南京大学本科生毕业论文（设计）中文摘要

毕业论文题目： FER 系统中基于决策树的表情分类方法研究

软件学院 院系 软件工程 专业 06 级本科生姓名： 顾惟祎

指导教师（姓名、职称）： 刘 峰 讲师

摘要：

人脸表情识别（FER）系统通过人脸定位，特征提取和表情分类三个步骤来识别人类面部表情。

表情分类模块接收特征提取模块处理过后提取出来的参数，包括特征点坐标，灰度等信息。采用特定的分类算法对输入进行处理，输出表情分类。

本文研究 FER 系统中采用决策树分类算法来实现表情分类模块。首先介绍了决策树分类算法的概念和发展过程，给出了算法的详细过程，分析了几种分枝划分标准和连续变量离散化方法，并介绍了 ID3，C4.5,CART 等经典决策树算法。随后文章介绍了 FER 系统中表情分类模块的详细设计，介绍了主要的类和函数。在工程性内容介绍完毕后，文章代入实际实验数据，就如何选择测试条件，如何进行连续变量离散化进行了探讨。实际构建决策树过程中通过信息熵增益准则安排测试条件，并描述了决策树的表义。随后使用构建好的决策树对测试样本集进行了测试，评价了测试结果。文章在最后对论文工作进行了总结和展望。

关键词：人脸表情识别，决策树，信息熵增益，分枝

南京大学本科生毕业论文（设计）英文摘要

THESIS：Research on Buiding an Expression Classification Module in FER by Using the Algorithm of Decision Tree

DEPARTMENT： Software Institution

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Gu Weiyi

MENTOR: Instructor Liu Feng

ABSTRACT：

Facial Expression Recognition System recognize expressions from images or video streams. Its work can be divided into three steps, locating faces, abstract features and classify expressions.

Expression classification module get parameters from dispatcher, which consist of feature points' coordinate and gray scale. This module use certain algorithms to handle the input and output expression classification.

This article research how to use decision tree to classify facial expression. First of all, it introduce the concept and development of decision tree. Then it shows how decision tree works in detail and give some standards, for example entropy gain, to evaluate the effect of branch and discretization. Article presents several classical decision tree algorithms, such as ID3, C4.5, CART and so on. Then it talks about the design of facial expression classification module. After that, article display how to build a decision tree, choosing test statements and separation points for discretizaion in deed. Finishing content about abstract engineer and algorithm, this article shows a complete process of building a decision tree to function expression classification. Module is tested by random photos taken from students and it has a good performance on classification. At last, article summarize previous work and make a prospect.

KEY WORDS: FER, decision tree, entropy gain, branch

目　　录

第一章 绪论

1.1 引言

人脸识别属于生物特征识别技术，是使用人本身的生物特征来区分个体的技术。

生物特征识别技术所研究的生物特征包括脸、指纹、手掌纹、虹膜、视网膜、声音（语音）、体形、个人习惯（例如敲击键盘的力度和频率、签字、步态）等，相应的识别技术就有人脸识别、指纹识别、掌纹识别、虹膜识别、视网膜识别、语音识别（用语音识别可以进行身份识别，也可以进行语音内容的识别，只有前者属于生物特征识别技术）、体形识别、键盘敲击识别、签字识别、步态识别等。

人脸识别的优势在于其自然性和不被被测个体察觉的特点。

所谓自然性，是指该识别方式同人类（甚至其他生物）进行个体识别时所利用的生物特征相同。因为人类本身是可以通过观察比较人脸区分和确认身份的，所以人脸识别是具有自然性的生物特征。另外具有自然性的识别还有语音识别、体形识别等，而指纹识别、虹膜识别等都不具有自然性，因为人类或者其他生物并不通过此类生物特征区别个体。

不被察觉的特点对于一种识别方法也很重要，这会使该识别方法不令人反感，并且因为不容易引起人的注意而减少被刻意欺骗的可能。人脸识别具有这方面的特点，它完全利用可见光获取人脸图像信息，而不同于指纹识别或者虹膜识别，需要利用电子压力传感器采集指纹，或者利用红外线采集虹膜图像，这些特殊的采集方式很容易被人察觉，从而更有可能被伪装欺骗。[1]

基于人脸识别的表情识别是一项更加深入的探索。

表情是人类用来表达情绪的一种基本方式,是非语言交流中的一种有效手段。人们可通过表情准确而微妙地表达自己的思想感情,也可通过表情辨认对方的态度和内心世界。关于表情传递信息的作用,心理学家 Mehrabian 给出了一个公式:感情表露=7%的言词+38%的声音+55%的面部表情。人脸表情识别(FER)所要研究的就是如何自动、可靠、高效地利用人脸表情所传达的信息。

由于以上的特性，人脸表情识别可能拥有以下广阔和多种多样的应用前景：

（1）人机自然交互，为计算机能够识别情绪并进行自然反馈提供技术前提；

（2）机器人制造，使得指定特征后机器人可以自然地模拟人类表情，例如上海世博会西班牙馆的机器婴儿“米格林”；和 4 月 3 日在日本公开展示的，可同步模仿人类表情的机器人“Geminoid TMF”，由大阪大学智能机器人学教授石黑浩带领的科研小组开发；

（3）计算机游戏，虚拟主角；

（4）医学，诊断面部神经瘫痪，人脸图像实时传输；

（5）摄影，捕捉人脸特定表情并自动触发快门，例如 Sony 数码相机的笑脸模式专利；

（6）安全，使用人脸表情进行加密，通过同样的表情才能获得访问资格认证，例如联想的 VeriFace；智能监控；

（7）汽车领域，我们工程最初的目的是实现车载的困倦表情识别，这将大大减少疲劳驾驶导致的车祸事故的发生。而据不充分的统计，当前车祸中，疲劳驾驶为起因的事故率在 40%左右。

