**深圳大学本科生课程论文**

题目 基于CNN和PCA的人脸识别 成绩

专业 机器人，自动化 课程名称、代码 机器人学与控制

年级 2022 姓名 周家辉 郑臻 高俊威 欧阳亿

学号 2022110214 2022110146 2021090168 2022110130时间2024 年10月9日

任课教师 马跃

**深圳大学本科生课程论文学术诚信承诺书**

本人在此声明所提交的课程论文基于CNN和PCA的人脸识别是本人独立完成的，具有原创性，并且未抄袭、剽窃他人成果或侵犯他人的知识产权。本声明书详细阐述以下内容：

1.本人郑重声明，课程论文的所有内容和观点均源自本人的研究和分析，未从其他来源直接复制或翻译。

2.对于其他作者或研究人员的观点、数据、图片、图表等引用和参考，本人已按照学校规定的引用标准进行准确的引用和注明，并在文中明确标明了引用部分。

3.本人保证，课程论文中使用的所有文献、资料和其他来源均已在参考文献部分列出，且准确无误地注明了相关信息，包括作者、出版年份、出版社或期刊名称等。未使用生成式人工智能文本生成工具。

4.本人明确知晓学术不端行为的严重性，包括但不限于抄袭、剽窃、造假、篡改数据等。本人承诺，在课程论文的整个研究和撰写过程中，坚守学术道德原则，维护学术诚信。

我郑重承诺以上内容的真实性，并愿意为我所提交的课程论文的原创性负全部责任。

论文作者签名： 日期： 年 月 日

目 录

[【摘要】 7](#_Toc179410343)

[1 绪论 8](#_Toc179410344)

[1.1 研究背景及意义 8](#_Toc179410345)

[1.2 国内外研究现状 8](#_Toc179410346)

[2 PCA人脸识别实现 9](#_Toc179410347)

[2.1 PCA介绍 9](#_Toc179410348)

[2.2 PCA数学原理 9](#_Toc179410349)

[2.3 人脸识别和重建实验中的PCA算法 9](#_Toc179410350)

[2.4 KNN 10](#_Toc179410351)

[3 CNN人脸识别 10](#_Toc179410352)

[3.1 CNN与全连接网络的异同 10](#_Toc179410353)

[3.2 推导CNN 11](#_Toc179410354)

[3.2.1 . CNN前向传播 12](#_Toc179410355)

[3.2.2 CNN反向传播： 12](#_Toc179410356)

[3.3 CNN设计与技术实现 13](#_Toc179410357)

[3.3.1 神经网络设计 13](#_Toc179410358)

[3.3.2 检验网络性能 14](#_Toc179410359)

[3.3.3 移植网络 14](#_Toc179410360)

[3.3.4 分类部分 16](#_Toc179410361)

[3.4 GUI设计 16](#_Toc179410362)

[3.4.1 界面设计 16](#_Toc179410363)

[3.4.2 交互设计 16](#_Toc179410364)

[4 测试实验与数据分析 19](#_Toc179410365)

[4.1 实验设计 19](#_Toc179410366)

[4.1.1 实验目的 19](#_Toc179410367)

[4.1.2 实验方式 19](#_Toc179410368)

[4.2 实验过程 19](#_Toc179410369)

[4.3 实验结果 19](#_Toc179410370)

[5 总结和展望 20](#_Toc179410371)

[5.1 总结 20](#_Toc179410372)

[5.2 展望 20](#_Toc179410373)

[【Abstract】 23](#_Toc179410374)

图 目 录

[图 2‑1中心脸 图 2‑2重构脸 9](#_Toc179410195)

[图 3‑1 全连接网络结构 10](#_Toc179410196)

[图 3‑2 CNN网络结构 10](#_Toc179410197)

[图 3‑3 13](#_Toc179410198)

[图 3‑4 卷积神经网络训练结果 13](#_Toc179410199)

[图 3‑5 GUI界面排版 15](#_Toc179410200)

[图 3‑6 KNN分类效果 17](#_Toc179410201)

[图 3‑7 CNN分类效果 17](#_Toc179410202)

[图 4‑1 蜡笔小新KNN识别结果 图 4‑2 张元英KNN识别结果 18](#_Toc179410203)

[图 4‑3 蜡笔小新CNN识别结果 图 4‑4詹姆斯CNN识别结果 19](#_Toc179410204)

**基于CNN和PCA的人脸识别**

**【摘要】**：随着深度学习技术的快速发展，人脸识别技术在安全、监控、金融等多个领域得到了广泛应用。本文针对人脸识别中的关键问题，研究了基于卷积神经网络（CNN）和主成分分析法（PCA）的混合模型。首先，本文系统介绍了PCA和CNN的基本原理，分别应用这两种方法对人脸图像进行特征提取和分类。PCA通过降维技术减少了数据冗余，从而提高了计算效率，而CNN作为一种深度学习算法，通过多层卷积和池化操作，自动提取图像中的高维特征，展现出较高的识别精度。

此外，本文结合了K近邻算法（KNN）对PCA降维后的人脸特征进行分类，并设计了基于Matlab的图形用户界面（GUI），实现了人脸图像的预处理、分类和结果显示。实验部分采用了不同来源和场景下的人脸图像数据集，分别对PCA和CNN模型进行了训练和测试。结果表明，PCA结合KNN方法在人脸识别的特定场景中具有一定的优势，但在大规模数据处理上表现相对有限。而CNN在面对复杂的图像数据时，展现出更高的准确率和鲁棒性，特别是在多类别分类任务中表现突出。

通过对比分析，本文认为CNN由于其强大的特征提取能力和较少的参数依赖性，未来在人脸识别领域将有更广阔的应用前景。同时，结合PCA、KNN等传统算法的混合模型或将成为未来优化算法性能的一种重要研究方向。本文的研究为人脸识别技术的发展提供了理论支持，并在实践中为复杂场景下的身份识别问题提供了可行的解决方案。

【关键词】： 卷积神经网络（CNN），主成分分析（PCA），人脸识别，K近邻算法（KNN），深度学习，图像分类

# 绪论

## 研究背景及意义

人脸识别技术是一种利用人脸图像进行身份识别的技术，广泛应用于安全、监控、金融、医疗等领域。近年来，随着深度学习技术的发展，人脸识别技术取得了显著进展。卷积神经网络（CNN）因其在图像处理方面的强大能力，被广泛应用于人脸识别中。而K-近邻算法（KNN）作为一种简单但有效的分类算法，也在某些情况下展现出不错的性能。

## 国内外研究现状

KNN作为一种简单且有效的分类算法，在人脸识别中的应用较为普遍。研究者们通常将KNN与特征提取方法结合使用，如主成分分析（PCA）和线性判别分析（LDA），以提高识别精度。

例如，2016年孙志帅利用PCA和均匀采样去弥补Gabor小波变换造成维度增加的劣势[1]，既保证了Gabor小波变换识别率高的优势,又弥补了维度增加的劣势,达到了优势互补，并通过分析K最近邻法和支持向量机法各自优势，把两者相结合,利用K最近邻法初步定范围，用支持向量机最终识别分类,达到了优势互补,得到了 KNN-SVM分类算法。

2018年简彩仁等人利用主成分分析和核主成分分析对基于降维的人脸识别方法进行了研究[2]。实验结果表明降维可以有效地提高人脸识别准确率，而非线性降维方法更适合人脸图像数据的降维，并且降维后的数据可以明显提高识别效率。

2018年Mustafa Zuhaer Nayef Al-Dabagh等人提出了一种使用核判别分析（KDA）和支持向量机（SVM）结合K近邻（KNN）方法的人脸识别系统[3]。核判别分析用于从输入图像中提取特征，随后使用SVM和KNN根据提取的特征对人脸图像进行分类。

近年来，随着深度学习的发展，CNN在人脸识别中的应用逐渐成为主流。国内许多研究者致力于优化CNN架构，以提升其在复杂场景下的人脸识别性能。

例如，2019年李瑛将基于CNN的人脸识别技术应用在课堂考勤系统中[4]。研究利用K-Means聚类方法从大量标注数据中学习到人脸候选区域的若干典型值，并以此为基础优化Faster R-CNN人脸检测算法的RPN网络候选区域提取策略。且在已有基于CNN的人脸识别算法基础上，对其网络结构和参数进行有针对性的优化，得到一个该应用场景下的高效人脸识别模型。

