

标题:基于 Pytorch 的 Facenet 和 RetinaFace 两种算法比较。

1 课设简介

面部识别是计算机视觉中的一个重要领域，在安全、社交媒体和个人电子产品中有着广泛的应用。本报告探讨了面部识别中两个关键算法的基于 Python 的实现——用于特征提取的 Facenet 和用于面部检测的 RetinaFace，在 PyTorch 机器学习框架中实现。Facenet 侧重于从人脸中提取丰富的特征以进行识别和验证，而 RetinaFace 侧重于准确和稳健的人脸检测，包括边界框回归和面部标志检测。

这些最先进的算法共同提供了一个强大而高效的系统，用于识别和区分图像中的人脸。通过对算法的 `detect_image` 进行比较分析，比较两种算法对不同场景图片的识别准确率。在本次算法比较中，RetinaFace 明显优于 Facenet 算法。

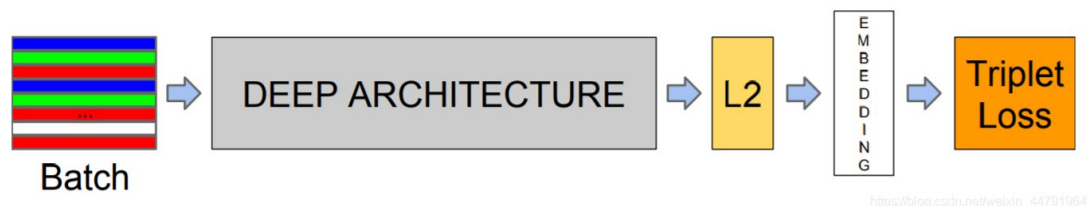
2 关键代码

1 Facenet

谷歌人脸识别算法，发表于 CVPR 2015，利用相同人脸在不同角度等姿态的照片下有高内聚性，不同人脸有低耦合性，提出使用 `cnn + triplet mining` 方法，在 LFW 数据集上准确度达到 99.63%。

通过 CNN 将人脸映射到欧式空间的特征向量上，实质上：不同图片人脸特征的距离较大；通过相同个体的人脸的距离，总是小于不同个体的人脸这一先验知识训练网络。

测试时只需要计算人脸特征 EMBEDDING，然后计算距离使用阈值即可判定两张人脸照片是否属于相同的个体。



源码: <https://github.com/bubbliiiing/facenet-pytorch>

2 RetinaFace

RetinaFace 是今年(2019 年)5 月份出现的人脸检测算法, 当时取得了 state-of-the-art。虽然在未受控制的人脸检测方面取得了巨大进步, 但野外准确有效的面部定位仍然是一个开放的挑战。这篇文章提出了一个强大的单阶段人脸检测器, 名为 RetinaFace, 它利用联合监督和自我监督的多任务学习, 在各种人脸尺度上执行像素方面的人脸定位。具体来说, 我们在以下五个方面做出了贡献:

- (1) 我们在 WIDER FACE 数据集上手动注释五个面部标志, 并在这个额外的监督信号的帮助下观察硬面检测的重要改进。
- (2) 我们进一步增加了一个自监督网格解码器分支, 用于与现有的受控分支并行地预测像素三维形状的面部信息。
- (3) 在 WIDER FACE 硬测试装置上, RetinaFace 的性能优于现有技术平均预测 (AP) 1.1% (达到 AP 等于 91.4%)。
- (4) 在 IJB-C 测试集上, RetinaFace 使最先进的方法 (ArcFace) 能够改善他们在面部验证中的结果 ($FAR = 1e-6$ 的 $TAR = 89.59\%$)。
- (5) 通过采用轻量级骨干网络, RetinaFace 可以在单个 CPU 内核上实时运行, 以实现 VGA 分辨率的显示。

源码: <https://github.com/bubbliiiing/retinaface-pytorch>

3 实验设置

实验使用两种不同的人脸识别和检测算法进行：**Facenet** 和 **RetinaFace**。这两种算法都是使用 **PyTorch** 库实现的,对于两种算法的定量分析，通过对算法的 `detect_image` 进行比较分析，比较两种算法对不同场景图片的识别准确率。

FaceNet:

Facenet 算法处理人脸识别任务。它处理两张图像，并通过一系列步骤计算出一个距离度量，该距离度量指示图像中人脸的相似性。此输出可用于人脸验证和识别任务。

首先，该函数 `detect_image` 将两张图像作为输入。预处理从根据 `input_shape` 参数调整图像大小开始，确保尺寸符合模型要求。`letterbox_image` 在这里，可以控制参数的使用，当设置为 `true` 时，在调整大小时保持图像的原始纵横比，必要时添加填充。

