目 录

摘　要

本课程项目旨在完成一个书籍推荐系统, 通过用户的脸部识别出用户的性别和年龄, 根据性别年龄分类与书籍的映射关系推荐书籍, 并完成了一个与用户交互的 demo. 本课程项目报告主要包括介绍问题背景和用户分类、数据集获取、人脸检测、人脸校准、深度学习方法、HOG+SVM 方法、方法对比和 demo.

问题背景介绍了课程项目的动机、研究方法整体思路框架图、书籍推荐系统使用逻辑和可拓展的场景. 用户性别分为男性和女性, 年龄分为婴儿、儿童、青少年、青年、中年和老年.

由于应用场景为中国人, 与公开数据集不符, 因此我们通过对视觉中国和 Veer 两个网址的爬虫, 通过很标准的爬取和命名方式, 制作了自己的中国人脸数据集, 共 78627 张图像.

项目通过 CPU 计算速度、是否正脸、检测尺寸等方面考虑选用了合适的检测算法.

对于人脸校准, 调研了多种方法, 结合 DEX[6] 的旋转算法和 DeepFace[5] 基于人脸关键点的方法提出了自己原创的人脸校准方法——基于人脸关键点的校准算法, 并通过实验展示了其合理性, 且选用为本课程项目中的人脸校准算法.

识别阶段首先选用了深度学习方法, 采用了 AlexNet 和 VGG16 两种不同的网络, 改变其全连接层适应性别和年龄分类, 根据自己的理解, 从损失函数、预训练、不同的学习率、归一化、dropout、数据增强等 7 个方面说明其原理, 加入大量自己对技巧的思考和感受.

深度学习方法实验, 展示了实验过程中的损失函数和准确率的变化, 从训练时间、模型大小、训练集准确率、测试集准确率和单张推断时间对比了两种不同的网络结构, 并说明了以上指标对书籍推荐系统部署的影响. 以 VGG16, 尤其对年龄识别的混淆矩阵进行了详细分析, 说明了深度学习算法模型在交叉熵损失函数的情况下学习到年龄渐变规律的结论. 最终在验证集上得到性别分类 95.89% 和年龄分类 77.28% 的准确率.

作为对比, 我们选用 HOG+SVM 传统方法进行性别和年龄分类, 我们回顾了 HOG 特征的提取原理和 SVM 在核函数选择、最优超参数寻优和多分类算法上的常用方法. 在实验阶段, 我们实现了 HOG 特征的提取, 使用 OpenCV 的 SVM 分类器对其进行分类. 在已知传统算法精度很难与深度学习方法媲美下, 不同于对特征提取进行改进, 我们主要研究 SVM 分类器算法寻找最优核函数和超参数. 针对未来得及实现的想法, 我们在改进展望中提出.

在方法对比阶段, 我们从特征提取、训练过程 (硬件需求、训练时间、端对端、调参难度、分类器和数据集数量要求)、训练结果 (训练集和测试集上的精度、对应硬件上的单张推断时间和模型大小) 和可解释性四个方法进行对比. 这些对比的论述是基于本次课程作业完成过程和结果的很个人的观点.

最后, 我们给出最终的书籍推荐系统 demo. 并对本次课程致谢.

需要特别注意的两点是:

1. 课程项目报告有很大一部分内容是我对深度学习、机器学习、计算机视觉算法的感悟, 这些感悟来源于论文阅读、课程、收集的资讯和自己在研究过程中的感悟. 本次课程给了我很大启发, 课程项目也验证了很多想法, 带给了我很多新的感受.
2. 课程项目报告除去人脸关键点图引用 openpose 的 github, 其余图表均为自己制作.

1 问题背景

性别和年龄是人的基本属性, 有广泛的应用场景, 例如书籍、衣服推荐, 场所 (例如网吧) 人员限制等. 本课程报告的目标是通过单张图像预测图像中人的性别和年龄, 将应用场景设定为书店的书籍推荐系统. 即通过调用摄像头获取用户的人脸图像, 通过识别算法识别用户的性别和年龄分类, 最终根据性别年龄分类的预设的映射关系, 为用户推荐适合自己的书籍, 定向投放书籍广告, 引起用户兴趣, 提高特定书籍的销售量.

1.1 思路来源

近些年, 人脸识别领域蓬勃发展. 但是大多数人们对人脸识别的认识还在人脸匹配, 因为其在门禁、手机屏幕解锁和上班打卡上有广泛的应用. 其实, 人脸识别还有性别、年龄识别等任务. 在本次课程作业中, 我主要完成人的性别、年龄的识别, 灵感来源于曾使用旷视 (face++) 的 detect API, 其可以实现人脸的性别、年龄、表情、人种、眼镜等多个属性的识别. 我在使用中发现经常会有识别失败的案例, 处于好奇, 我希望自己实现一个版本的 detect API, 通过自己的实现, 探索该领域的难点, 做出自己的评价和改进.

第二个动机来源于我对推荐系统的认识, 曾在 2017 年实习的时候参加过移动公司的套餐推荐算法的研究, 在实习中将性别和年龄作为很重要的用户画像刻画的属性. 因为不同的性别对世界的认知和承担的社会工作往往不同, 用户喜好和想法也存在明显的不同年龄段的差异性和相同年龄段的聚类现象. 我们在日常生活中将其称为” 男女有别” 和” 代沟”. 以书籍为例, 儿童肯定适合看童话而非青少年和青年喜欢的东野圭吾, 政治类书籍男性受众更多, 女性可能对盆栽园艺等更有兴趣. 类似的例子我们在生活中经历了太多, 我们所用的大多数 APP, 几乎都会让我们填写性别和年龄信息方便初始化推荐, 甚至你走进一家服饰店, 导购员也是根据性别年龄来给你推荐衣服的.

那么, 如果将两者结合, 做一个基于摄像头视觉的性别年龄识别算法, 应用该算法到基于性别年龄分类的推荐系统呢?

1.2 视觉任务描述

与单一的视觉任务不同, 我的课程项目将实现完整的视觉算法推荐系统, 因此涉及多方面的视觉任务. 由于一学期时间有限, 像人脸检测等任务直接使用 python 的库, 我们将主要精力放在数据集的制作、两种人脸年龄性别识别算法和两种识别算法比较上.

本课程项目的流程示意图见1.

Step 1. **爬虫阶段**. 为加深对数据集的理解, 也为适应国内部署的应用场景, 我通过爬虫制作了自己的中国人脸数据集. 首先选取合适图像网址, 这里选择的是视觉中国 (https://www.vcg.com/) 和 veer(https://www.veer.com/); 然后选定图像筛选条件, 为了获取标

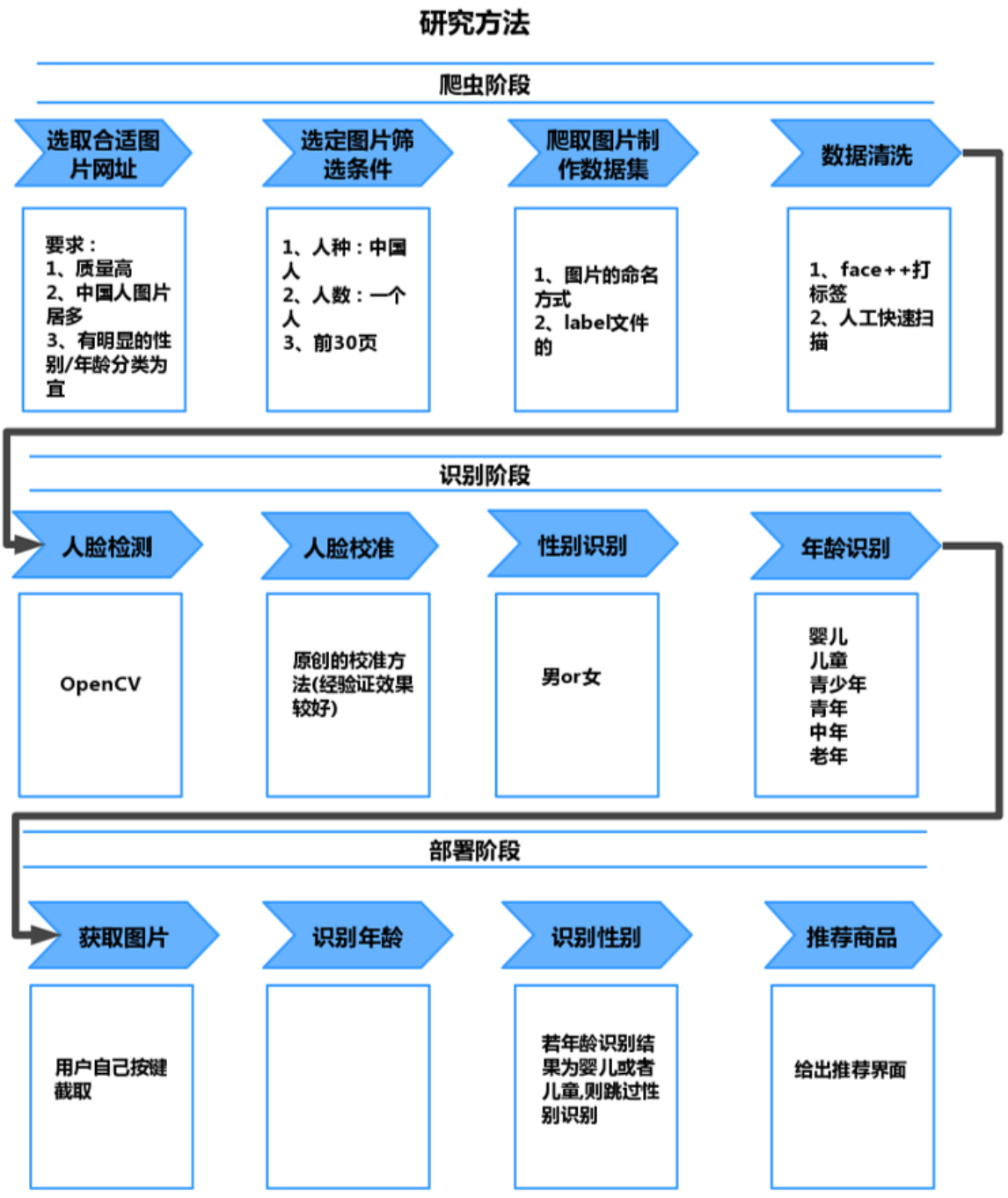

图 1: 视觉任务描述流程图

签方便, 只爬取图像中仅有一张人脸的图像; 前期工作结束后后使用 python 对图像进行爬虫; 最后对数据进行清洗并用 face++ API 对标签进行确认, 获取高质量的中国人脸数据集.

Step 2. **识别阶段**. 该阶段任务为本课程项目最重点的阶段, 首先检测图像中的人脸; 然后使用**自己原创的方法**对人脸进行校准; 再通过两种算法对性别和年龄进行识别, 对两种算法分别进行算法改进, 并对两种算法进行横向比较.

Step 3. **部署阶段**. 最后给出一个部署阶段的 demo 应用, 即用户交互获取图像、识别年龄、识别性别和推荐书籍, 作为课程项目, 仅展示 demo 以及提出部署阶段需要考虑的问题. 实际并没有渠道进行部署.

1.3 应用场合详述

1.3.1 基本应用场景——书籍推荐系统

本课程项目目标, 也是已经可以实现的是基本应用场景——书籍推荐系统. 该系统需要主机 (cpu/gpu+windows/linux+python 环境), 摄像头和显示屏 (按钮或触屏). 安置在书店入口处, 屏幕用于用户交互, 摄像头用于获取用户头像, 主机用于推断. 预想的应用过程为 (其实可以画个流程图):

Step 1. 用户出于猎奇心理走向屏幕, 查看系统说明和操作指南.

