

基于SVD与随机森林算法的人脸性别识别与颜值机器学习

21377251-庞熙芃 21377199-林星竹

基于SVD与随机森林算法的人脸性别识别与颜值机器学习

Introduction

- 1.1 研究背景
- 1.2 研究目的
- 1.3 现有研究：传统算法（人工标注特征）
 - 1.3.1 几何特征
 - 1.3.2 面积特征
 - 1.3.3 表观特征
- 1.4 数据来源

Methodology

- 2.1 奇异值分解 (SVD)
 - 2.1.1 基本原理
 - 2.1.2 在图像处理中的应用
 - 2.1.3 降维和特征提取
- 2.2 主成分分析 (PCA)
 - 2.2.1 理论背景
 - 2.2.2 数学原理
 - 2.2.3 在图像分析中的应用
 - 2.2.4 在实验中的应用
- 2.3 随机森林 (Random Forest)
 - 2.3.1 理论背景
 - 2.3.2 数学原理
 - 2.3.3 公式表示
 - 2.3.4 在实验中的应用

Experiments

3.1 实验一

- 3.1.1 数据收集：描述数据收集的过程和训练集/测试集的选择
 - 3.1.1.1 数据集描述
 - 3.1.1.2 训练集和测试集的选择
 - 3.1.1.3 训练数据矩阵创建

3.1.2 RGB转灰度图像转换：介绍图像处理的步骤和理由

3.1.2.1 数据预处理

3.1.2.2 图像预处理的具体实施

3.1.3 SVD在本研究中的应用

3.2 实验一延伸：人脸识别算法

3.2.1 线性回归最小残差法在人脸识别中的应用

3.2.1.1 算法概述

3.2.1.2 实施步骤

3.2.1.3 关键考量

3.2.2 Logistic回归在人脸识别中的应用

3.2.2.1 算法概述

3.2.2.2 实施步骤

3.3 实验二

3.3.1 实验目的

3.3.2 预处理及评分对应

3.3.2.1 数据集与预处理

3.3.2.2 图像读取与转换

3.3.2.3 评分数据读取

3.3.2.4 结构检查

3.3.2.5 结果解释

3.3.3 图像评分分析

3.3.3.1 平均图像的计算

3.3.3.2 平均图像的可视化

3.3.3.3 结果解释

3.3.4 图像展示与评分可视化

3.3.4.1 实验方法

3.3.4.2 结果呈现

3.3.4.3 结果分析

3.3.5 PCA和随机森林的应用

3.3.5.1 目的与方法

3.3.5.2 PCA应用

3.3.5.3 随机森林模型训练

3.3.5.4 结果与分析

3.3.6 图像颜值评分预测

3.3.6.1 实验方法

3.3.6.2 结果展示

3.3.6.3 分析

Discussion

4.1 实验结果解释

4.1.1 算法比较：性能和准确率比较

4.1.2 实验一结果分析：深入分析和讨论

4.1.2.1 实验过程

4.1.2.2 性别分类结果

4.1.2.3 影响因素讨论

4.1.3 SVD在性别分类中的优势和局限性

4.1.4 实验二结果分析：深入分析和讨论

4.1.5 实验局限性

4.2 与现有研究的比较

Conclusion

5.1 结论与未来的工作

5.1.1 实验一结论

5.1.2 实验二结论

参考文献

Introduction

1.1 研究背景

人脸识别技术已成为当今社会极其重要且应用广泛的一项技术。它在许多领域都有着显著的应用，包括但不限于案件侦破、个人设备安全解锁、网络信息安全等。这项技术的核心在于能够识别和分析人脸特征，进而进行身份验证或其他相关处理。随着技术的发展，人脸识别不仅仅局限于传统的安全领域，还开始在社交媒体、广告、健康监测等领域展现其潜力。

近年来随着人脸识别技术的发展，颜值打分也受到了广泛的关注与研究。可即使人来打分，大家也口味各异，御姐萝莉各有所爱。计算机又岂能判断人的美丑呢？实际上科学家研究过人脸的“颜值”，并一直在开发相对应的“颜值算法”。五官匀称，轮廓对称，肤色美观的脸更容易受到大众的喜爱，这一点在颜值中可算达到脸共识，也就是“丑人多作怪，美人一个胚”。正因如此，颜值算法才有了可行性，国内各大公司也相继开发了颜值打分应用。

1.2 研究目的

尽管人脸识别技术的发展迅速，但在处理大规模图像数据时仍面临诸多挑战，尤其是在进行特定属性识别，如性别分类时。图像识别的主要难点之一在于处理的图像数据量通常很大，且每张照片的像素和比例各不相同。这就需要一种能够有效处理并分析这些大规模数据的方法。

本研究第一部分旨在探讨如何使用R语言进行图像处理和数据分析，特别是在RGB格式向灰度图像的转换以及奇异值分解（SVD）方面。在本文中，我们使用了三种不同的方法来进行人脸图像中的性别判别：线性回归最小残差法、Logistic回归和欧几里得距离法。这些方法各有特点，适用于处理图像数据中性别分类的不同方面。

我们的目标是展示这些方法在人脸识别任务中的有效性，并比较它们在处理实际问题时的优缺点。通过这项研究，我们希望为人脸识别领域提供更有效的工具，尤其是在性别分类这一具体应用方面。

本研究第二部分旨在探索PCA和随机森林在图像处理中的联合应用，特别是在涉及图像压缩和分类的场景中。通过利用PCA的降维能力和随机森林的分类强度，旨在提高图像分析任务的效率和准确性。这种组合被假设在处理图像数据固有的复杂性（如高维度和变量相关性）方面特别有效。本研究的目标是开发更加高效的图像处理方法，为数字图像分析和计算机视觉领域做出重要贡献。

1.3 现有研究：传统算法（人工标注特征）

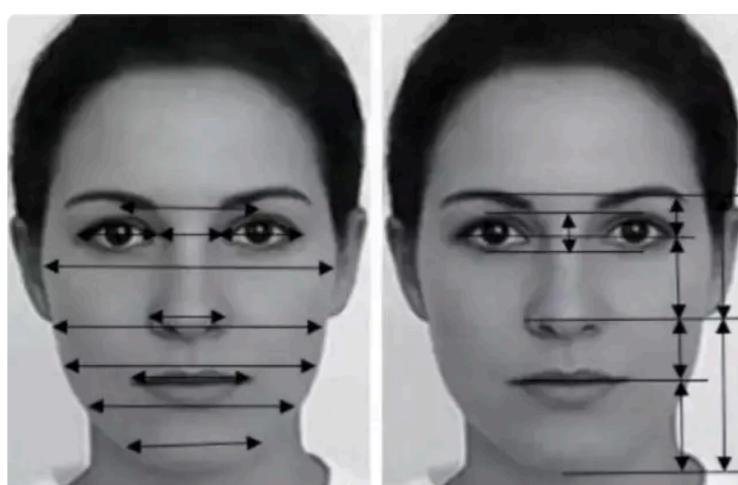
SVD作为一种高效的代数特征提取方法，在数据压缩、信号处理、模式分析等多个领域得到广泛应用。人脸识别中首次应用SVD是由洪子泉提出的，结合主成分分析（PCA）和线性判别分析（LDA），在公共数据库上实现了良好的识别率。

基于SVD的人脸识别方法主要通过提取图像的奇异值向量来识别人脸。这类方法通常存在识别率较低的问题，原因被归咎于小样本影响或奇异值向量未包含足够的识别信息，但这些解释并未得到理论支持。SVD作为图像的代数特征，具有稳定性以及对位移、旋转等具有不变性，但单独使用奇异值向量识别人脸效果不佳。

以往研究通常从几何特征、面部特征和表冠特征等进行预测。

1.3.1 几何特征

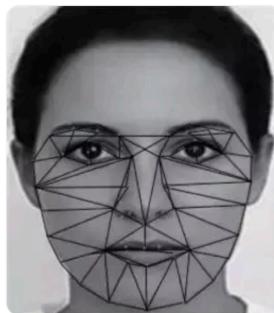
以ASM-68模型提取的特征点为基准，定义18个距离特征。



特征量	特征量描述	特征量	特征量描述
F1	两眼中心间距	F10	过下唇的脸宽度
F2	左眼长度	F11	下巴的宽度
F3	两眼内测眼角间距	F12	眉毛和眼睛的距离
F4	右眼长度	F13	眼睛的高度
F5	人脸面宽	F14	眼睛到鼻子的距离
F6	鼻子的宽度	F15	鼻子到嘴巴的距离
F7	过鼻尖的脸宽度	F16	嘴巴和下巴的距离
F8	过上唇的脸宽度	F17	眉毛到鼻子的距离
F9	嘴巴的长度	F18	鼻子到下巴尖距离

1.3.2 面积特征

根据 ASM 定位的关键点找到表征各器官面积的三角形，如眼睛、鼻子、下巴、嘴等，将得到的 54 个三角形面积特征归一化后就可以得到三角形面积特征。



将上图中的 18 个距离特征组成一个 18 维向量作为输入，分别用线性回归、高斯过程回归、支持向量机三种方式来进行测试，采用 10 折交叉验证的方式进行训练，测试效果用 PC（皮尔逊相关系数）、MAE（平均绝对误差）和 RMSE（均方根误差）来表示，结果如下图：

	Asian Female			Asian Male		
	LR	GR	SVR	LR	GR	SVR
PC	0.6771	0.7057	0.7008	0.6348	0.6923	0.6816
MAE	0.402	0.387	0.3876	0.3894	0.3572	0.356
RMSE	0.5246	0.5057	0.5089	0.5085	0.4752	0.4823

FACIAL BEAUTY PREDICTION USING GEOMETRIC FEATURE WITH SHALLOW MODELS FOR THE WHOLE DATASET

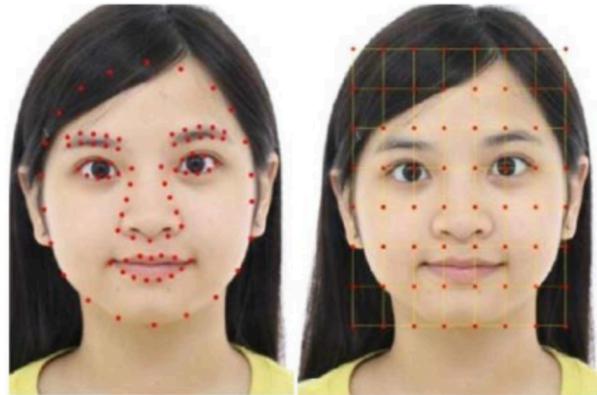
	Linear Regression	Gaussian Regression	SVR
PC	0.5948	0.6738	0.6668
MAE	0.4289	0.3914	0.3898
RMSE	0.5531	0.5085	0.5132

可以看到，线性回归的效果要略差于高斯过程回归和 SVM，后两者相差无几，但 0.67 的相关系数还不够好。

1.3.3 表观特征

使用包含5个尺度、8个方向的Gabor滤波器对图片进行滤波，得到40张特征图，使用两种采样方式来对这40张特征图进行采样。对86个特征点进行采样，组合成 $86 * 40 = 3440$ 维向量；按64UniSample的方式进行采样（详见下图右），组合成 $64 * 40 = 2560$ 维向量。采样完成后使用主成分分析法来对输入向量进行降维，同样采用10折交叉验证的方法来进行训练，结果如下图：

	86-keypoints		64UniSample	
	GR	SVR	GR	SVR
PC	0.7472	0.6691	0.6764	0.8065
MAE	0.3554	0.3891	0.4014	0.3976
RMSE	0.4599	0.5065	0.5177	0.5126



1.4 数据来源

华南理工SCUT-FBP5500人脸美学数据集提供了一种新的多元化基准数据集 SCUT-FBP5500，来实现多范式颜值预测。该数据集共有 5500 个人脸正面照片，这些照片具备不同属性（男性／女性，年龄等）和不同标签（面部地标、颜值得分（1~5）、颜值得分分布）使用不同的特性和预测器组合、不同的深度学习方法可以评估 SCUT-FBP5500 数据集。

每一张图由60个人进行评分，共评为5个等级，这60个人的年龄分布为18~27岁，均为年轻人。适用于基于appereance/shape等的模型研究。同时，每一个图都提供了86个关键点的标注。



这里，我们选用该数据集中亚洲男女性共1000张图片进行分类建模。

Methodology

2.1 奇异值分解 (SVD)

2.1.1 基本原理

奇异值分解 (Singular Value Decomposition, SVD) 是一种重要的矩阵分解技术，它能将任意一个矩阵分解为三个特定矩阵的乘积形式：

$$A = U\Sigma V^T \quad (1)$$

。在这个分解中，(U)和(V)是正交矩阵，分别代表原始数据矩阵的行空间和列空间的基础，而对角矩阵 Σ 包含了奇异值，这些奇异值提供了关于矩阵中数据变化的重要信息。

