

# 机器学习面部认证实验报告

## 1 模型简介

本报告中介绍了本组面部认证的模型工作原理，实现细节，以及实验结果的分析。该模型由两部分构成，包括**面部关键点检测**以及**高维 LBP 特征**两大部分。首先，面部关键点检测使用集成学习的思想，利用随机森林对面部特征点进行回归；高维 LBP 特征在检测到的面部关键点的基础上，在图像金字塔中在以关键点为中心提取 LBP 特征；再对高维特征进行降维从而在保留多数信息的基础上，提高模型的准确性。在实验中，我们在 Labeled Faces in the Wild (LFW) 数据集上实现了 **xx.xx%** 的准确率，F1 分数为 **XX.XX%**。

本报告按照一项方式展开：第2章介绍了面部关键点检测与高维 LBP 特征的工作原理和实施细节；第3章介绍了实施细节、实验结果以及结果的分析。

## 2 模型结构

本章介绍模型面部关键点检测与高维 LBP 特征的工作原理。其中，面部关键点检测的实现参照 [1] 的工作；高维 LBP 特征的实现参照 [2] 的工作。

### 2.1 面部关键点检测

该模型的工作框图参见图 1。该模型对面部特征点进行回归，在给定一个初始面部关键点集合，模型会经过一个双层结构的随机森林对面部关键点不断进行修正，从而实现准确的关键点估计。本节按照工作框图对模型各个部分进行介绍。

#### 2.1.1 平均形状迭代估计

我们假设给定的训练集为  $\{I_i, \hat{S}_i\}_{i=1}^N$ ，其中  $I_i$  是给定的图片， $\hat{S}_i$  是标注好的关键点集合，也成为形状。一个关键的概念叫做**正规形状**：在给定平均形状  $\bar{S}$  的情况下，一个形状  $S$  的正规形状是经过满足下式的仿射变换变换后得到的形状  $M \circ S$ ：

$$M_S = \arg \min_M \|\hat{S} - M \circ S\|_2$$

而所谓的**平均形状**是指所有正规形状的平均值。在实际的实现中，由于我们不知道平均形状  $\bar{S}$ ，因此采用一种迭代的方式进行估计。具体来说，我们首先制定任意一个形状，比如  $S_1$  作为平均形状，然后将其他所有形状向  $S_1$  正规化；然后在取所有形状的平均值作为新的平均形状，如此迭代计算。实际的实现中，大约 4 次迭代就已经可以保证平均形状的收敛。

### 2.1.2 形状相关特征

与随机森林协同使用的特征最经典的是**像素强度差**，即两个像素点之间的强度差。这种特征简单但是也具有较弱的区别能力。由于这种特征需要应用各种各样的人脸上，提取的像素强度差需要具有一定的几何不变性——一方面由于要进行仿射变换，另一方面要考虑到人姿态、表情等的不同。**形状相关特征**可以满足这种需要：所谓形状相关特征指的是以关键点为坐标原点，在坐标点附近选择像素。图??展示了这种局部坐标的表示方法；虽然两张人脸的五官尺度不同，但是在采用局部坐标表示的情况下都能表示相似的意义，比如眉毛前端的颜色的深浅以及左侧嘴角附近的颜色（比如是否有痣等）；而采用全局坐标则由于面部尺度的不同导致像素点之间表达的含义不同。因此也保证了一定的语义不变性。

在实际的实现中，记局部坐标为  $\delta^l = (\delta_x^l, \delta_y^l)$ ，其中  $l$  表示选取的关键点  $l = (x^l, y^l)$ ，那么  $\delta_x^l \sim U[x^l + \kappa, x^l - \kappa]$ ， $\delta_y^l \sim U[y^l + \kappa, y^l - \kappa]$ ，其中  $U[a, b]$  是定义在  $[a, b]$  上的均匀分布。 $\kappa$  是一个超参数，决定了这种波动的范围。在实际中定义为两眼瞳孔间距的 0.3 倍。同时，超参数  $P$  决定了选择的像素点的个数，模型会从关键点中随机选择 1 个点，然后在这个点的基础上采一个像素。这个过程会重复  $P$  次。因此，一共可以产生出  $P \times P$  个特征（含像素自身与自身的差）。