Step 2. 用户出于兴趣或者好奇心按下拍摄按钮, 摄像头拍摄用户图像.

Step 3. 算法进行人脸检测, 若因光线不好等原因检测不到人脸, 提醒用户重新拍一次; 若拍摄到多张人脸, 基于用户离摄像头最近的假设, 选取人脸框最大的人脸检测.

Step 4. 算法进行人脸校准, 检测性别和年龄, 根据映射关系推断出应该推荐的书籍.

Step 5. 单页只显示一本推荐的书籍及介绍, 可投入吸引用户的广告, 用户可按键或滑动翻页.

Step 6. 用户按键退出推荐界面或一段时间后自动退出或新用户退出上一个推荐界面.

1.3.2 拓展应用场景——基于视觉的推荐系统

本课程项目可做多种拓展, 基于视觉的人的属性不止性别、年龄, 常用的属性还有表情、衣服、眼镜、人种甚至身高等. 属性越全面, 对用户的刻画越准确, 可用来推荐的商品也越多, 推荐也更加精准. 一学期时间有限, 但是我的预想中是要对人的性别、年龄、表情、衣服进行识别, 衣服又分为很多细节例如上半身类型、下半身类型等全局的属性和是否有拉链是否有纽扣等局部的数据, 一个综合以上属性的基于视觉的推荐系统可以完成以下类型的商品推荐.

a. **扩展版的书籍推荐系统**. 除了根据性别、年龄外, 还可以通过眼镜和衣服大致推断用户的文化水平, 可以通过是否是西装来推断工作类型, 例如对于商务人士可能倾向于推荐投资理财书籍, 可以通过表情来判断用户实时的心情, 根据不同的心情推荐不同的书籍等.

b. **服装店的衣服推荐系统**. 根据用户的性别年龄确定基本的推荐板式, 根据用户所穿衣服的细节推荐同类的衣服, 例如: 用户穿的是有拉链, 短袖, 条纹图案, 衬衫, 那么给用户推荐带有拉链的短袖的条纹图案的衬衫. 或者挖掘更加深层次的语义关系, 按时尚逻辑进行推荐.

c. **推荐系统用来做调研**. 可以放在商场店铺入口, 对密集人群 (要加上密集人群的算法) 进行属性统计, 还是以书店为例, 统计每天进入书店的人的性别和年龄, 以此指定针对某个年龄段和性别书籍的进货方案, 以及对人数较少的年龄段和性别进行定点广告投放.

2 用户分类

在课程项目中, 将应用场景限定在书籍推荐系统, 考虑到对于婴儿和儿童, 区分性别推荐书籍意义不大, 也考虑到制作数据集时在要爬取的网址中, 婴儿和儿童的分类存在重叠现象, 即婴儿儿童标签不准确. 因此, 对婴儿儿童不区分性别.

最终的年龄性别分类如表1.

表 1: 用户标签分类

年龄分类	性别	描述
baby	gender	婴儿,gender 表示不区分性别
child	gender	儿童, 不区分性别
early_youth	male/female	男青少年/女青少年.
youth	male/female	男青年/女青年
middle_age	male/female	男中年/女中年
older	male/female	男老年/女老年

3 数据集获取

3.1 为什么要制作自己的数据集

在制作自己的数据集之前, 我找到了人脸性别和年龄领域常用的数据集 IMDB-WIKI 数据集[7]. 决定使用自己的数据集原因有两个, 一个是发现 IMDB-WIKI 数据集中白人居多, 黄种人尤其是中国人很少, 与我们的应用场景不同, 预计训练好模型在中国人脸上测试准确率较低 (例如白人的青少年给人感觉明显比国内青少年成熟); 一个是在做了初步的训练后发现数据集

质量很差, 这与其数据集获取过程有关.IMDB-WIKI 是从 IMDB 和维基百科网站上爬取的明星图像, 性别资料参考明星的性别, 年龄资料根据拍摄图片日期减去明星生日. 经过详细研究, 我发现这种方式存在以下 2 个问题:

a. 明星公开的照片大多化妆, 因此与实际年龄相比图像年龄偏小;

b. 明星多为合影方式出现, 数据集制作过程中若检测到多张人脸, 默认选取 face score 最高的脸作为标签的脸, 这样造成很多错误;

因此, 我开始制作自己的中国人脸数据集.

3.2 数据集制作过程

我主要爬取了视觉中国和 veer 两个网址上的中国人图片. 选取这两个网址的原因是图像质量较高, 国内网址可以找到大量的中国人图像, 有明确的性别年龄分组, 可以获取初步的年龄段和性别标签. 需要注意的一点是, 我在爬取这两个网址时视觉中国版权问题还未发生, 需要说明的一点是: **我爬取的图像仅做课程作业研究使用, 不做任何商业用途**.

为了避免 IMDB-WIKI 数据集的第 2 个问题, 我将图片中的人数检索条件设定为" 一个人", 最大程度确保获取的人脸标签即为检索标签.

我使用 python 对图像进行爬取, 主要使用到了 request 库和正则表达式库, 直接按照关键词搜索解析源网页, 使用正则表达式库找到图像网址, 保存图像. 需要注意的两点是:

a. 关键词检索往往是相关度排序, 因此图像页数越往后, 相关度越差, 会出现多人图像或者年龄明显错误图像, 因此对于超过 30 页 (3000 张) 的图像类别, 只截取前 30 页.

b. 图像命名方式为: 年龄分类标签 + 性别分类标签 + 数据来源 + 原网页页码 + 原网页页码中的图片次序. 这样命名是为了方便在出现断网或者其它错误时可以找到出错的位置修正或者继续爬取.(然后后面视觉中国关闭了一个月, 恢复后原网页被更新掉了.)

3.3 数据集制作结果

爬取的图像举例见图2.

最终, 共爬取 78627 张图像, 制作成中国人脸数据集, 数据集的各个类别图像数目统计见图3.

通过示例图2和统计图3可以看出, 从视觉中国和 VEER 爬取的图像有以下特点:

a. 风格偏艺术照, 非生活照, 这是这个数据集最大的问题, 我们不能确保在应用场景下的光线等条件有艺术照那样好. 但是我认为, 如果我们真的到了应用阶段, 那需要对特定场景下采取的数据做 fine-tune.

图 2: 爬取数据集举例

b. 青少年与青年很相似, 肉眼难以区分青少年和青年. 这是因为不管是在视觉中国还是在 VEER 中, 或者是其它高清图像网址中, 艺术照的青少年偏年龄大, 青年偏年轻. 这是导致识别过程中很容易将其识别错误的一个重要原因.

4 人脸检测

要实现人脸的性别和年龄识别, 首先是人脸检测, 即将人脸从图像中截取 (crop) 出来. 这不是本课程项目的主要任务, 属于人脸性别和年龄识别的下游任务. 在本项目报告中, 不详细比较各种人脸检测算法, 仅给出几种常见的检测原理及其**优缺点**, 阐述选用人脸检测方法的理由.

4.1 系统应用需求

对于预想的书籍推荐系统应用场景, 对人脸检测要求并不是特别高, 具体需求如下:

1. 正脸占大多数, 侧脸很少. 应用场景为用户在摄像头前交互拍摄图像, 相当于摄像头设置在屏幕上 (可以参照笔记本电脑的摄像头), 因此拍摄的多为正脸, 只有在用户按完拍摄按钮且把人脸移开时会出现侧脸.

2. 遮挡较少. 大部分用户都不会在遮挡的场景下使用设备. 即用户在已知要进行人脸识别时, 不会有意戴口罩等.

3. 尺度相对固定. 大部分用户距离屏幕的距离应该是相近的, 不会存在特别小的人脸.

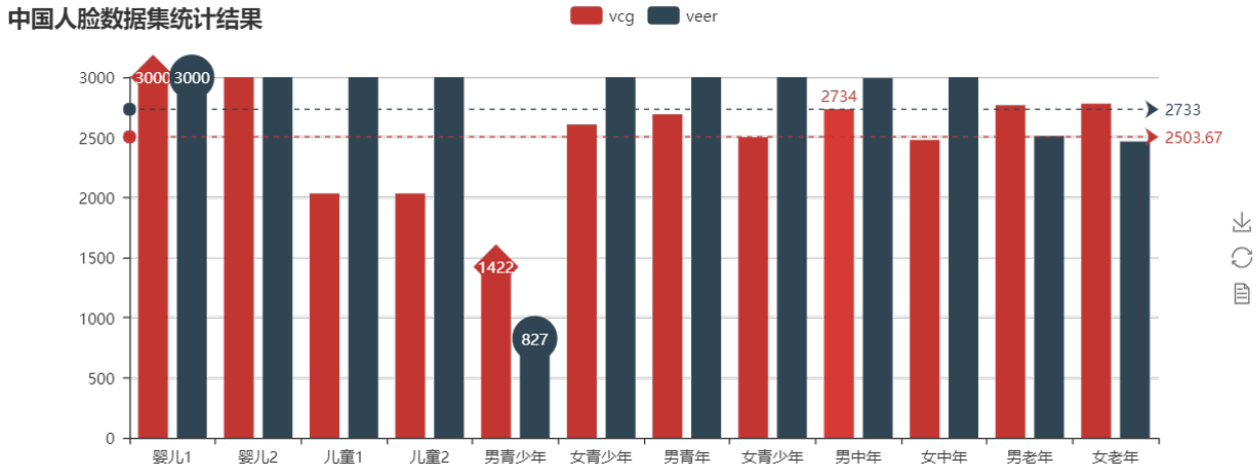

图 3: 爬取数据集统计

4. 人脸数不会太多. 在书店门口的设备不会有多个人一起使用, 由于屏幕朝上, 不会拍摄到过得过往行人. 仅在结伴时会出现两三人情况.

5. 要求延迟低. 人脸识别会占据系统使用的大部分时间, 因此对于检测, 我们要求延迟低, 即计算复杂度小, 运行速度快. 这决定了用户体验.

综合以上需求考虑, 我们需要一款适合正脸检测、人脸尺寸大小相对固定、最主要是运行速度快的人脸检测器.

4.2 常见的人脸检测算法

虽然人脸检测效果会影响人脸性别和年龄识别精度, 但是在本课程作业中我们仅考虑算法的优劣和能够正确使用算法, 不过, 对算法的原理做简要的理解也有助于我们对识别结果的分析.

这里, 我们主要找到了 4 种有较好开源实现的方法:**OpenCV 中的 Haar Cascade 人脸检测器、基于深度学习的 OpenCV 人脸检测器、dlib 中的 HoG + SVM 人脸检测器和 dlib 中的 CNN 人脸检测器**.

由于我们的主要任务是分类而非检测, 因此只需要按照应用需求选用合适的人脸检测器获取高质量的人脸即可. 作为课程项目, 我希望可以尽可能考虑全面, 做一个整体的规划 (虽然这个规划短期内没有条件实现, 却是一个很好的锻炼的机会). 在对比这几种人脸检测算法时, 我主要从使用场景、运行速度两个方面考虑, 而系统的设置为用户在设备前自行截取脸部, 不存在密集的人脸、严重侧脸、尺度变化很大等问题. 因此, 对以上四种人脸算法对比, 我们有以下结论:

方法 1. OpenCV 中的 Haar Cascade 人脸检测器: 适合检测正脸, 几乎可以实时工作, 对遮挡的检测效果不好.