人脸表情识别技术可以在以上领域中体现出其非凡的价值，并拥有一个不断扩大的应用前景，这就是我们进行这项研究的动力。

1.2 研究现状

人们对表情识别的研究可以追溯到 20 世纪 70 年代, 早期主要集中在从心理学和生物学方面进行研究和分析。Darwin 首先在《The Expression of the Emotions in Animals and Man（1872）》一书中揭示了表情在不同性别, 不同种族的人群中的一致性，这使得我们可以通过研究特定群体的表情样本来寻找普适的表情规律。[2]

Ekman和Frisen提出面部表情编码系统(FACS)，用44个运动单元（Action Unit）来描述人脸表情变化，并定义了6种基本情感类别：惊奇、恐惧、厌恶、愤怒、高兴、悲伤。这一系统得到了广泛的认同，并成为后来很多表情识别研究工作的基础。

现在的人脸表情识别系统（Facial Expression Recognition System）一般包括 3 个环节：人脸检测与定位，特征提取，表情分类。如（图 1.1）所示：

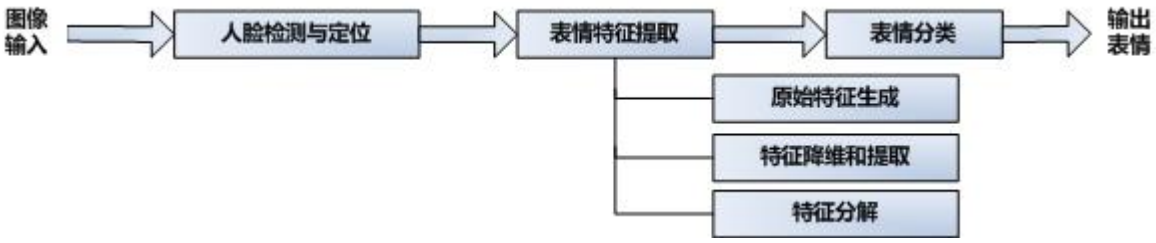


图 1.1 人脸表情识别系统

建立一个 FER 系统，第一步进行的是人脸检测与定位，这一环节的研究实际上已经成为一个独立的方向[3]。第二步从一张人脸图像或者图像序列中提取能够表征输入表情本质的信息，在提取特征数据的过程中，为了避免维数危机，可能还需要特征降维、特征分解等进一步处理；第三部分析特征之间的关系，将输入的人脸表情分类到相应的类别，如运动单元组合或基本感情类别。

1.2.1 表情特征提取的方法

（1）使用原始特征生成的方法

基于几何特征的方法：使用训练后的人脸几何模型进行匹配，从正面人脸提取若干特征点。几何特征的识别效果对特征点的准确性邀请很高，在图像质量低和背景复杂的情况下难以实现。同时几何特征的提取忽略了脸部其他部分的信息，比如皮肤的纹理变化等，因此在识别表情时存在着一定的误差。比如实验中 MK.FE3.133.bmp 图像（图 1.2）的众多几何特征均表征为高兴（happy），但是人类感知很容易辨认出来是恐惧（fear）表情，其中人类认知中额头的纹理起到了十分关键的作用。

图 1.2 JAFFE 数据库 MK.FE3.133.bmp

基于外貌特征的方法：外貌特征泛指使用全部人脸图像像素的特征，反映了人脸图像底层的信息。这类方法基于局部特征，利用一组滤波器对图像进行滤波，结果可以反映局部像素之间的关系（梯度、相关性、纹理等）。

基于混合特征的方法：几何特征能够简洁地表示出人脸宏观的结构变化，而外貌特征侧重于提取局部的细微变化，一些研究将两者结合起来，用混合的特征进行表情识别。《使用动态信息融合理解图像序列中的人脸表情》一文提出使用多种特征融合的方法进行表情识别，其中包括：几何特征提取，法令纹检测，前额区域边缘检测，后两者均为脸部瞬时变化出现的特征[4]。如图 1.3 所示：

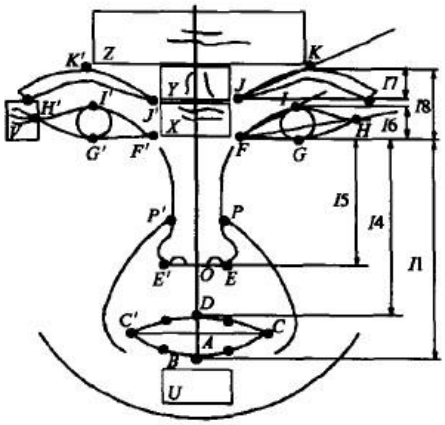


图 1.3 特征点的几何关系（方形区域表示皱纹）

（2）特征降维和提取的方法

由于特征表示方法尤其是外貌特征的空间维数通常非常巨大，因此需要通过一些映射或变换将它们转换到低维的子空间来表示。这样不仅可以使特征的维数明显降低，同时这些低维空间特征的有效性也将得到提高。常用的特征降维及提取的方法有主元分析（PCA），线性判别分析（LDA），ICA等。这些方法在进行特征提取时各有优势，如PCA提取了最有代表性的特征，可以有效地消除冗余，降低维数，但它没有考虑不同类别数据之间的区分性。而LDA则通过最大化数据的类间离散度和最小化类内离散度来选择合适的投影方向，侧重于寻找具有最大分辨力的方向。

（3）特征分解的方法

人脸图像包含了丰富的信息，对不同的识别任务来说，所利用的信息也各不相同。人脸检测寻找的是

人脸图像共有的一致性，人脸识别需要利用表示人脸个体差异的信息，而表情识别则需要表示各种表情之间差异的信息，对一种识别任务有利的信息有可能反而对其他识别任务造成干扰。近来，一种新的解决思路是把人脸不同的因素如表情因素和个体因素分离开来，使得识别能够在相应的子空间中进行，避免其他因素的干扰。