2021年冯友兵等人针对人脸识别在实际应用中存在姿态变化、表情、遮挡等问题[5]，采用了基于facenet卷积神经网络的特征提取算法以及svm分类算法，在此基础上设计并实现了人脸识别系统。

2017年Musab Coşkun等人提出了一种基于CNN架构的人脸识别算法[6]。该算法的主要特征是在第一层和最后一层卷积层的输出上应用了批量归一化，从而使网络能够达到更高的准确率。在全连接层步骤中，使用Softmax分类器对人脸进行分类。

KNN和CNN在国内的人脸识别研究中各有侧重。KNN作为传统算法，在特定场景中与其他算法的结合显示出良好的效果，而CNN凭借深度学习的优势，逐渐成为人脸识别的主流方法。随着技术的发展，结合这两种方法的混合模型和更为复杂的深度学习架构将可能成为未来研究的重要方向。

# PCA人脸识别实现

## PCA介绍

主成分分析（Principal components analysis, PCA），顾名思义，其目的在于提取数据中的主要成分信息，因此，常用于对数据的提炼，例如：降维（使用最多的领域之一），异常值检测等，是数据分析中的一种重要方法。

## PCA数学原理

我们希望投影后投影值尽可能分散，而这种分散程度，可以用数学上的方差表述，于是用两个字段的协方差表示其相关性。

当协方差为0时，表示两个字段完全独立。为了让协方差为0，我们选择第二个基时只能在与第一个基正交的方向上选择。因此最终选择的两个方向一定是正交的。协方差矩阵是一个对称矩阵，其对角线分别是各个字段的方差。

要进行优化，等价于将协方差矩阵对角化：即除对角线外的其他元素化为0，并且在对角线上将元素按大小从上到下排列，这样我们就达到了优化目的。

优化目标变成了寻找一个矩阵P，满足PCPT是一个对角矩阵，并且对角元素按从大到小依次排列，那么P的前K行就是要寻找的基，用P的前K行组成的矩阵乘以X就使得X从N维降到了K维并满足上述优化条件。

## 人脸识别和重建实验中的PCA算法

1. 降维：所有图片构成的3维矩阵降维成2维，提升运算速度；
2. 标准化：求所有图像各像素点的平均值，即平均脸；
3. 中心化：原始数据减去平均量，中心化每一列为一张图；
4. 由中心化向量得到协方差矩阵；
5. 求解主成分：求解协方差矩阵的特征值与特征向量，对特征值和特征向量进行降序排序，取前d个特征向量，这d个特征向量组成的矩阵X即为选取的主成分
6. 根据选取的d个特征向量构造特征脸W，W\*X映射到脸空间，重构人脸

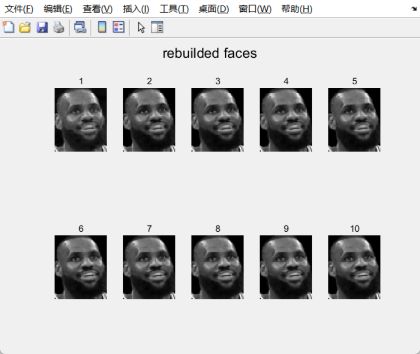
中心化后: 重构后的人脸：  
 

图 2‑1中心脸 图 ‑2重构脸

## KNN

算法核心思想为：算法的核心思想为：给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的K个实例，这K个实例的多数属于某个类，就把该输入实例分类到这个类中。

本实验对测试集的验证选择了KNN近邻类分类，算法步骤如下：

1. 计算距离：本实验中距离定义为测试集与训练集映回脸空间后的矩阵相减的差值，差值越大距离越大，即越不可能为同一张脸
2. 对距离进行升序排序
3. 选取前k个距离，获取对应的标签（即对应的人脸），选取出现次数最多的label作为预测脸
4. 预测label与测试脸label进行对比，最后用正确的人脸数除去测试集大小即为对应k值下的正确率。

# CNN人脸识别

## CNN与全连接网络的异同

1. 结构

全连接网络的内部，层与层之间是全连接的，也就是说，第i层的任意一个神经元一定与第i+1层的任意一个神经元相连。全连接网络不具备对输入数据的空间结构处理能力。它将输入数据展平为一维向量，并丢失了原始数据的空间信息。

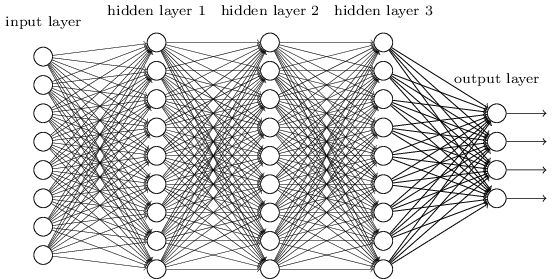
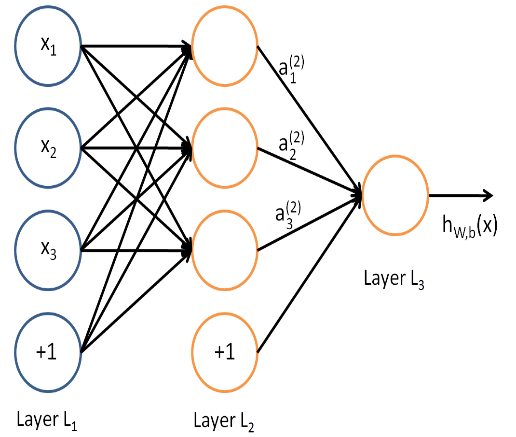


图 ‑1 全连接网络结构

而CNN的隐藏层由卷积层、池化层和全连接层组成。卷积层用于提取输入数据中的局部特征，池化层用于降低特征图的尺寸并保留主要特征，全连接层用于将提取的特征映射到输出类别上。CNN的输入通常是张量，例如RGB图像，能够处理具有空间结构的数据。

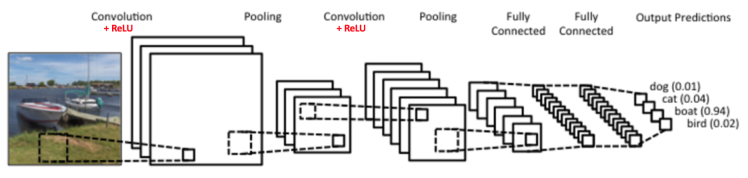


图 3‑2 CNN网络结构

（2）**参数和数据效率：**

使用卷积核在输入上滑动，可以共享权重来检测相同的特征模式，CNN通常具有较少的参数数量，尤其适用于处理大规模数据，例如图像。这也降低了计算复杂度，使得CNN更容易训练和部署。而全连接网络的参数数量较大

，尤其是在输入维度较高的情况下。CNN以从数据中提取出更少但更有代表性的特征，因此对于大规模数据集的训练效果较好。全连接网络通常需要更多的数据样本来获得好的泛化性能。否则，容易出现过拟合现象。并且，全连接网络由于每个神经元都有独立的权重，很难评估网络中每个神经元对结果的贡献。

1. **算法上：**

由于CNN多了卷积层和池化层，这两层的前向传播和后向传播的数学推导和具体实现上也不同。

## 推导CNN

### . CNN前向传播

输入是张量,卷积层激活函数是ReLU;隐藏层前向传播到池化层只需对输入的矩阵进行缩小，池化标准一般是MAX或者Average；隐藏层前向传播到全连接层和DNN相同，但最后一层的输出层激活函数是softmax。

是卷积层： \*表示卷积运算

是池化层： ，pool表示按池化大小和标准将输入张量缩小

是全连接层：

对于输出层第层:

### CNN反向传播：

求出中间变量和的传递关系，来推导出上一层的梯度。

全连接层反向传播与DNN相同，主要探讨卷积层和池化层。

**池化层的推导上一隐藏层的：**

反向传播时，我们首先会把的所有子矩阵矩阵大小还原成池化之前的大小，然后如果是MAX，则把的所有子矩阵的各个池化局域的值放在之前做前向传播算法得到最大值的位置。如果是Average，则把的所有子矩阵的各个池化局域的值取平均后放在还原后的子矩阵位置。这个过程一般叫做upsample（上采样）。

 （1）

**卷积层的推导上一隐藏层的：**

 （2）

与DNN不同就是使用了卷积预算\*并且卷积核被旋转了180度（先上下翻转一次再左右翻转一次）。

求出每层的梯度后可以求改成的的梯度，池化层没有就不用求，仅考虑卷积层。

由 ，得出  （3）

对于b,则有些特殊，因为是高维张量，而b只是一个向量，不能像DNN那样直接和相等。通常的做法是将的各个子矩阵的项分别求和，得到一个误差向量，即为b

的梯度：  （4）

所以CNN反向传播使用梯度下降更新表示为

1）若为全连接层（则与DNN相同）： ，

2）若为卷积层： ，

## CNN设计与技术实现

### 神经网络设计

本网络的设计目标是实现图像的分类，根据需求，我们使用了matlab的深度神经网络设计器来构建卷积神经网络。

具体设计如下：

1. imageInputLayer：输入图像层，定义了输入图像的大小和通道数。
2. convolution2dLayer：卷积层，执行二维卷积操作，用于提取特征。
3. batchNormalizationLayer：批归一化层，标准化输入以加速训练和提高稳定性。
4. reluLayer：激活层，应用ReLU（线性整流单元）激活函数，增加非线性。
5. fullyConnectedLayer：全连接层，将前一层的所有神经元连接到该层的每个神 经元，通常用于分类。
6. softmaxLayer：Softmax层，将输出转化为概率分布，适用于多类分类。
7. classificationLayer：分类层，计算损失函数并评估分类性能。

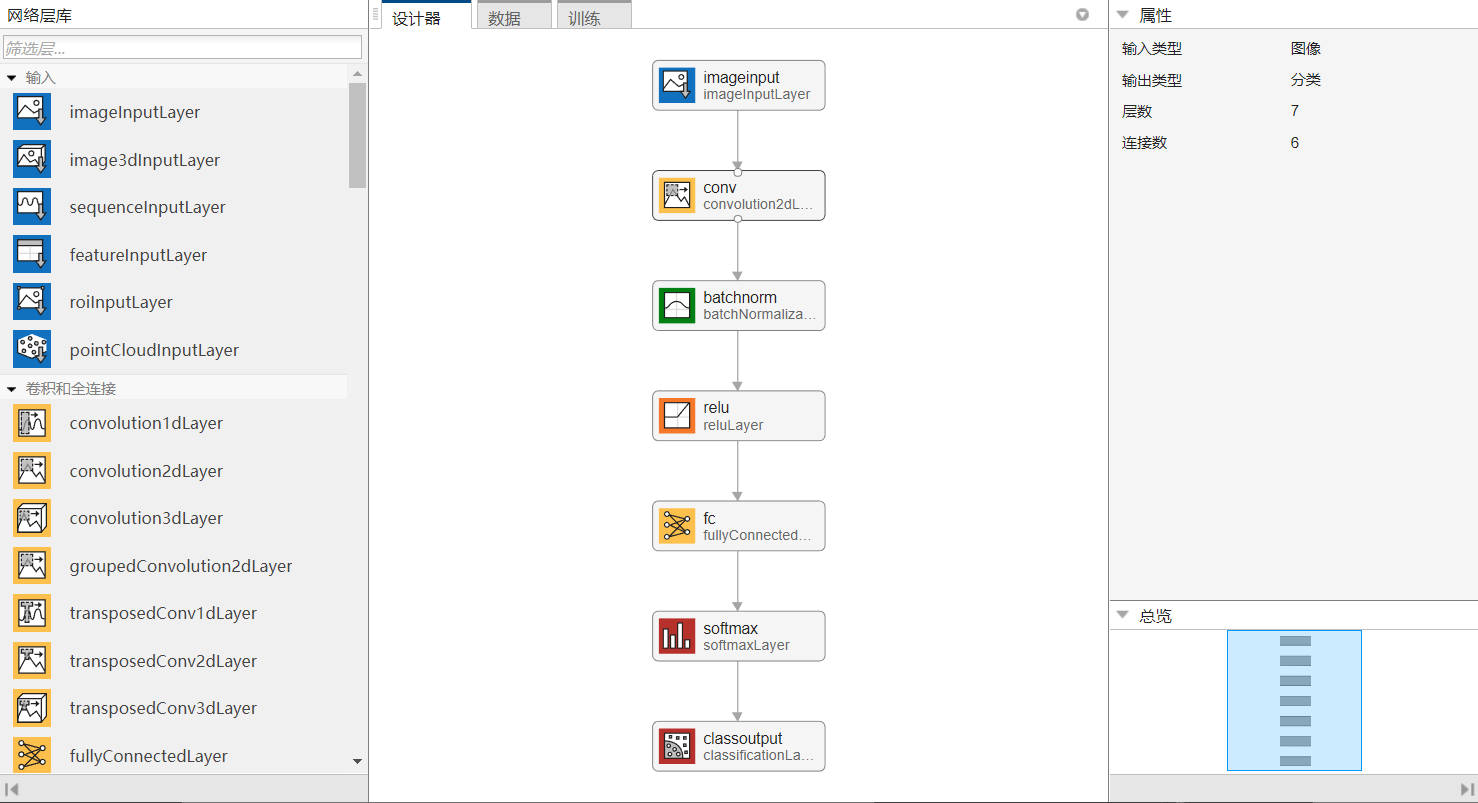


图 3‑3

### 检验网络性能

设计好网络后，在APP中直接导入数据集，选取了30%的数据作为测试集，训练，并得到训练图及分类性能。经过30轮迭代后可以看到最终准确度达到100%，损失函数收敛，并且验证准确度达到100%，符合预期分类效果。

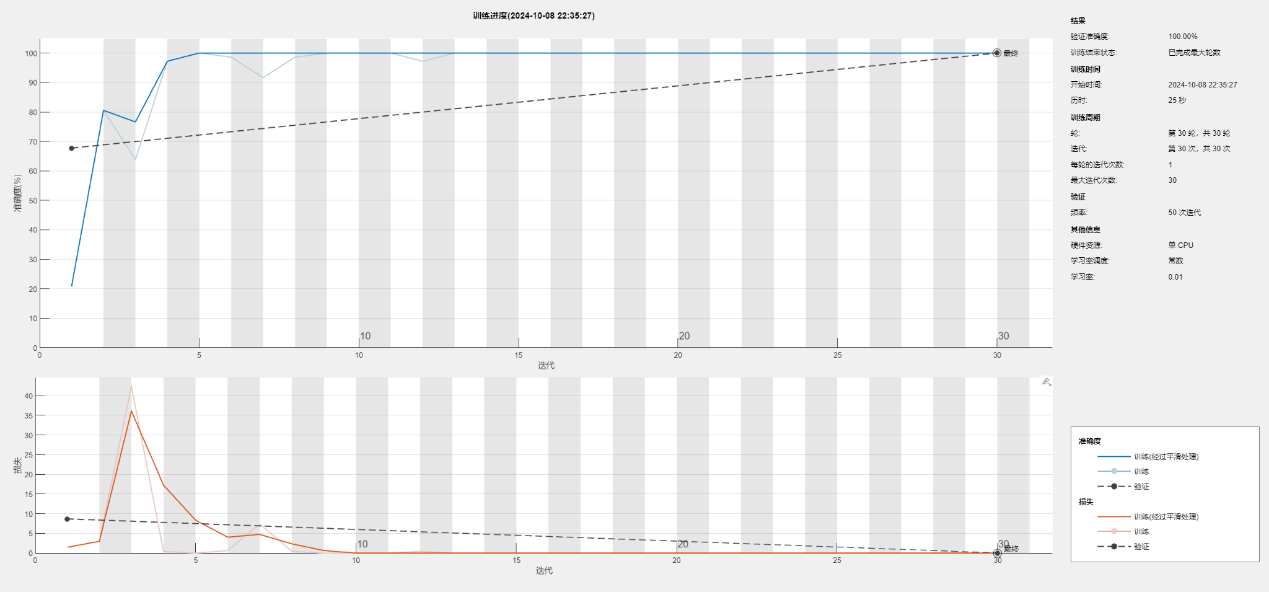


图 3‑4 卷积神经网络训练结果

### 移植网络

为了后续GUI界面的开发，还需要将设计好的网络封装为函数，当调用时训练并返回训练好的网络和训练结果。每训练一轮就会调用progressFcn，方便后续在GUI界面上更新训练进程。函数代码如下。

function [net,traininfo] = cnn\_train(app, train\_folderPath)

imdsTrain = imageDatastore(train\_folderPath, "IncludeSubfolders", true, "LabelSource", "foldernames");

% 分割数据集

[imdsTrain, imdsValidation] = splitEachLabel(imdsTrain, 0.7);

% 训练选项，包括 OutputFcn

opts = trainingOptions("sgdm",...

