Paper Reading: Head Pose Estimation

2020-08-06

Intro

- 头部位姿估计（HPE）：从一张人脸图片回归出其空间的三维角度（欧拉角）。
- 不同文章解决的关键问题各有不同，如上次讲到的keypoints-free问题(HopeNet\FSA-Net)
- 这周分享的两篇文章：
  - 使用生成图像训练（迁移学习）

    (Felix ICCV19)
  - 多任务学习

    (Deng CVPR20)

Deep Head Pose Estimation Using Synthetic Images and Partial Adversarial Domain Adaption for Continuous Label Spaces

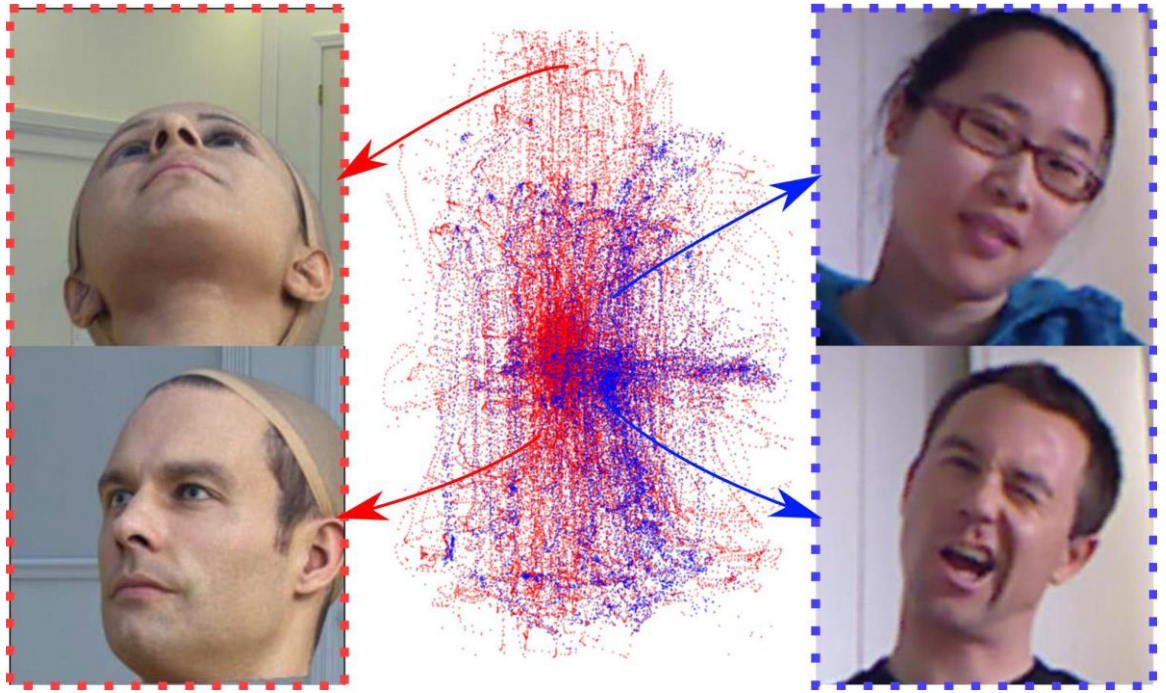
FELIX KUHNKE,
JORN OSTERMANN

INSTITUT FUR
INFORMATIONSVERARBEITUNG LEIBNIZ
UNIVERSITY HANNOVER, GERMANY

ICCV19

问题描述

- 在HPE领域，给深度网络提供足量的有标签的数据是一项很困难的工作：
  - 使用传感器（如深度、惯性）采集数据的话，标签会受到传感器本身噪声影响。如BIWI数据库平均就有1°左右的误差。
  - 使用手标关键点作为标签，则会受到未知3D模型和相机参数的影响，导致结果有误。
  - 使用生成数据（如人脸模型进行旋转）似乎可行，但是直接使用生成数据进行训练会导致测试结不佳，原因是训练库的标签分布和真实数据的分布有所区别。

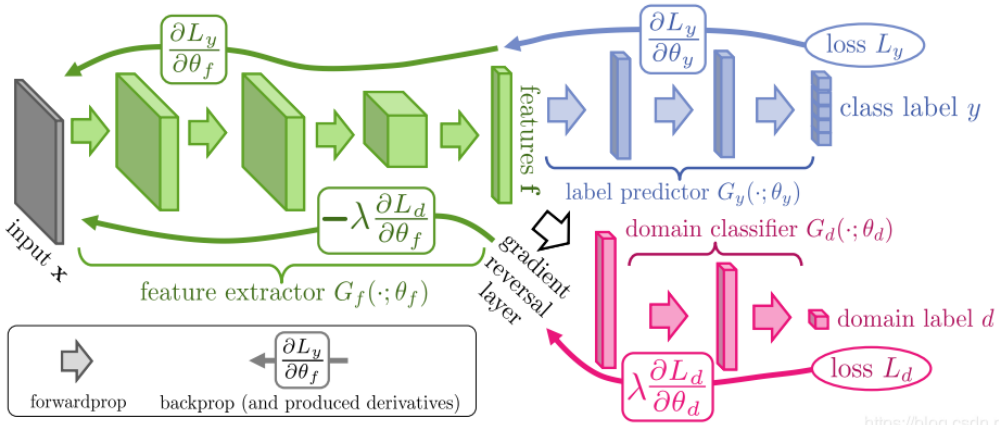


本文贡献

- 为解决标签分布有异的问题，本文沿用了视觉任务中另一个常见的算法：域适应学习（DA）。
- 文章对域适应学习进行了改写，将其适用于HPE任务（回归任务，标签连续，源域标签分布多于目标域）
- 文章进一步对域适应学习进行了实验分析，提出了一套评价基准。