基于特征分解的方法在分类过程中需要对已知表情类别的样本库进行遍历搜索。首先假设测试样本的个体因素来自第1个训练的个体，分解得到相应的表情分量，计算测试样本表情分量与该训练样本表情分量的相似性(如余弦距离或欧式距离)，对所有训练个体重复这一过程，最终将测试人脸分类到最相近的表情类别。

1.2.2 表情分类的方法

（1）线性分类器

假设不同类别的模式空间线性可分，引起可分的主要原因是不同表情之间的差异。

（2）人工神经网络（Artificial Neural Network）

人工神经网络是一种模拟人脑神经元细胞的网络结构，它是由大量简单的基本元件——神经元，相互连接成的自适应非线性动态系统。将人脸特征的坐标位置及其相应的灰度值作为神经网络的输入，ANN 可以提供很难想象的复杂的类间分界面。神经网络分类器主要有：多层感知器、BP 网、RBF 网。

其缺点在于，需要大量的训练样本和训练时间，不能满足实时处理要求。

（3）隐马尔科夫模型（Hiden Markov Model）

用HMM方法进行识别需要确定其初始和结束的状态，因此一般用于单独的表情序列或分割好的表情序列。Cohen提出多层次的HMM，第1层为分别针对6种表情的6个HMM模型；第2层为表示6种表情之间状态变化的Markov模型[5]。将单个表情HMM的状态输出联合起来作为高层Markov模型的输入，并通过训练得到6种表情之间的转移概率，从而可以自动将视频分割为不同的表情段进行分析。

（4）支持向量机（Support Vector Machine）

支持向量机是在统计学习理论的基础上发展起来的一种分类方法，在解决小样本、非线性和高维模式识别问题上有很多优势。目前支持向量机是机器学习中应用最多的分类器之一，近年来也被应用于表情识别中[6]。

其基本思想是对于非线性可分样本，首先通过非线性变换将输入空间变换到一个高维空间，然后在这个新空间中求取最优线性分界面。这种非线性变换通过定义适当的内积函数实现，常用的三种内积函数为：多项式内积函数、径向内积函数、Sigmoid内积函数。

（5）其他

基于人脸物理模型的识别方法，将人脸图像建模为可变形的3D网格表面，把空间和灰度放在一个3D空间中同时考虑。

基于模型图像解码的方法是使用遗传算法来编码、识别与合成各种不同的表情。

1.2.3 人脸表情数据库

目前在人脸表情识别研究中使用最广泛的是CMU的Cohn-Kanade数据库 ，在很多研究中被列为算法比较的标准。Cohn-Kanade数据库是基于AU编码的数据库，含有210个对象的大约2000个图像序列，每个人有一系列的脸部活动，包括单个AU或者AU组合。

其次，日本ATR的女性表情数据库（JAFFE）也得到较多使用。JAFFE是以7种基本表情为基础的数据库，包括10位日本女性，每种表情有三至四幅图像，总共213幅图像。此外，还有一些数据库也可应用于人脸表情识别，如Ekman和Friesen的人脸表情数据库，Yale人脸数据库，Purdue的AR人脸数据库，CMU 的PIE数据库等。

本项目中使用JAFFE数据库进行训练。

1.2.4 研究展望

（1）鲁棒性有待提高：外界因素，主要是头部偏转及光线变化的干扰需要提出合理的解决方案。目前采用多摄像头技术、色彩补偿技术予以解决，有一定效果，但并不理想。

（2）表情识别计算量有待降低，以确保实时性的要求。

（3）加强多信息技术的融合。

面部表情不是唯一的情感表现方式，综合语音语调、脉搏、体温等多方面信息来源来更准确的推测人的内心情感，将是表情识别技术需要考虑的问题。[7]

1.3 论文的主要工作及论文组织结构

本文研究了如何使用决策树来进行表情分类，用来训练的表情数据库为日本女性表情数据库 JAFFE，而用来测试的有 JAFFE 和自行拍摄的一组表情照片。输入为系统特征提取出来的特征点坐标，输出为表情分类。在选择测试条件过程中，通过实验筛选出具有区分性的几个特征向量。其次得到样本的几个和测试条件相关的统计量，根据直接划分、信息熵增益、信息熵增益比三种方式来安排测试条件，构建决策树。最后对在摄影中具有实用意义的笑脸模式进行了分析。

本文论文组织结构如下:

第一章对人脸表情识别的研究背景、研究现状以及本文主要工作做了一个概要的介绍。

第二章对决策树算法的概念及发展进行了介绍，并阐述了一些经典算法是如何实现的。

第三章进行毕业设计项目概述，对系统需求、设计及实现进行了描述。

第四章介绍了在人脸表情识别系统中应用决策树算法进行表情分类。

第五章对毕业设计和毕业论文工作进行了总结，并对进一步的工作进行了展望。

最后对在工作中给予帮助的朋友师长们表示了感谢。

第二章 决策树算法

2.1 决策树的概念和发展

决策树学习着眼于从一组无序、无规则的已知样本中归纳出决策树表示形式的分类知识。该方法生成一棵决策树表示学习所获得的假设，用于近似描述训练样本所确定的未知函数。在分类时，利用生成的决策树预测未知样本的类别。

代表了对象属性与对象值之间的一种映射关系。树中每个内部结点对一个或多个属性的取值进行某种测试比较，并根据不同的比较结果确定该节点的分枝；每个分叉路径就代表了某个可能的属性值；决策树的叶结点给出相应的类别标志，表示到达该叶结点的样本所属的类别。