### 2.1.3 相关特征选择

形状相关特征一共会产生  $P^2$  个特征，过多的特征不利于拟合，同时也有较高的计算开销。模型在  $P^2$  个特征中选择  $F$  个特征，选择的原则是 1) 特征要与训练目标有较高的相关度；2) 各个被选择特征之间的相关度要尽量小。模型使用基于相关度的方法对特征进行筛选。给定  $Y \in \mathbb{R}^{N \times 2N_{fp}}$ ，其中  $N$  是悬链样本个数， $N_{fp}$  是选择的面部关键点的个数，记  $X \in \mathbb{R}^{N \times P^2}$ ，即像素差矩阵。选择  $v \sim \mathcal{N}(0, I_{2N_{fp}})$  对  $Y$  进行随机采样，选择：

$$j_{opt} = \arg \min_j \text{cor}(Yv, X_j)$$

列作为最优特征。将此步骤重复  $K$  次即可选择出最相关的  $K$  个特征。

在实际的实现中，由于这样的运算复杂度在  $O(NP^2)$ ，利用下面式子的关系：

$$\begin{aligned} \text{corr}(Yv, \rho_m - \rho_n) &= \frac{\text{cov}(Yv - \rho_m) - \text{cov}(Yv - \rho_n)}{\sqrt{\sigma(Yv)\sigma(\rho_m - \rho_n)}}, \\ \sigma(\rho_m - \rho_n) &= \text{cov}(\rho_m, \rho_m) + \text{cov}(\rho_n, \rho_n) - 2\text{cov}(\rho_m, \rho_n) \end{aligned}$$

如果预先计算好  $\sigma(\rho_m - \rho_n)$ ，可以将复杂度减少到  $O(NP)$ 。

### 2.1.4 双层随机森林回归器

在训练时，模型连续地训练一系列弱回归器来优化目标函数。假设模型使用  $T$  个连续的弱回归器  $\{R^1, \dots, R^T\}$ ，在训练每一个回归器的阶段， $R^t$  需要：

$$\begin{aligned} \min_{R^t} \sum_{i=1}^N \|y_i - R^t(I_i \cdot S_i^{t-1})\|_2, \\ y_i = M_{S_i^{t-1}} \circ (\hat{S}_i - S_i^{t-1}) \end{aligned}$$

也即,  $R^t$  不断地最小化正规化后上一回归器回归结果与真实关键点位置之间的差距。

在实现中,  $R^t$  应该是一个较弱的回归器。模型选择了序列的随机蕨 [1] 回归器, 也就是说  $R^t$  是由连续的随机蕨  $r_1, \dots, r_K$  组成, 其中  $K$  是一个超参数。在训练的时候, 每一个  $R^t$  会在图像中提取新的特征, 即  $F$  个像素强度差, 而对于  $r_k$ , 这些特征不改变。对于每一个蕨, 它会将样本按照每个特征上的取值分成  $2^F$  类, 每一类会产生一个输出  $y_b$ 。对于蕨, 它的优化目标为:

$$y_b = \arg \min_y \sum_{i \in \Omega_b} \|\hat{y}_i - y_i\|_2$$

其中  $\Omega_b$  是被分到同一类的样本下标集合。显然, 最优解是均值。但这可能会造成过拟合, 因此采用

$$y_b = \frac{1}{1 + \beta/|\Omega_b|} \frac{\sum_{i \in \Omega_b} \hat{y}_i}{|\Omega_b|}$$

作为最优解。

对于蕨, 学习输出一方面, 同事也要学习如何确定样本在哪一类中, 这是通过阈值来实现的。预知的选取是从均匀分布中随机选值, 从而避免过拟合。

## 2.2 高维 LBP 特征

图??展示了高维 LBP 特征提取的工作流程。首先, 模型基于图片中人脸的关键点, 在关键点附近对对蹄片进行采集。这来自于一个直观的观点: 人的五官和轮廓含有更丰富的区分信息。但是这样的高纬度会使得训练变得困难, 一方面不利于最优值的求解, 另一方面会使训练的速度下降。因此, 模型进一步利用 PCA 进行降维, 这种方法的关键之处在于没有过早地引入数据降维, 而是在利用高维度信息之后在进行降维, 从而一定程度上保留了高维度特征中较为丰富的信息。本节将对该模型的实现细节进行介绍。