方法 2. 基于深度学习的 OpenCV 人脸检测器: 较为准确, 实时性较好, 适合于不同方向, 不同尺度的图像.

方法 3. dlib 中的 HoG + SVM 人脸检测器: 适合检测正脸,CPU 上毫秒级, 不能检测小尺寸的脸.

方法 4. dlib 中的 CNN 人脸检测器: 可以检测非正脸,GPU 上毫秒级, 可以检测不同方向的脸.

对比需求和算法的特点, 由于我们的 demo 希望在本地电脑上使用 cpu 运行, 且书籍推荐系统存在用户交互过程, 用户与屏幕之间的距离相对固定, 人脸较大, 正脸偏多, 因此采用 dlib 中的 HoG + SVM 人脸检测器.

5 人脸校准

人脸校准可以有效地提高人脸属性识别的精度[6]. 人脸校准是把角度不是很正的人脸对齐的过程. 虽然我们的应用场景中侧脸很少, 但是难免用户会出现歪头的场景. 为了提高人脸性别和年龄识别的精度, 这里我们对人脸进行校准操作.

我调研了多种人脸校准方法, 最后**提出了自己的检测方法——一种基于人脸关键点的易操作的效果较好的方法**. 在课程项目中采用自己的方法, 是对自己本学期在课堂上与老师讨论各种思路学习到东西的总结, 希望在以后的科研路上可以多多创新, 做未来视觉领域的先驱.

5.1 DEX 的旋转算法

DEX[6] 是在本课程项目开始时主要参考的一篇论文, 其启发了我要做人脸校准. 我将文献提出的人脸校准方法如下:

step 1. 对原图进行以步长为 5° 度的旋转, 旋转范围为 $-60°$ 到 $60°$.

step 2. 根据正前向脸的位置, 采纳最高检测分数的脸作为正脸.

这实际上是一种遍历的方法, 本课程项目在实验之后选择放弃这种笨重的方法. 但是该方法给了**旋转图像以校准人脸的基本思路**.

5.2 DeepFace

DeepFace[5] 是深度学习用在人脸识别领域的奠基之作, 其用到的人脸校准方法较为复杂, 采用 3D 对齐的方式.

其基本步骤总结为:

step 1. 检测出人脸及对应的 6 个基本点;

step 2. 对人脸进行二维对齐;

step 3. 使用狄罗尼三角划分在 2D 人脸上划分出 67 个关键点，并在边缘处采用添加三角形的方式避免不连续;

step 4. 通过 3D 模型产生的 67 个基准点进行分段映射使人脸变弯曲，对人脸进行对齐处理;

step 5. 处理生成的 2D 人脸和 3D 人脸.

这种复杂的方法超出了我的实现能力范围, 因此没有采用. 但其**启发了我做人脸校准时可以根据人脸关键点进行校准**.

5.3 基于人脸关键点的原创算法

常见的人脸关键点检测有 dlib 的 68 人脸关键点和 openpose 的 70 人脸关键点, 以 openpose 为例 (https://github.com/CMU-Perceptual-Computing-Lab/openpose), 其 70 关键点分布如图4.

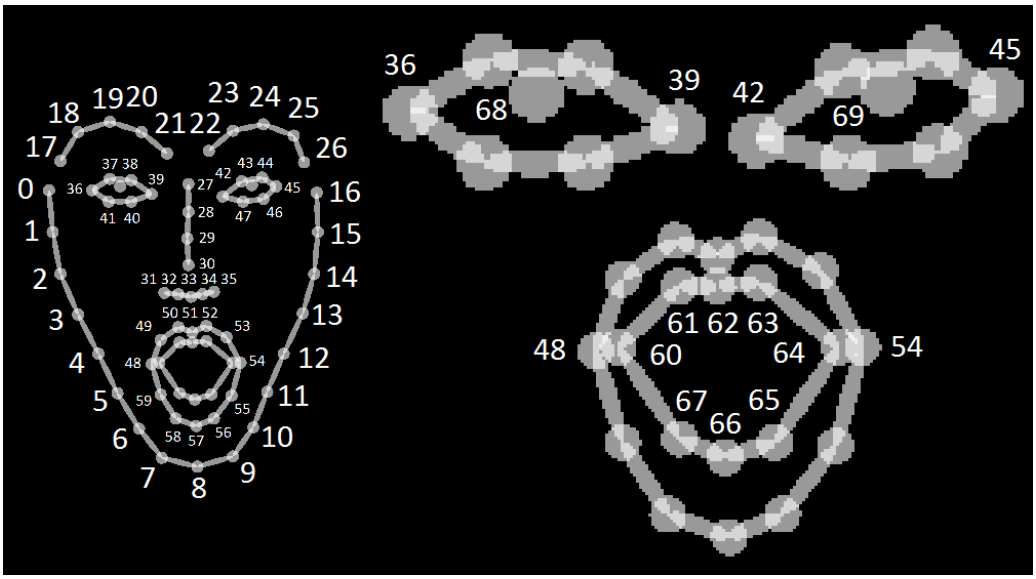

图 4: openpose 人脸 70 关键点检测

通过人脸关键点我们发现, 人的鼻梁 (点 27,28,29,30) 基本是人脸的对称轴, 我们可以通过计算点 27 与点 30 之间连线与图像垂直方向逆时针方向的夹角 θ , 对人脸进行顺时针旋转 θ, 即可完成人脸校准. 检测校准示意图见5.

该方法实现起来比较简单, 但还是有一些细节需要说明, 因此将其写成算法描述如算法1: 特别强调的一点是, 这里是对原图进行旋转, 对检测的人脸框选外接矩形 (即选择包含原人脸框的最小的边为水平和垂直的矩形), 再截取人脸. 而非对人脸截取后进行旋转, 这是因为若采用后者 (实际上我也采用过), 会发现非原人脸框部分没有像素点, 这样对识别会产生困扰, 而

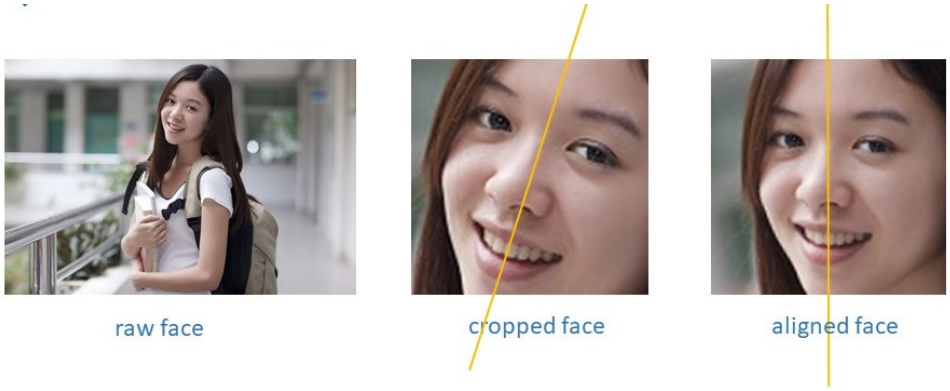

图 5: **原创的**检测和校准算法示意图

Data: 原图像 + 检测的人脸框
Result: 校准后的人脸
begin
Step1 检测人脸关键点;
Step2 若检测到点 27 和 30, 计算点 27 与点 30 之间连线与图像垂直方向逆时针方向的夹角 θ, 否则输出原检测到的图像;
Step3 对检测到的人脸框在**原图上**顺时针旋转 θ, 这样保证不会出现缺失背景;
Step4 对旋转后的人脸框取外接矩形, 即取横纵坐标的极大极小值, 生成新的校准后的人脸框;
Step5 按照校准后的人脸框截取原图, 输出校准后的人脸.
end

对原图进行旋转, 虽然会带些非原来人脸框的边缘, 但是有文献表示这种边缘可以增加人脸识别的鲁棒性[6].

图2经过检测和校准后的人脸如图6, 其中**第二行第二列的人脸 (即红圈圈出的人脸)** 因检测不到鼻子关键点被换为新的人脸, 这些校准后的人脸将被用来识别和分类.

5.4 基于人脸关键点的原创算法讨论

由于是原创算法, 因此我对算法体会颇深, 有许多改进的思路, 囿于时间有限, 没法一一实现, 仍想在这里讨论.

1. **关键点冗余**. 通过 openpose 或者 dlib 检测到的关键点有 70 个/68 个, 而我们原创的算法只使用了 2 个, 这样导致了计算的冗余. 改进思路是使用其它更加简单的人脸关键点检测算法, 例如只检测人左右眼和嘴巴三个关键点, 选择嘴巴与左右眼连线的垂线作为垂直方向进行旋转.

2. **存在校准失败的情况**. 人脸关键点检测有时并不会完整地检测出所有关键点, 若关键点 27 或者点 30 检测不到, 目前算法是不做校准, 直接输出检测图像. 这样会导致校准失败的情

图 6: 图2经检测和校准后的人脸

况. 解决办法是若原算法校准失败, 即图4中点 27 或者 30 检测不到, 使用点 28 或 29 代替缺失的点. 当然, 如果鼻子关键点完全检测不到, 使用眼睛、嘴巴等水平方向的连线也可以.

3. **对设计创新算法的体会**. 之所以会想出这样的算法, 是因为在一次偶然的机会看到图4, 直观感觉鼻子很直, 又自己测试了几张图像, 发现确实关键点检测到的鼻子都比较垂直. 在细读文献 [DEX] 时发现其用到了旋转, 不过用的是一种遍历的方法, 个人觉得效率低下, 可以用关键点辅助旋转. 综上, 提出算法需要直观感受, 也需要前人工作的启发. 因此在接下来的科研生活中, 我会多多读文献, 多多独立思考, 结合实际, 勇于提出自己的想法, 并通过实验验证它, 这也是这门课对我最大的启发.

6 方法一: 深度学习方法原理

通过对人脸的检测和校准, 我们已经得到一个高质量的人脸数据集. 文献 [DXP] 在 2015 年首次提出了可以用 CNN 的方法实现对人脸年龄的端对端 (end-to-end) 训练. 因此, 针对深度学习方法, 我们主要使用经典的 AlexNet 和 VGG16, 使用其 CNN 网络结构实现 feature map 的提取, 保留其部分全连接层, 将其最后一层全连接层换为我们的分类器. 即相当于使用经典的网络自动提取特征, 用全连接层对性别和年龄进行分类.

另外, 我们用到了 CNN 网络结构用于分类常用的一些技巧, 这里做简短的说明, 不同于一般处理的方法我们会强调.

6.1 网络结构

6.1.1 AlexNet

AlexNet 深度学习网络是 Alex 和 Hinton 参加 ILSVRC2012 比赛的卷积网络论文，本网络结构也是开启 ImageNet 数据集更大，更深 CNN 的开山之作，本文对 CNN 的一些改进成为以后 CNN 网络通用的结构；在一些报告中被称为 Alex-Net[4].

我们保留了 AlexNet 的大部分结构, 只改变了最后一层卷积层. 我们本项目中使用的 AlexNet 结构如图7, 其中改变的最后一层已经被用红色标注出来.