决策树仅有单一输出，若想要复数输出，可以建立独立的决策树以处理不同输出。

决策树方法在机器学习领域曾经得到广泛而深入的研究。最早的决策树算法是 Quinlan 提出的 ID3 算法，该方法以数据集中各字段的信息增益为依据，以信息增益最大的字段作为决策树的根结点；并依次对各个子树进行类似的操作，直到确定决策树的所有结点。在 ID3 算法提出之后，决策树方法出现了很多改进型的算法，如 ID4，ID5，C4.5 等。决策树方法可用于数据挖掘中的数据分类。

2.2 数据挖掘分类算法[8]

分类技术在很多领域都有应用，例如可以通过客户分类构造一个分类模型来对银行贷款进行风险评估；当前的市场营销中很重要的一个特点是强调客户细分。客户类别分析的功能也在于此，采用数据挖掘中的分类技术，可以将客户分成不同的类别，比如呼叫中心设计时可以分为：呼叫频繁的客户、偶然大量呼叫的客户、稳定呼叫的客户、其他，帮助呼叫中心寻找出这些不同种类客户之间的特征，这样的分类模型可以让用户了解不同行为类别客户的分布特征；其他分类应用如文献检索和搜索引擎中的自动文本分类技术；安全领域有基于分类技术的入侵检测等等。机器学习、专家系统、统计学和神经网络等领域的研究人员已经提出了许多具体的分类方法。分类算法的训练和分类过程简单描述如下：

训练：训练集——>特征选取——>训练——>分类器
分类：新样本——>特征选取——>分类——>判决

最初的数据挖掘分类应用大多都是在这些方法及基于内存基础上所构造的算法。目前数据挖掘方法都要求具有基于外存以处理大规模数据集合能力且具有可扩展能力。以下是当前几个主要的分类算法：

（1）决策树

决策树归纳是经典的分类算法。它采用自顶向下递归的各个击破方式构造决策树。树的每一个结点上使用特定标准选择测试属性。可以从生成的决策树中提取规则。

（2）KNN 法（K-Nearest Neighbor）

KNN 法即 K 最近邻法，最初由 Cover 和 Hart 于 1968 年提出的，是一个理论上比较成熟的方法。该方法的思路非常简单直观：如果一个样本在特征空间中的 k 个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

KNN 方法虽然从原理上也依赖于极限定理，但在类别决策时，只与极少量的相邻样本有关。因此，采用这种方法可以较好地避免样本的不平衡问题。另外，由于 KNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN 方法较其他方法更为适合。

该方法的不足之处是计算量较大，因为对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的 K 个最近邻点。目前常用的解决方法是事先对已知样本点进行剪辑，事先去除对分类作用不大的样本。另外还有一种 Reverse KNN 法，能降低 KNN 算法的计算复杂度，提高分类的效率。

该算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分。

（3） SVM 法

SVM 法即支持向量机(Support Vector Machine)法，由 Vapnik 等人于 1995 年提出，具有相对优良的性能指标。该方法是建立在统计学习理论基础上的机器学习方法。通过学习算法，SVM 可以自动寻找出那些对分类有较好区分能力的支持向量，由此构造出的分类器可以最大化类与类的间隔，因而有较好的适应能力和较高的分准率。该方法只需要由各类域的边界样本的类别来决定最后的分类结果。

支持向量机算法的目的在于寻找一个超平面 H(d), 该超平面可以将训练集中的数据分开，且与类域边界的沿垂直于该超平面方向的距离最大，故 SVM 法亦被称为最大边缘(maximum margin)算法。待分样本集中的大部分样本不是支持向量，移去或者减少这些样本对分类结果没有影响，SVM 法对小样本情况下的自动分类有着较好的分类结果。

（4）VSM 法

VSM 法即向量空间模型(Vector Space Model)法，由 Salton 等人于 60 年代末提出。这是最早也是最出名的信息检索方面的数学模型。其基本思想是将文档表示为加权的特征向量：$D=D(T_1, W_1; T_2, W_2; \ldots; T_n, W_n)$，然后通过计算文本相似度的方法来确定待分样本的类别。当文本被表示为空间向量模型的时候，文本的相似度就可以借助特征向量之间的内积来表示。

在实际应用中，VSM 法一般事先依据语料库中的训练样本和分类体系建立类别向量空间。当需要对一篇待分样本进行分类的时候，只需要计算待分样本和每一个类别向量的相似度即内积，然后选取相似度最大的类别作为该待分样本所对应的类别。

由于 VSM 法中需要事先计算类别的空间向量，而该空间向量的建立又很大程度的依赖于该类别向量中所包含的特征项。根据研究发现，类别中所包含的非零特征项越多，其包含的每个特征项对于类别的表达能力越弱。因此，VSM 法相对其他分类方法而言，更适合于专业文献的分类。

（5）Bayes 法

Bayes 法是一种在已知先验概率与类条件概率的情况下的模式分类方法，待分样本的分类结果取决于各类域中样本的全体。

Bayes 方法的薄弱环节在于实际情况下，类别总体的概率分布和各类样本的概率分布函数(或密度函数)常常是不知道的。为了获得它们，就要求样本足够大。另外，Bayes 法要求表达文本的主题词相互独立，这样的条件在实际文本中一般很难满足，因此该方法往往在效果上难以达到理论上的最大值。

（6）神经网络

神经网络分类算法的重点是构造阈值逻辑单元，一个值逻辑单元是一个对象，它可以输入一组加权系数的量，对它们进行求和，如果这个和达到或者超过了某个阈值，输出一个量。如有输入值 $X_1, X_2, ..., X_n$ 和它们的权系数：$W_1, W_2, ..., W_n$，求和计算出的 X_i*W_i ，产生了激发层 $a = (X_1 * W_1)+(X_2 * W_2)+...+(X_i * W_i)+...+ (X_n * W_n)$，其中 X_i 是各条记录出现频率或其他参数，W_i 是实时特征评估模型中得到的权系数。神经网络是基于经验风险最小化原则的学习算法，有一些固有的缺陷，比如层数和神经元个数难以确定，容易陷入局部极小，还有过学习现象，这些本身的缺陷在 SVM 算法中可以得到很好的解决。