2.1.2 在图像处理中的应用

在图像处理领域，SVD被用于识别和提取图像的关键特征。对于图像数据矩阵，SVD帮助我们确定哪些特征（对应于较大的奇异值）对图像表示最为重要。这种方法在降低数据维度和强调重要特征方面特别有效。

2.1.3 降维和特征提取

使用SVD进行降维和特征提取包括以下关键步骤：

1. 选择奇异值和奇异向量：

- 选取前几个较大的奇异值及其对应的奇异向量（来自左奇异向量 (U) 的列）。这样做的目的是保留数据中最重要的方面，同时去除可能代表噪声的较小奇异值。

2. 特征向量的提取：

- 提取的奇异向量可视为图像数据的压缩表示，每个向量捕捉数据中的关键特征或模式。

2.2 主成分分析（PCA）

2.2.1 理论背景

主成分分析（PCA）是一种统计方法，它通过正交变换将一组可能相关的变量转换为一组线性不相关的变量，这些新变量称为主成分。PCA是多元数据分析中的一种工具，常用于探索性数据分析和预测建模。

2.2.2 数学原理

PCA的核心思想是将原始数据在新的坐标系下重构，以使得重构后的数据在某些维度上具有最大的方差（数据的变异性），从而捕捉数据中的主要变化趋势。数学上，PCA涉及以下步骤：

1. 标准化数据：

数据标准化是PCA的重要步骤，确保每个特征的贡献是等价的。对于数据集 X ，每个特征列的值减去其均值并除以标准差。

2. 计算协方差矩阵：

$$C = \frac{1}{n-1} X^T X \quad (2)$$

，其中 X 是标准化后的数据， n 是样本数量。

3. 求解特征值和特征向量：

协方差矩阵 C 的特征值和特征向量被计算出来。特征向量（即主成分）确定了数据在新坐标系中的方向，而特征值确定了这些方向的重要性。

4. 选择主成分：

根据特征值大小排序特征向量，并选择前 k 个特征向量作为主成分，其中 k 是一个预先设定的数目，表示保留的主成分数量。

5. 转换到新的坐标系：

使用选定的主成分将原始数据转换到新的坐标系。

2.2.3 在图像分析中的应用

在图像分析中，PCA可以用于降低图像数据的维度。每张图像可以视为一个高维数据点，PCA使我们能够提取最重要的特征（主成分），这些特征捕捉了图像数据中的主要变化。这对于图像压缩、图像识别和其他图像处理任务非常有用。

2.2.4 在实验中的应用

在该实验中，PCA用于减少图像数据的维度，从而简化随后的机器学习任务。通过仅保留最重要的特征，PCA帮助减轻了计算负担，并可能提高了模型训练的效率。在将图像数据降维后，这些数据被用作随机森林模型的输入，以进行进一步的分析和预测。

2.3 随机森林 (Random Forest)

2.3.1 理论背景

随机森林是一个强大的机器学习方法，用于分类和回归问题。它是一个集成学习方法，主要思想是构建多个决策树并将它们合并起来以获得更准确和稳定的预测。随机森林的优势在于它能够处理大型数据集，能够处理具有高维特征的数据，并且能够评估变量的重要性。

2.3.2 数学原理

随机森林的建立涉及以下步骤：

1. 自助抽样 (Bootstrap sampling) :

对原始数据进行重复抽样以构建多个训练集。每个训练集用于训练一个决策树。这种抽样方法允许一些样本在一个训练集中多次出现，而另一些则可能根本不出现。

2. 构建决策树：

对于每个训练集，构建一个决策树。在每个决策树的分裂过程中，选择最佳分裂是基于一个随机选择的特征子集。这种随机性有助于增加模型的多样性，减少过拟合风险。

3. 聚合预测：

- 分类问题：每个决策树给出一个预测结果，最终的预测结果由多数树的预测结果决定（多数投票）。
- 回归问题：每个决策树给出一个数值预测，最终的预测结果是所有树预测结果的平均值。

2.3.3 公式表示

设 Y 为目标变量， X 为特征变量，随机森林由 N 棵决策树 $\{T_1, T_2, \dots, T_N\}$ 组成。每棵树 T_i 在一个自助样本上训练，并对输入向量 x 进行预测 Y_i 。

- 分类：

$$Y_{RF}(x) = \text{mode}\{T_1(x), T_2(x), \dots, T_N(x)\} \quad (3)$$

- 回归：

$$Y_{\text{RF}}(x) = \frac{1}{N} \sum_{i=1}^N T_i(x) \quad (4)$$

2.3.4 在实验中的应用

在该实验中，随机森林被用于对降维后的图像数据进行分析和预测。利用多棵树的集成学习方法，随机森林能够处理复杂的数据结构，提高预测的准确性，并且通过聚合多个模型的预测结果来减少过拟合的风险。这在处理高维数据（如图像）时尤其有用，因为这些数据通常包含复杂的特征结构。

综合PCA和随机森林的优点，您的项目能够有效处理图像数据，提取重要特征，并进行准确的分类或回归分析。

Experiments

3.1 实验一

3.1.1 数据收集：描述数据收集的过程和训练集/测试集的选择

3.1.1.1 数据集描述

- 图像集来源：**数据集由一系列标记为男性和女性的人脸图像组成。这些图像从华南理工大学的人脸美学图像库中精心挑选，以确保在性别分类任务中的有效性和代表性。数据集包括两部分：训练集和测试集，分别用于模型的训练和评估。
- 性别标记方法：**图像的性别通过文件名前缀进行标记。具体来说，“AF”前缀表示图像为女性，“AM”前缀则表示图像为男性。这种标记方法简洁且便于自动化处理，确保了快速有效地分类和组织图像数据。
- 图像尺寸：**为保证实验的统一性和数据一致性，所有图像在预处理阶段被调整至统一尺寸（350x350像素）。这一步骤关键在于确保在进入分类算法之前，所有图像具备相同的维度和比例。



3.1.1.2 训练集和测试集的选择

- 训练集的选择：**训练集的选择旨在提供广泛且均衡的性别分布，以确保模型能够学习到从男性到女性的广泛特征。训练集包含1000张图像，其中男性和女性图像的数量相等。
- 测试集的选择：**测试集包含25张图像，从班级同学获得，以评估模型在未见数据上的性能。测试集的选择也遵循性别平衡的原则，以确保测试结果的公正性和准确性。



3.1.1.3 训练数据矩阵创建

创建训练数据矩阵是构建分类模型的关键步骤，涉及以下过程：

- 图像尺寸统一：**确保所有图像被处理至统一尺寸（350x350像素），以维持数据的一致性。
- 矩阵初始化：**初始化一个矩阵，其行数等于图像数量（1000张），列数等于每张图像的像素总数（350x350）。
- 性别标记与存储：**为每张图像分配性别标签（男性为1，女性为0），并将这些标签存储在一个单独的向量中，用于后续的分类模型训练。

通过这一过程，确保数据集的质量和一致性，为接下来的图像处理和分析奠定了坚实的基础。

在完成了对照片的男女判断分析后，针对男女不同的数据集以及收集到的每张图片来自60位志愿者打分综合的“颜值评分”，对判断好的男女性别图片进行进一步的“颜值评分”分析。

```
# 第1部分：初始化和导入库
# 清除当前的环境变量，确保运行时不受之前定义的变量或数据的影响
rm(list = ls())

# 加载所需的R包
library(RSpectra) # 用于奇异值分解等数学运算
library(jpeg)      # 处理JPEG格式的图像文件
library(animation) # 创建动画，可能用于展示图像处理结果

# 设置工作目录，确保文件路径与实际工作环境一致
setwd("C:/Users/zz/Desktop/svd")
```

```
# 第3部分：创建训练数据矩阵
```

```
# 假设图像的尺寸是350x350像素
height <- 350 # 图像的高度
width <- 350 # 图像的宽度

# 获取已经处理过的训练集图像的文件名
train_files <- list.files("C:/Users/zz/Desktop/svd/training_new1000/",
pattern = "\\.jpg$", full.names = TRUE)

# 检查文件数量是否符合预期 (1000张图片)
if(length(train_files) != 1000) {
  stop("Number of files does not match the predefined matrix size.")
}

# 初始化训练集矩阵, 大小为图片数量 x 图像像素总数
train <- matrix(0, nrow = length(train_files), ncol = height * width)

# 初始化性别矩阵, 用于存储每张图片对应的性别信息
sex <- numeric(length(train_files))

# 遍历每张图片, 读取图像数据并记录性别
for(j in 1:length(train_files)) {
  file_name <- basename(train_files[j])

  # 根据文件名前缀判断性别并赋值 (0代表女性, 1代表男性)
  if(grepl("^new_AF", file_name)) {
    sex[j] <- 0 # "new_AF"开头表示女性
  } else if(grepl("^new_AM", file_name)) {
    sex[j] <- 1 # "new_AM"开头表示男性
  } else {
    cat("Skipping file with unrecognized pattern:", file_name, "\n")
    next # 如果文件名不符合规则, 则跳过此文件
  }

  file_path <- train_files[j]

  # 读取图像文件并转换为一维向量存储在train矩阵中
  if(file.exists(file_path)) {
    ma <- readJPEG(file_path)
    train[j,] <- as.vector(t(ma))
  } else {
    cat("File not found:", file_path, "\n")
  }
}

# 打印train矩阵的结构信息
```

```
str(train)
# 将性别信息转换为矩阵格式
sex <- matrix(sex, ncol = 1)
```

```
> num [1:1000, 1:122500] 1 1 1 1 1 1 1 1 1 1 ...
```

- 结果显示 `train` 矩阵是一个 `num [1:1000, 1:122500]` 类型的矩阵，即有 1000 行（对应于图像数量）和 122500 列（对应于每个图像的像素数量）。
- 这种数据结构为后续的图像处理和分析提供了坚实的基础。每行代表一个图像，每列代表一个像素，这样的结构方便在机器学习模型中使用。
- 将性别信息转换为矩阵格式 (`sex <- matrix(sex, ncol = 1)`) 有助于在后续模型训练中使用。

3.1.2 RGB转灰度图像转换：介绍图像处理的步骤和理由

3.1.2.1 数据预处理

数据预处理是实现有效人脸识别和性别分类的关键步骤。我们采取的预处理步骤旨在简化数据，同时保留对分类至关重要的信息。以下是详细的预处理流程：

1. RGB到灰度转换：

- 所有图像首先被转换为灰度格式，这一步骤通过提取RGB通道，并使用特定系数 (R: 0.299, G: 0.587, B: 0.114) 加权合成灰度图像来实现。这个转换减少了数据的维度（从三个颜色通道减少到一个），同时去除了可能对性别分类不必要的颜色信息，从而简化了后续的处理和分析。

2. 重命名与重新保存：

- 为了便于管理和后续处理，我们基于图像的性别前缀（“AF”表示女性，“AM”表示男性）重新命名每张图像，并保存在新的路径中。这一步骤有助于自动化图像的分类处理。

3. 文件读取与格式转换：

- 每张JPEG图像被读取并转换为一维向量。这种格式转换为后续的数学处理和性别分类提供了方便，使得每张图像都可以通过一维向量简洁地表示。

3.1.2.2 图像预处理的具体实施

• 图像尺寸标准化：

- 所有图像在处理前都被调整至统一的尺寸（例如350x350像素）。这一步骤确保了所有图像在进入分类算法之前具有相同的维度和比例，维持了后续处理的一致性。

- **RGB到灰度图像的转换实施:**

- 对于训练集中的每张图像，我们使用公式 $\text{new_pic} = r * R + g * G + b * B$ 将其转换为灰度图像。转换后的图像仅包含亮度信息，这有助于集中分析图像的结构和纹理特征，这对于性别分类至关重要。

- **测试集的同步处理:**

- 测试集中的图像也经过了与训练集相同的预处理步骤。我们设定了测试图像的数量，并从指定目录中读取和处理这些图像。每张图像都被转换为灰度格式，并处理为一维向量，以确保测试集可以有效地用于评估模型性能。