"ExecutionEnvironment","auto",...

"InitialLearnRate",0.01,...

"Shuffle","every-epoch",...

"Plots","none",...

"ValidationData",imdsValidation,...

"OutputFcn", @(info)progressFcn(app, info)); % 设置输出函数用于更新进度条

layers = [

imageInputLayer([128 100 3],"Name","imageinput")

convolution2dLayer([3 3],32,"Name","conv","Padding","same")

batchNormalizationLayer("Name","batchnorm")

reluLayer("Name","relu")

fullyConnectedLayer(3,"Name","fc")

softmaxLayer("Name","softmax")

classificationLayer("Name","classoutput")];

% 训练网络

[net, traininfo] = trainNetwork(imdsTrain, layers, opts);

end

### 分类部分

最终要实现对输入图像进行分类，调用classify(net,img)对输入图像进行分类。图像被输入到神经网络 net。网络从输入层开始，依次通过卷积层、激活层、批归一化层、全连接层等，进行前向传播计算。在最后一层（softmaxLayer），网络生成每个类别的预测概率。classify 函数根据这些概率输出预测的类别标签。

## GUI设计

### 界面设计

GUI使用Matlab的App Designer进行设计，本GUI界面功能要求：

1. 导入数据集
2. 预览数据集
3. KNN分类
4. CNN网络训练
5. CNN分类

根据要求设计的GUI界面如下。

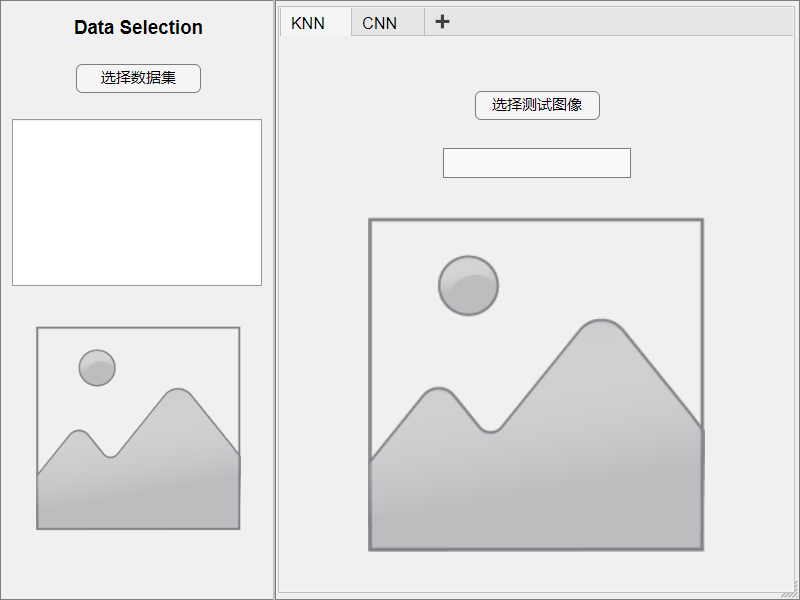
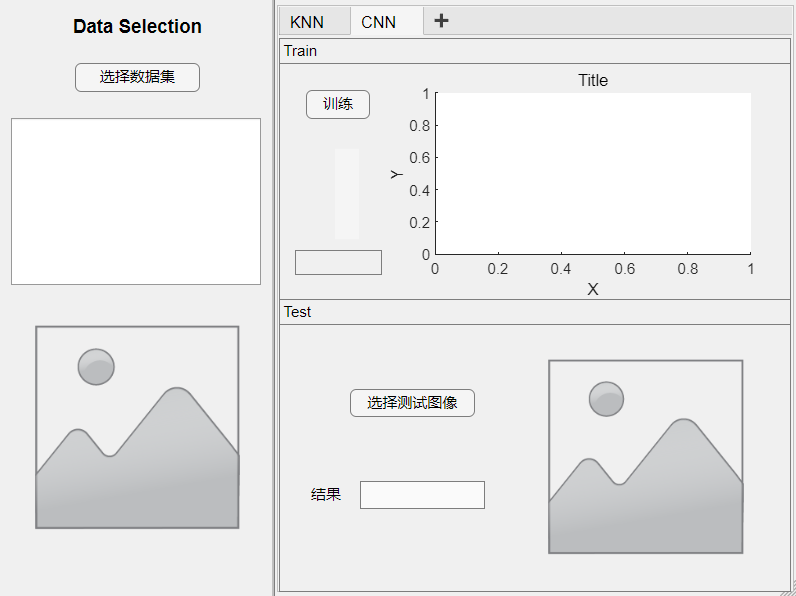
 

图 3‑5 GUI界面排版

### 交互设计

通过设计回调函数，实现组建之间的交互。

其中，点击左侧的“选择数据集按钮”来选择数据集存储的文件夹路径，并调用buildTree函数来更新下方显示文件夹结构的树，在树中选择对应的图片对数据集进行预览。以下是buildTree函数代码。

% 递归函数来构建树

function buildTree(app, parentNode, folderPath)

    % 获取文件夹中的内容

    items = dir(folderPath);

    % 遍历每个项目

    for i = 1:length(items)

        if strcmp(items(i).name, '.') || strcmp(items(i).name, '..')

            continue;

        end

        % 创建新的节点

        fullPath = fullfile(folderPath, items(i).name);

        if items(i).isdir

            % 如果是文件夹，递归添加子节点

            newNode = uitreenode(parentNode, 'Text', items(i).name, 'NodeData', fullPath);

            buildTree(app, newNode, fullPath); % 递归调用

        else

            % 如果是文件，直接添加节点

            if isImageFile(app,items(i).name)

                newNode = uitreenode(parentNode, 'Text', items(i).name, 'NodeData', fullPath);

            end

        end

    end

end

对于KNN界面，点击“选择测试图像按钮”选择要测试的分类图像，然后会在下方显示分类结果及所选的图像。效果如下图。



图 3‑6 KNN分类效果

对于CNN界面，点击“训练”后开始训练，左侧显示训练进度条，训练完成后右侧则显示训练的结果的准确度和损失函数图像。点击“选择测试图像”选择要分类的图像后，在下方显示分类结果，右侧显示选择对应的图像。效果如下图。