调整大小后，图像将转换为 **PyTorch** 张量，并进一步准备输入到模型中。这涉及维度的标准化和转置以适应模型的预期输入格式。如果可用并指定，**GPU** 加速用于计算。

然后将处理过的图像传递给 **Facenet** 模型。该模型为每个输入图像输出一个 128 维的特征向量，代表面部特征。在这两个向量之间计算 **L2** 范数（欧氏距离），提供两个面之间的相似性度量。该距离随后由函数返回并用于人脸比较。

RetinaFace:

RetinaFace 执行人脸检测任务。它识别和定位图像中的人脸，并提供关键的面部标志。

RetinaFace 的函数 `detect_image` 将一张图像作为输入。它首先创建用于绘制检测结果的输入图像的副本。然后将图像转换为 `numpy` 数组，并存储尺寸以供以后重新缩放模型输出时使用。与 `Facenet` 一样，图像预处理涉及调整大小和规范化，并可选择 `letterbox_image` 保持纵横比。

处理后的图像被输入 `RetinaFace` 模型，该模型输出面部位置、置信度分数和面部标志的预测张量。使用存储的图像尺寸将这些预测解码为原始图像比例中的边界框、置信度分数和地标坐标。

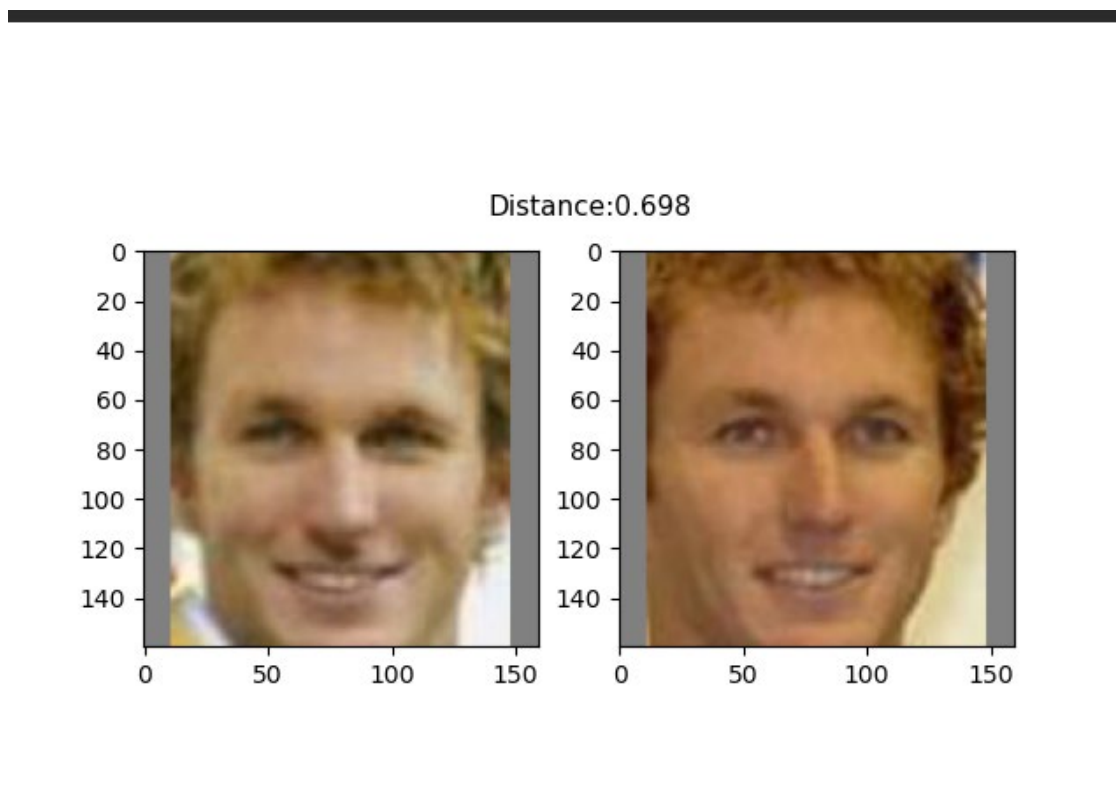
对解码输出进行非极大值抑制以进行滤波器重叠检测。然后将最终的边界框和地标叠加到原始图像副本上。

数据集使用详情：

这些实验中使用的数据集由 `Facenet` 的成对图像和 `RetinaFace` 的单个图像组成。理想情况下，数据集应该是多样化的，涵盖不同的光照条件、角度、面部表情和人口统计差异，以实现稳健和可推广的模型性能。