### 2.2.1 图形矫正

首先, 利用之前的回归方法, 模型确定两个瞳孔、以及鼻子的关键点, 按照图??的方式对图像进行旋转。这个方法将面部上三角区域的中心点的纵坐标调整到与鼻尖相同, 从而将面部摆正。在面部竖直的情况下, 模型更容易抓住相似的特征, 从而提升模型的表现效果。

### 2.2.2 高维 LBP 特征提取

图??介绍了我们如何对面部特征进行提取。模型利用较为**稠密、准确**的关键点, 以关键点为中心选择一块  $P \times P$  大小的区域, 将其分割为  $S \times S$  个网格。在每一个网格内, 模型提取其中的 LBP 特征。LBP 特征的计算如图??所示, 它将一片区域转化为二进制的表示, 因此称为 Local binary pattern (LBP)。

为了适应面部尺度大小的变化, 在单纯的 LBP 基础上, 模型构建了  $L$  级的图形金字塔, 在每个尺度都进行这样的特征提取。虽然图像的尺度在变化, 但是一特征为中心的区域大小不发生改变, 这可以进一步帮助模型提取到从局部到全局的信息。

### 2.2.3 数据降维

假设我们选择了  $N_{fp}$  个关键点，那么最终每个样本特征的维度是  $N_{fp} \times S^2 \times L$  维，再考虑到样本的数目，我们需要对特征进行降维。我们选择方法是 PCA。

## 3 实验与评估

### 3.1 具体实现

在2一章中，我们介绍了模型的工作原理和部分实验细节，这里将这些内容总结一下并补充一些细节。

首先，对于面部关键点检测，我们使用的超参数如??所示。

P	T	F	K	$N_{aug}$	$\beta$	$\kappa$
400	10	5	500	20	1000	0.3 TupleDist

## 4 讨论

YOLO 是一个很好的上手项目，首先他的结构较为简单，但是有包含目前主流的多种结构，比如跳跃链接，批归一化层等；另一方面 YOLO 从最初的版本一直发展到 YOLOv3，吸纳了很多计算机视觉领域的新的方法，比如使用锚框，使用多尺度的图像金字塔。因此能够较为全面地对计算机视觉领域主流的方法进行一个简要的回顾，也能很好的学习计算视觉领域常见的网络结构。虽然结构简单，但是 YOLO 中也需要手动对一些新定义的功能层进行实现，比如 shortcut, route, 以及 yolo 层，这也能帮助了解利用 `nn.Module` 创建新的类的过程，以及更加深入的了解了 `nn.ModuleList`, `nn.Sequential` 等常见容器的使用和适用场景。

另一方面，YOLO 不需要很高强度的训练过程。作者在网站上已经提供了训练好的参数，可以在完成网络框架后，立即进行测试和观察效果。最后，YOLO 能够给实践者一个很好的反馈，因为任务的结果仍然是一个图像，可视化性。

在本项目中，还可以对利用 `conda` 管理环境和包的管理进行练习。在这个项目的实践中，我还是学习到了很多。

## 参考文献

- [1] PyTorch 中的 ModuleList 和 Sequential: 区别和使用场景。  
<https://zhuanlan.zhihu.com/p/64990232>
- [2] 目标检测 |YOLO 原理与实现。 <https://zhuanlan.zhihu.com/p/32525231>
- [3] Tutorial on implementing YOLO v3 from scratch in PyTorch: Part 1.  
<https://medium.com/paperspace/tutorial-on-implementing-yolo-v3-from-scratch-in-pytorch-part-1-a0054d38ec78>

- [4] You Only Look Once: Unified, Real-Time Object Detection. Joseph Redmon and Santosh Divvala and Ross Girshick and Ali Farhadi. 2015. 1506.02640 arXiv.
- [5] YOLO9000: Better, Faster, Stronger. Joseph Redmon and Ali Farhadi. 2016. 1612.08242 arXiv.
- [6] YOLOv3: An Incremental Improvement. Joseph Redmon and Ali Farhadi. 2018. 1804.02767 arXiv.