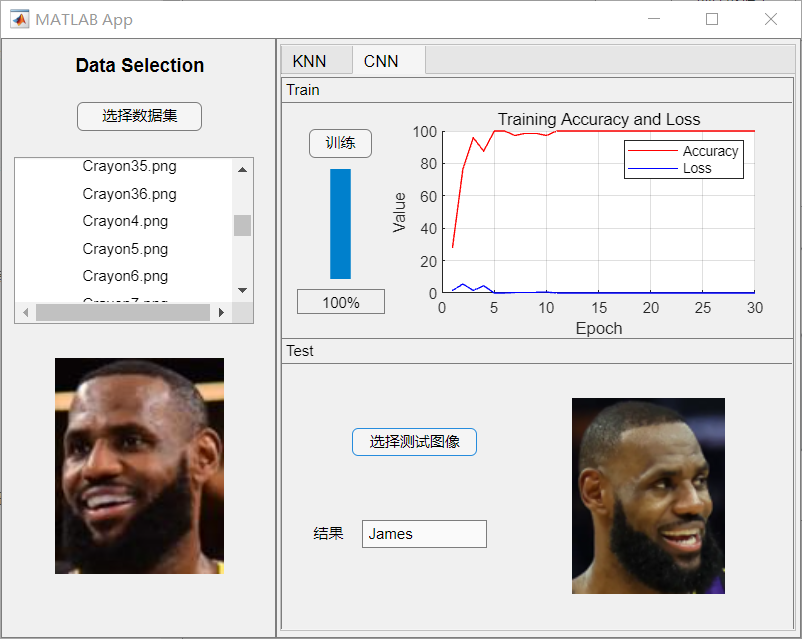


图 3‑7 CNN分类效果

# 测试实验与数据分析

## 实验设计

### 实验目的

检验本研究所设计KNN和CNN人脸识别架构对于实际人脸识别的识别效果。

### 实验方式

选定了詹姆斯，张元英和蜡笔小新作为识别对象， 从互联网或个人图库中收集每个对象的图片，确保每个对象至少包含30张不同场景的照片。对收集到的图片进行统一处理，包括调整大小、去噪声和标准化，使其适合输入到KNN和CNN模型中。使用KNN模型进行训练，并选择适当的K值。构建CNN模型，并进行训练。最后评估模型的性能，记录每种算法的识别准确率、错误率及处理时间。

## 实验过程

1. 在网络上收集不同来源的图片，确保多样性。将所有图片进行标签化，标记每张图片对应的对象。
2. 将所有图片调整为统一的尺寸。应用数据增强技术（如旋转、翻转、缩放等）来增加数据集的多样性。
3. 使用数据集进行模型训练。
4. 选取测试集测试模型的准确度

## 实验结果

导入数据集后，随机选取图像进行识别，KNN在导入图像时直接计算与数据集最匹配的标签，识别效果如下图。

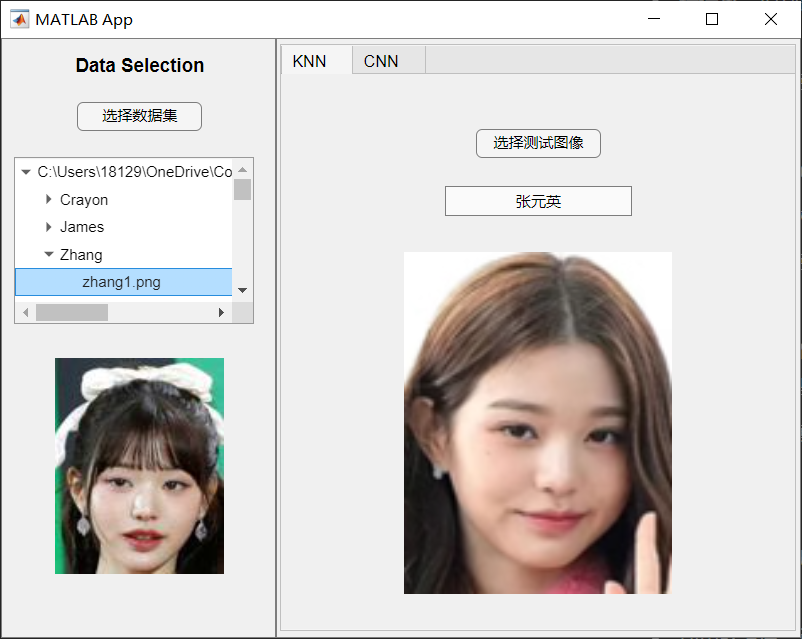
 

图 4‑1 蜡笔小新KNN识别结果 图 ‑2 张元英KNN识别结果

CNN在导入数据后，点击训练等待训练结束，选择测试图像得出最匹配的分类标签，识别效果如下。

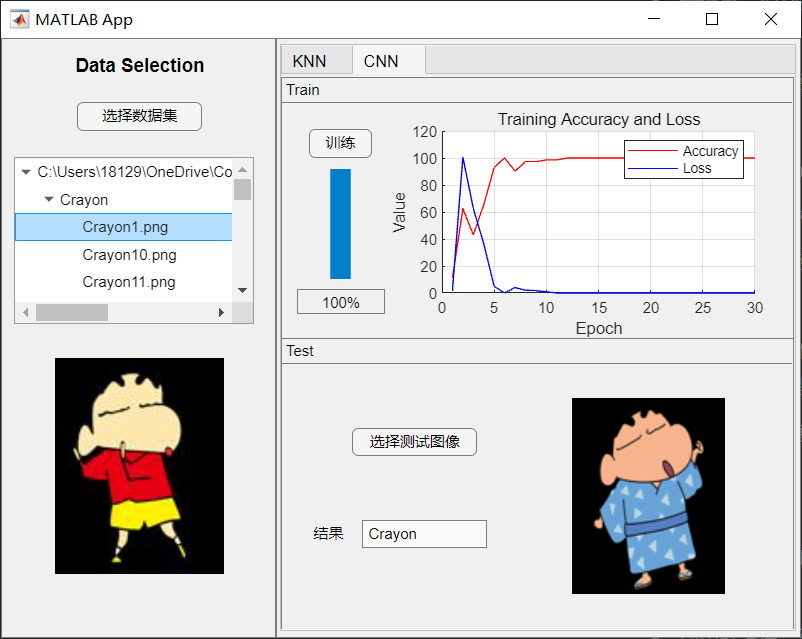
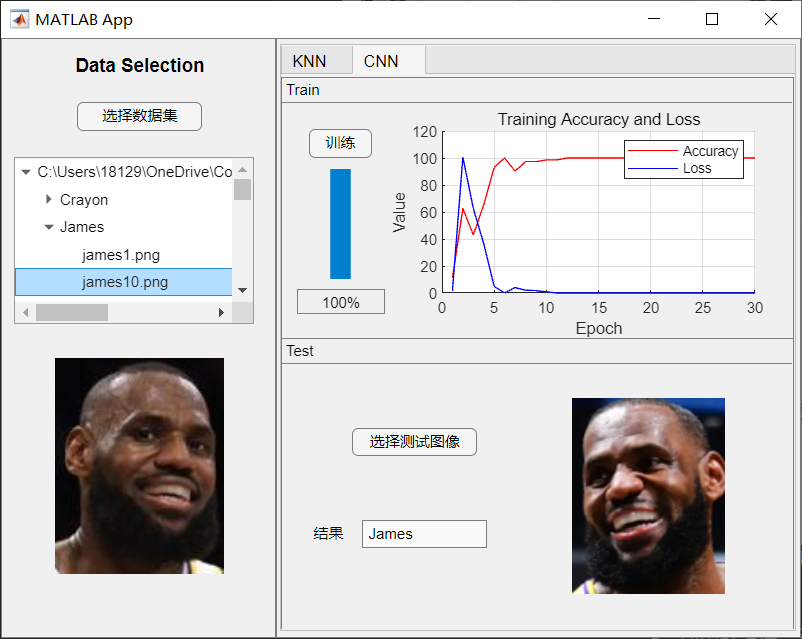
 

图 4‑3 蜡笔小新CNN识别结果 图 ‑4詹姆斯CNN识别结果

经过测试和统计，PCA和CNN的识别准确度都在90%以上，符合预期标准。

# 总结和展望

## 总结

本文研究了基于卷积神经网络（CNN）和主成分分析法（PCA）的人脸识别技术，系统分析了两种算法在人脸识别中的应用效果。通过对PCA和K近邻算法（KNN）的结合，本文实现了对人脸特征的降维与分类，在特定数据集上取得了较好的分类准确率。然而，随着数据量的增加，PCA的局限性逐渐显现，特别是在处理大规模图像数据时，其特征提取能力不足。相较之下，CNN通过自动提取图像的高维特征，展现出更强的识别能力和更高的精度，尤其适用于处理复杂场景下的人脸识别任务。通过实验验证，CNN在人脸识别的表现优于传统的PCA-KNN组合，并在数据处理的效率和准确性方面具备显著优势。