2.3 决策树分类的评价

相对于其他数据挖掘算法，决策树在以下几个方面拥有优势[9]：

（1）决策树易于理解和实现。人们在通过解释后都有能力去理解决策树所表达的意义。

（2）对于决策树，数据的准备往往是简单或者是不必要的。其他的技术往往要求先把数据一般化，比如去掉多余的或者空白的属性。

（3）能够同时处理数据型和常规型属性，即同时具有处理离散属性或连续属性的能力。其他的技术往往要求数据属性的单一。

（4）使用白盒模型。如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。

（5）易于通过静态测试来对模型进行评测。表示有可能测量该模型的可信度。

（6）在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

而相对于常规统计方法来说，决策树算法本身具有的劣势有：

（1）难以预测连续属性，并给出合理的离散化。

（2）对有时间顺序的数据，需要很多预处理的工作。

（3）当类别太多时，错误可能就会增加的比较快，这是因为深层的决策树往往缺乏有效地泛化能力。

（4）一般的算法分类的时候，只是根据一个属性来分类，后期需要大量的规则化工作来使得决策条件清晰明了。

2.4 决策树实现

决策树的工作方式：自上而下生成。生成过程中选择分割的方法有好几种，但是目的都是一致的：对目标类尝试进行最佳的分割。从根到叶子结点都有一条路径，这条路径就是一条“规则”。

其中对每个结点的衡量有：

（1）通过该结点的记录数；

（2）如果是叶子结点的话，分类的路径；

（3）对叶子结点正确分类的比例。

实现决策树学习的基本步骤包括：决策树创建、分枝划分标准、连续属性离散化、决策树剪枝。

2.4.1 决策树创建

使用分而治之[10]的思想，利用训练样本集 T 创建一棵决策树的过程可分 3 种情况进行：

（1）T 中包含一个或多个样本，所有的样本均属于同一种类别 c_j。T 所对应的决策树为一个叶子结点，类别表示为 c_j。

（2）T 中没有样本。T 所对应的决策树仍为一个叶子结点，但是由于根据 T 中的信息无法确定该叶子结点的类别标识，因此类别标识需要有其他信息确定。可以根据应用领域的某些背景信息，或利用大多数原则，如所有训练样本中大多数样本所属类别，或父亲结点所对应的样本几种大多数样本所属类别。

（3）T 中的样本属于多个类别。在这种情况下，需要对 T 进行细分，尽量使每个子集中的样本只属于一种类别。此时，需要设置一个测试准则，根据某一属性或某些属性的信息得到若干互不相交的测试输出条件$\{O_1,O_2,\cdots,O_n\}$，每个测试输出条件 O_i 对应着 T 的一个样本子集 T_i $(1\leqslant i\leqslant n)$，这些子集构成对样本集 T 的一个划分。

T 所对应的决策树即由具有上述测试输出地内部结点以及 n 个分枝组成，每个分枝对应着一个测试输出。然后对每个子集递归地采用上述决策树创建过程。整个过程流程如图 2.1 所示：

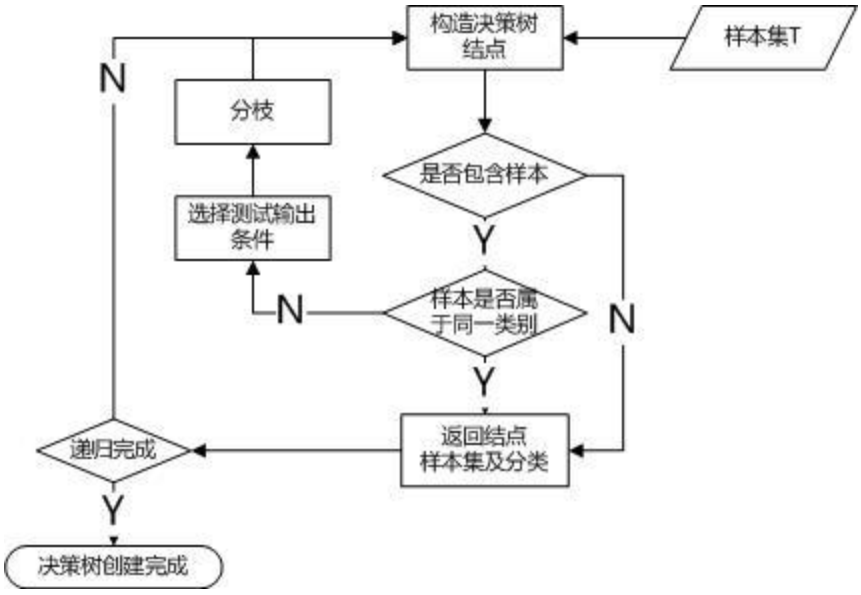


图 2.1 决策树创建流程

2.4.2 分枝划分标准

在决策树创建过程中，如何确定内部结点的分枝是最为关键的问题。对决策树的当前结点进行分枝，对应着当前样本集的一个划分，样本集的划分是问题的本质。在确定内部结点的分枝过程中，需要根据某

个划分标准获得一个最佳测试，该测试的不同输出对应着当前结点的各个分枝。采用不同的划分标准，得到的决策树会有很大的差异。

(1) 直接划分法

直接划分法没有明确的划分标准，其思想是：按照属性在样本的属性向量中的出现顺序，选择从根结点到当前结点的路径中尚未出现的下一个属性，作为当前结点的测试属性，根据该属性的不同取值，将当前训练样本集划分为相应的子集；如果该属性不能对当前样本集进行进一步的划分，则顺序选择下一个属性。