属性	层	输入维度	卷积 kernel	卷积 步长	池化 kernel	池化 步长	输出通道数
卷积 1	cov1	227*227	11*11	4*4	3*3	2*2	96
卷积 2	cov2	27*27	5*5	1*1	3*3	2*2	256
卷积 3	cov3	13*13	3*3	1*1			384
卷积 4	cov4	13*13	3*3	1*1			384
卷积 5	cov5	13*13	3*3	1*1	3*3	2*2	256
全连接层 6	fc6						4096
全连接层 7	fc7						4096
全连接层 8	fc8						2 或 6

图 7: **适应本项目的修改后**的 AleNet 网络结构

我们选择使用 AlexNet 是因为其在深度学习领域具有划时代的意义, 首次在 CNN 中成功应用了 ReLU 激活函数、Dropout 和 LRN 等训练技巧,CNN 层数较之后提出的网络较浅, 参数量也较少. 我们将 AlexNet 的创新点总结如下:

- 使用 **ReLU 激活函数**解决了当网络比较深时梯度弥散的问题, 并在性能上超过了 Sigmoid. 而之后的 ReLU 函数成为最常用的激活函数. 我个人理解的 ReLu 激活函数比 Sigmoid 更优的原因是在保证了非线性的基础上, 尽量保证了梯度不变. 而 Sigmoid 函数在函数的两端形状已经趋向于水平, 出现了梯度弥散的现象.

- 使用 **Dropout** 避免模型过拟合.AlexNet 通过在全连接层中使用 Dropout 验证了其效果. 这成为之后设置全连接层网络防止过拟合的经典方法. 我个人感觉 Dropout 的成功在于减少了参数量, 根据奥朗姆剃刀法则, 模型越简单泛化能力越强, 而全连接层参数很多 (例如 4096×4096), 表达能力太强, 在训练集很少的情况下很容易过拟合.

- 全部用**最大池化层**. 记得在 2017 年第一次接触深度学习的时候, 一些书中最大池化和平均池化还是等价的地位, 现在已经广泛使用最大池化层. 最大池化层避免了平均池化的模糊化效果. 这也是 AlexNet 的突出贡献之一.

6.1.2 VGG16

VGG 卷积神经网络[8] 是牛津大学在 2014 年提出来的模型. 当这个模型被提出时，由于它的简洁性和实用性，马上成为了当时最流行的卷积神经网络模型. 它在图像分类和目标检测任务中都表现出非常好的结果. 在 2014 年的 ILSVRC 比赛中，VGG 在 Top-5 中取得了 92.3% 的正确率. 为了适应我们的人脸分类问题, 我们保留其大多数网络结构, 只改变最后一层全连接层, 网络结果如图8, 其中, 改变结构的部分用红色标注出来.

属性	层	输入维度	kernel	步长	输出通道数
卷积 1 cov1	cov1_1	64*64	3*3	1*1	64
	cov1_2		3*3	1*1	64
	max_pool1		2*2	2*2	——
卷积 2 cov2	cov2_1		3*3	1*1	128
	cov2_2		3*3	1*1	128
	max_pool2		2*2	2*2	——
卷积 3 cov3	cov3_1		3*3	1*1	256
	cov3_2		3*3	1*1	256
	cov3_2		3*3	1*1	256
	max_pool3		2*2	2*2	——
卷积 4 cov4	cov4_1		3*3	1*1	512
	cov4_2		3*3	1*1	512
	cov4_3		3*3	1*1	512
	max_pool4		2*2	2*2	——
卷积 5 cov5	cov5_1		3*3	1*1	512
	cov5_2		3*3	1*1	512
	cov5_3		3*3	1*1	512
	max_pool5		2*2	2*2	——
全连接层 6	fc6		——	——	4096
全连接层 7	fc7		——	——	4096
全连接层 8	fc8		——	——	2 或 6

图 8: **适应本项目的修改后**的 VGG16 网络结构

采用 VGG16 的原因是, 在我目前的研究方向——检测和追踪算法中,VGG16 往往是准确率最高的主干网络, 因为我相信其一定可以在人脸分类中达到最优的准确率. 当然,VGG16 最

大的不足在于网络参数较多, 模型巨大, 推断时间慢. 这里, 详细指出 VGG16 的特点和我个人对 VGG16 的理解.

- 网络较深, 探索了**网络层数的加深有助于其性能的提升**.VGG16 达到 16 层, 在当时算是较深的网络结构. 其精度的表现给我们留下了一个印象: 网络的加深有助于网络的表达和精度的提升.
- 通过 3×3 小卷积堆叠来获取较大的感受野, 并降低参数. 这一点个人感觉是 VGG16 最有贡献的一点, 之后的很多网络结构都采用了这种小的卷积核堆叠的结构.

6.2 损失函数

对于分类问题, 最常用的仍然是 softmax 输出的交叉熵损失函数, 这里再给出 softmax 激活函数和交叉熵损失函数的公式:

$$softmax(y)_i = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}} \tag{1}$$

$$H(p,q) = -\sum_{x} p(x) \log q(x) \tag{2}$$

其中,y_i 是最后一层全连接层的输出, 例如在年龄分类中有 2 个输出, 那么分母的 $n = 2$, 通过 softmax 激活后, 输出即为图片为该类的概率.

而交叉熵损失公式中,$p(x)$ 表示输入对应的正确答案,$q(x)$ 代表预测值, 求和是对每个 batch 而言的.

以上是通用的损失函数公式, 但在我们的年龄分类中, **我发现存在以下问题**: 若年龄标签为 0, 即为婴儿; 第一个 softmax 输出为 $[0.5, 0, 0, 0, 0, 0.5]$, 即 0.5 概率是婴儿,0.5 概率是老年人; 第二个 softmax 输出为 $[0.5, 0.5, 0, 0, 0, 0]$, 即 0.5 概率是婴儿,0.5 概率是儿童. 在计算损失时, 这两种输出损失都是相同的:$-(1 \times \log 0.5 + 0 \times \log 0.5)$, 但是**实际上儿童和婴儿的年龄更加接近, 相似度更高, 第二个输出应该更优**.

我并没有在其它论文上发现有讨论这个问题的, 但是我相信这是个值得研究的问题. 不过, 好在我们的 VGG16 顺利地学习到了年龄变化趋势.

6.3 预训练模型

我的理解是 CNN 网络 (这里只包括卷积核池化操作) 取代了传统方法的人手工提取特征, 即 CNN 实际上是提取特征的作用. 不管是 ImageNet 上的 1000 类分类问题还是这里的性别/年龄分类, 甚至是像 yolo 和 faster RCNN 这样的检测网络, 在提取图像特征上都有共性, 因此在使用经典的 CNN 来提取特征时, 都会使用预训练模型. 这样会比随机初始化权重 (正态分布的)

要更快速地收敛, 我认为当然不可避免地容易陷入局部最优, 限制了模型的表达能力. 但是总体来说是利大于弊的.

预训练模型的主要思想是, 在少量的标注检测数据集上训练一个高效的模型，就需要借助神经网络浅层学习低级特征，而深层网络学习高级特征的特点. 在样本量较大的标注数据集上预先训练图像分类模型，把其作为预训练模型载入，再用该预训练模型在小型数据集上进行特定领域的微调. 也就是用迁移学习的方法来克服标注数据少的问题.

需要注意的一点是: 对于新加入的全连接层, 我们**仍用随机初始化权重**的方式.

6.4 不同的学习率

这里我们不必再从头介绍学习率. 在我个人理解里, 由于我们已经采用了在卷积神经网络中常用的预训练模型, 因此初始学习率不会太大. 而不同的学习率从两方面来说的.

- 不同卷积层学习率不同: 底层的卷积学习到的纹理、轮廓等图像分类问题中较为低级的语义特征, 对于 ImageNet 上和我们自己数据集上的学习的参数应该存在很大的共性, 因此我在训练时, 不对低层的卷积层进行训练, **固定原预训练的参数**; 对于**剩余的卷积层**, 其提取的是较为高级的语义特征, 个人感觉**随着卷积层的变深, 人脸分类任务与 ImageNet 分类任务的差别越来越大**. 在这里我初始学习率设置为 $10^{-4} - 10^{-3}$, 实验多组, 最终选用 10^{-4} 来多训练些轮次; 对于最后新添加的全连接层, 由于其实际上做到一个分类器的作用, 而 ImageNet 上的分类与人脸分类任务差距较大, 因此需要大的学习率. 另外对于新添加的最后一层卷积层, 采用随机初始化的参数, 需要大的学习率来达到收敛, 因此使用较大的学习率, 对应实验了 $10^{-3} - 10^{-2}$, 最终选用 10^{-3}.
- 不同训练轮数学习率不同: 为了防止训练后期参数在局部最优解附近左右摇摆, 往往需要进行学习率的衰减. 这是常用的方法, 原理不再赘述, 我的代码中也没有对此进行创新, 选择每五轮衰减 0.9.

6.5 归一化

由于我们用到了在 ImageNet 上预训练好的 CNN 模型, 因此需要对我们的数据集进行归一化处理. 即对 RGB 三个通道的数据分别减去 ImageNet 数据集三个通道的均值, 除以三个通道的标准差. 实际上相当于一个 Batch Normalization 的操作.

在课程项目中, 是用 pytorch 的 transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225])]) 实现的.

6.6 dropout

由于我们爬取的图像仅有几万张, 而 VGG16 等又是体量巨大, 表达能力很强的网络结构, 因此训练过程中容易出现过拟合的现象, 在实际应用中可能会出现泛化能力比较差的问题.

dropout 是图灵奖得主, 人工智能三巨头之一的 Hinton 提出的. 即在训练时, 将某些神经元暂时移除, 此外, 和这些神经元的连接也暂时移除. 也就是说, 在训练过程中停掉某些神经元的作用随机地移除网络层中的神经元，停止该神经元和别的神经元的所有联系，从而达到通过训练一个网络，得到 2^n 网络集成的效果.

在本课程项目中, 使用常规的 dropout 防止过拟合, 每个神经元被 dropout 的概率设置为 $p = 0.5$.

6.7 数据增强

数据增强与 dropout 相似, 也是一种常用的防止过拟合的方法, 即利用原始数据集, 生成新的数据, 以增加数据的样本量. 在图像分类过程中, 对图像进行翻转和旋转并不影响图像自身的分类, 而我们期望的输入本来就是校准后的人脸, **因此只做镜像翻转**, 旋转和随机剪裁并没有在被使用.

除了镜像翻转, 光照条件也会影响图像的识别, 通过随机改变图像的亮度、对比度、饱和度和色相, 图像的年龄和性别标签不应该发生太大变化.

最终, 我们选择了以下 5 种数据增强的方式:

方式 1. **以 0.5 概率随机镜像旋转**. 通过 pytorch 的 transforms.RandomHorizontalFlip() 实现.

方式 2. 随机调整**对比度**为原图的 [0.9,1.1] 倍. 通过 pytorch 的 transforms.ColorJitter 实现, 下同.

方式 3. 随机调整**对比度**为原图的 [0.9,1.1] 倍.

方式 4. 随机调整**色相**为原图的 [0.9,1.1] 倍.

方式 5. 随机调整**饱和度**为原图的 [0.9,1.1] 倍.

7 方法一深度学习方法实验

7.1 性别识别

7.1.1 实验参数

由于 AleNet 与 VGG16 使用的实验参数相近, 我们仅展示 VGG16 的实验参数, 需要注意的一点是: 超参数是在常用参数基础上经过多次实验中得到的较优参数. 训练过程的其余参数设置和硬件条件如下:

- **数据集**: 自己制作的中国人脸数据集.
- **硬件参数**: 操作系统 ubuntu16.04 64 位操作系统，CUDA 的版本为 8.0.61，选用深度学习框架 pytorch 搭建程序. 网络在 1 块型号为 TITAN Xp 的 GPU 上完成训练.

- **模型超参数**:Batch size: 100; 轮数:20 轮.
- **训练集组成 mini-batch 的方式**: 随机组合, 即不按照训练集 txt 文件读入顺序, 避免一些 mini-batch 中仅有一类数据.
- **激活函数**:ReLU.
- **优化器**:SGD.