本研究还设计了基于Matlab的图形用户界面（GUI），使用户能够便捷地进行人脸数据的预处理、分类和结果展示。

## 展望

随着人工智能技术的不断进步，人脸识别领域仍有广阔的发展前景。未来的研究可以在以下几个方面展开：

1. 优化CNN结构：虽然CNN在人脸识别中的表现已经非常优秀，但其在训练过程中对硬件资源的要求较高。未来可以进一步研究轻量化网络结构，如MobileNet、ShuffleNet等，以提升模型的计算效率，降低训练成本。
2. 混合模型的探索：尽管CNN已经在图像识别领域占据主流，但传统的PCA、LDA等算法在特定场景下依然具有优势。未来可以探索更多混合模型，将传统方法与深度学习技术相结合，以提高模型的鲁棒性和适应性。
3. 数据增强与迁移学习：为提升模型的泛化能力，未来可以结合数据增强技术和迁移学习，通过扩展训练数据集的多样性来增强模型在不同应用场景下的表现。
4. 应用领域的扩展：人脸识别技术已广泛应用于安防、考勤等领域，但随着深度学习和计算机视觉技术的发展，未来可以探索更多垂直行业的应用场景，如医疗诊断、情感识别等，进一步拓展人脸识别的应用边界。

通过不断优化模型结构、融合更多先进技术，人脸识别技术将在未来的智能社会中扮演更加重要的角色，并为更多领域提供创新性的解决方案。

参考文献

1. 孙志帅.基于Gabor-PCA和KNN-SVM的人脸识别算法[D].null,2016.
2. 简彩仁,庄凌宇,林智鹏.基于降维的人脸识别方法研究与应用[J].信息技术与网络安全,2018,37(12):75-78.DOI:10.19358/j.issn.2096-5133.2018.12.017.
3. Al-Dabagh M Z N, Alhabib M H M, Al-Mukhtar F H. Face recognition system based on kernel discriminant analysis, k-nearest neighbor and support vector machine[J]. International Journal of Research and Engineering, 2018, 5(3): 335-338.
4. 李瑛.基于CNN的人脸识别技术在课堂考勤系统中的应用[D].null,2019
5. 冯友兵，陆轶秋，仲伟波.基于CNN 和SVM 的人脸识别系统的设计与实现∗[J].计算机与数字工程,2021,第49卷(2): 378-382，420
6. Coşkun M, Uçar A, Yildirim Ö, et al. Face recognition based on convolutional neural network[C]//2017 international conference on modern electrical and energy systems (MEES). IEEE, 2017: 376-379.

【Abstract】With the rapid development of deep learning technologies, facial recognition has been widely applied in various fields such as security, surveillance, and finance. This paper addresses key issues in facial recognition by studying a hybrid model based on Convolutional Neural Networks (CNN) and Principal Component Analysis (PCA). First, the basic principles of PCA and CNN are systematically introduced, and these two methods are respectively applied to extract and classify features from facial images. PCA reduces data redundancy through dimensionality reduction, thus improving computational efficiency, while CNN, as a deep learning algorithm, automatically extracts high-dimensional features from images through multiple layers of convolution and pooling, showing high recognition accuracy.

Furthermore, the paper combines the K-Nearest Neighbors (KNN) algorithm to classify facial features after PCA dimensionality reduction and designs a Graphical User Interface (GUI) based on MATLAB to implement facial image preprocessing, classification, and result display. The experiments utilized facial image datasets from various sources and scenes to train and test the PCA and CNN models. The results indicate that the PCA combined with KNN method has certain advantages in specific scenarios of facial recognition, but its performance is relatively limited in large-scale data processing. On the other hand, CNN demonstrated higher accuracy and robustness when handling complex image data, particularly excelling in multi-class classification tasks.

Through comparative analysis, this study concludes that CNN, due to its powerful feature extraction capabilities and reduced reliance on parameters, has broader future applications in facial recognition. Additionally, the combination of traditional algorithms such as PCA and KNN with deep learning techniques may become an important research direction for optimizing algorithm performance in the future. This research provides theoretical support for the development of facial recognition technology and offers practical solutions for identity recognition challenges in complex scenarios.

**【Keywords】**Convolutional Neural Network (CNN); Principal Component Analysis (PCA); facial recognition; K-Nearest Neighbors (KNN); deep learning, image classification

1.系统建模

对于机器人的平衡，纵向以及转向运动，可以将模型简化为三维空间中的类一阶倒立摆模型，即忽略腿长的变化，转向运动由两个驱动轮差速运动控制。

1.1模型定义

称机器人机体质心到两个驱动轮转轴中心的连线为摆杆，得到如图1所示平衡车模型。

该模型的变量与参数定义如表1、表2所示。

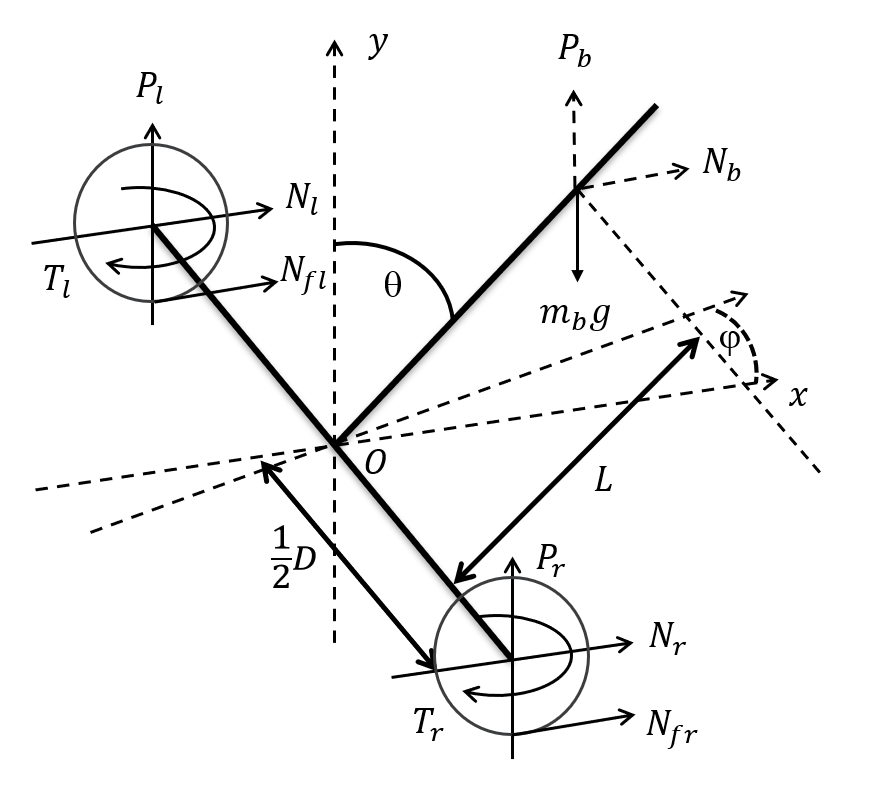


图1 平衡车模型

表1 平衡车模型变量定义

|  |  |  |  |
| --- | --- | --- | --- |
| 符号 | 含义 | 正方向 | 单位 |
| θ | 摆杆与竖直方向夹角 | 图示为正方向 | rad |
| ϕ | 摆杆偏航角 | 图示为正方向 | rad |
|  | 驱动轮位移 | X轴正方向 | m |
|  | 机体位移 | X轴正方向 | m |
|  | 驱动轮对摆杆力的竖直分量 | 箭头所示 | N |
|  | 摆杆受到的竖直方向力 | 箭头所示 | N |
|  | 驱动轮对摆杆力的水平分量 | 箭头所示 | N |
|  | 摆杆受到的水平方向力 | 箭头所示 | N |
|  | 驱动轮输出力矩 | 箭头所示 | N·m |