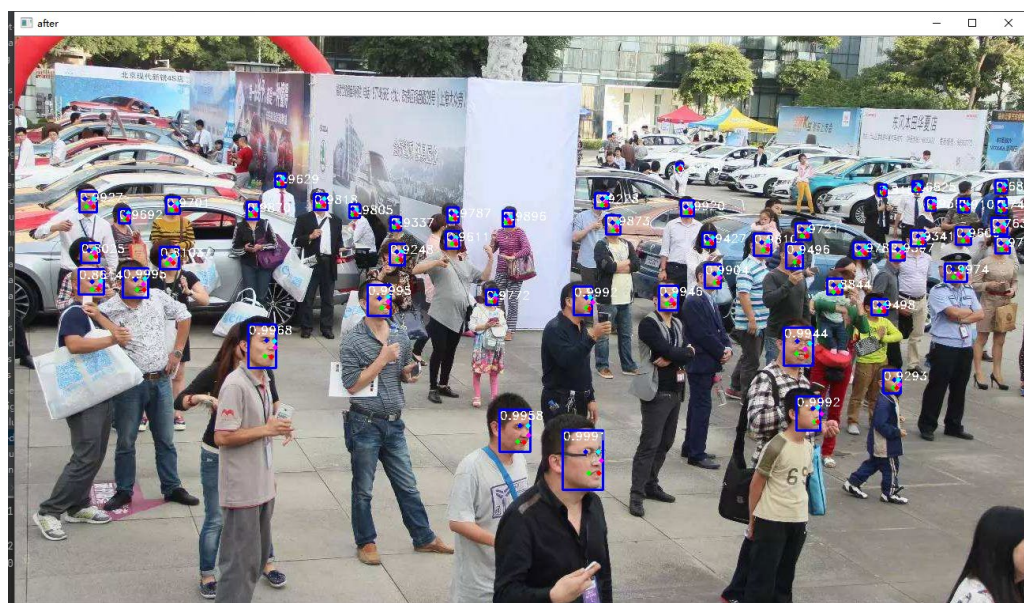
4 视觉和定量实验结果的显示和解释

`Facenet` 通过显示输入图像和计算出的人脸相似性度量来可视化结果。对于 `Facenet`，同一个人的人脸嵌入之间的平均距离小表明面部识别性能高，而不同人的面部嵌入之间的平均距离大则表明具有良好的辨别能力。



RetinaFace 将检测到的人脸边界框和地标叠加在输入图像上，有效地将检测和地标预测可视化。RetinaFace 检测中的高平均置信度分数表明人脸检测具有良好的性能，而这些分数的低方差则表明一致性。

在这两种情况下，视觉反馈都是理解和验证模型输出的直观方式



5 实验结果原理分析

Facenet

Facenet 性能背后的原理取决于它从输入面部图像中提取高质量特征的能力，它通过将它们传递给深度卷积神经网络来实现。然后将这些特征映射到欧几里德空间。Facenet 方法的创新之处在于，它不仅仅是识别不同的特征，还旨在确保特征之间的欧几里德距离代表面部相似性。换句话说，与不同人的图像相比，同一个人的两个不同图像在该空间中的特征之间的距离应该更小。

在检查 Facenet 的输出时，两幅图像的特征向量之间的 L2 范数或欧几里德距离时来作为判断两幅图是否相似的依据。如果这个距离很小，说明两张图片是同一个人；如果很大，则表明他们可能是不同的人。从上图的结果来看，这两个人的不同值达到了 0.68，可以判断为不相似。但是本质上这两张图片都是属于同一个人，所以 Facenet 无法取得很好的效果。

RetinaFace

RetinaFace 的操作原理有些不同；它的重点是检测图像中的人脸并准确映射面部特征。它利用特征金字塔网络主干结合多任务分支进行联合人脸检测和地标定位。

由于使用锚机制和金字塔特征，RetinaFace 的优势在于它能够处理各种尺度、姿势和表情。它检测小脸和大脸的能力来自于使用特征金字塔网络。然后它可以使用该网络的不同特征图级别来预测不同比例的人脸。

在多个尺度和纵横比下使用锚点有助于模型检测图像中不同方向和尺度的人脸。联合面部检测和面部标志定位使 RetinaFace 不仅可以检测面部，还可以准确估计面部标志，例如眼睛、鼻子和嘴巴的位置。

RetinaFace 模型对于上面图中中的各个人脸都做到了有效识别，基本识别的准确率都维持在了 90%以上，可以看出有极佳的效果。

6 结论

总之，该实验展示了 Facenet 和 RetinaFace 算法在面部识别和检测任务中的能力。通过对实验结果的可视化和定量分析，RetinaFace 明显优于 Facenet 算法。。它们的性能在很大程度上取决于训练数据的质量和多样性，而它们的基本原理使它们能够从原始图像数据中提取有意义的信息，这是它们成功不可或缺的一部分。同样重要的是要注意，虽然这两种算法都很强大，但它们的效用将取决于手头任务的具体要求和约束，例如计算资源、实时处理需求和精度要求。