7.1.2 实验过程

由于 AlexNet 和 VGG16 的实验过程类似, 我们在这里只展示 VGG16 的训练过程, 下面是训练过程中的损失函数和每个 batch 的准确率图11, 这里的 Test accuracy 量纲是 $1/10000$,Test loss 量纲是 $1/1000$.

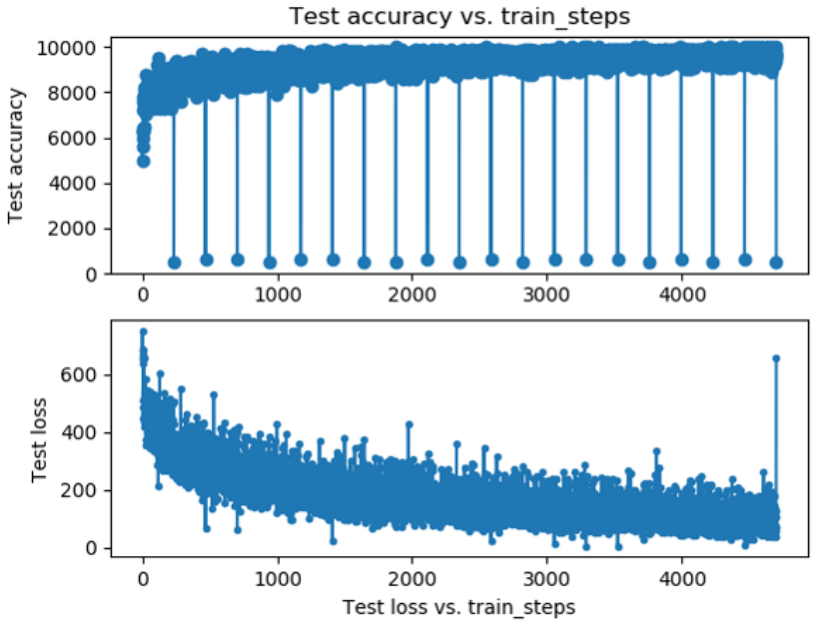

图 9: VGG16 性别识别训练过程每个 batch 上的准确率和 Loss 图

需要特别指出的一点是: 图中等间距的出现错误点的原因是, 在写代码时, 我的准确率公式为:

$$test\_accuracy = \frac{true}{batch\_size}$$

而在每一轮训练的最后一个 batch 中, 剩余的图像数目小于 batch size, 例如在某轮训练中, 共 20 张图像, 正确 18 张, 准确率为 90%, 但我的代码中计算出准确率为 $20/100 = 20\%$. 由于来不及改代码了, 因此特在此说明.

根据图11, 我们可以发现以下两点:

- 训练中精确随着训练步数持续提高, 以我们的模型的测试精度不断提升, 说明训练是正确的.
- 训练中损失函数随着训练步数持续下降, 说明我们的训练过程是收敛的.

7.1.3 不同网络实验结果对比

对于 AlexNet 和 VGG16 的训练时间、最终模型大小、训练集和测试集上准确率以及 Titan xp 上单张推断时间可见表2.

表 2: AlexNet 和 VGG16 性别识别结果比较

项目	AlexNet	VGG16
训练时间	$55120.23s$	$93476.04s$
模型大小	$217.57MB$	$512.22MB$
训练集准确率	$22609/23406 = 96.59\%$	$22809/23406 = 97.45\%$
测试集准确率	$5438/5859 = 92.81\%$	$5618/5859 = 95.89\%$
单张推断时间	$0.0039s$	$0.0094s$

通过以上结果比较, 我们发现有以下特点:

- AlexNet 模型较为轻量, 相同轮数下训练时间短, 模型较小, 单张推断时间短, 但是训练集和测试集上的准确率较低.
- VGG16 模型较大, 相同轮数下训练时间长, 单张推断时间较长, 但是在训练集和测试集上的准确率较高.

以上比较结果同我们对两种网络的认识是相同的, 层数较多, 参数较多的网络往往具有更好的表达能力, 因此往往准确率更高, 但是参数较多也导致训练时间长, 推断时间长, 模型较大等问题, 这就涉及实际应用中的选型问题. 在这里, 我们考虑使用 VGG16 得到更高的精度, 因此以下分析以 VGG16 为主,AlexNet 的分析与 VGG16 类似.

7.1.4 VGG16 年龄识别的混淆矩阵分析

这里, 我们给出测试集上 VGG16 对性别识别的混淆矩阵, 可以直观地观察我们的模型在男女分类中的表现, 见图10. 对于混淆矩阵的详细分析, 我们会在年龄识别结果中详细分析.

通过混淆矩阵图我们可以发现,VGG16 模型对于女性的分类精度较高, 对于男性的分类精度较低.

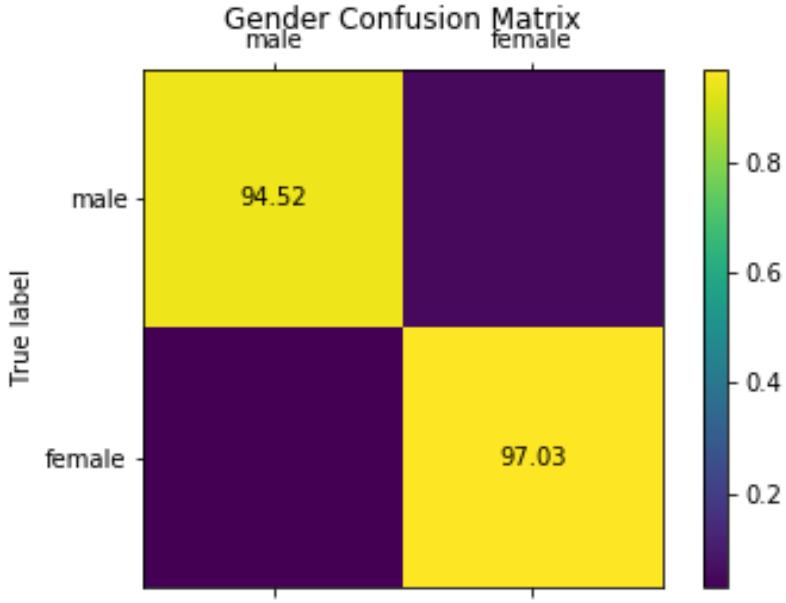

图 10: VGG16 性别分类在测试集上的混淆矩阵

7.2 年龄识别

除掉网络结构外, 年龄识别的实验参数与性别识别相同, 我们同样对 AlexNet 和 VGG16 做了实验, 得出了与性别识别相似的对比结果, 为了报告的精炼和可读性, 我们在本节中只展示 VGG16 的实验结果. **尤其对 VGG16 在测试集上的混淆矩阵做详细分析**.

7.2.1 实验过程和结果

下面是 VGG16 对年龄识别训练过程中的损失函数和每个 batch 的准确率图**??**, 这里的 Test accuracy 量纲是 1/10000,Test loss 量纲是 1.

在这里, 我们修复了性别识别中由于 batch size 不能整除训练集总数带来的图像中的波动.

与性别识别相近, 我们可以看到训练过程是收敛的. 除此之外, 与性别识别图11相比, 我们还发现以下结论:

- 比起性别识别的二分类问题, 年龄识别的 6 分类问题训练难度更大, 体现在精度较低、损失较大, 以及收敛过程波动.

VGG16 年龄识别训练结果和分析如下:

1. 训练时间为 77283s. 训练时间较性别识别训练时间短, 我们推断是因为 GPU 节点状态导致的.

2. 在训练集上准确率:22809/23406=97.45%.

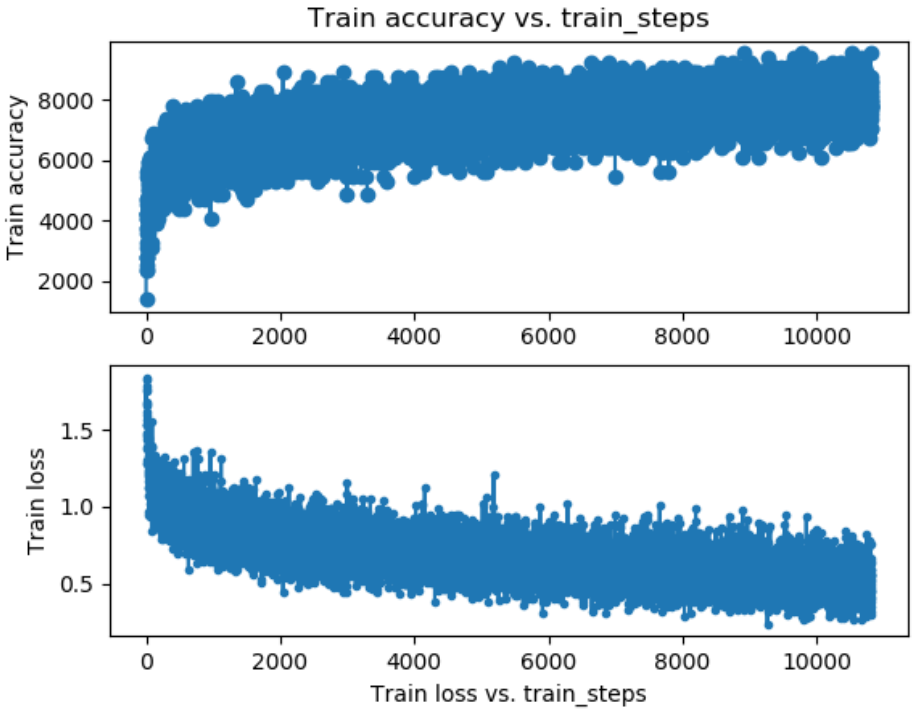

图 11: VGG16 年龄识别训练过程每个 batch 上的准确率和 Loss 图

3. 在测试集上准确率: 测试集准确率:6698/8667=77.28%. 对比训练集上准确率和测试集上准确率, 我们发现, 虽然采用了数据增强, 但是由于数据量仍较少, 而 VGG16 的表达能力很强, 仍有一定程度的过拟合现象.

4. 单张推断时间:0.010061s. 与性别识别相同, 推断时间为在 Titan X 上的推断时间, 时间不包含检测和校准时间.

5. 模型大小:524578KB. 仅比性别识别模型大一点点.

7.2.2 VGG16 性别识别的混淆矩阵分析

这里, 我们给出测试集上 VGG16 对年龄识别的混淆矩阵, 可以直观地观察我们的模型在年龄分类中的表现, 见图12. 由于混淆矩阵给出了令人激动的结论, 因此我们将对混淆矩阵进行详细的分析.

混淆矩阵的颜色表示了占比, 正如右侧颜色标度表示的, 颜色越亮则占比越高. 我们发现以下几点激动人心的结论.

1. **婴儿和老人识别精度最高, 儿童和中年人其次, 青少年和青年人识别精度最低**. 这是由以下原因导致的:

   (a) 婴儿和老人的面部特征最为明显, 而且仅有 1 个相邻的年龄段, 识别较为容易.