(2) 信息熵增益

基于 Shannon 在 1948 年提出了信息理论以数学的方法度量并研究信息，使用信息量来度量具有确定概率事件发生（图 2.2）时所传递的信息量。Quinlan 于 1979 年提出以信息熵的下降速度作为选择测试属性的标准，该方法称为信息熵增益。

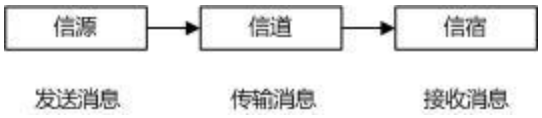


图 2.2 消息传输模型

在消息传输模型中，消息 a_i 的自信息量 $I(a_i)$ 是对信源向信宿发出消息 a_i 的不确定性的度量，定义为：

$$I(a_i) = -\log_2 p(a_i)$$

公式 2.1

其中 $p(a_i)$ 为信源发送消息 a_i 的概率（直接使用以 2 为底的对数，单位为 bit，整篇论文里均这样使用，不做特殊说明）。

信息熵可以用来度量整个消息集合的不确定性，定义如下：

$$H(X) = -\sum_{i=1}^{r} p(a_i) * \log_2 p(a_i)$$

公式 2.2

其中 r 为所有互不相同的消息数目，X 为这些消息的集合。公式 2.2 定义了 X 的信息熵的实际含义就是传输消息集合 X 中的一个消息所需要的平均信息量。

如果这样一种情况，信源发送的随机变量 X 与随机变量 Y 不是互相独立的，且信宿已经接收到信息 Y。这时候，使用条件熵 H (X | Y) 来度量信宿收到随机变量 Y 之后，信源传输随机变量 X 仍然存在的不确定性。记 X 所对应的消息用 $a_i(1 \leqslant i \leqslant r)$ 表示，Y 所对应的消息用 $b_j(1 \leqslant j \leqslant s)$ 表示，则 X 相对于 Y 的条件熵如公式 2.3 所示：

$$H(X \mid Y) = \sum_{i=1}^{r} \sum_{j=1}^{s} p(a_i, b_j) * \log_2 p(a_i \mid b_j)$$

公式 2.3

其中，$p(a_i,b_j)$表示 X 为 a_i 且 Y 为 b_j 的概率；$p(a_i \mid b_j)$表示当 Y 为 b_j 时，X 为 a_i 的概率。

信息熵增益是有 Quinlan 提出的，它首先在 ID3 系统中得以应用，其思想是将样本的分类问题看成一个通信问题，假设我们从样本集合 T 中随机选取一个样本，且这个样本属于类别 c_j，则这一消息传输的信息量为：

$$-\log_2\left(\frac{freq(c_j,T)}{|T|}\right)\ bit$$

公式 2.4

其中 $freq(c_j, T)$表示样本集 T 中属于类别 c_j 的样本数目。

定义上述消息对所有类别成员的信息量的期望为：

$$Info(T) = -\sum_{j=1}^{m}\frac{freq(c_j,T)}{|T|} * \log_2\left(\frac{freq(c_j,T)}{|T|}\right)\ bit$$

公式 2.5

其中 m 为类别数；Info(T)就是样本集合 T 的信息熵，它表示为了标识 T 中某个样本的类别所需要的平均信息量。

对于一个产生 k 个输出，即将样本集 T 划分为 k 个子集的测试条件φ，划分后的信息量等于各个样本子集的信息量的加权和，其值为：

$$Info_{\varphi}(T) = \sum_{i=1}^{k}\frac{|T_i|}{|T|} * Info(T_i)\ bit$$

公式 2.6

在此基础上产生的信息熵增益定义为：

$$Gain(\varphi) = Info(T) - Info_{\varphi}(T)\ bit$$

公式 2.7

信息熵增益表示利用测试条件φ对样本集合 T 进行分割后所获得的信息量。使用信息熵增益的标准来进行分枝，就是选择信息熵增益最大的测试条件来进行决策树当前结点的分枝。

(3) 增益比

在特定情境下，信息熵增益法存在着一个严重的缺陷：即它偏向于指定输出分枝较多的测试条件。例如在确定体育周举办什么类型的球类比赛的时候，会以学生的运动爱好作为一个合理的测试条件，然而信息熵增益却会偏向于使用分枝更多的学号作为第一个使用的测试条件，同理在其他样本的分类中会使用一些唯一标识作为首选测试条件。因为这样的分类结果使得每个子集中只包含一个样本，即在公式 2.5 中，

m=1，freq(cj, T)=1，|T|=1，所以 Info(T)=0 bit，即每个子集的信息熵均为 0 bit。由此学号的信息熵增益

$$Gain(ID) = Info(T)-InfoID(T)=Info(T)$$

最大。

但是这样生成的决策树不仅产生了大量的分枝，导致决策树规模庞大，而且该决策树对预测新样本的类别毫无用处，因为学号属性是一个与类别无关的属性，对判定运动爱好没有任何作用，任何未知样本的学号属性值均与训练样本集中各个样本的学号值不同。

针对此缺陷，Quinlan 进一步提出了增益比的划分标准。增益比标准对有较多条件分枝的情况进行了调整，该方法考虑的消息中的信息内容不仅仅是样本的类别，还包括了该样本落入测试条件的哪个输出分枝。与 Info(T)的定义方式类似，Quinlan 定义了测试条件φ所带来的分割信息熵 SplitInfo(φ)：

$$SplitInfo(\varphi) = -\sum_{i=1}^{k} \frac{|T_i|}{|T|} * \log_2\left(\frac{|T_i|}{|T|}\right) \text{ bit}$$