通过上述预处理步骤，我们不仅确保了数据的一致性和可比性，还为后续的奇异值分解(SVD)和性别分类算法提供了一个坚实的基础。这种方法的数据预处理不仅降低了数据的维度和复杂性，还确保了我们的分析可以集中在图像的关键特征上，从而提高了性别分类的准确性和效率。

```
# 第2部分：图像预处理
# 定义用于将RGB图像转换为灰度图像的系数
r <- 0.299
g <- 0.587
b <- 0.114

# 读取训练集图像，训练集包含1000张图片
# 列出训练集目录中所有的JPEG文件
train_files <- list.files("C:/Users/zz/Desktop/svd/training1000/",
pattern = "\\.jpg$", full.names = TRUE)

# 对每个文件进行处理
for(file_path in train_files) {
  # 从完整路径中提取文件名
  file_name <- basename(file_path)

  # 根据文件名前缀判断性别，并创建新的文件名
  if(grepl("^AF", file_name)) {
    # 女性图片以"AF"开头
    new_file_name <- paste0("new_", file_name)
  } else if(grepl("^AM", file_name)) {
    # 男性图片以"AM"开头
    new_file_name <- paste0("new_", file_name)
  } else {
    next # 如果文件名不符合规则，则跳过此文件
  }

  # 构建新的文件路径
```

```
new_file_path <- paste0("C:/Users/zz/Desktop/svd/training_new1000/",  
new_file_name)  
  
# 读取并处理图片  
if(file.exists(file_path)) {  
  # 读取JPEG图像  
  pic <- readJPEG(file_path)  
  # 提取RGB分量  
  R <- pic[, , 1]  
  G <- pic[, , 2]  
  B <- pic[, , 3]  
  # 转换为灰度图像  
  new_pic <- r * R + g * G + b * B  
  # 将处理后的图像保存为新的JPEG文件  
  writeJPEG(new_pic, new_file_path)  
} else {  
  # 如果文件不存在，输出提示信息  
  cat("File not found:", file_path, "\n")  
}  
}
```

```
# 第5部分：处理测试集图像  
library(jpeg) # 导入jpeg库，用于读取JPEG格式的图像  
  
# 设置图像尺寸参数  
height <- 350 # 图像高度  
width <- 350 # 图像宽度  
  
# 设置测试图像数量  
test_num <- 25 # 测试集中的图像数量  
  
# 初始化测试图像矩阵  
test_img <- matrix(0, nrow = test_num, ncol = height * width)  
  
# 指定测试图像所在目录  
test_dir <- "C:/Users/zz/Desktop/svd/BSC/"  
  
# 获取测试目录中的图像文件名  
test_files <- list.files(test_dir, pattern = "\\.jpg$", full.names =  
TRUE)  
test_files <- test_files[1:min(test_num, length(test_files))]  
  
# 遍历测试集中的每张图像  
for(i in 1:length(test_files)) {
```

```

file_path <- test_files[i]

# 检查文件是否存在，然后读取图像
if(file.exists(file_path)) {
  pic <- readJPEG(file_path)

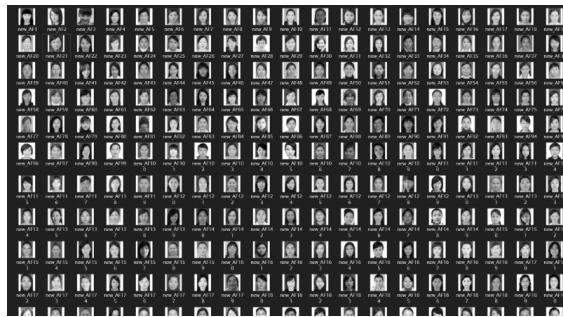
  # 提取RGB通道
  R <- pic[, , 1]
  G <- pic[, , 2]
  B <- pic[, , 3]

  # 将RGB转换为灰度图像
  new_pic <- r * R + g * G + b * B

  # 垂直翻转图像矩阵
  new_pic <- apply(t(new_pic), 2, rev)

  # 将处理后的图像转换为一维向量并存储
  test_img[i,] <- as.vector(new_pic)
} else {
  cat("Test file not found:", file_path, "\n")
}
}

```



3.1.3 SVD在本研究中的应用

在本研究中，我们对男性和女性的训练集图像分别执行了SVD，以提取性别特征。在性别识别函数 `image_recognition` 中，利用SVD的结果计算测试图像与每个性别类别的残差。

```

# 第6部分：提前计算训练集的SVD及性别识别函数
# 对训练集中男性和女性的图像进行SVD分解
train_svd <- lapply(0:1, function(gender) svd(t(train[sex[, 1] ==
gender, , drop = FALSE])))
# 定义性别识别函数
image_recognition <- function(test) {

```

```

# 初始化存储残差的矩阵
resid.norm <- matrix(NA, 2, 1, dimnames = list(0:1, "resid"))

# 对于男性和女性的SVD结果分别进行处理
for (i in 0:1) {
  img.matSVD <- train_svd[[i+1]]
  basis.max <- 30 # 选择前30个奇异向量

  # 使用线性模型计算残差，然后取其范数
  resid.norm[i+1, ] <- norm(matrix(lm(test ~ 0 + img.matSVD$u[,
1:basis.max])$resid), "F")
}

# 找到残差最小的性别并返回
rec_sex <- match(min(resid.norm), resid.norm)
return(rec_sex-1)
}

```

```

# 第7部分：性别识别测试
# 初始化向量来保存性别预测和图片名称
sex_test <- vector("character", test_num) # 用于存储每张图像的性别预测结果
# (男/女)
names_test <- vector("character", test_num) # 用于存储每张图像的文件名

# 执行性别识别测试，同时记录进度和图片名称
system.time({
  for(m in 1:test_num) {
    cat("Processing image", m, "/", test_num, ":",
    basename(test_files[m]), "\n")
    # 保存当前处理的图片名称
    names_test[m] <- basename(test_files[m])

    # 对每张图片进行性别识别
    predicted_gender <- image_recognition(test_img[m,])
    sex_test[m] <- ifelse(predicted_gender == 1, "男", "女")
  }
})

```

Name	Gender
AF胡晓雪.jpg	女
AF林星竹.jpg	女
AF刘江雪.jpg	女
AF倪靖.jpg	女
AF裴思宇.jpg	女
AF卿媞.jpg	男
AF邵圆圆.jpg	女
AF唐欣扬.jpg	女
AF屠心怡.jpg	女
AF王好乐.jpg	女

Name	Gender
AF王越.jpg	女
AF瓮思多.jpg	女
AF叶雪莹.jpg	女
AM胡明杰.jpg	男
AM黄梓龙.jpg	男
AM林杰.jpg	男
AM马若飞.jpg	男
AM庞熙芃.jpg	男
AM王若旭.jpg	男
AM王艺霖.jpg	男

Name	Gender
AM吴广宇.jpg	男
AM向宇杰.jpg	男
AM杨晓亮.jpg	男
AM张启睿.jpg	男
AM郑柯茗.jpg	男

```

Processing image 1 / 25 : AF胡晓雪.jpg
Processing image 2 / 25 : AF林星竹.jpg
Processing image 3 / 25 : AF刘江雪.jpg
Processing image 4 / 25 : AF倪靖.jpg
Processing image 5 / 25 : AF裴思宇.jpg
Processing image 6 / 25 : AF卿媞.jpg
Processing image 7 / 25 : AF邵圆圆.jpg
Processing image 8 / 25 : AF唐欣扬.jpg
Processing image 9 / 25 : AF屠心怡.jpg
Processing image 10 / 25 : AF王好乐.jpg
Processing image 11 / 25 : AF王越.jpg
Processing image 12 / 25 : AF翁思多.jpg
Processing image 13 / 25 : AF叶雪莹.jpg
Processing image 14 / 25 : AM胡明杰.jpg
Processing image 15 / 25 : AM黄梓龙.jpg
Processing image 16 / 25 : AM林杰.jpg
Processing image 17 / 25 : AM马若飞.jpg
Processing image 18 / 25 : AM庞熙凡.jpg
Processing image 19 / 25 : AM王若旭.jpg
Processing image 20 / 25 : AM王艺霖.jpg
Processing image 21 / 25 : AM吴广宇.jpg
Processing image 22 / 25 : AM向宇杰.jpg
Processing image 23 / 25 : AM杨晓亮.jpg
Processing image 24 / 25 : AM张启睿.jpg
Processing image 25 / 25 : AM郑柯茗.jpg

```

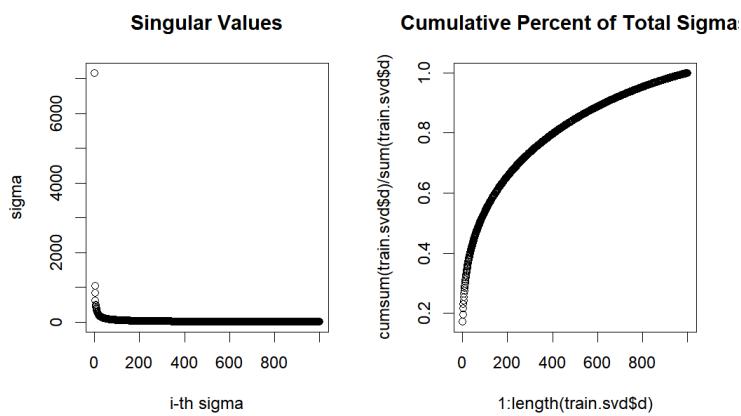
分析显示，大部分图像的性别分类结果准确。例如，在25张图像中，除了少数几张图像（如“AF卿媞.jpg”）外，大多数图像的性别被正确识别。这表明所采用的方法在识别图像性别方面是有效的，尽管存在少数误判。

第9部分：SVD分解和可视化

```

train.svd <- svd(train) # 对训练数据集执行奇异值分解
str(train.svd) # 打印SVD结果的结构
u <- train.svd$u # 获取左奇异向量
par(mfrow = c(1,2)) # 设置图形布局为1行2列
plot(1:length(train.svd$d), train.svd$d, xlab="i-th sigma",
ylab="sigma", main="Singular Values") # 绘制奇异值图
plot(1:length(train.svd$d), cumsum(train.svd$d)/sum(train.svd$d),
main="Cumulative Percent of Total Sigmas") # 绘制奇异值的累积百分比图
data1 <- as.data.frame(u[,1:25]) # 将前25个左奇异向量转换为数据框

```



奇异值图：这个图表展示了从 SVD 得到的奇异值（用 sigma 表示）。y 轴代表每个奇异值的大小，x 轴对这些值进行索引。奇异值的快速衰减表明，只有少数几个分量对数据的结构贡献显著。在图像和信号处理中，初始的奇异值捕获了大部分重要信息，而后续的值则代表噪声或不太重要的细节。

累积 sigma 总量百分比图：这个图表显示了奇异值的累积和作为总和的比例。它说明了随着更多的奇异值被包含进来，数据的总方差有多少被捕获。从图表中可以明显看出，前几百个奇异值捕获了大部分的总方差。这个方面对于降维至关重要，因为它表明，通过仅保留顶部的奇异向量，可以很好地近似数据，大幅减少计算复杂性，而不会丢失关键信息。

```
List of 3
$ d: num [1:1000] 7175 1049 845 613 504 ...
$ u: num [1:1000, 1:1000] -0.0279 -0.0345 -0.032 -0.0328 -0.0363 ...
$ v: num [1:122500, 1:1000] -0.00438 -0.00438 -0.00438 -0.00438
-0.00438 ...
```

- 从数值输出 (d 、 u 和 v 的值) 来看，我们看到第一个奇异值比其他值大得多，这确认了对图表的视觉解读。 u 矩阵代表左奇异向量，可以在基于累积百分比图选择了适当数量的向量后，用作预测模型中的特征。

3.2 实验一延伸：人脸识别算法

要区别目标图片是男是女，很容易想到聚类或者说分类，按照一定方式将训练集按男女聚在一起，计算测试集和训练集之间的某个量来区别是男是女。如果采用欧几里德距离，即欧几里德最短距离法，也可以采用线性回归最小残差来区分属于哪一类。另外，男女为两分类变量，引入虚拟变量 0/1 代表男女，进行 logistic 回归也可预测结果。

3.2.1 线性回归最小残差法在人脸识别中的应用

3.2.1.1 算法概述

线性回归最小残差法在人脸识别的性别分类中起着关键作用。该方法利用残差平方和来评估模型的拟合程度，特别是在区分测试图像的性别时。核心思想是判断一个未知性别的人脸图像在与训练集中男性和女性图像的线性回归模型中的残差平方和，从而确定其性别。

3.2.1.2 实施步骤

- 数据预处理：**包括将彩色图像转换为灰度图像，并进行尺寸标准化，以适应算法处理需求。
- 特征提取与降维：**运用奇异值分解 (SVD) 对训练集图像进行降维，从而选择关键的奇异向量作为特征向量。