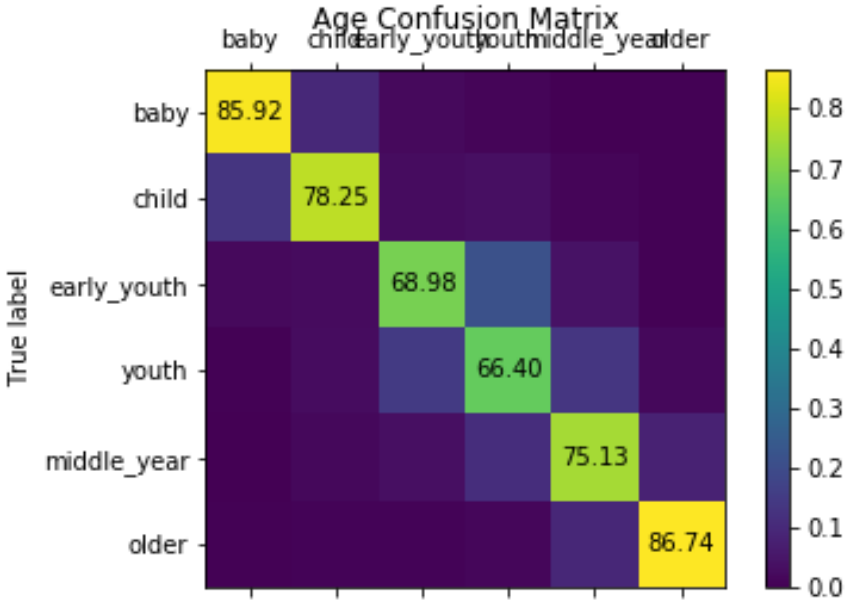

图 12: VGG16 年龄分类在测试集上的混淆矩阵

(b) 在对数据集的分析中我们已经说明, 爬取的数据集偏艺术照, 青少年偏老, 青年偏年轻. 对于人来讲也很难区分, 这也是我们制作自己数据集的初衷, 如果仅仅是使用别人的数据集, 那么很难完全理解数据集. 只有自己制作, 在出现问题的时候才知道从哪些方面考虑.

2. **相邻年龄段的误分类率比较高**. 也就是说, 将婴儿识别为儿童的概率要远远大于婴儿识别为青年或者老年的概率. 这可由对角线两侧的颜色较浅观察到. 我们在损失函数那一节有分析到, 在定义损失函数时, **将婴儿识别为儿童和将婴儿识别为老人有相同的损失函数**. 而我们的模型还是学习到了人的年龄随时间渐变的规律, 这是令人激动的. 再次说明了我们模型的合理性.

8 方法二:HOG+SVM 方法原理

众所周知, 传统的人工提取特征的方法在视觉任务上往往不如卷积神经网络提取特征. 作为本次课程项目的第二种方法, 为了更好地体会传统方法的优缺点, 我们选取了 HOG+SVM 的方法. 虽然增加特征的种类和维度会提高 SVM 的识别精度, 但我们本次课程项目着重于充分理解 HOG 和 SVM, 通过对参数的一些改进尝试提高分类精度, 以感受人工提取传统特征的过程和 SVM 分类器与神经网络的不同. 这里, 我们以性别识别为主要讨论对象, 年龄识别类似. 针对年龄识别, 仅提出我们是如何将 SVM 二分类器用于多分类算法的.

8.1 HOG

HOG+SVM 用于行人检测由论文[2] 提出, 文献[1] 将其用于人的性别识别等任务.HOG 采用了直方图的形式提取基于梯度的特征. 其基本思路是将图像局部的梯度统计特征拼接起来作为总的特征. 局部特征指的是图像被分为的多个 Block, 先对每个 Block 提取特征, 然后将其联合为最终的特征. 根据原论文及常用的 HOG 特征提取流程,HOG 特征提取步骤如下:

step 1. Normalize gamma and colour(标准化 gamma 空间和颜色空间): 将 RGB 图像转化为灰度图, 采用 Gamma 校正法对输入图像进行颜色空间的标准化（归一化）; 目的是调节图像的对比度，降低图像局部的阴影和光照变化所造成的影响，同时可以抑制噪音的干扰. 这里,Gamma 矫正公式为:

$$H(x,y) = H(x,y)^{gamma} \tag{3}$$

这里,$H(x,y)$ 是像素点 (x,y) 处的像素值.

step 2. Compute gradients(计算梯度): 计算水平梯度 g_x 和竖直梯度 g_y, 然后总的梯度强度和方向由下式给出: 设坐标为 (x,y) 的像素点的像素值为 $H(x,y)$, 那么水平梯度 $G_x(x,y)$ 和竖直梯度 $G_y(x,y)$ 分别为:

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \tag{4}$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \tag{5}$$

那么在像素点 (x,y) 处的梯度大小 $G(x,y)$ 和方向 $\theta(x,y)$ 分别为:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{6}$$

$$\theta(x,y) = \arctan \frac{G_x(x,y)}{G_y(x,y)} \tag{7}$$

step 3. Weighted vote into spatial and orientation cells(权重投票): 直方图的方向 bin 在 0°-180°(无符号梯度) 或者 0°-360°(有符号梯度) 之间均分。为了减少混叠现象，梯度投票在相邻 bin 的中心之间需要进行方向和位置上的双线性插值。投票的权重根据梯度幅值进行计算，可以取幅值本身、幅值的平方或者幅值的平方根. 本课程项目在 8×8 的 cell 里面计算, 采用的无符号梯度, 使用 9 个 bins, 直接使用梯度本身进行投票.

step 4. Contrast normalize over overlapping spatial blocks(对比度归一化): 为了降低光照的影响, 对同一个 block 里的梯度直方图进行归一化. 在本课程项目中, 选取 block 的大小为 $16\times$

16, 即对长度为 36 的 vector 进行归一化, 滑动步长为 8×8，采用的是 L2-Hys, 即先进行 L2 归一化, 再对结果进行截断, 然后再重新归一化, 其中 L2 归一化为:

$$v^{'} = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \tag{8}$$

其中,v 是归一化前的向量,$x^{'}$ 是归一化后的向量,ϵ 是很小的常数, 防止分母出现 0.

step 5. Collect HOG's over detection window(生成 HOG 特征): 将所有 block 的 HOG 描述符组合在一起, 形成最终的 HOG 特征向量.

8.2 SVM

支持向量机 SVM（Support vector Machine）由 Cortes 和 Vapnik 在 1995 年提出，它是一种基于统计学习技术，可以被用来做模式分类和变量之间的非线性关系推断. 该方法已经成功被应用到探测、验证和识别脸部、物体、手写字符和数字，文本，语音和说话人识别以及信息和图像.

它的主要思想可以总结为：首先通过核函数映射输入域，然后在映射之后的域中寻找超平面将数据分开，并且使误差最小增益最大. 最终, 超平面被转换回输入域以获得可能是非线性的决策边界. 它是建立在统计学习理论的 VC 维理论和结构风险最小化原理基础上的一种有监督的统计学习方法[3]. 由于 SVM 是基于线性可分模式下的最优超平面，因此它本质上是一种线性分类器.

作为课程项目, 我们不从支持向量机的推导介绍其原理, 但着重给出在课程项目中我们用到的支持向量机使用的几个技巧.

8.2.1 核函数的选择

由于我们的年龄/性别分类是线性不可分问题, 因此需要选择合适的核函数. 选择合适的核函数 $k(x_i, x_j)$ 是建立 SVM 优化问题的一个关键步骤, 以将原始输入 HOG 特征空间的样本映射到高维特征空间. 常见的核函数有以下几种:

1. **线性核函数 (Linear)**.

$$k(x_i, x_j) = (x_i \bullet x_j) \tag{9}$$

2. **多项式核函数 (Polynomial)**.

$$k(x_i, x_j) = (\gamma(x_i \bullet x_j) + c)^d \tag{10}$$

其中,γ, c, d 为超参数.

3. **径向基核函数 (RBF)**.

$$(x_i, x_j) = (-\gamma \|x_i - x_j\|^2) \tag{11}$$

.

其中,γ 为超参数.

4. **S 型核函数 (Sigmoid Function Kernel)**.

$$k(x_i, x_j) = \tan(\gamma(x_i \bullet x_j) + c) \tag{12}$$

其中,γ, c 为超参数.

以上核函数中,RBF 核函数最为常用, 因为其能将样本非线性地映射到一个更高维的空间, 与线性核函数相比的优势在于,RBF 核函数能够很好地处理分类标注和属性的非线性关系. 另外,RBF 核函数拥有更少的超参数.

在实际实验中, 我们使用线性核函数、RBF 核函数、S 型核函数来训练 SVM, 除此之外, 我们还使用了 OpenCV 提供的 INTER 核函数, 以确定哪种核函数更加适合我们的年龄性别分类问题.

8.2.2 网格搜索法确定最优超参数

选定核函数后, 在求解最优超平面之前, 需要对核函数的超参数进行赋值. 超参数 C 用来协调分类器的复杂性和不可分离点数之间的平衡, 决定模型的识别准确率和泛化能力. 关于 SVM 超参数的优化选取, 目前学术上还有没有的最优方案, 目前常用的是网格搜索法. **网格搜索法是指需要优化的超参数在一定参数范围中, 按照一定步长, 划分为网格, 通过遍历网格中所有的点来寻找最优参数**. 理论上, 当范围足够大、步数足够多时可以找到参数的全局最优解. 网格搜索法的缺点在于优化过程耗时, 优化精确度受到取定范围及步长的影响. 在 SVM 的实际使用过程中, 常用的超参数取值为 2 的指数.

8.2.3 多分类算法

支持向量机适用于二分类问题，而**年龄分类为多分类问题 (6 分类问题)**, 将二分类器用于多分类问题的基本思路是将多分类问题拆分为多个二分类问题, 通过建立多个二分类器解决多分类问题.

最常用的算法是**一对一（1-v-1）多分类算法、一对余（1-v-R）多分类算法**.

1. **一对一多分类算法**.k 分类问题中，每两类之间都单独构造 SVM 分类器，总共需要构造 $\frac{k(k-1)}{2}$ 个分类器. 当判定测试样本所属类别时，使用所有分类器对该样本进行分类，统计各个类别的得票数，选择得票数最高的分类器作为该样本的类别.

2. **一对余多分类算法**.k 分类问题中，对于第 i 类样本，首先都标记为正类，将其余 $k-1$ 类都标记为负类，通过训练构造对应的第 i 个最优的 SVM 二分类器，共需要构造 k 个分类器.

9 方法二 HOG+SVM 方法实验

9.1 HOG 特征提取

我们使用 python 对 HOG 算法进行复现, 参考了 CSDN 博主的代码, 在此致谢. 我们与原代码不同的是, 原代码使用 python2 写的, 我们使用 python3 复写了, 且修改了个别参数 (例如原代码中 $bin_size = 8$ 显然是写错了).

表 3: HOG 基本参数

参数	数值
block 尺寸	16×16
block 滑动步长	8×8
bin 数	9
是否高斯滤波	否
是否 gamma 校准	是

我们的图像是按照一级来源目录 ('vcg_CN_1_crop/','veer_CN_1_crop/'), 二级年龄目录 ('baby/','child/','early_youth/','youth/','middle_age/','older/') 和三级性别目录 ('female/','male/') 存放的, 因此需要通过三重循环遍历所有文件夹, 对每张图像提取 HOG 特征, 根据图像路径确定 label, 最后拼接所有图像的特征和 label 作为我们的训练集 + 测试集.

我们提取的 HOG 特征可视化如图13:

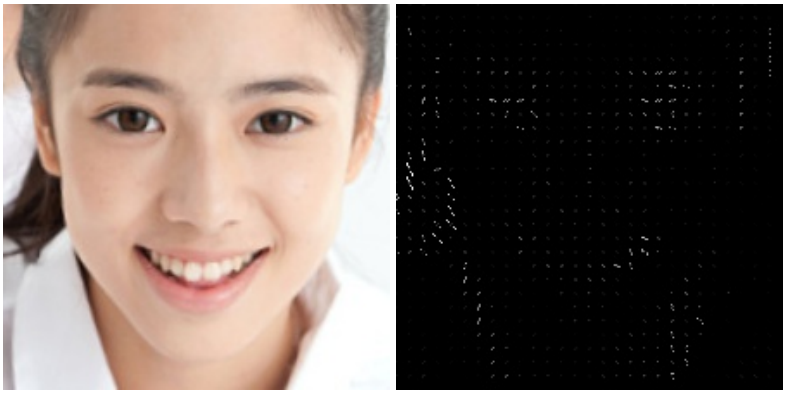

图 13: HOG 特征可视化

可以看出我们的 HOG 特征可以很好地捕捉到梯度显示边缘特征.