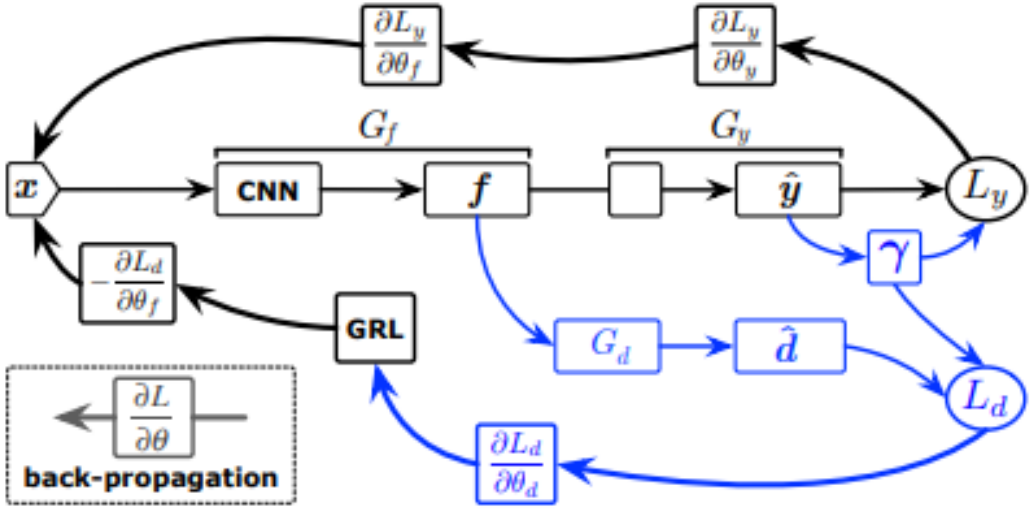
域适应学习

- 域适应学习：假定有一组有标签的数据D_S以及一组无标签的数据D_T，学习的目标就是学习出D_T中的标签。
- 一个典型算法是DANN：利用对抗的思想进行学习，以学习到域不变的特征



$$C_0(\theta_f,\theta_y,\theta_d)=\frac{1}{n_s}\sum_{\mathbf{x}_i\in\mathcal{D}_s}L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right),y_i\right)$$
$$-\frac{\lambda}{n_s+n_t}\sum_{\mathbf{x}_i\in\mathcal{D}_s\cup\mathcal{D}_t}L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right),d_i\right)$$

- Partial ADA：传统的DANN只能处理源域和目标域相同的问题，对于目标域小于源域的情况，需要将源域中不存在的类别去掉



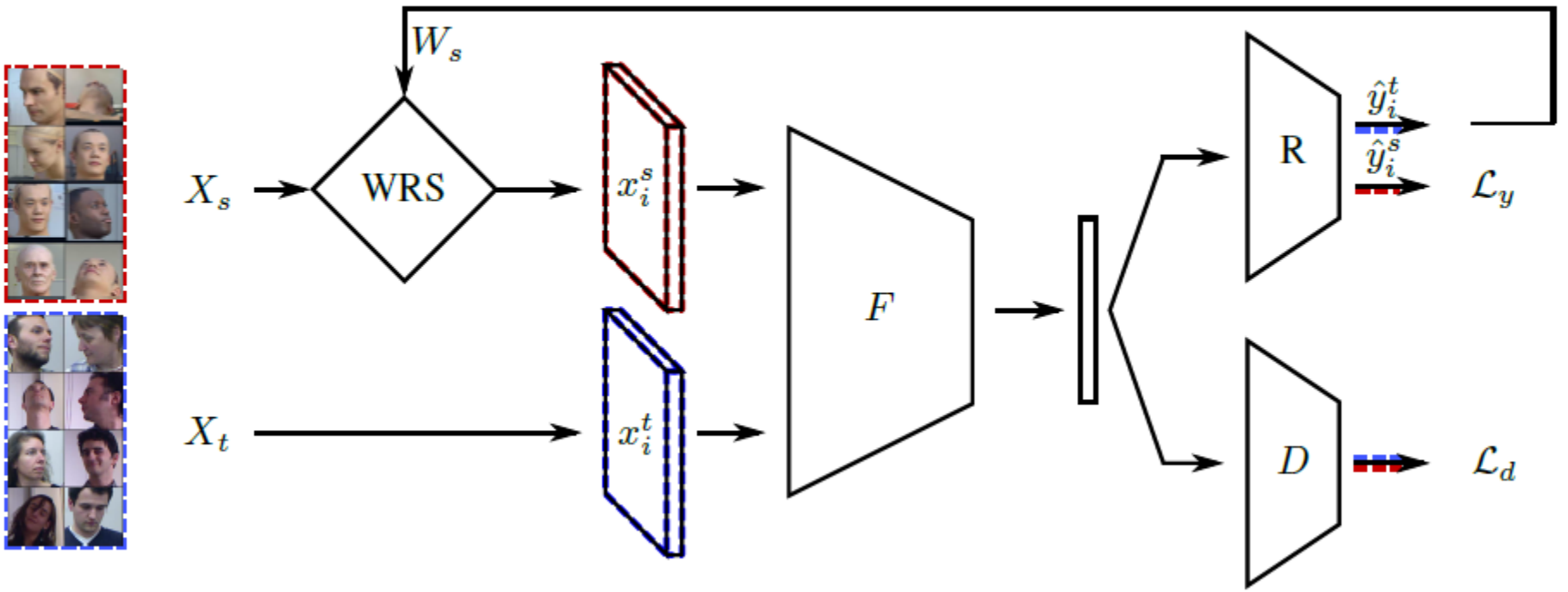
$$C\left(\theta_f, \theta_y, \theta_d\right) = \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \gamma_{y_i} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), y_i\right)$$
$$- \frac{\lambda}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \gamma_{y_i} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right)$$
$$- \frac{\lambda}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right)$$

$$\boldsymbol{\gamma} = \frac{1}{n_t} \sum_{i=1}^{n_t} \hat{\mathbf{y}}_i,$$

回到HPE问题

- 仿照PADA的思路搭建网络，其中筛选源域的系数改为：

$$w_i^s = \begin{cases} 0, \text{ if } \min_{\hat{y}_i^t \in \hat{Y}_t} \mathcal{L}_y(\hat{y}_i^t, y_i^s) \geq t \\ t - \min_{\hat{y}_i^t \in \hat{Y}_t} \mathcal{L}_y(\hat{y}_i^t, y_i^s), \text{ otherwise.} \end{cases}$$