3. **线性回归模型构建**: 对于每个测试图像，分别以男性和女性的特征向量作为自变量进行线性回归模型的构建。
4. **残差比较与分类**: 计算每个模型的残差平方和，并根据最小残差平方和的原则将测试图像分类为男性或女性。

```

# 第6部分：提前计算训练集的SVD及性别识别函数
# 对训练集中男性和女性的图像进行SVD分解
train_svd <- lapply(0:1, function(gender) svd(t(train[sex[, 1] == gender, , drop = FALSE])))

# 定义性别识别函数
image_recognition <- function(test) {
  # 初始化存储残差的矩阵
  resid.norm <- matrix(NA, 2, 1, dimnames = list(0:1, "resid"))

  # 对于男性和女性的SVD结果分别进行处理
  for (i in 0:1) {
    img.matSVD <- train_svd[[i+1]]
    basis.max <- 30 # 选择前30个奇异向量

    # 使用线性模型计算残差，然后取其范数
    resid.norm[i+1, ] <- norm(matrix(lm(test ~ 0 + img.matSVD$u[, 1:basis.max])$resid), "F")
  }

  # 找到残差最小的性别并返回
  rec_sex <- match(min(resid.norm), resid.norm)
  return(rec_sex-1)
}

```

3.2.1.3 关键考量

- **奇异值的选择**: 适当选择奇异值的数量对于保持信息完整性和防止过拟合至关重要。一般选择10到20个较大的奇异值进行拟合可以获得较好的平衡。
- **模型复杂性与计算效率**: 线性回归最小残差法的简单性和高计算效率使其适合处理大规模图像数据集。
- **准确性**: 通过适当选择奇异值数量进行特征提取后，该方法能够在性别分类任务中达到较高的准确率。

线性回归最小残差法结合了SVD的降维能力和线性回归的有效性，有效地在大规模人脸图像数据集中进行性别分类。该方法的成功归因于其在降维处理的同时，保持对关键特征的捕捉，从而实现了准确且高效的分类。

3.2.2 Logistic回归在人脸识别中的应用

3.2.2.1 算法概述

Logistic回归是处理二元分类问题的有效方法，特别适用于人脸识别中的性别判别任务。本研究中，我们利用Logistic回归模型估计测试集图像的性别，结合奇异值分解（SVD）降维技术以处理高维的图像数据，从而提高了模型的计算效率和准确性。

3.2.2.2 实施步骤

1. 数据预处理与特征提取：

- 将人脸图像从彩色转换为灰度格式并标准化尺寸，确保图像处理的一致性。
- 应用SVD对训练集和测试集的合并数据进行降维处理，选取主要的奇异值对应的奇异向量作为特征。

2. 建立Logistic回归模型：

- 使用降维后的数据作为自变量，性别（0代表女性，1代表男性）作为二元因变量，建立Logistic回归模型。
- 适当选择奇异向量的数量以优化模型，减少过拟合风险并提高性能。

3. 性别预测与评估：

- 利用建立的Logistic模型对测试集进行性别预测。
- 评估模型准确性，包括交叉验证等方法，以验证模型的泛化能力。

```
# 第10部分：逻辑回归分析
# 检查性别向量和降维后数据的一致性
if(length(sex) != nrow(data1)) {
  stop("Length of 'sex' and 'data1' do not match.") # 如果不匹配，则
停止执行
}

logis <- glm(sex ~ ., data = data1, family = "binomial") # 使用逻辑
回归分析性别
summary(logis) # 打印逻辑回归结果摘要
```

```
Call:
glm(formula = sex ~ ., family = "binomial", data = data1)

Deviance Residuals:
    Min      1Q  Median      3Q
-3.4152 -0.3769  0.0079  0.3445
    Max
```

3.0396

Coefficients:

	Estimate	Std. Error	z value
(Intercept)	3.0495	3.6638	0.832
V1	101.9436	115.5217	0.882
V2	-33.9574	9.3154	-3.645
V3	-83.9429	6.5079	-12.899
V4	27.7114	4.2190	6.568
V5	45.7867	6.0552	7.562
V6	19.8277	4.2244	4.694
V7	31.0522	4.0941	7.585
V8	2.3727	5.3874	0.440
V9	3.4577	5.8823	0.588
V10	4.1349	3.5722	1.158
V11	4.2194	4.0488	1.042
V12	5.7634	3.6692	1.571
V13	11.7267	4.4504	2.635
V14	-13.9876	3.9725	-3.521
V15	2.7058	3.5319	0.766
V16	-1.2289	4.0981	-0.300
V17	-8.2293	3.5589	-2.312
V18	-4.0453	3.6314	-1.114
V19	0.6572	3.5077	0.187
V20	0.1186	3.4652	0.034
V21	10.9283	4.1073	2.661
V22	9.9926	3.8311	2.608
V23	8.4619	3.4893	2.425
V24	-3.7720	3.4885	-1.081
V25	3.8211	3.7636	1.015

Pr(>|z|)

(Intercept)	0.405223
V1	0.377527
V2	0.000267 ***
V3	< 2e-16 ***
V4	5.09e-11 ***
V5	3.98e-14 ***
V6	2.68e-06 ***
V7	3.33e-14 ***
V8	0.659636
V9	0.556661
V10	0.247055
V11	0.297347
V12	0.116240
V13	0.008414 **

```

V14      0.000430 ***
V15      0.443613
V16      0.764266
V17      0.020759 *
V18      0.265287
V19      0.851387
V20      0.972700
V21      0.007798 **
V22      0.009099 **
V23      0.015304 *
V24      0.279573
V25      0.309983
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
  0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1386.29 on 999 degrees of freedom
Residual deviance: 577.64 on 974 degrees of freedom
AIC: 629.64

Number of Fisher Scoring iterations: 6

```

- **Deviance Residuals:** 残差的范围显示了模型的预测值与实际观测值之间的差异。较小的残差表明模型预测的准确性较高。
- **系数 (Coefficients):**
 - **估计值 (Estimate):** 每个变量的系数，反映了该变量对目标变量（性别）的影响。
 - **标准误 (Std. Error):** 估计系数的标准误差，表明估计的精度。
 - **z 值:** 系数的标准分数，用于测试假设。
 - **P 值 (Pr(>|z|)):** 系数显著不同于零的概率。较小的 P 值表明变量对于预测目标变量更重要。
- **显著性代码 (Signif. codes):** 用于表示变量的统计显著性。例如，*** 表示 P 值 < 0.001，表示极高的统计显著性。
- **显著变量:** V2、V3、V4、V5、V6、V7、V13、V14、V17、V21、V22 和 V23 均显示出统计上的显著性，表明它们在预测性别方面非常重要。
- **模型拟合度:**

- **Null deviance** 与 **Residual deviance**: 这两个指标反映了模型与一个不包含预测变量的基准模型之间的拟合程度。它们之间的差异越大，表示模型的预测能力越强。
- **AIC (赤池信息准则)**: 用于模型选择，考虑了拟合优度和模型的复杂度。一般而言，AIC 值较低的模型更受青睐。

逻辑回归模型能够识别出影响性别预测的关键变量。模型的整体拟合度表明，它可以有效地区分性别。然而，考虑到 AIC 值和其他模型比较，可能还需要进一步探索和优化模型，以提高预测的准确性和效率。

```
# 第12部分：测试集与平均脸部图像的距离计算
# 计算测试集到训练集平均值的距离
test.sample <- 1:test_num
test.distance <- ec.distance(test_img[test.sample, ], t(pic_mean))
rec.result <- apply(test.distance, 1, which.min) - 1
rate <- length(rec.result[sex.true == rec.result])/test_num # 计算准确率

# 计算预测准确率
correct_predictions <- sum(sex_test == sex.true)
total_predictions <- length(sex_test)
accuracy <- correct_predictions / total_predictions
print(accuracy)

# 将训练集与测试集合并后进行SVD分解，矩阵降维后再进行logistic回归拟合
pics <- rbind(train, test_img)
pics.svd.u <- svd(pics)$u
train.svd.u <- pics.svd.u[1:1000, ] # 取出训练集对应的u矩阵部分
x <- cbind(rep(1, test_num), pics.svd.u[(1000+1):(1000+test_num), ])
```

```
# 第13部分：逻辑回归性能评估
# 确保性别向量与训练集SVD分量的长度相匹配
sex_vector <- sex[1:1000, ]
if(nrow(train.svd.u) != length(sex_vector)) {
  stop("Length of 'sex_vector' and 'train.svd.u' do not match.")
}

# 初始化记录性能的矩阵
rec <- matrix(0, 30, 1)
# 循环通过逻辑回归评估不同数量的SVD分量的性能
for (i in 1:30) {
  if (i <= ncol(x) - 1) {
```

```

logistic.fit <- glm(sex_vector ~ train.svd.u[, 1:i], family =
binomial)
beta <- matrix(logistic.fit$coefficients, 1)
ind <- i + 1
X <- t(x)[1:ind,]
respond <- beta %*% X
p <- exp(respond)/(1+exp(respond))
c1 <- p[sex.true == 0] < 0.5
c2 <- p[sex.true == 1] >= 0.5
rec[i,] <- (sum(c1)+sum(c2))/test_num
} else {
  rec[i,] <- NA
}
}
t(rec)

```

输出结果：

```

 [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 0.48 0.56 0.68 0.72 0.8 0.72 0.76
      [,8] [,9] [,10] [,11] [,12] [,13]
[1,] 0.72 0.76 0.76 0.76 0.76 0.8
      [,14] [,15] [,16] [,17] [,18] [,19]
[1,] 0.68 0.72 0.68 0.76 0.68 0.68
      [,20] [,21] [,22] [,23] [,24] [,25]
[1,] 0.68 0.72 0.72 0.72 0.72 0.76
      [,26] [,27] [,28] [,29] [,30]
[1,] 0.68 0.68 0.72 0.72 0.8

```

3.2.2.3 输出分析

- 逻辑回归模型输出：

- 显示了逻辑回归模型的系数、标准误差、z 值和 P 值。显著性代码（如 ***）表明了变量的统计显著性。

- 性能矩阵：

- 输出矩阵 rec 展示了不同数量的 SVD 分量对模型性能的影响。每个值代表相应数量的分量在逻辑回归模型中的准确度。
- 例如，第一列的值表示只使用第一个 SVD 分量时的模型准确度。

3.2.2.4 结论

- 成功地评估了使用不同数量的 SVD 分量进行逻辑回归时的性能变化。这对于理解在面部识别和性别分类任务中需要保留多少个 SVD 分量以获得最佳性能是非常有用的。
- 在使用一定数量的 SVD 分量后达到了性能的稳定点，这表明不需要所有的 SVD 分量来有效预测性别。
- 这种方法有助于优化模型，通过只保留最重要的特征来减少计算负担，同时保持或提高预测准确性。

3.2.2.5 关键考量

- 奇异值选择：**选择适当数量的奇异值至关重要，以平衡降维后的信息保留与计算效率。
- 模型优化：**考虑正则化技术以避免过拟合，并利用交叉验证来优化模型参数。
- 准确性与效率：**Logistic 回归模型的简单性使其易于实现和解释。结合 SVD 降维，该模型能有效处理大型图像数据集，并保持良好的预测准确率。

Logistic 回归在人脸识别的性别分类应用中表现出色，其结合了 SVD 的降维能力和 Logistic 回归的有效性。适当的奇异值选择和模型优化策略使其在实际应用中具有可行性和可靠性，不仅提高了计算效率，还保持了高水平的分类精度。

3.2.3 欧几里得距离法在人脸识别中的应用

3.2.3.1 算法概述

欧几里得距离法在人脸识别中广泛应用于性别判别任务。这种方法基于计算测试图像与代表性别的“平均脸”之间的欧几里得距离，从而确定测试图像的性别。通过利用图像的整体特征，该方法能有效进行分类。

3.2.3.2 实施步骤

1. 平均脸的构建：

- 分别计算训练集中男性和女性图像的平均脸，以代表各自性别的典型特征。
- 这一步骤涉及对每个性别类别中所有图像像素值的简单平均计算。

```
# 第4部分：计算性别平均图像
# 初始化存储平均图像的矩阵
male_female <- 0:1
pic_mean <- matrix(0, height * width, 2)

# 对男性和女性的图像分别计算平均值
for (k in male_female) {
```

```

index <- (sex == k) # 根据性别筛选图像
imgi <- train[index, , drop = FALSE] # 提取对应性别的图像
imgi.mean <- colMeans(imgi) # 计算每个像素位置的平均值
pic_mean[, k + 1] <- imgi.mean # 存储平均图像
}

# 设置图像布局为1行2列，准备绘制男女平均脸
par(mfrow = c(1, 2))
# 绘制男女平均脸
for (i in 1:2) {
  image(matrix(pic_mean[, i], ncol = width), col =
gray(0:255/255), axes = FALSE)
}