公式 2.8

SplitInfo(φ)表明将训练样本集 T 分割为 k 个子集产生的潜在信息量，而信息熵增益 Gain(φ)度量的是划分所带来的与分类相关的信息量。在某种程度上讲，分割信息熵可以看作实现相应划分所需的代价，因为条件分枝越多，用户需要获取的信息也就越多，划分的代价就越高。在信息熵增益相同的情况下，分割信息熵越小越好。Quinlan 定义的增益比就是信息熵增益与分割信息熵的比值：

$$GainRatio(\varphi) = Gain(\varphi) \Big/ SplitInfo(\varphi)$$

公式 2.9

增益比标准进行分枝划分就是，选择增益比比值最大的属性作为当前内部结点的测试属性进行分枝。

需要说明的是，采用增益比标准，当划分产生的分割信息熵很小的时候，增益比比值不稳定：

第一种情况，划分后只有一个自己种有样本，那么分割信息熵为 0，即公式 2.9 中的分母为 0；

第二种情况，对于某些划分，虽然产生的信息熵增益很小，但是由于其分割信息熵也很小，往往能够获得很大的增益比。

为了避免上述上述情况的干扰，增益比标准设定了一个约束条件，即信息熵增益必须足够大，不低于所有测试条件的平均信息熵增益。

在实际使用中，增益比标准时在满足约束条件的情况下，选择增益比最大的测试条件进行划分。

(4) Gini 系数

如下定义一个 Gini 系数：

$$Gini(T) = 1 - \sum_{i=1}^{m} p_i^2$$

公式 2.10

其中 m 为样本集 T 中样本类别的数目，p_i 为类别 c_i 在样本集 T 中出现的频率。

如果测试条件φ将集合 T 划分成为 k 个子集，分别为 $T_1,T_2,\cdots,T_k$，那么该划分的 Gini 系数为

$$Gini_{split}(T,\varphi) = \sum_{i=1}^{k} \frac{|T_i|}{|T|} * Gini(T_i)$$

公式 2.11

根据公式 2.10 函数的大体走势来说，每个样本集的类别成分越单纯，Gini 系数值越小。而相应的测试条件划分所产生的决策树分枝越能代表不同类别之间的差异，因此造成错误分类的几率也越低。故在选择内部结点的测试条件时，Gini 系数越大的属性，包含信息量越少。

Gini 系数进行分枝划分就是，在所有可能的划分中，选择 $Gini_{split}$ 最小的测试条件进行划分。

（5）其他

最短距离划分，使用样本集之间的 Mantaras 距离作为分枝划分标准；

最短距离划分。

论文实验过程中没有使用这两种分枝划分方法，在此不做详述。

2.4.3 连续属性离散化

参与分类的样本属性有两种：离散属性和连续属性。离散属性的取值之间没有明确的顺序关系。连续属性的取值之间具有一定的顺序关系，一般为某数域内的各种取值，如果在决策树的学习中，直接像处理离散属性一样处理连续属性，则一旦某连续属性被选为决策树内部结点的测试属性，将产生很多分枝，最后生成的决策树不仅结构庞大，而且难以应用到实际的分类问题中，因此需要在决策树生成前或生成过程中对连续属性进行离散化。

连续属性离散化是将连续属性的取值划分为若干区间，从而将连续属性作为离散属性来处理，好在决策树的每个结点根据使用连续属性进行分枝。这些区间即为该离散属性的取值。

（1） 典型离散化处理过程

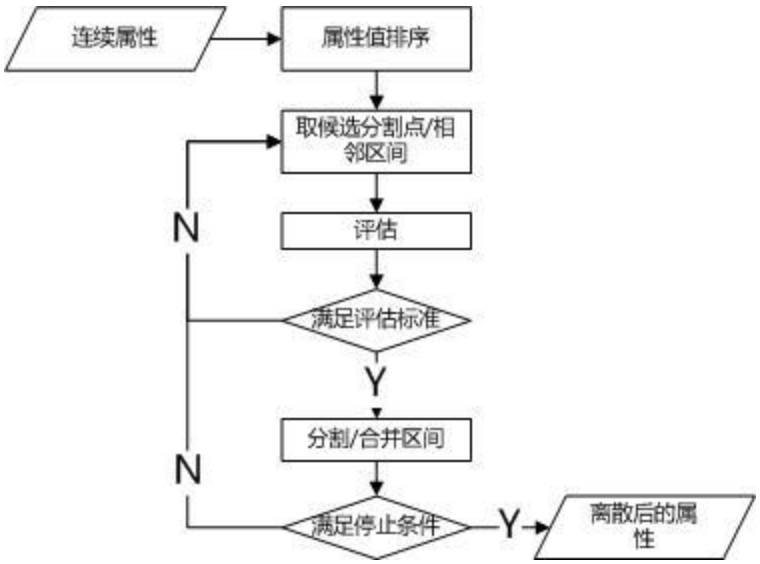


图 2.3 连续属性离散化处理流程

1、将连续属性的值按递增或递减的顺序进行排序。排序时离散化过程中计算量大，耗时长的步骤，对其进行快速处理是提高算法性能的关键。

2、分割点/相邻区间评估：排序结束后，根据评价准侧对分割点（分割法）或相邻区间（合并法）进行评估。这些评价准则可以是信息熵增益、最小错误率等等。

3、区间分割或合并：对于分割法，根据最佳分割点将一个连续值区间划分为两个子区间，并得到落入这两个子区间的样本集。对于合并法，则将两个最佳相邻区间合并为一个区间，对应样本集也进行合并。