进一步处理连续标签的问题

- 虽然用到了权重来压制源域中不存在的样本，但这部分样本还是会进入到网络的前传部分，或者说BN部分，这对网络训练同样会有不利的影响。
- 文章将权重的部分进一步提前，不用在计算Loss步骤，而是用于对源样本进行重采样。
- 重采样的步骤利用了K近邻算法：对于目标域的每个样本，找到源域中最接近的N个样本，进行累加，最终得到源域样本的权重值W，最终根据W的概率进行采样。
- 重采样的权重W只在第一步进行计算而不是迭代得到，原因是使用迭代法的话网络不总是收敛。

训练过程

Algorithm 1: Training procedure

Input: labeled source samples X_s, Y_s
unlabeled target samples X_t
parameter λ_{max}, N_n

Output: $\hat{\theta}_F, \hat{\theta}_R$

Stage-1:

$\hat{\theta}_F, \hat{\theta}_R \leftarrow$ pre-train F and R on X_s with Y_s

$\hat{\theta}_D \leftarrow$ random initialization

Stage-2:

$\hat{Y}_t \leftarrow$ evaluate target data $R(F(X_t))$

$W_s \leftarrow$ calculate weights using N_n, Y_s, and $\hat{Y}_t$

while $\lambda < \lambda_{max}$ **do**

- $b_s \leftarrow$ sample source batch with weighted sampling from X_s using W_s
- $b_t \leftarrow$ sample target batch from X_t
- $\hat{\theta}_F, \hat{\theta}_R \leftarrow$ train F and R with b_s
- $\hat{\theta}_F, \hat{\theta}_D \leftarrow$ train F and D with b_s and b_t using adversarial training [12]
- $\lambda \leftarrow$ update λ according to a schedule

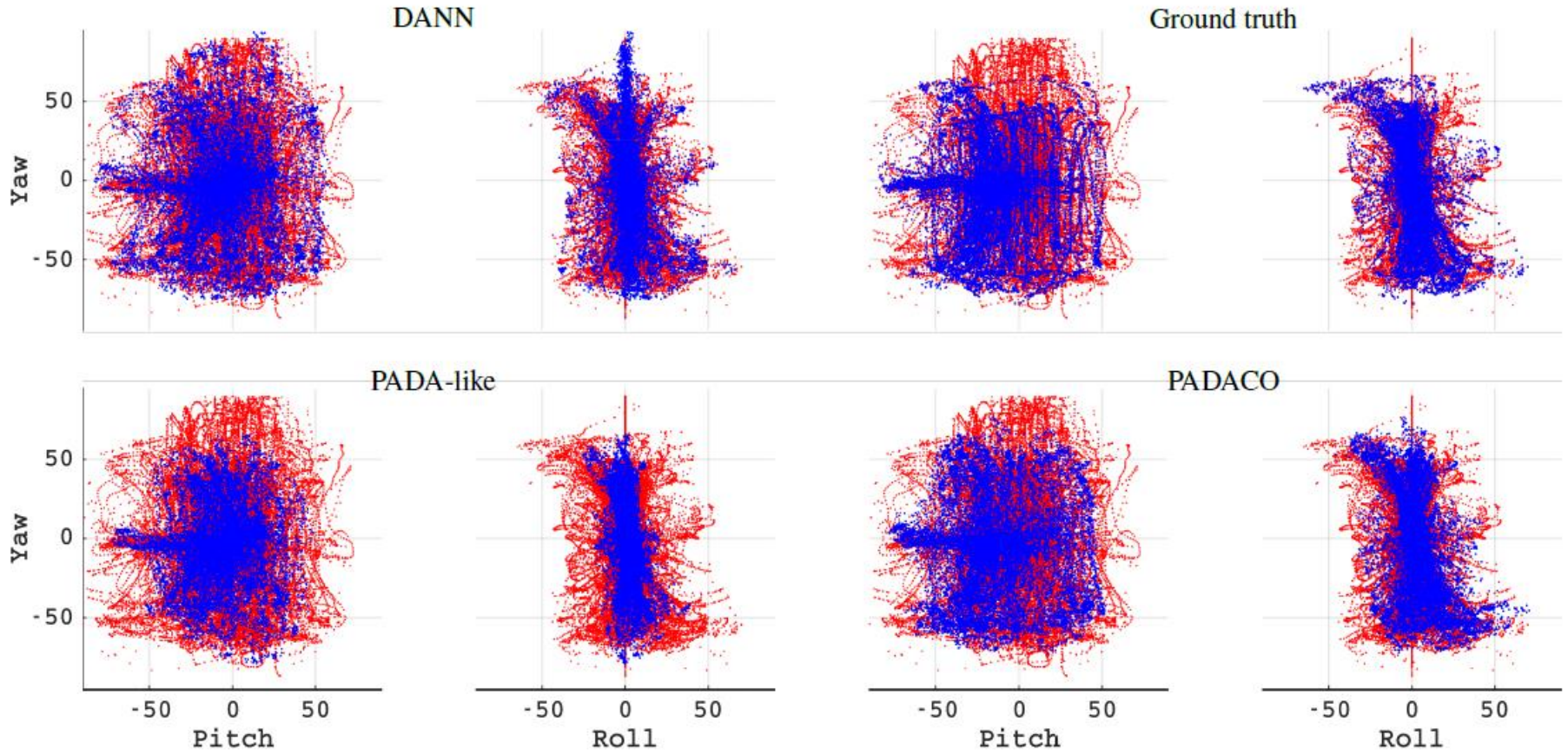
数据库

- 测试库：
  - BIWI：20个人的24个视频序列，使用kinect采集
  - BIWI+:添加了人脸的BondingBox
- 生成库：
  - SynHead：使用10个头部模型在不同角度下得到的图像和角度标签。注意这里由于欧拉角的性质，旋转轴的顺序会对图像造成影响，导致同一组欧拉角对应的实际角度不同，这会导致某些角度在BIWI中存在，但在SynHead中对应的图像已经没有正脸了。
  - SynHead+：在SynHead中找到所有有正脸的图，按照BIWI的欧拉角重新计算标签。
  - SynBIWI+：对于每张BIWI中的图像，将SynHead+中的图像进行旋转，得到对应最相似的10张图像（10个头部模型），构成数据集，用于测试迁移学习性能。
  - SynHead++：上述两个数据库的并集，用于测试局部迁移的性能。