9.2 SVM 分类

首先, 我们将提取的特征和年龄或性别 label 随机划分为训练集和测试集, 使用 scikit-learn 的 model_selection 模块的 ms.train_test_split 函数, 将所有的数据集划分为 23412 张训练集和 5853 张测试集, 由于设定了随机种子, 在每次实验中获得的划分都是相同的.

SVM 的实现我们借助 cv2.ml 完成.

如上文所言, 核函数的选择非常重要, 我们共测试了四种核函数, 线性核函数,RBF 函数,S 型核函数, 以及 OpenCV 提供的 INTER 核函数.

在优化 SVM 过程中, 我们采用了网格搜索法对选取的线性核函数超参数进行寻优, 通过循环完成.

对于年龄分类 6 分类问题, 我们采用一对余的方式,OpenCV 的 svm 已经自动实现该功能.

9.3 实验改进历程和结果

9.3.1 最优核函数的选择

在不进行网格搜索法寻找最优超参数 (即使用默认超参数) 的情况下, 使用 OpenCV 提取的 HOG, 以性别分类为例, 我们给出四种核函数的比较如表4.

表 4: 四种核函数在性别识别任务中的表现

核函数类型	训练集上的精度	测试集上的精度	训练时间	测试时间
线性核函数	0.6048	0.6102	$257s$	$< 1s$
RBF 核函数	0.6098	0.5830	$1031s$	$38s$
S 型核函数	0.5406	0.5393	630	25
INTER 核函数	0.5693	0.5646	$67s$	$2s$

根据上表, 我们发现, 综合精度和速度, 以及考虑到 HOG 原论文使用的线性核函数, 我们最终**选用线性核函数**作为最优核函数做进一步优化.

9.3.2 网格搜索法寻优结果

在选用线性核函数后, 我们对其超参数 C 进行寻优, 此时, 由于只有一个超参数, 网格搜索法寻优**退化为一维**的按步长寻优, 我们给出性别识别的部分遍历的参数的 SVM 分类表现如表5.

通过超参数寻优, 我们找到在以 10 倍为步长的 C 的寻优中, 最优的超参数 $C =$, 在测试集上进一步提高了精度. 但是提高幅度不大, 因此, 我们还是考虑由于 HOG 特征表达能力不够导致的.

表 5: 线性核函数超参数寻优结果

C 的值	训练集上的精度	测试集上的精度
0.001	0.6008	0.5811
0.01	0.6489	0.6217
0.1	0.6247	0.5994
1	0.6167	0.6028
10	0.5861	0.5768

9.4 HOG+SVM 进行年龄分类

我们已经通过对性别的分类改进研究发现 HOG 特征太过单一, 难以有很高的精度. 这也是人工提取特征的一大缺点：提取有效的特征十分困难. 因此我们对 HOG+SVM 对年龄分类期望不大. 但是, 我们仍想通过实验明确传统 CV 方法与深度学习方法在年龄分类任务上的差距.

我们采用线性核函数, 参数选用性别分类时实验出的较优参数, 特征提取与数据集划分与性别分类完全相同, 多分类算法由 OpenCV 自动实现, 训练时间为 3503s, 测试时间可以忽略, 在训练集上的精度为 36.46%, 在测试集上的精度为 32.77%. 确实远远不如卷积神经网络的方法.

9.5 改进展望

由于时间有限, 再加已经知道 HOG+SVM 方法是几乎不可能比卷积神经网络的方法更优的, 因此没做进一步改进. 但是我已经构思好如果需要改进时, 我会对 HOG+SVM 方法进行哪些方面的改进, 在这里做简要介绍和记录.

改进 1. **组合更多的传统 CV 特征**. 实验已经证明,HOG 特征对于人脸分类来说是不够具有表达性的, 根据阅读论文的经验, 可以增加颜色直方图等 0 阶特征, 或者边缘线等 1 阶特征, 或者局部协方差矩阵等 2 阶特征. 这里的阶即对像素点求几阶导. 特征又分为局部特征和全局特征. 对于多种特征, 需要设计最优的组合方案. 这也是一个庞大的工程.

改进 2. **提取特定局部的 HOG 特征**. 有论文曾指出, 在提取人脸的 HOG 特征时, 可以根据任务提取特定区域的 HOG 特征, 例如对于年龄识别, 往往额头、两侧脸颊和下巴. 而眼睛、鼻子等提供的信息很弱, 这种思路是可取的.

改进 3. **特征的降维**. 当特征提取过多, 维度过高时, 需要采用 PCA 等方法对特征进行降维, 这样有利于组合特征.

改进 4. **更好的分类器**.SVM 对于小样本来说往往分类效果较优, 但是我们自己制作的中国人数据集来说数据集还是太大, 可以考虑随机森林等其它分类器.

10 两种方法对比

10.1 特征提取的对比

特征提取方面, 在本次课程作业中体会的是:

1. 深度学习方法, 即通过卷积神经网络提取特征的方法, 不需要人工设计特征, 从浅层提取边缘等低层次特征到深层提取富含语义的特征完全自动, 只需要设置好卷积神经网络结构即可;

   传统 CV 方法, 即通过提取 HOG 等梯度特征或者角点、边缘点等特征的方法, 需要人工设计特征, 这就需要选取合适的特征表达, 提取过程中设计很多技巧. 例如 HOG 特征提取过程中的规范化过程, 需要结合对图像和任务的理解.

2. 深度学习方法, 对特定的分类任务不需要太强的背景知识, 相似分类任务之间的可迁移性强. 例如我们的年龄和性别分类, 这两个相似的任务需要相似的特征. 在我们的课程作业中采用相同的卷积神经网络结构和相同的预训练参数提取到了高质量的特征;

   传统 CV 方法, 通过性别和年龄两个任务在相同卷积特征下的识别精度对比, 结合查阅的论文, 提取 HOG、SIFT 等特征用于检测、匹配和分类任务是传统 CV 常用的特征, 我认为也是有一定可迁移性的. 与深度学习方法不同, 在课程项目中, 我感觉传统 CV 方法需要很强的背景知识, 这也是我在本学期课程中学到的一些东西. 例如梯度的作用 (可以表示边缘), 角点的特点等. 在课程项目中由于缺乏对 CV 背景更深层的理解, 是我对 HOG 缺乏理解的主要原因.

3. 深度学习方法, 尤其是层数较多的网络, 拥有大量参数, 可以拟合的函数多, 表达能力强, 提取的特征比传统 CV 方法有效. 作为课程项目, 我只试用了一种传统 CV 的特征提取方法, 但明显感觉深度学习方法提取的特征更加有效. 这与其庞大的参数量是有关的.

4. 传统 CV 方法也有其优点, 首先是不需要训练, 特征提取过程直接计算即可, 例如我们的 HOG, 可以不依赖于 label, 直接对给定的图像进行提取;

   而深度学习的方法依赖于 label 和训练过程, 通过反向传播来更新卷积层参数, 这样带来特征提取模型训练难度大, 也导致对 label 准确性依赖度高. 而传统 CV 提取特征不需要训练.

5. 传统 CV 方法提取特征可解释性强, 可视化方便, 我们在实验过程中已经展示到. 针对可解释性, 我们将在下面详细讨论.

10.2 训练过程的对比

对于训练过程, 以性别分类为例, 我们将其对比汇总为表6.

表 6: 深度学习和传统 CV 方法训练过程对比

角度	深度学习方法 (本课程项目参数)	传统 CV 方法 (本课程项目参数)
硬件需求	GPU(Titan Xp)	CPU(6 核 6 线程)
训练时间	长 (93476s)	短 (257s)
端对端	是	否
调参难度	较容易	与特征提取协同调参, 较困难
分类器	相当于全连接层, 类型固定	多种类型可选 (当前 SVM)
数据集数量要求	需要海量数据防止过拟合	要求不高, 过拟合不严重

上表中的内容详细表述如下:

1. **硬件需求**. 深度学习算法往往需要 GPU 的支持, 否则容易内存溢出. 例如在本课程项目中, 需要用到 1 块 NVIDIA Titan Xp 显卡, 在 CPU 上训练是不可取的;

   传统 CV 方法在一块普通 CPU 上运行训练即可, 例如本课程项目中我在本地电脑训练, 训练集数目与深度学习方法相同, 特征为 8100 维. 需要注意的是, 我并没有尝试是否可以把传统 CV 方法部署到 GPU.

2. **训练时间**. 虽然 GPU 算力要远远高于 CPU, 对比两者训练时间是不公平的. 但是在 GPU 算力强的情况下, 深度学习方法训练时间仍远远高于传统 CV 方法 (93476s VS. 257s). 训练时间长带来的坏处是每次调参需要等待时间, 影响了优化模型的效率.

3. **端对端**. 深度学习方法是一种端对端 (end-to-end) 的训练方法, 在训练过程中只需要给定优质的输入, 观测其输出调参即可;

   而传统 CV 方法将特征提取过程和分类器区分开来, 在训练过程中需要考虑两者的协同关系, 即如何配合才能达到最优的分类效果, 这带来了训练过程的困难.

4. **调参难度**. 综合是否端对端和训练时间等因素, 以及考虑超参数的数目, 个人认为传统 CV 方法调参难度仍是比深度学习方法难. 因为其不止要对分类器的超参数进行寻优, 还要考虑特征提取过程中的超参数. 例如在本课程项目中,bin size 和 block size 等都会影响特征提取的好坏, 但是我们仍选用了经典的超参数.

5. **分类器**. 传统 CV 方法非端对端的特性带来的好处是, 分类器可选种类较多. 例如在本课程项目中,SVM 分类器可换为决策树分类器, 甚至随机森林等集成学习的方法;

而深度学习方法, 在某种视角下, 目前常用的分类器仍是全连接神经网络或者全卷积神经网络. 这也限制了分类器的种类.

6. **数据集数量要求**. 卷积神经网络参数巨大, 表达能力强带来的问题是容易过拟合, 因此需要大量的数据, 在数据不足时, 像本课程项目中那样, 需要对数据进行增强, 以防止过拟合;

   而传统 CV 方法不存在这个问题, 尤其是本课程项目用到的 SVM 分类器, 非常适用小样本问题.

10.3 训练结果的对比

对于训练结果, 以性别分类为例, 我们从训练集上精度, 测试集上精度, 单张推断时间, 模型大小四个方法展开对比, 这四个方法是我们在书籍推荐系统中要考虑的精度问题、泛化能力问题、用户体验问题和部署难度问题的关键指标.

对比详情如表**??**.

表 7: 深度学习和传统 CV 方法训练结果对比

角度	深度学习方法 (本课程项目参数)	传统 CV 方法 (本课程项目参数)
训练集上的精度	高 (0.9745)	低 (0.6489)
测试集上的精度	高 (0.9589)	低 (0.6217)
对应硬件上单张推断时间	较快 (0.0094s)	快 (<0.0002s)
模型大小	很大 (512.22MB)	较小 (204MB)

对于上表的详细解释及其对应的书籍推荐系统的需求如下:

1. **测试集上的精度**. 测试集上的精度代表了我们的模型对于性别或者年龄的识别准确率, 这是书籍推荐系统的最为关键的指标. 通过对比可以发现, 深度学习方法达到 95.89% 的测试精度, 要远远高于传统的方法 (62.17%). 精度高是深度学习方法比传统 CV 方法最大的优势. 针对我们的书籍推荐系统, 深度学习方法达到的精度已经可以满足实用需求.