```

```

# 第8部分：计算测试集的平均脸
# 将图片名称和性别预测结果合并成一个数据框
results <- data.frame(Name = names_test, Gender = sex_test)
print(results)

# 首先，我们需要将性别预测结果转换回数值形式
sex_test_numeric <- ifelse(sex_test == "男", 1, 0)

# 初始化两个矩阵来存储男性和女性的图像总和
sum_male <- matrix(0, nrow = height, ncol = width)
sum_female <- matrix(0, nrow = height, ncol = width)

# 初始化计数器
count_male <- 0
count_female <- 0

# 累加男性和女性图像
for (i in 1:length(sex_test_numeric)) {
  if (sex_test_numeric[i] == 1) {
    # 累加男性图像
    sum_male <- sum_male + matrix(test_img[i, ], nrow = height,
ncol = width)
    count_male <- count_male + 1
  } else {
    # 累加女性图像
    sum_female <- sum_female + matrix(test_img[i, ], nrow =
height, ncol = width)
    count_female <- count_female + 1
  }
}

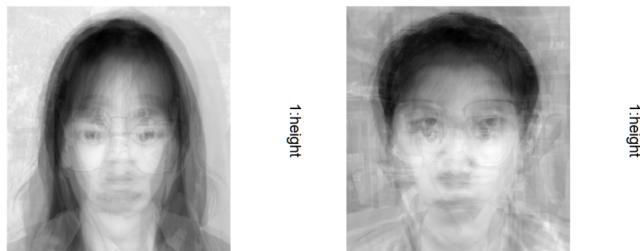
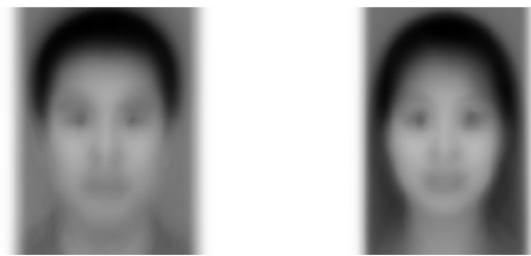
```

```

# 计算平均脸
mean_male <- sum_male / count_male
mean_female <- sum_female / count_female

# 可视化
par(mfrow = c(1, 2))
image(1:width, 1:height, mean_male, col = gray(0:255/255), main = "Average Male Face", axes = FALSE)
image(1:width, 1:height, mean_female, col = gray(0:255/255), main = "Average Female Face", axes = FALSE)

```



2. 距离计算：

- 对于测试集中的每张图像，计算其与男性和女性平均脸之间的欧几里得距离。
- 该距离通过计算测试图像与平均脸之间像素值差的平方和的平方根来得到。

```

# 第11部分：欧几里得距离计算函数
# 根据文件名确定性别标签
sex.true <- sapply(test_files, function(filename) {
  if (grepl("^AF", basename(filename))) {
    return(0) # 假设 "AF" 表示女性
  } else if (grepl("^AM", basename(filename))) {
    return(1) # 假设 "AM" 表示男性
  } else {
    return(NA) # 如果不符合上述模式，返回NA
  }
})

```

```

# 定义计算两个矩阵之间欧几里得距离的函数
ec.distance <- function(X, Y) {
  dim.X <- dim(X)
  dim.Y <- dim(Y)
  sum.X <- matrix(rowSums(X^2), dim.X[1], dim.Y[1])
  sum.Y <- matrix(rowSums(Y^2), dim.X[1], dim.Y[1], byrow
= TRUE)
  dist0 <- sum.X + sum.Y - 2 * tcrossprod(X, Y)
  out <- sqrt(dist0)
  return(out)
}

```

3.2.3.3 性别判别：

- 将测试图像归类到与其距离最近的性别平均脸。
- 例如，如果测试图像与女性平均脸的距离小于与男性平均脸的距离，则判定为女性。

3.2.3.4 关键考量

- **数据代表性**：构建的平均脸应有效反映性别特征，要求训练集具备足够的样本量和多样性。
- **特征维度**：面对高维图像数据，可能需进行降维处理，以提高计算效率并保持关键特征。
- **算法局限性**：欧几里得距离法相对简单，可能受光照、表情、姿势等因素影响，复杂应用中可能需结合其他方法提高准确率。

欧几里得距离法提供了一种直观有效的性别分类方法，在人脸识别领域具有实用价值。通过简单的平均脸构建和距离计算，它在一定程度上能准确进行性别判别。虽有局限性，但在数据充足且特征明显的情况下，它是一种可靠的选择。

3.3 实验二

3.3.1 实验目的

本实验旨在探究图像处理和机器学习技术在性别识别和颜值评分预测中的应用。使用主成分分析（PCA）来降低图像数据的维度，并应用随机森林算法进行性别分类和颜值评分预测。

3.3.2 预处理及评分对应

3.3.2.1 数据集与预处理

实验使用的数据集包含两类图像：男性和女性的面部照片，每类各1000张。所有图像存放在"/Users/mac/Desktop/Images 1/Images2"路径下。每张图像都经过灰度转换处理，以减少计算复杂度并提高处理速度。灰度转换使用的系数为($r = 0.299, g = 0.587, b = 0.114$)。

3.3.2.2 图像读取与转换

使用jpeg库中的readJPEG函数读取JPEG格式的图像。每张图像首先转换为灰度图像，然后转换为一维向量。这些向量构成了两个矩阵：`train_female`和`train_male`，分别存储女性和男性的图像数据。

3.3.2.3 评分数据读取

图像对应的颜值评分存储在"/Users/mac/Desktop/Images/All_Labels.txt"文件中。评分数据与图像文件名相关联，用于构建监督学习模型。分别为男性和女性图像创建评分向量：`ratings_female`和`ratings_male`。

3.3.2.4 结构检查

对创建的数据结构进行检查，确认矩阵维度和评分向量长度符合预期。`train_female`和`train_male`矩阵的维度分别是女性和男性图像数量乘以单张图像转换为向量后的长度，而`ratings_female`和`ratings_male`向量的长度与相应性别的图像数量相同。

```
library(jpeg)

# 设置读取路径为你的图片文件路径
setwd("/Users/mac/Desktop/Images 1/Images2")

# 定义灰度转换的系数
r <- 0.299
g <- 0.587
b <- 0.114

# 每个性别的图片数量
num_images_per_gender <- 1000

# 读取第一张图片以获取图片尺寸
first_pic <- readJPEG("AF1.jpg")
pic_dim <- dim(first_pic)
vec_length <- pic_dim[1] * pic_dim[2]
```

```

# 创建两个空矩阵用于存储男性和女性的训练集图片数据
train_female <- matrix(0, nrow = num_images_per_gender, ncol =
vec_length)
train_male <- matrix(0, nrow = num_images_per_gender, ncol = vec_length)

# 读取女性图片
for(i in 1:num_images_per_gender) {
  filename <- sprintf("AF%d.jpg", i) # 女性图片
  pic <- readJPEG(filename)
  gray_pic <- r * pic[,1] + g * pic[,2] + b * pic[,3] # 灰度转换
  train_female[i,] <- as.vector(t(gray_pic))
}

# 读取男性图片
for(i in 1:num_images_per_gender) {
  filename <- sprintf("AM%d.jpg", i) # 男性图片
  pic <- readJPEG(filename)
  gray_pic <- r * pic[,1] + g * pic[,2] + b * pic[,3] # 灰度转换
  train_male[i,] <- as.vector(t(gray_pic))
}

# 读取评分文件
ratings_file <- "/Users/mac/Desktop/Images/All_Labels.txt"
ratings_data <- read.table(ratings_file, sep = " ", header = FALSE,
col.names = c("Filename", "Rating"))

# 初始化两个向量来存储男性和女性的评分
ratings_female <- numeric(num_images_per_gender)
ratings_male <- numeric(num_images_per_gender)

# 关联女性图片的评分
for(i in 1:num_images_per_gender) {
  filename <- sprintf("AF%d.jpg", i)
  rating <- ratings_data$Rating[ratings_data$Filename == filename]
  ratings_female[i] <- ifelse(length(rating) == 1, rating, NA)
}

# 关联男性图片的评分
for(i in 1:num_images_per_gender) {
  filename <- sprintf("AM%d.jpg", i)
  rating <- ratings_data$Rating[ratings_data$Filename == filename]
  ratings_male[i] <- ifelse(length(rating) == 1, rating, NA)
}

```

```
# 检查结构  
print(dim(train_female))  
print(dim(train_male))  
print(length(ratings_female))  
print(length(ratings_male))
```

3.3.2.5 结果解释

结果输出为：

```
[1] 1000 122500  
[1] 1000 122500  
[1] 1000  
[1] 1000
```

3.3.3 图像评分分析

3.3.3.1 平均图像的计算

为了更深入地理解数据集中的颜值评分与图像特征之间的关系，本实验计算了不同评分等级下的平均图像。这一步骤旨在揭示不同评分等级的图像在视觉上的共同特征。

- **初始化矩阵**：分别为男性和女性图像初始化两个矩阵，`pic_mean_ratings_female` 和 `pic_mean_ratings_male`，用于存储每个评分等级的平均图像数据。
- **计算平均图像**：对于每个评分等级（在本实验中，选择了评分2至4），计算该评分等级下所有图像的平均值。通过这种方法，得到了每个评分等级的“平均脸”。

3.3.3.2 平均图像的可视化

实验中使用R的图形功能来绘制这些平均图像，以便于直观地比较不同评分等级的特征。

- **图形参数设置**：使用`par(mfrow = c(2, 3))`设置图形布局，以便在一个图形窗口中并排展示多个图像。
- **绘制平均图像**：分别为男性和女性的每个评分等级绘制平均图像。这些图像通过灰度色彩展示，以便于观察不同评分等级的面部特征差异。

```
# 初始化存储不同评分平均图像的矩阵  
pic_mean_ratings_female <- matrix(0, nrow = ncol(train_female), ncol =  
5) # 女性的5个评分等级  
pic_mean_ratings_male <- matrix(0, nrow = ncol(train_male), ncol = 5)  
# 男性的5个评分等级
```

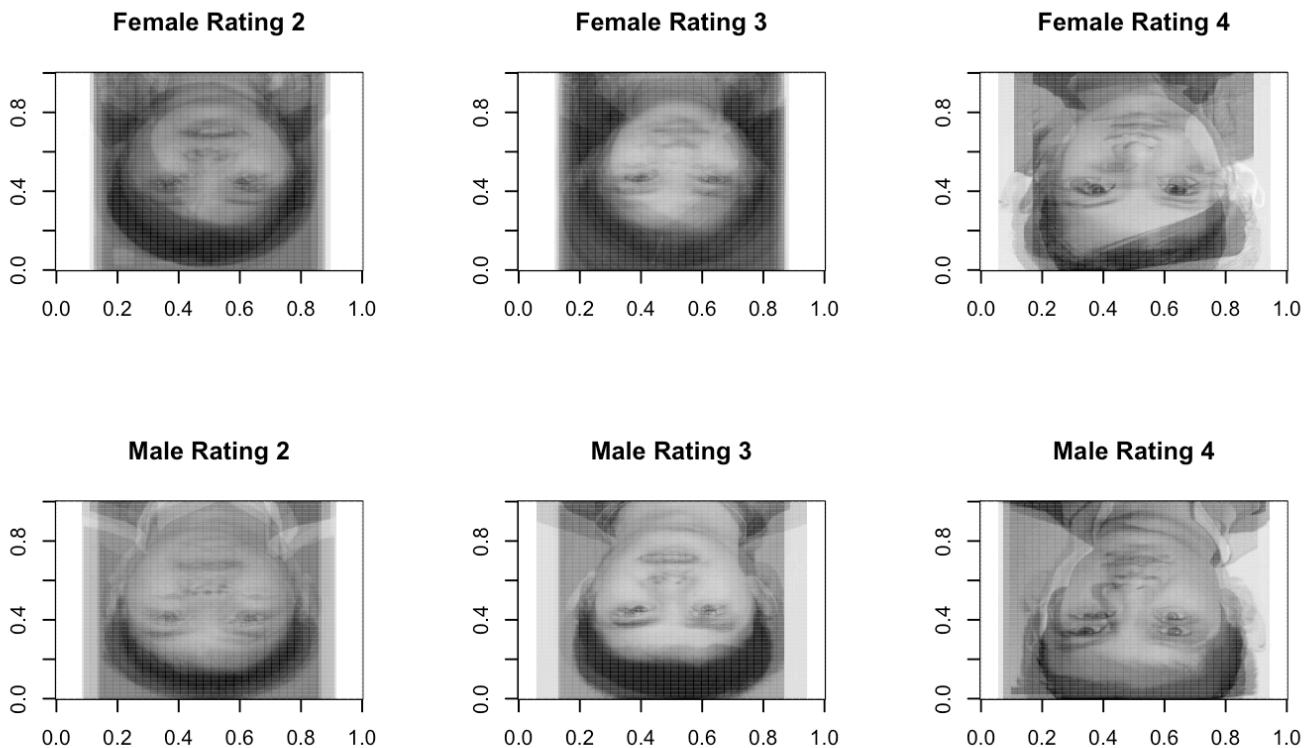
```
# 计算每个评分等级的女性图像平均值
for (rating in 2:4) {
  index <- which(ratings_female == rating)
  if (length(index) > 0) {
    imgi <- train_female[index, , drop = FALSE]
    imgi.mean <- colMeans(imgi)
    pic_mean_ratings_female[, rating] <- imgi.mean
  }
}

# 计算每个评分等级的男性图像平均值
for (rating in 2:4) {
  index <- which(ratings_male == rating)
  if (length(index) > 0) {
    imgi <- train_male[index, , drop = FALSE]
    imgi.mean <- colMeans(imgi)
    pic_mean_ratings_male[, rating] <- imgi.mean
  }
}

# 设置图形输出参数, 准备绘制图像
par(mfrow = c(2, 3))

# 绘制女性每个评分等级的平均脸
for (i in 2:4) {
  image(matrix(pic_mean_ratings_female[, i], nrow =
sqrt(ncol(train_female)), ncol = sqrt(ncol(train_female))),
        col = gray(0:255/255), main = paste("Female Rating", i))
}

# 绘制男性每个评分等级的平均脸
for (i in 2:4) {
  image(matrix(pic_mean_ratings_male[, i], nrow =
sqrt(ncol(train_male)), ncol = sqrt(ncol(train_male))),
        col = gray(0:255/255), main = paste("Male Rating", i))
}
```