表2 平衡车模型参数定义

|  |  |  |
| --- | --- | --- |
| 符号 | 单位 | 含义 |
|  |  | 车体转向直径 |
|  |  | 驱动轮半径 |
|  |  | 摆杆质心到驱动轮转轴中心的直线长度 |
|  |  | 驱动轮质量 |
|  |  | 摆杆质量 |
|  |  | 摆杆绕质心关于Z轴的转动惯量 |
|  |  | 驱动轮转子转动惯量 |
|  |  | 整车质心关于Y轴的转动惯量 |

1.2 经典力学分析

对驱动轮，有：

对摆杆，有：

对整车，有：

其中：

1.3 状态空间模型

定义非线性模型：

利用*MATLAB*符号运算工具，联立式（1）到式（8），通过函数solve对方程组消去中间变量，最终得到非线性表达式（9），式（10），和式（11）

定义状态向量x与控制向量u分别为：

根据式状态向量x与控制向量u，求非线性模型在平衡点处的雅可比矩阵并对其线性化，即将，则代入式（9）到（11）后，求：

其中，x、u是系统平衡点，即方程的解：

最终求得的状态方程为：

由于表达式略为复杂，此处用符号代替，具体内容可通过运行MATLAB代码在工作空间中查看。

所有变量均可通过平衡车上各个传感器（如驱动轮编码器，IMU模块等）直接测量或通过数据融合解算得到，故可得系统输出，其中为单位阵。

代入平衡车实车模型参数后可以得到该状态空间模型的状态矩阵A和控制矩阵B，在MATLAB中使用rank函数求得其可控矩阵满秩，系统可控。系统输出矩阵C为单位矩阵，系统可观。

1.4 仿真验证

将Solidwork中搭建的模型分模块导入到仿真软件Webots中，通过对关节电机设置其允许位移为0，得到一个定腿长的平衡车模型。



图2 仿真环境中的整车模型

通过使用MATLAB中的lqr函数计算得到反馈增益矩阵K。因为在仿真中主要目的是验证模型建立是否正确，用于计算K矩阵的Q和R权矩阵的数值选取只关注机器人俯仰角的稳定。经检验，得到模型正确。

2 控制器设计

基于本项目的平衡车构型，除了机器人的平移、转向和保持平衡的控制外，还需要控制平衡车双腿的长度及其横滚角。

2.1 LQR控制器

基于上述求得的状态空间方程，通过一个LQR控制器可以较好地对机器人平移、旋转以及机体俯仰角进行控制。

设计控制律为系统状态的线性组合：

通过线性二次型调节器计算反馈矩阵，定义代价函数为

其中：

* 是系统状态向量；
* 是控制输入向量；
* 是对状态变量的加权矩阵，通常是半正定矩阵，反映了对状态偏离的惩罚；
* 是对控制输入的加权矩阵，通常是正定矩阵，反映了对控制能量的惩罚。

**最优控制律：**

为了使代价函数达到最小，控制输入 应满足以下关系：

其中：

* 是代数 Riccati 方程的解。

**反馈增益矩阵：**

通过上述控制律，反馈增益矩阵 可以表示为：

代数 Riccati 方程**:**

矩阵需要满足以下代数 Riccati 方程：

其中：

* 和 是系统状态空间方程中的系统矩阵和输入矩阵。

为了可以控制平衡车的移动和转向，还需要实现对位移和偏航角两个状态变量的轨迹跟踪。在系统输入中加入参考输入，即状态期望：

其中为：

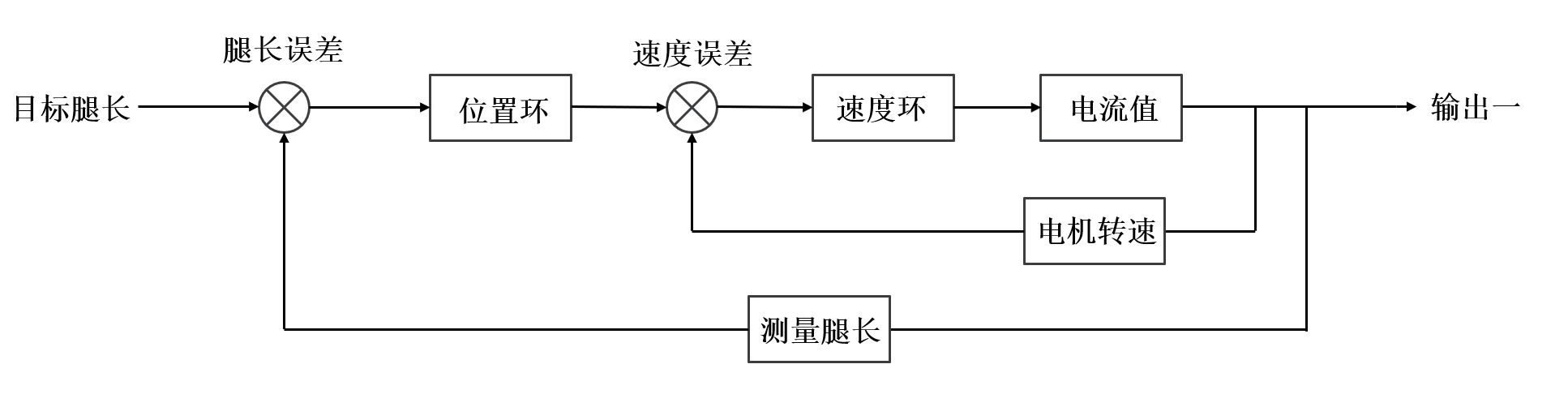
为期望位移，为期望偏航角，两者均由其期望一阶微分量对时间积分得到。

2.2 双环PID控制器

基于该项目平衡车的四连杆腿部构型，可以通过关节电机的输出来调整机器人的腿部长度；同时，为了让机器人在进行视觉识别时机体保持横滚角水平，还需要根据机器人实时的横滚角值来调整两边腿的长度。

为了让机器人的腿长控制有更快的响应速度和鲁棒性，引入两个双环PID控制器分别对机器人的腿长和横滚角进行控制，最后并行输出到电机控制电流值。

腿长的控制框图如图3所示：

图3 机器人腿长双环PID控制框图

横滚角的控制框图如图4所示：

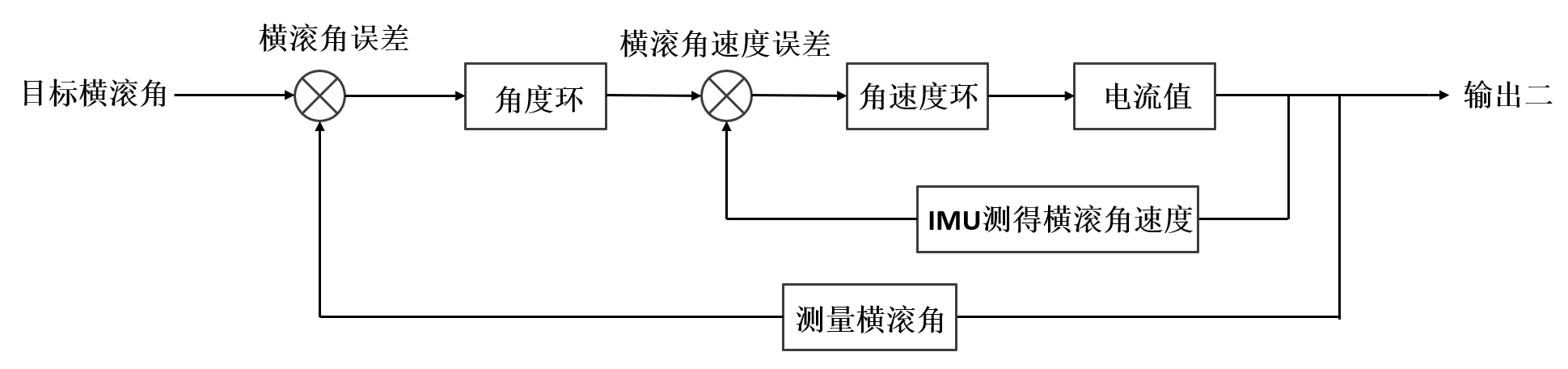


图4 机器人横滚角双环PID控制框图

最终发送的电机电流值为两个控制器的输出之和。

2.3 增益矩阵K的参数拟合

在建模阶段，我们将倒立摆的杆长假设为定值，但实际上，该值会随着机器人腿长的变化而改变，其对应的模型也会有差异。由于求得解析解较为困难，我们将机器人腿长作为拟合的自变量，对增益矩阵K进行四阶多项式拟合。