实验结果

Experiment	Method	Network	Training set	Test set	MAE	Pitch	Yaw	Roll
Intra domain	Anh [1]	Custom CNN	Biwi*	Biwi*	2.93	3.4	2.8	2.6
	Liu [22]	Custom CNN	Biwi◇	Biwi◇	5.93	6.0	6.1	5.7
	Ruiz [29]	ResNet50	Biwi†	Biwi†	3.23	3.39	3.29	3.00
	Gu [15]	VGG16 [30]	Biwi†	Biwi†	3.66	4.03	3.91	3.03
Inter domain	Ruiz [29]	ResNet50	300W-LP [35]	Biwi†	4.90	6.61	4.81	3.27
	Liu [22]	Custom CNN	*unavailable*	Biwi	3.73	4.3	4.5	2.4
Inter domain	BaselineDA	ResNet18	SynBiwi+	Biwi+	4.58	4.99	4.85	3.89
Domain adaptation	**DANN** [12]	ResNet18	SynBiwi+	Biwi+	**3.34**	**3.56**	**3.43**	**3.03**
	PADACO (proposed)	ResNet18	SynBiwi+	Biwi+	4.04	4.47	4.11	3.56
Inter domain	BaselinePDA	ResNet18	SynHead++	Biwi+	4.53	4.97	4.61	3.97
Partial DA	DANN [12]	ResNet18	SynHead++	Biwi+	6.05	8.08	6.17	3.91
	PADA-like	ResNet18	SynHead++	Biwi+	6.41	8.14	6.86	4.22
	PADACO (proposed)	ResNet18	SynHead++	Biwi+	**4.13**	**4.51**	**4.11**	**3.78**

实验结果


DANN
Ground truth
PADA-like
PADACO
Yaw
50
0
-50
-50
0
50
Pitch
Roll

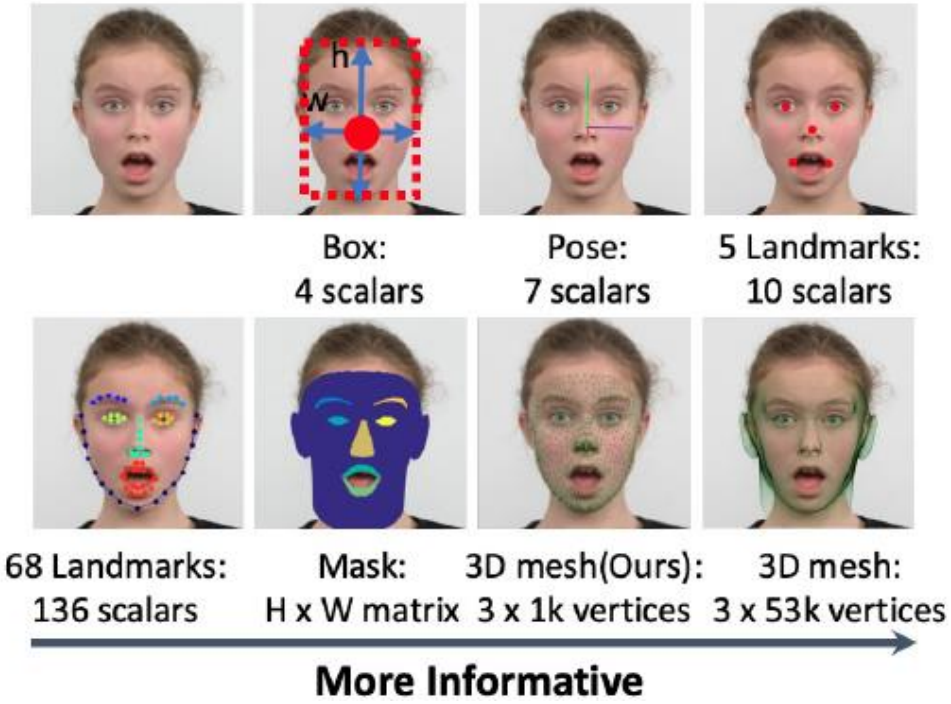
RetinaFace: Single-shot Multi-level Face Localisation in the Wild