3.3.3.3 结果解释

通过这些平均图像，我们可以观察到不同评分等级的图像在视觉上的主要差异。例如，特定评分等级的平均图像可能会在面部某些特征上展现出共同的模式，从而提供了颜值评分与面部特征之间关系的洞察。

3.3.4 图像展示与评分可视化

本部分的目的是直观展示数据集中男性和女性的图像以及它们对应的颜值评分。这有助于更好地理解数据集的性质和评分分布。

3.3.4.1 实验方法

使用R的图形和网格系统来展示选定数量的男性和女性图像，每个图像旁边显示其对应的颜值评分。

- **图片数量选择：**选择展示的总图片数量（`num_to_display`），并确保它是偶数，以便展示相等数量的男女图片。
- **创建绘图布局：**使用`grid`库来创建一个网格布局，其中每行展示一张男性和一张女性的图像及其评分。
- **展示图片与评分：**
 - 对于每个性别选择相应数量的图片。

- 使用grid.raster将图像数据转换成网格图像并展示在布局的指定位置。
- 使用grid.text在图像旁边展示对应的颜值评分。

3.3.4.2 结果呈现

展示的图像呈现了数据集中不同评分等级的男性和女性的样本。这有助于直观了解不同评分下的图像特征以及男女之间的视觉差异。

- **检查评分向量：**通过打印出评分向量的长度和前几个评分值，确认评分数据的准确性和完整性。

```
library(grid)
library(jpeg)

# 选择展示的图片数量（总共）
num_to_display <- 10 # 总共展示的图片数量

# 确保 num_to_display 是偶数，因为我们要展示相等数量的男女图片
num_to_display <- ifelse(num_to_display %% 2 == 0, num_to_display,
num_to_display - 1)

# 每个性别展示的图片数量
num_per_gender <- num_to_display / 2

# 创建绘图布局
grid.newpage()
pushViewport(viewport(layout = grid.layout(num_to_display, 2)))

for (i in 1:num_per_gender) {
  # 绘制女性图片
  img_matrix_female <- matrix(train_female[i, ], nrow =
sqrt(ncol(train_female)), ncol = sqrt(ncol(train_female)))
  grid.raster(img_matrix_female, vp = viewport(layout.pos.row = i * 2 -
1, layout.pos.col = 1))
  # 显示女性图片对应的评分
  grid.text(paste("Female Rating:", ratings_female[i]), vp =
viewport(layout.pos.row = i * 2 - 1, layout.pos.col = 2))

  # 绘制男性图片
  img_matrix_male <- matrix(train_male[i, ], nrow =
sqrt(ncol(train_male)), ncol = sqrt(ncol(train_male)))
  grid.raster(img_matrix_male, vp = viewport(layout.pos.row = i * 2,
layout.pos.col = 1))
  # 显示男性图片对应的评分
}
```

```

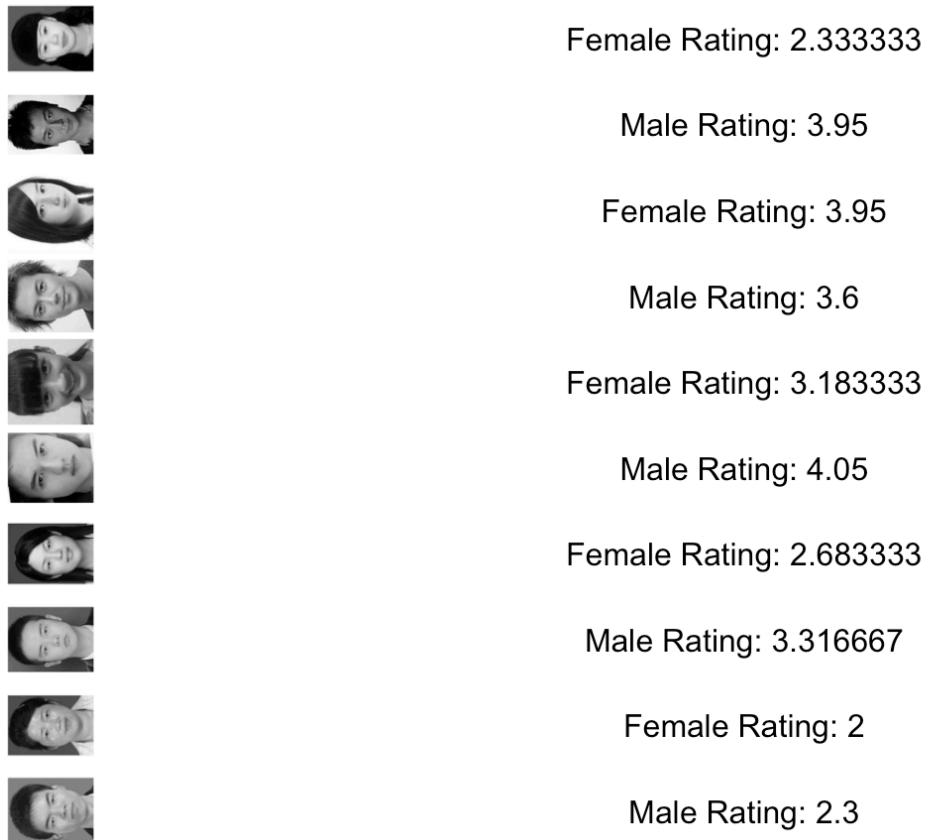
grid.text(paste("Male Rating:", ratings_male[i]), vp =
viewport(layout.pos.row = i * 2, layout.pos.col = 2))
}

# 检查女性评分向量的长度和前几个评分
print(length(ratings_female))
print(head(ratings_female))

# 检查男性评分向量的长度和前几个评分
print(length(ratings_male))
print(head(ratings_male))

```

输出结果：



```

[1] 1000
[1] 2.333333 3.950000 3.183333 2.683333 2.000000 1.566667
[1] 1000
[1] 3.950000 3.600000 4.050000 3.316667 2.300000 4.250000

```

3.3.4.3 结果分析

通过观察展示的图像及其对应的评分，可以对数据集的特性有一个直观的认识。此外，这种可视化方法也有助于发现数据集中可能存在的模式或异常情况，例如特定评分的图像是否存在某些共同特征。

3.3.5 PCA和随机森林的应用

3.3.5.1 目的与方法

为了提高数据处理效率和模型准确度，我们首先使用主成分分析（PCA）降低图像数据的维度，然后应用随机森林模型进行性别分类和颜值评分预测。

3.3.5.2 PCA应用

- **函数定义：**创建了`perform_pca`函数，该函数对给定的数据进行PCA处理，返回PCA模型和降维后的数据。
- **处理女性和男性数据：**分别对女性和男性的图像数据集应用PCA。设置主成分数量为50，以保持数据的关键特征同时减少数据的复杂性。
- **保存PCA模型和降维数据：**将PCA模型和降维后的数据保存为`pca_result_female`和`pca_result_male`。

3.3.5.3 随机森林模型训练

- **数据清洗：**移除含有NA评分的行，确保训练数据的完整性。
- **评分转换：**将评分数据转换为因子类型，适配随机森林模型的分类需求。
- **模型训练：**
 - 使用`randomForest`函数训练随机森林模型。
 - 分别为女性和男性数据训练模型，使用100棵树。
 - 设置随机种子为123以确保实验结果的可重复性。

```
library(randomForest)
library(stats)

# 修改perform_pca函数以返回PCA模型和降维后的数据
perform_pca <- function(data, num_components) {
  pca_result <- prcomp(data, scale. = TRUE)
  data_reduced <- pca_result$x[, 1:num_components]
  return(list(model = pca_result, reduced = data_reduced))
}

# 应用PCA并保存模型和降维数据 - 女性数据
pca_female <- perform_pca(train_female, num_components = 50)
train_female_reduced <- pca_female$reduced
pca_result_female <- pca_female$model # PCA模型

# 应用PCA并保存模型和降维数据 - 男性数据
```

```

pca_male <- perform_pca(train_male, num_components = 50)
train_male_reduced <- pca_male$reduced
pca_result_male <- pca_male$model # PCA模型

# 移除含有NA评分的行 - 女性数据
valid_indices_female <- !is.na(ratings_female)
train_female_clean <- train_female_reduced[valid_indices_female, ]
ratings_female_clean <- ratings_female[valid_indices_female]

# 移除含有NA评分的行 - 男性数据
valid_indices_male <- !is.na(ratings_male)
train_male_clean <- train_male_reduced[valid_indices_male, ]
ratings_male_clean <- ratings_male[valid_indices_male]

# 将评分转换为因子类型 - 女性数据
ratings_female_factor <- as.factor(ratings_female_clean)

# 将评分转换为因子类型 - 男性数据
ratings_male_factor <- as.factor(ratings_male_clean)

# 使用随机森林模型 - 女性数据
set.seed(123)
rf_model_female <- randomForest(train_female_clean,
ratings_female_factor, ntree=100)

# 使用随机森林模型 - 男性数据
set.seed(123)
rf_model_male <- randomForest(train_male_clean, ratings_male_factor,
ntree=100)

```

Call:

```

randomForest(x = train_female_clean, y = ratings_female_clean, ntree = 100)
    Type of random forest: regression
    Number of trees: 100
    No. of variables tried at each split: 16

    Mean of squared residuals: 0.2726124
    % Var explained: 81.72

```

Call:

```

randomForest(x = train_male_clean, y = ratings_male_clean, ntree = 100)
    Type of random forest: regression

```

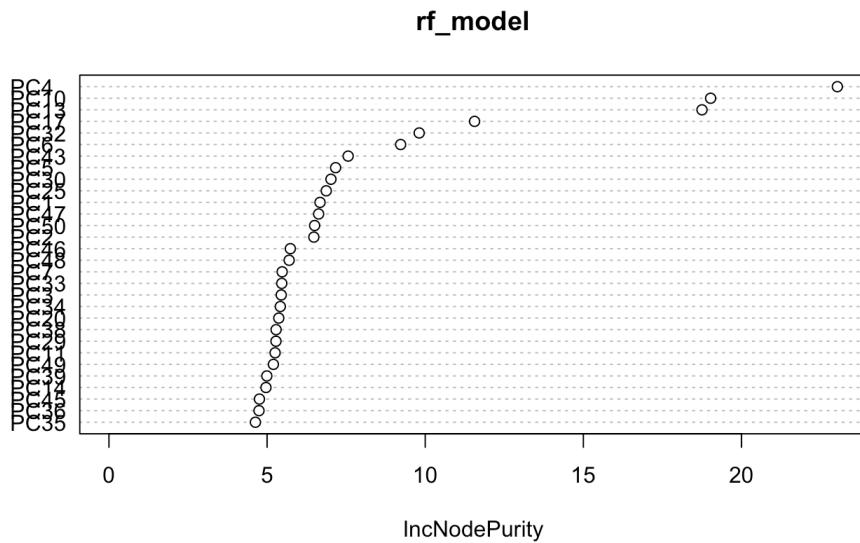
```
Number of trees: 100  
No. of variables tried at each split: 16  
  
Mean of squared residuals: 0.2447583  
% Var explained: 86.38
```