2. **训练集上的精度**. 通过训练集上的精度和测试集上的精度对比, 我们可以推断模型的过拟合程度. 就性别分类任务来讲, 两种方法在测试集上精度比训练集上低了 2 个点. 这都表明对于性别分类任务的过拟合程度不严重.

   对于年龄分类任务则不然, 深度学习方法在测试集上准确率达 97.45%, 而测试集上仅有 77.28%, 下降了 20 个点, 说明深度学习方法在年龄分类任务上是存在一定程度过拟合的. 传统 CV 方法虽然只下降了 3.7 个点, 但是由于其精度本来就比较低, 说明也是存在一定过拟合的. 综上说明年龄分类任务比性别分类任务困难的多, 需要更多的数据去训练.

3. **对应硬件上的单张推断时间**. 这决定了用户的体验, 即用户按下按键后获取结果的时间延迟.

   对于深度学习方法, 在 GPU 上单张推断时间 0.0094s, 可以忽略. 在本地 CPU 上所有流程耗时相加 2s 左右, 可以接受.

   传统 CV 方法在不考虑提取特征过程时推断时间可以忽略不计, 增加特征提取过程在 CPU 上仍比深度学习方法快. 这说明, 在 CPU 上的应用场合, 传统 CV 方法往往更快, 这与其模型小有关.

4. **模型大小**. 模型大小除了影响推断速度外, 还决定了部署难度. 深度学习方法参数量 512.22MB, 两个模型 (性别 + 年龄) 至少需要 1G 的硬盘存储, 这在很多嵌入式设备中都是很困难的, 需要训练轻量级网络.

   而传统 CV 方法提取特征过程无存储好的参数, 只需要脚本,OpenCV 保存的 SVM 模型 (.mat 文件) 也相对较小, 整体模型较小, 适合部署到嵌入式设备中.

我们的 demo 想在本地 CPU 上跑, 更希望精度高, 而且深度学习方法延迟 2s 左右也可以接受, 因此 demo 阶段选用精度高的深度学习方法.

10.4 可解释性对比

深度学习的可解释性低一直是其最大的弱点, 因此在此单独说明, 以下阐述以下我对深度学习方法和传统 CV 方法可解释性的对比感受.

1. **传统 CV 方法**. 在本次课程项目中, 我更加真切地感受到传统 CV 方法建立在严密的科学推理之上.

   在特征提取阶段,HOG 特征不管是梯度的计算还是直方图的统计, 还是一些归一化方法, 都是建立在严密的数学公式和原理之上的. 虽然这些方法有一定随意性, 取决于算法设计者对问题的理解. 但每一步都可很好地可视化出来, 通过可视化可以看到是否很好地完成了期望的特征提取. 因此传统 CV 方法是可靠的, 我们可以完全理解其特征提取的科学性.

   在分类阶段,SVM 有完善的理论支撑, 从理论上证明了 SVM 求解过程是严密的, 非局部最优的. 只是由于超参数的影响会寻到考虑进超参数后的局部最优.SVM 在数学上的严密推导是其在工程上广泛应用的原因之一, 个人理解是其有数学推导说明了其是可控的, 不会出现大的问题.

2. **深度学习方法**. 个人理解深度学习方法建立在直观感受、弱的科学推理和实验三者之后. 在本次课程项目中, 我阅读了大量文献, 同时自己也有在关注检测和追踪的文章, 感觉深度学习论文大多在讲网络结构为什么是合理的,module 设计的优势是什么. 而这些优势和合理性

体现在作者可以自圆其说和实验结果提高精度, 很少有数学推理. 这也再次印证了深度学习方法的缺陷: 可解释性弱.

如果让我们说出性别识别或年龄识别的科学依据, 我们只能说: 我们采用经典的卷积神经网络提取了人脸的高层次的语义特征, 通过全连接神经网络对其进行了分类. 到底卷积神经网络学到了什么样的语义特征, 为什么会学到这样的特征, 我们很难解释清楚.

可解释性差是深度学习方法在一些关键领域难以被广泛应用的原因, 据我与业界交流得知, 目前像银行、医学图像、自动驾驶等关乎财产和生命的领域, 仍高度依赖传统机器学习分类器, 对深度学习算法持保守态度. 但是这不影响深度学习算法在其它领域, 正如我们的书籍推荐系统中的应用.

11 demo

当前人脸识别最流行的方法仍是深度学习的方法, 我们使用方法一仅仅是为了与传统算法做对比. 因此 demo 阶段, 我们使用基于深度学习的方法, 给出我们的书籍推荐系统的想象图. 由于最终以文档形式展示, 因此这里以截图方式说明, 视频已在汇报中展示.

demo 展示截图可见图14和15.

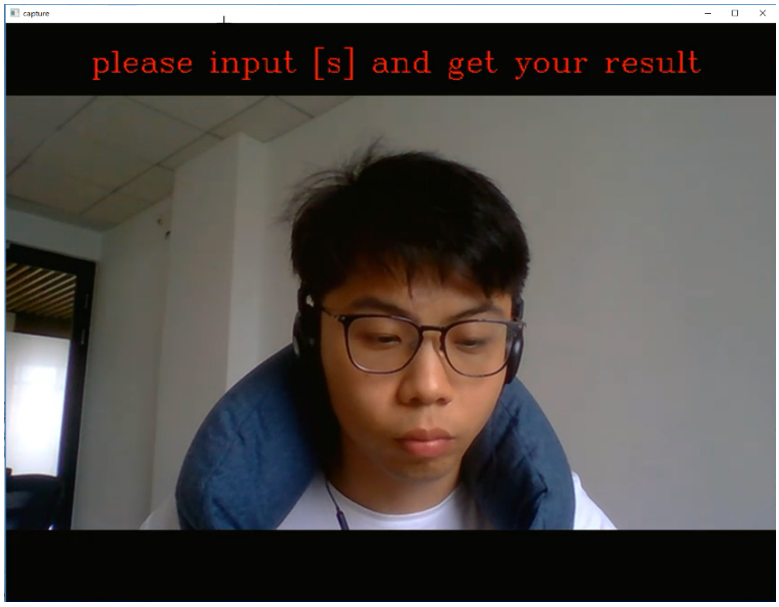

图 14: 书籍推荐系统用户界面

我们需要实现定义好书籍与年龄性别的映射关系, 为了在识别错误的情况下不引发不良用户体验, 我们不会在用户界面输出识别结果, 而是后台输出. 识别流程如下:

1. opencv 调用摄像头, 采集用户画面;

2. 用户按键'q' 进入识别阶段, 摄像头保存当前帧;

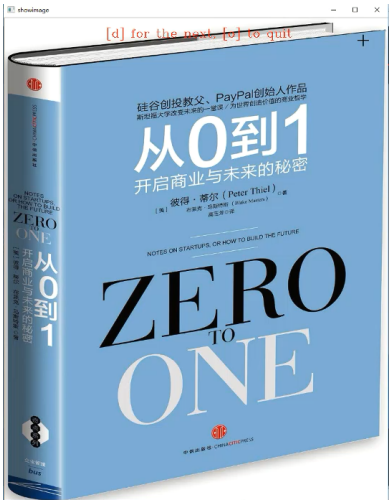

图 15: 书籍推荐系统推荐界面

3. 当前帧通过人脸检测、人脸校准, 输入到 VGG16 网络进行年龄识别, 若用户非婴儿或儿童, 输入到 VGG16 网络进行性别识别;

4. 打开展示推荐书籍窗口, 通过分类和书籍映射关系输出推荐书籍图像, 用户可按'u' 向上翻页, 可按'd' 向下翻页;

5. 用户按'o' 关闭推荐界面, 系统等待下一用户;

6. 管理员按'q' 退出系统.

12 感谢

感谢陈老师一学期的教导. 在本学期的课程中, 陈老师主讲的课我全部参加了, 无一翘课. 原因是陈老师讲课特别轻松, 也让我从思想上受到很多启发. 主要是三个方法: **创新精神、联系实际、勤于动手**. 陈老师有节课问我:' **你想做先驱吗**?', 我毫不犹豫地说:' 想'. 这也是我这节课收获的最大的鼓励.

创新精神. 陈老师在课上一直在引导我们思考, 想要让我们体会自己思考和创新的感觉. 我全程参与其中, 在与老师的互动的互动中收获良多. 从如何教小朋友识别两种不同的鱼我发现了鱼鳍的数目不同, 到对两张不同时期地图的匹配时通过形状识别出是上海, 再到追踪时考虑两条鱼的动力学. 我认为我的创新在慢慢萌芽. 在本次课程作业里, 我也尽量去融合自己的创新想法, 最令自己满足的是对人脸校准算法的提出. 虽然仅仅是一个很容易的观察, 即通过人脸关键点可以发现人脸的垂线基本上是与鼻子的连线平行的, 因此选用鼻子连线作为垂线做旋转, 最终获得了很好的效果. 这虽然只是一个很小的创新, 也许离先驱还差十万八千里. 但是通过本门课程, 我至少懂得了如果创新, 要善于观察, 勤于思考, 并将想法付诸实践.

联系实际. 在本次课程项目中, 我把自己定位在总策划的高度, 试图想从所有方面考虑书籍推荐系统的设计. 包括算法逻辑、数据集适用性、模型大小、推断速度、部署条件、用户体验等. 这十分考虑我联系实际的能力. 例如制作自己的中国人数据集的动机之一是认为公开数据集大多为白人, 不适合我们的实际应用场景, 因为人种的差异往往会导致模型泛化能力很差. 另外, 在研究算法同时, 我还考虑了尽量选取耗时较短的算法, 以此来缩短推断时间, 增加用户体验. 联系实际的能力很重要, 作为以后要参加工作的硕士, 做的东西往往是需要落地的, 因此必须考虑应用需求、已有条件等. 本次课程项目在这一方面对我进行了很好的锻炼.

勤于动手. 陈老师一学期都在强调, 一定要动手实践. 由于我项目的特殊性, 我在动手实践方法可能偏向于深度学习算法. 在实践之前, 我对深度学习算法各个 trick 的优劣的理解来源于书本和课程. 而在实际动手调参、写代码、debug 等过程中, 我才真正意识到预训练、学习率等深度学习基本技巧的用处, 也才真的意识到过拟合是个多么严重的问题, 意识到数据集对于深度学习来说是那么重要. **争做先驱**. 这句话我一直记在心里, 我把它当做老师的一种认可. 志当视觉领域的先锋, 希望把自己的奇思妙想都实现出来, 哪怕只是为视觉领域推动一小步, 也可以实现人生的价值. 这条路势必是困难的、曲折的, 我会以本次课程作业为起点, 一点点做起, 努力前行, 争做先驱.

最后, 再次感谢陈老师的谆谆教导！如果有机会希望跟老师有更多的交流！

参 考 文 献

[1] George Azzopardi, Antonio Greco, and Mario Vento. Gender recognition from face images using a fusion of svm classifiers. volume 9730, pages 533–538, 07 2016.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[5] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[6] R. Rothe, R. Timofte, and L. V. Gool. Dex: Deep expectation of apparent age from a single image. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 252–257, Dec 2015.

[7] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.