JIANKANG DENG,
JIA GUO,
EVANGELOS VERVERAS,
IRENE KOTSIA,
STEFANOS ZAFEIRIOU

CVPR20

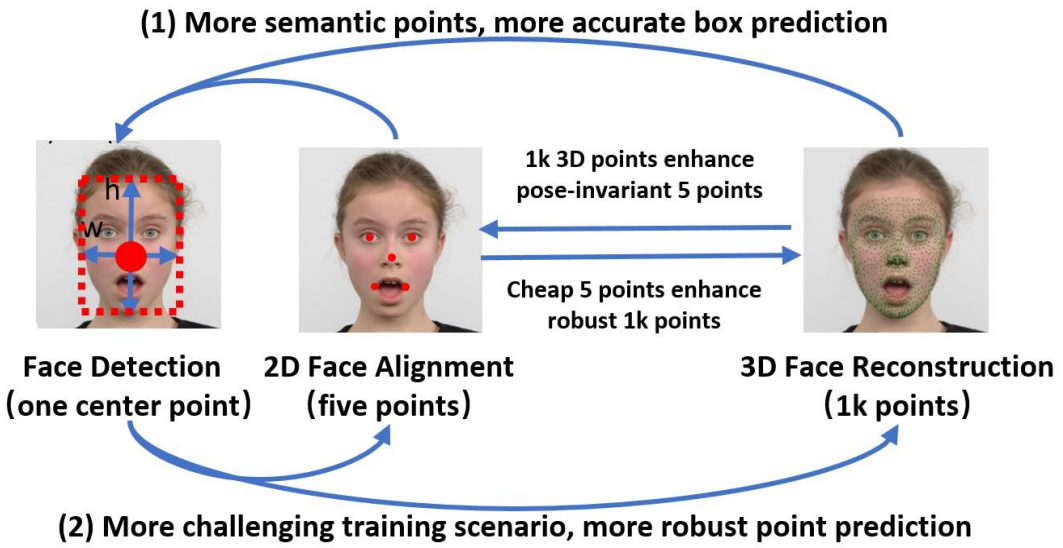
问题描述

- 人脸定位任务：本文中将人脸定位任务广义地定义为多个任务的总和，包括人脸检测、人脸姿态、人脸关键点检测、人脸分割以及三维人脸重建。

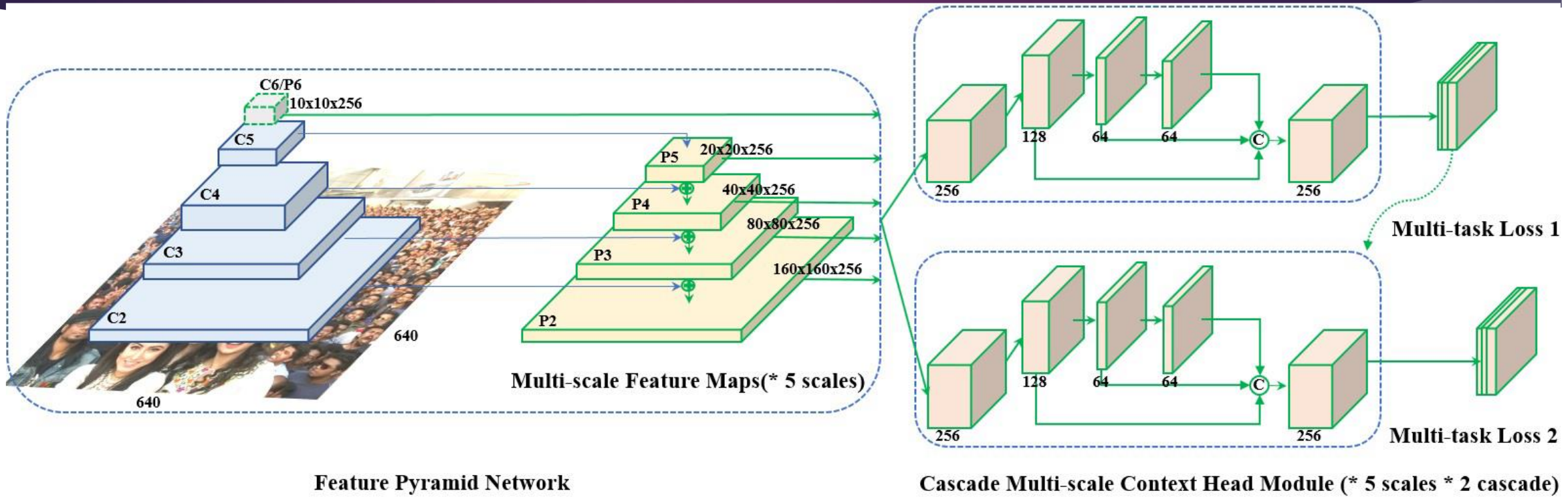


本文贡献

- 将多个人脸定位任务（人脸检测、关键点检测和3Dmesh回归，这些任务实际上可以转化为在人脸平面上点的回归问题）同时进行训练。
- 使用了一种互利的学习策略，保证不同的任务能够相互提高性能。
- 在各项任务中均能达到state-of-the-art的性能。