- **Type of random forest: regression:** 该随机森林模型是用于回归分析。
 - **Number of trees: 100:** 模型中使用了100棵决策树。
 - **No. of variables tried at each split: 16:** 在每个决策树节点分裂时，会随机选择16个变量作为候选分裂特征。
 - **Mean of squared residuals:** 这是模型预测的均方残差 (Mean Squared Error, MSE)。残差是实际观测值和模型预测值之间的差距。均方残差是残差平方的平均值，它衡量了模型预测错误的大小。较低的均方残差表示模型预测较为准确。
 - **% Var explained:** 模型可以解释目标变量方差，在回归分析中类似于R²，衡量了模型对数据变异性的解释能力。这个值越高越好，一个高的百分比表示模型可以很好地捕捉到数据中的变异性。根据结果，模型解释的变异性较高，可以捕捉到较多变异性。

3.3.5.4 结果与分析

通过PCA降维和随机森林模型的结合，实验能够有效处理图像数据，同时提高分类和评分预测的准确性。PCA有助于减少数据的噪声和冗余，而随机森林则通过集成多个决策树提高了预测的稳定性和鲁棒性。

```
importance <- importance(rf_model)  
varImpPlot(rf_model)
```



代码输出一个特征重要性图，这个图是用于展示随机森林模型中各个特征在预测过程中的重要性。图中的每个点代表一个特征，y轴显示了特征的名称，x轴代表了特征的重要性，这里使用的是增加节点纯度（IncNodePurity）这一度量。

- **特征名称**: y轴上的标签代表不同的特征。在图像处理的上下文中，这些通常代表像素位置或经过某种转换的特征。由于标签是 "PC1", "PC2" 等，这表明您可能使用了主成分分析（PCA）或类似方法来转换原始像素数据。
- **增加节点纯度（IncNodePurity）** : x轴上的数值代表了特征在构建随机森林模型的树时，通过分裂节点而导致的纯度增加。纯度可以用基尼指数（Gini index）或信息增益（Information Gain）等来衡量。一个特征的IncNodePurity越高，意味着它在模型中越重要。
- **特征重要性排序**: 图像中的特征按重要性降序排列，最顶部的特征对模型的影响最大。在您的图中，PC1到PC5似乎是最重要的特征。

通过特征重要性分析，可以采取几种方法来针对性改进模型：

1. 特征选择:

- 移除贡献小的特征：根据图表，可以选择移除那些对增加节点纯度贡献较小的特征，这可能有助于简化模型，减少计算时间，有时甚至可以提高模型性能。
- 保留最重要的特征：可以尝试只保留最重要的一些特征，比如前5或前10个，并重新训练模型来观察性能的变化。

2. 特征工程:

- 创建新特征：基于最重要的特征，可以尝试创建新的特征，比如这些特征的组合或转换。
- 对特征进行变换：对于那些重要的特征，可以尝试不同的数学变换，比如对数变换、平方/开方等，看是否能进一步提升模型性能。

3. 模型超参数调整:

- 可以根据重要特征的变化调整随机森林的超参数，比如 `mtry` 参数（在每个分裂中考虑的特征数量）。

4. 数据质量和预处理:

- 保证数据的质量，确保重要特征没有错误或异常值。
- 对数据进行标准化或归一化，以确保所有特征都在同一尺度上被评估。

3.3.6 图像颜值评分预测

现在利用先前训练的PCA模型和随机森林模型对一组新的图像数据进行颜值评分预测，并分析这些预测结果。

3.3.6.1 实验方法

- **数据准备**: 从指定目录读取JPEG格式的图像文件，并将这些图像转换为灰度图像。
- **灰度转换**: 使用预定义的灰度转换系数 ($r = 0.299, g = 0.587, b = 0.114$) 将彩色图像转换为灰度图像。
- **图像处理**: 将每张灰度图像转换为一维向量，以便进行PCA降维和随机森林模型的预测。
- **PCA降维**: 对每张图像应用PCA降维，使用先前训练的PCA模型将图像数据降至50个主成分。
- **颜值评分预测**: 使用随机森林模型对降维后的图像进行颜值评分预测。

```
library(jpeg)
library(randomForest)

# 初始化一个空的数据框来存储结果
results <- data.frame(Filename = character(), Gender = character(),
Score_Percentage = numeric(), stringsAsFactors = FALSE)

# 获取文件夹下所有JPG图片的路径
photo_directory <- "/Users/Mac/Desktop/test"
photo_paths <- list.files(photo_directory, pattern = "\\.jpg$",
full.names = TRUE)

# 定义灰度转换的系数
r <- 0.299
g <- 0.587
b <- 0.114

num_components_female <- 50
num_components_male <- 50

# 遍历所有图片，进行预测，并填充数据框
for(image_path in photo_paths) {
    # 读取单个图片文件
    image <- readJPEG(image_path)

    # 将图片转换为灰度图像
```

```
gray_image <- r * image[,1] + g * image[,2] + b * image[,3]

# 将图片转换为向量
image_vector <- as.vector(t(gray_image))

# 应用PCA降维到单张图片
gender_prefix <- substr(basename(image_path), 1, 1)
if (gender_prefix == "1") {
    image_reduced <- predict(pca_result_male, newdata =
matrix(image_vector, nrow = 1))[, 1:num_components_male]
    predicted_rating <- predict(rf_model_male, newdata =
image_reduced)
    gender <- "Male"
} else {
    image_reduced <- predict(pca_result_female, newdata =
matrix(image_vector, nrow = 1))[, 1:num_components_female]
    predicted_rating <- predict(rf_model_female, newdata =
image_reduced)
    gender <- "Female"
}
# 确保预测结果为数值类型
if(is.factor(predicted_rating)) {
    predicted_rating <- as.numeric(levels(predicted_rating))
[predicted_rating]
}

# 将评分转换为百分制
score_percentage <- predicted_rating
score_percentage_compare <- ((predicted_rating - 2.9)/6) * 100
# 获取图片文件名
file_name <- basename(image_path)
file_name_no_first_char <- substr(file_name, 2, nchar(file_name))
file_name_no_ext <-
tools::file_path_sans_ext(file_name_no_first_char)

# 将结果添加到数据框
results <- rbind(results, data.frame(Filename = file_name_no_ext,
Gender = gender, Score_Percentage = score_percentage, Compare =
score_percentage_compare))
}

# 显示结果
print(results)
```

3.3.6.2 结果展示

- 结果汇总：**将预测结果汇总到一个数据框中，包括文件名、性别、评分百分比以及与基准评分（2.9分）的比较值。
- 数据展示：**打印出整个数据框，展示所有测试图像的预测评分结果。

	Filename <chr>	Gender <chr>	Score_Percentage <dbl>	Compare <dbl>
1	关晓彤	Female	3.423800	8.7300001
11	刘亦菲	Female	3.408711	8.4785185
12	刘浩存	Female	3.018178	1.9696292
13	女1	Female	2.205378	-11.5770392
14	女2	Female	2.845219	-0.9130094
15	完颜慧德	Female	3.539117	10.6519444
16	杨幂	Female	3.399356	8.3225929
17	杨颖	Female	3.416397	8.6066207
18	白鹿	Female	3.424128	8.7354634
19	迪丽热巴	Female	3.411781	8.5296765

1-10 of 24 rows

Previous 1 2 3 Next

	Filename <chr>	Gender <chr>	Score_Percentage <dbl>	Compare <dbl>
110	吴磊	Male	3.151742	4.1956957
111	成龙	Male	3.478031	9.6338426
112	易烊千玺	Male	3.167119	4.4519917
113	王一博	Male	3.238114	5.6352322
114	王俊凯	Male	3.153742	4.2290292
115	王源	Male	3.373697	7.8949533
116	王鹤棣	Male	3.153686	4.2281032
117	男1	Male	1.932486	-16.1252284
118	男2	Male	2.291439	-10.1426869
119	肖战	Male	3.159481	4.3246773

11-20 of 24 rows

Previous 1 2 3 Next

	Filename <chr>	Gender <chr>	Score_Percentage <dbl>	Compare <dbl>
120	赵本山	Male	2.484450	-6.9258337
121	陈哲远	Male	3.162314	4.3718993
122	马嘉祺	Male	3.153742	4.2290292
123	龚俊	Male	3.312489	6.8748147

21-24 of 24 rows

Previous 1 2 3 Next

3.3.6.3 分析

- **评分分布**: 通过比较评分百分比和基准评分比较值，可以观察到模型对不同图像的评分差异。
- **模型性能**: 通过分析预测结果的分布和准确性，可以评估模型在新数据上的泛化能力和预测性能。

```
library(jpeg)
library(grid)
library(randomForest)

# 定义灰度转换的系数
r <- 0.299
g <- 0.587
b <- 0.114

num_components_female <- 50
num_components_male <- 50

# 假设 pca_result_male, pca_result_female, rf_model_male, rf_model_female
# 已经定义好

# 设置图片文件夹路径
photo_directory <- "/Users/Mac/Desktop/test"
photo_paths <- list.files(photo_directory, pattern = "\\.jpg$",
                           full.names = TRUE)

# 定义每行展示的图片数量
num_per_row <- 4

# 遍历所有图片，并展示图片、文件名和评分
for (i in seq_along(photo_paths)) {
  if (i %% num_per_row == 1) {
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(num_per_row, 2)))
  }

  # 读取图片
  image_path <- photo_paths[i]
  image <- readJPEG(image_path)

  # 处理图片并获得评分
  gray_image <- r * image[,,1] + g * image[,,2] + b * image[,,3]
  image_vector <- as.vector(t(gray_image))
```

```

gender_prefix <- substr(basename(image_path), 1, 1)

if (gender_prefix == "1") {
  image_reduced <- predict(pca_result_male, newdata =
matrix(image_vector, nrow = 1))[, 1:num_components_male]
  predicted_rating <- predict(rf_model_male, newdata =
image_reduced, type = "response")
} else {
  image_reduced <- predict(pca_result_female, newdata =
matrix(image_vector, nrow = 1))[, 1:num_components_female]
  predicted_rating <- predict(rf_model_female, newdata =
image_reduced, type = "response")
}

# 转换评分为百分制
score_percentage_compare <- ((predicted_rating - 2.9) / 6) * 100

# 获取并处理文件名
file_name <- basename(image_path)
file_name_no_first_char <- substr(file_name, 2, nchar(file_name))
file_name_no_ext <-
tools::file_path_sans_ext(file_name_no_first_char)

# 绘制图片
grid.raster(image, vp = viewport(layout.pos.row = (i - 1) %%
num_per_row + 1, layout.pos.col = 1))

# 展示处理后的文件名和评分
grid.text(paste("Name: ", file_name_no_ext, "\nRating: ",
format(score_percentage_compare, nsmall = 2)),
vp = viewport(layout.pos.row = (i - 1) %% num_per_row + 1,
layout.pos.col = 2))
}

```

输出结果：



Name: 关晓彤
Rating: 8.73



Name: 刘亦菲
Rating: 8.478518



Name: 刘浩存
Rating: 1.969629



Name: 女1
Rating: -11.57704



Name: 女2
Rating: -0.9130094



Name: 完颜慧德
Rating: 10.65194



Name: 杨幂
Rating: 8.322593



Name: 杨颖
Rating: 8.606621



Name: 白鹿
Rating: 8.735463



Name: 迪丽热巴
Rating: 8.529677



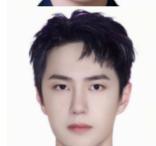
Name: 吴磊
Rating: 4.195696



Name: 成龙
Rating: 9.633843



Name: 易烊千玺
Rating: 4.451992



Name: 王一博
Rating: 5.635232



Name: 王俊凯
Rating: 4.229029



Name: 王源
Rating: 7.894953



Name: 王鹤棣
Rating: 4.228103



Name: 男1
Rating: -16.12523



Name: 男2
Rating: -10.14269



Name: 肖战
Rating: 4.324677



Name: 赵本山
Rating: -6.925834



Name: 陈哲远
Rating: 4.371899



Name: 马嘉祺
Rating: 4.229029



Name: 高俊
Rating: 6.874815

Discussion

4.1 实验结果解释

4.1.1 算法比较：性能和准确率比较

在本研究中，我们比较了三种不同的算法：线性回归最小残差法、Logistic回归和欧几里得距离法在性别分类任务中的性能和准确率。

1. 线性回归最小残差法：

- 显示了高效的计算性能，尤其是在使用SVD降维后。
- 准确率方面，该方法在选择了适当数量的奇异值后，展现了较高的分类精度。

2. Logistic回归：

- 提供了稳定的分类效果，并能给出概率估计，对于需要精确概率输出的场景特别有用。
- 通过适当的模型优化和奇异值选择，该方法在测试集上也表现出了良好的准确性。

3. 欧几里得距离法：

- 以其简单直观的方法在基于距离的分类任务中表现良好。
 - 尽管受到一定局限性的影响，但在数据质量充足的情况下，该方法仍能达到令人满意的准确率。
-
- 这三种方法在不同的应用场景中各有优势。线性回归最小残差法适用于需要快速有效分类的任务，而Logistic回归在处理复杂分类问题时表现出更好的灵活性和准确度。欧几里得距离法则在数据质量较高时能够提供一种简单有效的解决方案。
 - 结合SVD的数据降维处理，这些方法不仅提高了分类任务的计算效率，还保持了较高的准确率。适当的奇异值选择和参数调整对于优化这些方法的性能至关重要。