由于实车环境下我们无法直接测量机器人实时腿长，需要由一些角度关系推算得到。将模型简化至如图5所示。

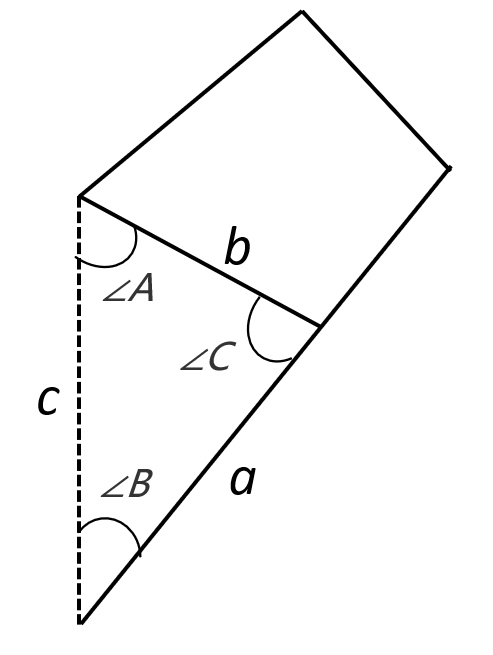


图5 腿部四连杆结构简化模型

其中，三角形边a、边b可以通过在机械图纸中测量得到，角A可以由电机编码器求得，边c则是期望求得的机器人腿长。由余弦定理得式（13）：

通过MATLAB的solve函数求解（排除负数解）得式（14）：

经测量，实车边，边，将数据代入并由0至90度遍历角A可以得到角A与腿长的关系曲线如图6所示。

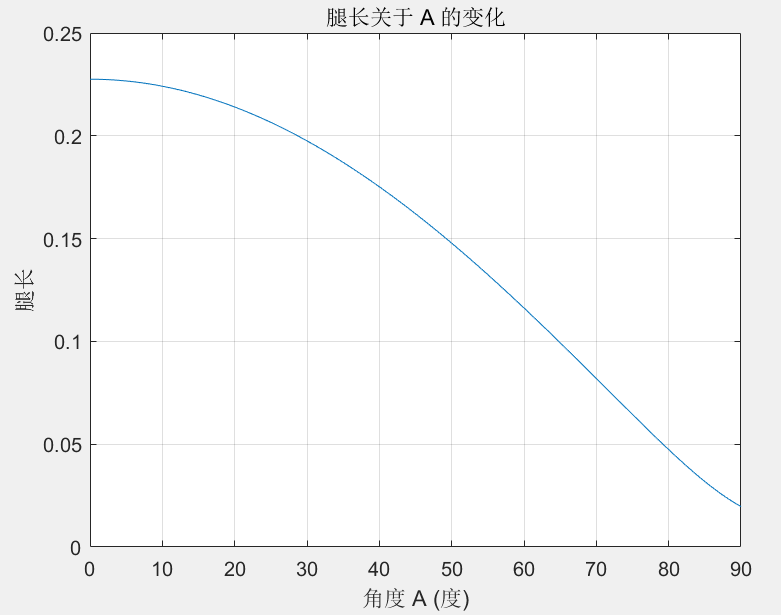


图6 机器人腿长与角度A的关系曲线

在MATLAB中遍历腿长，计算不同腿长值下的增益矩阵K，并对K矩阵每个元素进行关于角度A的多项式拟合，求得拟合系数并存到EXCEL表格中，以便将它们添加进实车代码中进行增益矩阵的实时计算。

拟合公式如式（15）：

其中y对应K矩阵的各个元素，x为测得角度A的值。

最终的拟合效果如图7所示，蓝色曲线为各个角度A值下的实际K矩阵值，紫色曲线为拟合值。

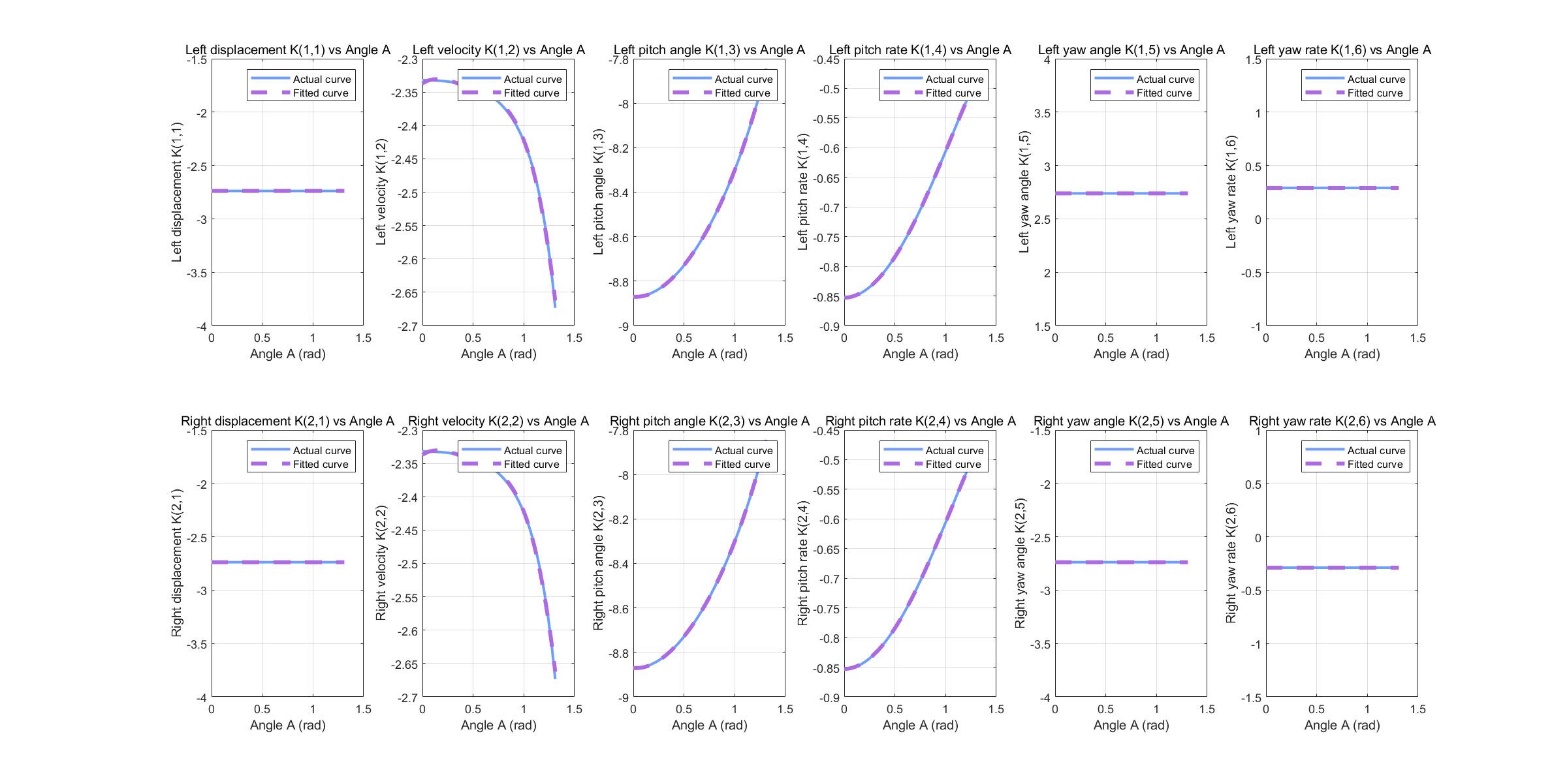


图7 增益矩阵K的实际曲线与拟合曲线对比

通过对比，可知拟合效果满足实车控制需求。

3. 实车硬件选型