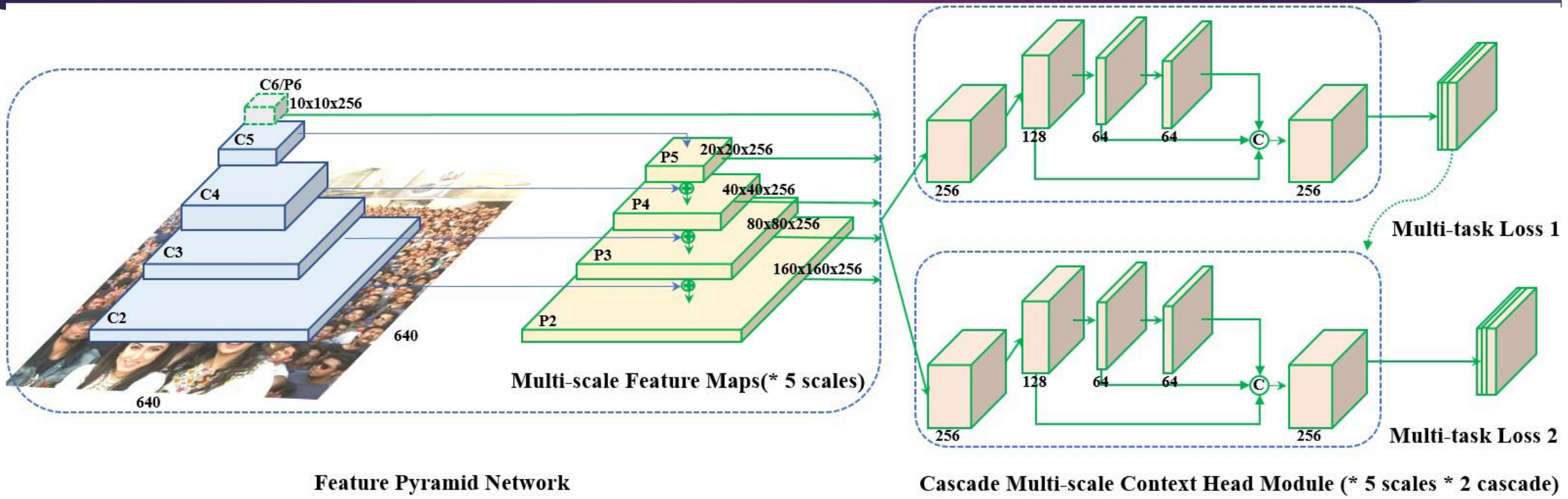


网络结构



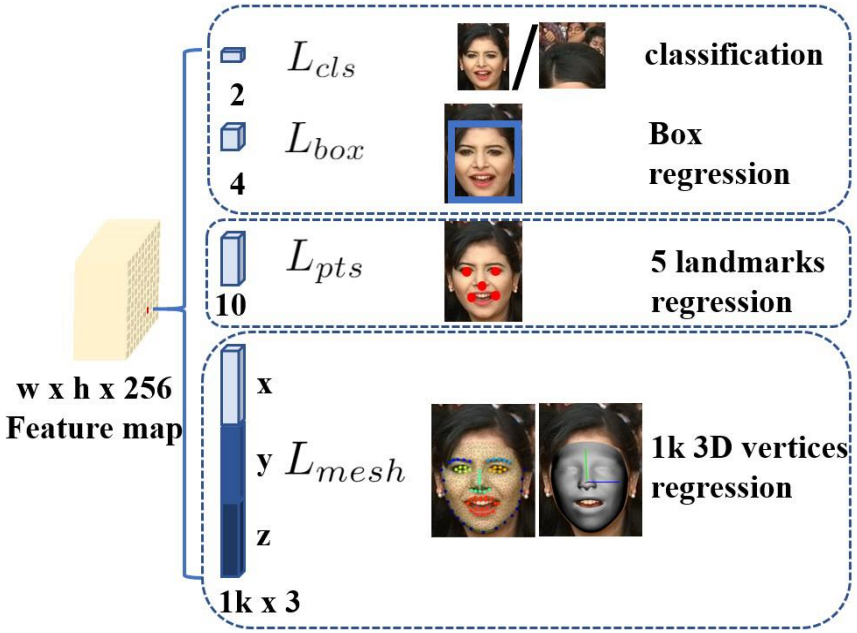
- 整个网络大致可以分为3个部分：FPN、CascadeContextHead和Loss
- FPN是一个典型的特征金字塔结构，共输出5层不同层级的特征
- ContextHead同样合并了不同维度的特征，提升了网络的感受野。卷积也用可变卷积核代替传统3*3核

网络结构



- Cascade部分要求第一次Head使用标准的anchor进行回归，而第二次使用第一次回归得到的人脸box进行进一步回归。

损失函数



$$\mathcal{L} = \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_1 p_i^* \mathcal{L}_{box}(t_i, t_i^*) + \lambda_2 p_i^* \mathcal{L}_{pts}(l_i, l_i^*) + \lambda_3 p_i^* \mathcal{L}_{mesh}(v_i, v_i^*).$$

- 整体的损失函数包含4部分：人脸分类、box回归、关键点回归以及3D点回归
- 其中需要说明两点：
  - 对于训练时每个正样本锚点，其训练目标按照锚点的坐标和尺度会进行如下的变换

$$(x_j{}^* - x^a_{center})/s^a,$$
$$(y_j{}^* - y^a_{center})/s^a,$$
$$(z_j{}^* - z^*_{nose-tip}))/s^a,$$

  - 对于3D点回归，除了常规的点坐标的差异之外，为了考虑到2D图像可能不包括所有点，还要加上三角面片边长的差异

$$\mathcal{L}_{vert} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{V}_i(x,y,z) - \mathbf{V}_i^*(x,y,z)||_1, \qquad \mathcal{L}_{edge} = \frac{1}{3M}\sum_{i=1}^{M} ||\mathbf{E}_i - \mathbf{E}_i^*||_1,$$

锚点设置和匹配策略

- 各层特征的锚点设置如表所示，输入图像大小为640*640，锚点可以覆盖从16*16到406*406大小的图像，保证各尺度的人脸都能被检测到。
- 对于cascade模块，第一次检测时，锚点的IOU高于0.7认为是正样本，第二次该阈值改为0.5。
- 在后续实验中，非网络直接得到的结果（如68关键点、姿态等）都可以通过3D点计算得到。

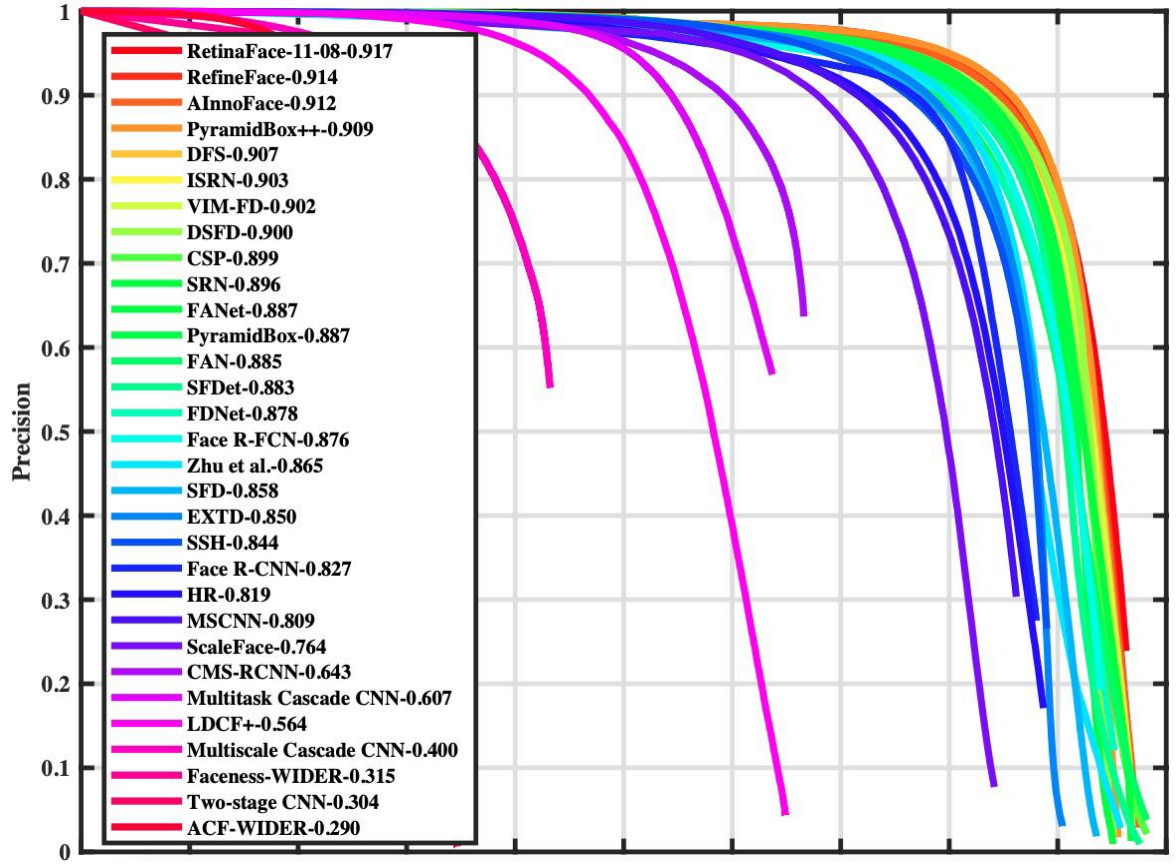
Feature Pyramid	Stride	Anchor
P_2 ($160 \times 160 \times 256$)	4	16, 20.16, 25.40
P_3 ($80 \times 80 \times 256$)	8	32, 40.32, 50.80
P_4 ($40 \times 40 \times 256$)	16	64, 80.63, 101.59
P_5 ($20 \times 20 \times 256$)	32	128, 161.26, 203.19
P_6 ($10 \times 10 \times 256$)	64	256, 322.54, 406.37