综上所述，线性回归最小残差法、Logistic回归和欧几里得距离法在人脸识别的性别分类任务中均展现出了各自的优势和应用价值。通过对这些方法的比较分析，我们得出结论：选择合适的方法应基于具体任务的需求和数据的特性。此外，SVD作为一种有效的降维技术，在提升这些分类算法性能方面发挥了关键作用。

4.1.2 实验一结果分析：深入分析和讨论

4.1.2.1 实验过程

在本研究中，我们采用了线性回归最小残差法、Logistic回归和欧几里得距离法对一组人脸图像进行性别分类。测试集包含了25张图像，分别用各种方法处理并预测性别。以下是对这些结果的详细分析。

4.1.2.2 性别分类结果

1. 分类准确率：

- 分析显示，大部分图像的性别分类结果准确。例如，在25张图像中，除了少数几张图像（如“AF卿媚.jpg”）外，大多数图像的性别被正确识别。
- 这表明所采用的方法在识别图像性别方面是有效的，尽管存在少数误判。

2. SVD分解的影响：

- SVD分解结果表明，数据集的主要变异可以由前几个奇异值捕获。这说明了降维在性别分类中的有效性。
- Logistic回归模型的系数分析显示了不同特征的重要性，其中一些特征对性别分类有显著影响。

4.1.2.3 影响因素讨论

- **数据质量和多样性：**图像的质量和多样性对性别分类结果有显著影响。例如，图像的清晰度、光照和面部表情可能影响分类的准确性。
- **特征选择：**在Logistic回归中，选择合适的特征对模型的性能至关重要。特征选择应基于其对分类任务的贡献程度。
- **算法局限性：**每种算法都有其局限性。例如，线性回归最小残差法可能不适用于复杂的分类任务，而欧几里得距离法可能受到图像条件的影响。

4.1.3 SVD在性别分类中的优势和局限性

• 优势：

- **计算效率：**通过降低数据处理量提高效率。
- **突出重要特征：**帮助更清晰地区分不同性别。
- **灵活性：**可根据需求选择不同数量的奇异值和向量。

• 局限性：

- **信息损失：**过度降维可能导致重要信息丢失。
- **选择难度：**确定保留多少奇异值和向量有时具挑战性。

- **过拟合风险**: 尤其在变量众多时可能导致对训练数据的过拟合。

在本研究中，我们通过执行奇异值分解（见代码部分 {r 1.9}），选择了前25个奇异向量进行逻辑回归分析（见代码部分 {r 1.13}），实现了有效的特征提取和降维。此外，我们还进行了测试集与平均脸部图像的距离计算（见代码部分 {r 1.12}），进一步展示了SVD在提取关键特征方面的价值。

4.1.4 实验二结果分析：深入分析和讨论

本实验应用了主成分分析（PCA）和随机森林模型来预测图像的颜值评分。结果显示，通过这种方法能够有效地处理高维图像数据并进行性别分类及颜值评分预测。PCA作为一种降维技术，有助于减少计算复杂性，同时保留了数据中最关键的变异信息。而随机森林作为一种集成学习方法，通过构建多个决策树并结合它们的预测结果，提高了预测的准确性和稳健性。

根据实验结果以及输出的对应评分，可以看出：男女性两组中，明星和几位“传统审美”普遍认为比较“丑”的男1、2；女1、2分数相差较大，明星中网络大众普遍认为较好看的得分也相对较高。每个人的审美标准不一样，不同人很难对于“杨幂、杨颖、刘亦菲、迪丽热巴谁更美？”或“王俊凯、易烊千玺、王源、王一博谁更帅？”达成共识性的评分，Compare组的评分只是放大了几者之间细微的分数差距但差异并不大，但你很容易判断“如花和迪丽热巴谁更好看？”，相对应的分数差距也很大，证明了针对多数人审美标准而言，该模型算法有一定的可取之处与准确性。当然比如完颜慧德的分数最高，这不一定符合大众对于这组测试组颜值的评判，证明了模型也有一定的误差和错值。

以下为将测试集更换为本班同学所得到的结果。从Score-Percentage看出，同学们的评分相对集中在2.3左右，与之前测试的明星组集中于3.4有一定差距，当然考虑到是否化妆、背景、拍摄角度、灯光等问题会造成一定的差异，但也证明了“明星靠脸吃饭”的评分有一定的参考性和真实性。

	Filename <chr>	Gender <chr>	Score_Percentage <dbl>	Compare <dbl>
1	倪婧	Female	2.307031	6.783842
11	刘新洁	Female	2.462600	9.376667
12	刘江雪	Female	2.399919	8.331990
13	卿	Female	2.422714	8.711898
14	叶雪莹	Female	2.394831	8.247176
15	唐欣扬	Female	2.374431	7.907176
16	屠心怡	Female	2.164281	4.404676
17	王好乐	Female	2.413389	8.556481
18	王越	Female	2.476747	9.612454
19	瓮思多	Female	2.044272	2.404537
110	胡晓雪	Female	2.217439	5.290648
111	裴思宇	Female	1.831247	-1.145880
112	邵圆圆	Female	2.417947	8.632454
113	龙乔欣	Female	2.371544	7.859074
114	向宇杰	Male	2.326989	7.116481
115	吴广宇	Male	2.168686	4.478103
116	张启睿	Male	2.153742	4.229029
117	杨晓亮	Male	2.153742	4.229029
118	林杰	Male	2.153742	4.229029
119	王艺霖	Male	2.168686	4.478103
120	王若旭	Male	2.153742	4.229029
121	胡明杰	Male	2.153742	4.229029
122	郑柯茗	Male	2.189739	4.828982
123	马若飞	Male	2.153742	4.229029
124	黄梓龙	Male	2.168686	4.478103

4.1.5 实验局限性

尽管实验结果令人鼓舞，但也存在一些局限性。首先，数据集的规模和多样性可能限制了模型的泛化能力。在一个相对较小且同质化的数据集上训练的模型可能无法有效地推广到更广泛多样的人群。其次，评分标准的主观性可能也会影响模型预测的准确性。此外，PCA虽然能有效降维，但可能会丢失一些对颜值评分预测至关重要的细微特征。

4.2 与现有研究的比较

与现有的图像处理和机器学习研究相比，本实验的方法在处理特定类型的图像数据集（面部图像）时展示了其有效性。这与一些现有文献中的发现一致，即PCA和随机森林在图像分类和特征提取方面的有效应用。然而，本实验特别关注于颜值评分的预测，这在现有文献中相对较少探索。

Conclusion

5.1 结论与未来的工作

5.1.1 实验一结论

通过本研究的分析，我们可以得出结论，线性回归最小残差法、Logistic回归和欧几里得距离法在人脸识别的性别分类中都是有效的方法。每种方法都有其优势和局限性，适用于不同的应用场景。为了提高性别分类的准确性，建议综合考虑数据质量、特征选择和算法的适用性。未来的工作可以探索这些方法的结合使用，以进一步提高分类的准确率和鲁棒性。

5.1.2 实验二结论

本实验通过结合PCA降维和随机森林模型，对新的图像数据集进行了颜值评分预测。预测结果的分析提供了对模型性能的初步认识。未来的工作可以包括：

- 改进模型：考虑使用更复杂的机器学习模型或深度学习方法来提高预测准确性。
- 扩展数据集：使用更大和更多样化的数据集来训练和测试模型，以提高其鲁棒性。
- 特征工程：探索更高级的图像处理和特征提取技术，以捕获更丰富的图像信息。

针对这些局限性，未来的研究可以在几个方面进行扩展。首先，可以通过使用更大和更多样化的数据集来改善模型的泛化能力。其次，探索结合PCA和其他高级特征提取技术（如卷积神经网络）可能进一步提高预测准确性。最后，未来的研究还可以探讨不同的评分机制，以减少主观评分带来的偏差，并提高评分的一致性和可靠性。

参考文献

[1] Lingyu Liang, L.j.Lin, L. Jin, D.Xie, M.Li. SCUT-FBP5500: A Diverse Benchmark Dataset for Multi-Paradigm Facial Beauty Prediction[J]Computer Science, 2018(9)

[2] D. Xie, L. Liang, L. Jin, J. Xu and M. Li, “SCUT-FBP: A Benchmark Dataset for Facial Beauty Perception,” in Proc. of IEEE SMC, pp. 1821– 1826, 2015.

[3] L. Liang, D. Xie, L. Jin, J. Xu, M. Li, and L. Lin, “Region-aware scattering convolution networks for facial beauty prediction,” in Proc. of IEEE ICIP, pp. 2861– 2865, 2017.

[4] J.Xu,L.Jin,L.Liang,Z.Feng,D.Xie, and H.Mao, “Facial attractiveness prediction using psychologically inspired convolutional neural network (PI-CNN),” in Proc. of IEEE ICASSP, pp. 1657–1661, 2017.

- [5] L. Liang, L. Jin, and D. Liu, “Edge-aware label propagation for mobile facial enhancement on the cloud,” IEEE Trans. on Circuits and Systems for Video Technology, vol. 27, no. 1, pp. 125–138, 2017.
 - [6] L. Liang, L. Jin, and X. Li, “Facial skin beautification using adaptive region-aware mask,” IEEE Trans. on Cybernetics, vol. 44, no. 12, pp. 2600–2612, 2014.
 - [7] J. Li, X. Chao, L. Liu, X. Shu, and S. Yan, “Deep face beautification,” Proc. ACM International Conference on Multimedia, pp. 793–794, 2015.
 - [8] T. Leyvand, D. Cohen-Or, G. Dror, and D. Lischinski, “Data-driven enhancement of facial attractiveness,” ACM Trans. Graph., pp. 28:1–10, 2008.
 - [8] H. Gunes, “A survey of perception and computation of human beauty,” in Proc. of J-HGBU, pp. 19–24, 2011.
 - [9] A. Laurentini and A. Bottino, “Computer analysis of face beauty: A survey,” Comput. Vision and Image Underst., vol. 125, pp. 184–199, 2014.
 - [10] D. Zhang, F. Chen, and Y. Xu, Computer models for facial beauty analysis. Springer International Publishing Switzerland, 2016.
 - [11] L. Liu, J. Xing, S. Liu, H. Xu, X. Zhou, and S. Yan, “Wow! you are so beautiful today!,” ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 11, no. 1s, p. 20, 2014.
 - [12] K. Scherbaum, T. Ritschel, M. Hullin, T. Thormählen, V. Blanz and H. Seidel, “Computer-suggested facial makeup,” Comput. Graph. Forum, vol. 30, no. 2, pp. 485–492, 2011.
 - [13] Y. Mu, “Computational facial attractiveness prediction by aesthetics-aware features,” Neurocomputing, vol. 99, pp. 59–64, 2013.
 - [14] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” Nature, vol. 251, pp. 436–444, 2015.
 - [15] Y. Eisenthal, G. Dror and E. Ruppin, “Facial Attractiveness: Beauty and the Machine,” Neural Computation, vol. 18, pp. 119–142, 2006.
 - [16] N. Murray, L. Marchesotti L, and F. Perronnin, “AVA: A large-scale database for aesthetic visual analysis,” in Proc. of CVPR, pp. 2408–2415, 2012.
 - [17] A. Laurentini and A. Bottino, “Computer analysis of face beauty: A survey,” Computer Vision and Image Understanding, vol. 125, pp. 184–199, 2014.
 - [18] I. Stephen, M. Law Smith, M. Stirrat, and D. Perrett, “Facial skin coloration affects perceived health of human faces,” International Journal of Primatology, vol. 30, no. 6, pp. 845–857, 2009.
- [19]高全学,梁彦,潘泉,陈玉春,张洪才.SVD用于人脸识别存在的问题及解决方法[J].中国图象图形学报,2006,11(12):1784–1791