E, F)

V={v_i}:关键点　E={e_{ij}}:学习到的点之间的联系　F={f_i}:特征

$$\mathbf{f}^i \leftarrow \mathbf{W}_1\mathbf{f}^i + \sum_j e_{ij}\mathbf{W}_2\mathbf{f}^j$$

特征集 F={f_i}：

1. 局部图像信息：从CNN中得到D维特征图H在每个坐标位置上v_i的插值

$$\mathbf{p}_i \in R^D$$

2. 全局形状信息：与其他所有关键点间的偏移量（flattened）

$$\mathbf{q}_i = \{\mathbf{v}_j - \mathbf{v}_i\}_{j\neq i} \in R^{2\times(N-1)}$$

3. concat $\mathbf{f}_i \in R^{D+2(N-1)}$.

Global Transformation GCN

透视变换，L1 loss

$$M = [a, b, c, d, e, f, g, h, i]^T \in \mathbb{R}^{9\times 1}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \cong \begin{bmatrix} rx' \\ ry' \\ r \end{bmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

对 $\{\mathbf{m}_i\}$

$$\mathbf{M} = \frac{1}{N} \sum_i \mathbf{m}_i,$$

对齐后的关键点坐标

$$V^1 = \{\mathbf{v}_i^1\} = \{\mathbf{M}\mathbf{v}_i^0\}$$

Local Refinement GCN

与global GCN区别仅在于最后一层输出为二维向量

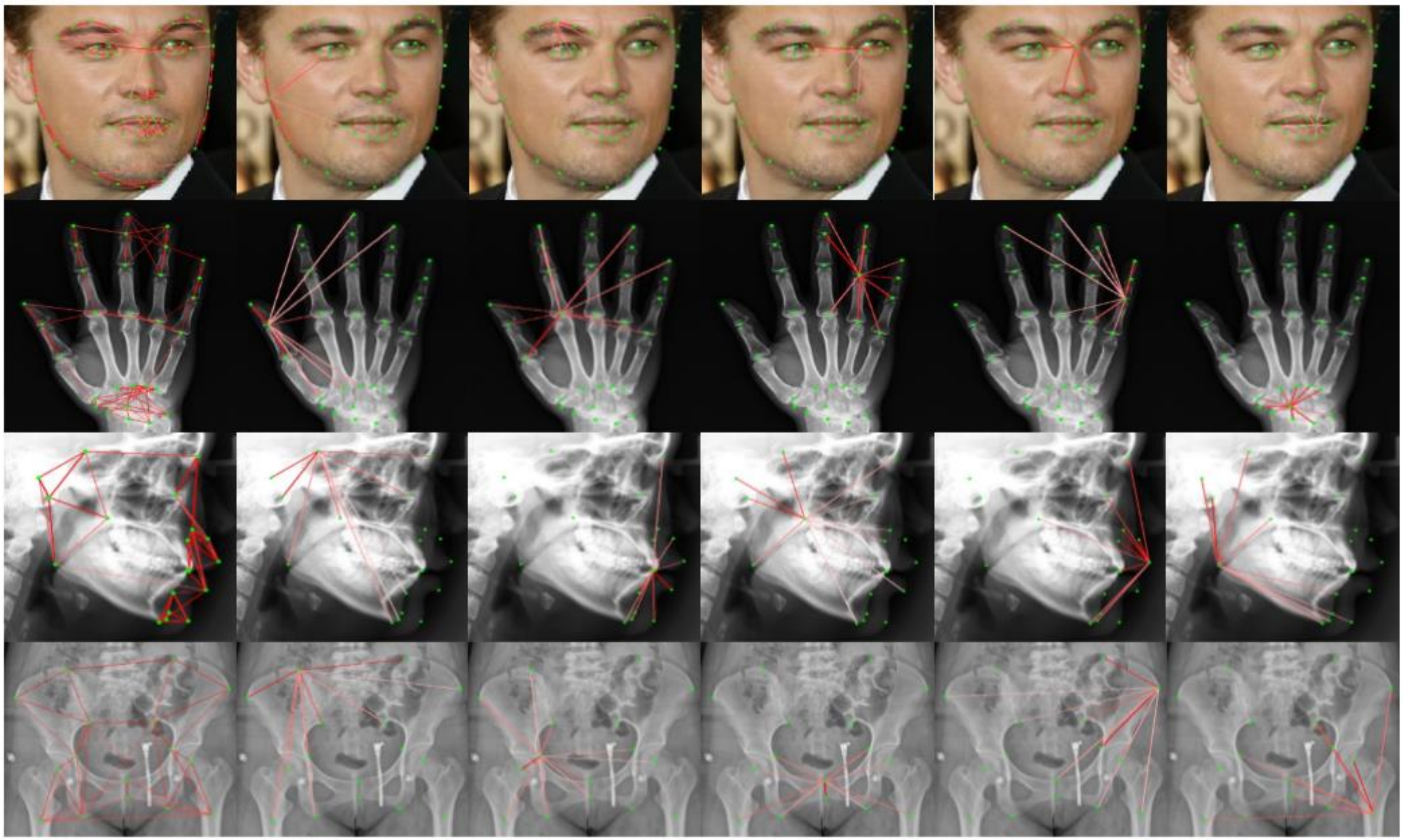
$$\Delta \mathbf{v}_i^t = (\Delta x_i^t, \Delta y_i^t)$$

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \Delta \mathbf{v}_i^t,$$

迭代多次 T=3

Results

Dataset: 3个人脸数据集（带遮挡）+ 3个x光数据集



谢谢大家!