Table 2. The details of feature pyramid, stride size, anchor in RetinaFace. For a 640×640 input image, there are 102,300 anchors in total, and 75% of these anchors are tiled on P_2.

数据库

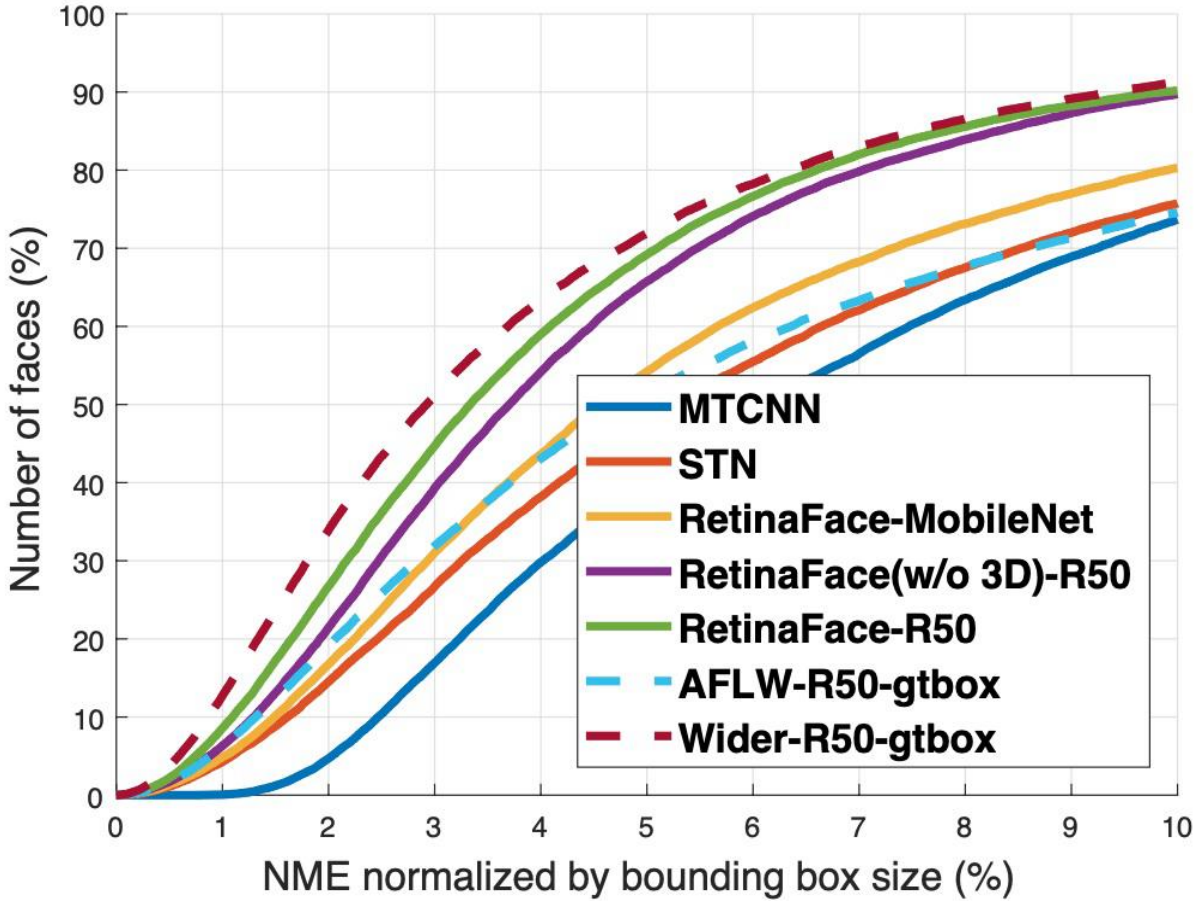
- WIDER FACE dataset：包含了32, 203张图像共393, 703个人脸标注
- 人工标注了84.6k个人脸的5关键点用于训练集18.5k人脸的5关键点用于验证集
- 半自动标注3D点：首先通过人脸图像回归68个3D点，使用这68个点生成一个53K顶点的mesh模型，为保证准确性，人工验证这个模型重新投影到人脸上是否准确，如果不准确，则人工重新选取68个关键点重复这一步骤，如果再不准确，则抛弃这个人脸。最终得到了22k个有3D点标注的人脸图像。
- 再从AFLW和FDDB两个数据库分别得到27.1k和39.3k个人脸图像（包含3D标注）

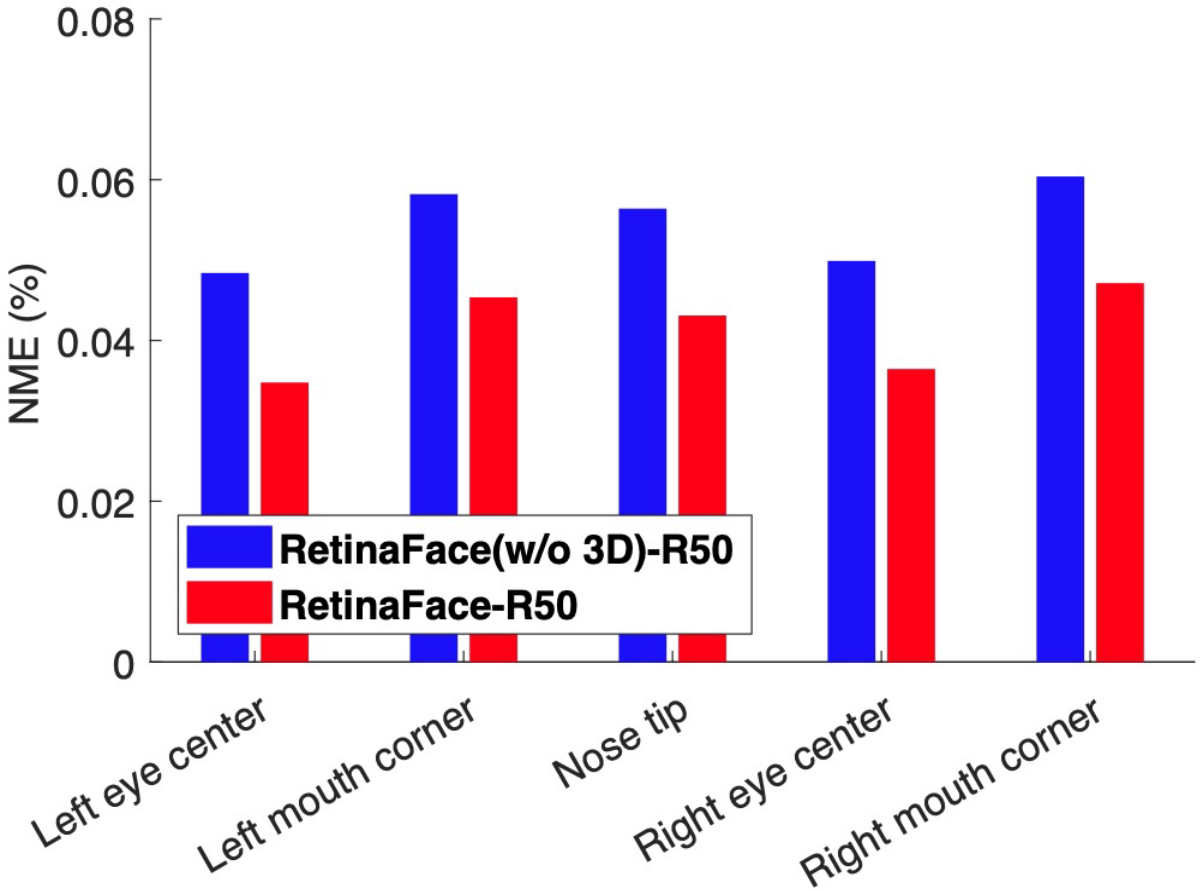
实验结果1：人脸检测



Method	Easy	Medium	Hard	average AP
Baseline	95.832	95.243	89.875	52.65
+DCN	96.149	95.568	90.286	53.36
+Cascade	96.233	95.679	90.642	54.20
$\mathcal{L}_{pts}$	96.570	95.913	91.161	54.73
$\mathcal{L}_{vert}$	96.512	95.805	90.983	54.55
$\mathcal{L}_{mesh}$	96.528	95.829	90.991	54.62
$\mathcal{L}_{5pts+mesh}$	**96.713**	**96.082**	**91.447**	**55.02**

实验结果2：5关键点检测





实验结果3：68关键点检测

Method	[0°,30°]	[30°,60°]	[60°,90°]	Mean
SDM [72]	3.67	4.94	9.67	6.12
3DDFA [72]	3.43	4.24	7.17	4.94
Yu *et al.* [61]	3.62	6.06	9.56	6.41
3DSTN [2]	3.15	4.33	5.98	4.49
PRN [19]	2.75	3.51	4.61	3.62
FAME [5]	3.11	3.84	6.60	4.52
SS-SFN [6]	3.09	4.27	5.59	4.31
MS-SFN [6]	2.91	3.83	4.94	3.89
$\mathcal{L}_{vert}$	2.77	3.70	4.95	3.81
$\mathcal{L}_{mesh}$	2.72	3.65	4.81	3.72
$\mathcal{L}_{5pts+mesh}$	**2.57**	**3.32**	**4.56**	**3.48**

实验结果4：语义分割&姿态

Method	Eyebrow	Eye	Nose	Lip
DenseReg [1]	47.62	74.29	87.71	72.35
$\mathcal{L}_{5pts+vert}$	71.3	76.85	90.90	75.43
$\mathcal{L}_{5pts+mesh}$	**72.23**	**78.51**	**92.21**	**77.55**

Method	[0°,30°]	[30°,60°]	[60°,90°]
DenseReg [1]	4.14 ± 3.93	5.96 ± 4.74	6.38 ± 4.90
PRN [19]	3.96 ± 3.43	5.75 ± 4.42	6.08 ± 4.41
$\mathcal{L}_{5pts+vert}$	3.79 ± 3.08	5.28 ± 3.83	5.60 ± 3.81
$\mathcal{L}_{5pts+mesh}$	3.69 ± 2.99	5.11 ± 3.73	5.41 ± 3.57

实验结果5：定性效果展示