4、终止条件判断：重复步骤 2 和步骤 3，对于分割法，每次增加一个区间，对于合并法，每次减少一个区间，直到满足终止条件，处理过程结束。

（2） 常用离散化方法

1.、等宽度划分（Equal Width）

不使用类别信息，即是非监督的离散化方法。用户给定区间数目 k，该方法将连续属性的取值区间划分为宽度相等的 k 个区间。k 取值对离散化结果的影响和很大，而且直接宽度划分可能使某些区间内的样本数目过少，因此效果往往不太理想。

2、等频率划分（Equal Frequence）

该方法也是一种非监督离散化方法，根据用户给定的区间数目 k，将连续属性的取值区间分成 k 个区间，使得落入每个区间内的样本数目相同。这种方法对 k 的取值很敏感，若某个取值频繁出现，会导致同一取值出现在不同的区间中，因此需要对区间边界进行特殊处理。

3、1R

Holte 提出的 1R 算法是一种监督离散方法，根据连续属性的取值对样本进行排序后，将整个取值区间划分成互不相交的若干个区间，并且利用类别信息确定区间的边界。每个区间至少包括一定数量的样本，如果下一个样本的类别与某区间内出现频率最高的类别相同，则此样本归纳于这个区间。最后一个区间包括剩余样本。该算法的终止条件是由每个区间内的最少样本数来隐式地决定的，最少样本数默认为 6。

4、二分法

二分法是决策树学习中最常用的方法，它采用信息熵增益或增益比作为评估标准，对连续属性的每次处理均是将其当前取值范围分为两个区间，因此只需寻找一个最佳分割点就可确定离散后的区间边界。二分法是一种动态局部离散化方法，离散化过程融合在决策树生成过程中，边学习边进行离散化处理。其终止条件是到达每个叶结点的样本属于同一类别，一般需要根据实际情况对该条件进行适当放宽，即存在一定的容错。

5、D2

Catlett 提出的 D2 算法是一种静态方法，它通过递归调用二分法得到多个区间，离散后的结果再用于创建分类器。当训练样本集的规模很大时，相对于二分法，该方法能有效地提高归纳学习的速度。D2 算法的终止条件可为下列条件之一：当前样本集内的样本数少于 14；区间数目大于 8；所有候选分割点获得的信息熵增益相同；区间内的所有样本属于同一类别。

6、Entropy(MDLP)

Fayyad 和 Irani 提出的 Entropy(MDLP)算法利用虽短描述长度准则（Minimum Description Length Principle, MLDP）作为区间分割点的选择标准与离散化过程的结束条件，并引入边界点的概念对离散化后的区间进行边缘优化。给定当前样本集 T 和连续属性 A，任意候选分割点 s 可将 T 划分为两个样本集 T_1 和 T_2，划分后的信息熵为：

$$\mathrm{Entropy}(A,s;T)=\frac{|T_1|}{|T|}\mathrm{Info}(T_1)+\frac{|T_2|}{|T|}\mathrm{Info}(T_2)$$

公式 2.12

选取划分后的信息熵最小的候选分割点作为属性 A 的的离散化分割点。然后对新生成的样本子集递归的进行类似的划分，直到如下基于最短长度的停止条件得到满足：

$$\mathrm{Info}(T)-\mathrm{Ent}(A,s;T)<\frac{\log_2(|T|-1)}{|T|}+\frac{\Delta(A,s;T)}{|T|}$$

公式 2.13

其中$\Delta(A,s;T)$定义如下：

$$\Delta(A,s;T)=\log_2\left(3^k-2\right)-[k*\mathrm{Info}(T)-k_1*\mathrm{Info}(T_1)-k_2*\mathrm{Info}(T_2)]$$

公式 2.14

k_1, k_2 分别表示在样本集 T_1, T_2 中出现的类别数目。

Entropy(MDLP)是一种监督式的离散化方法。

7、其他

Mantaras 距离评估分割点。选择使划分后的两个样本子集之间的 Mantaras 距离最小的候选分割点作为

最佳分割点。该方法采用 MDLP 作为停止条件。

Zeta 值。定义 k-1 个分割点将某取值区间划分为 k 个子区间的 Zeta 值为：

$$Z = \sum_{i=1}^{k} n_{f(i),i}$$

公式 2.15

其中 f(i)为第 i 个区间内出现频率最高的类别，$n_{f(i),i}$ 为第 i 个区间内类别为 f(i)的样本个数。Zeta 法选择划分后 Zeta 值最大的分割点作为最佳分割点。一般取 k=2，即每次选择一个最佳分割点，并不断划分，直到获得预定数目的区间。

ChiMerge 是一种合并离散化方法，初始时连续属性的每个不同取值均为一个分割点，然后不断将卡方值（χ^2，统计学概念）最小的相邻区间进行合并。一般选择显著性水平在 0.90-0.99 之间，最大区间数为 10-15。

Chi2 法对 ChiMerge 方法中的参数进行动态调整，即在合并的过程中，其显著性水平不断改变，直至达到某个不一致性的条件。两个样本不一致是指他们的属性值相同，但属于不同的类别。Chi2 法的另一个特点是可以剔除那些与类别无关的属性，这些属性由于对分类过程不起作用，故经过 Chi2 法处理后，最终得到一个区间。

RCAT。算法的思想是将待离散化的连续属性转为一个概率属性，该概率属性的二分法结果对应着原连续属性的一个多区间划分。

从决策树的预测精度、结构规模、生成时间、样本的一致性等方面的综合结果来看，MDLP 由于其他离散化方法。但是每种离散化方法都有自己的特点：如果不知道样本的类别，则应当选择非监督的方法；若要去除无用属性，则应选择 Chi2 发；若要在决策树生成过程中进行离散化，则应该选择动态离散化方法等等。需要根据应用场景选用不同的离散化方法。在本论文实验中，使用到了 D2 来对连续属性进行离散化，详细的在第四章实验描述中介绍。

（3） 离散化方法的一种归类模式

根据应用场景，选择离散化方法可按照图 2.4 所示：

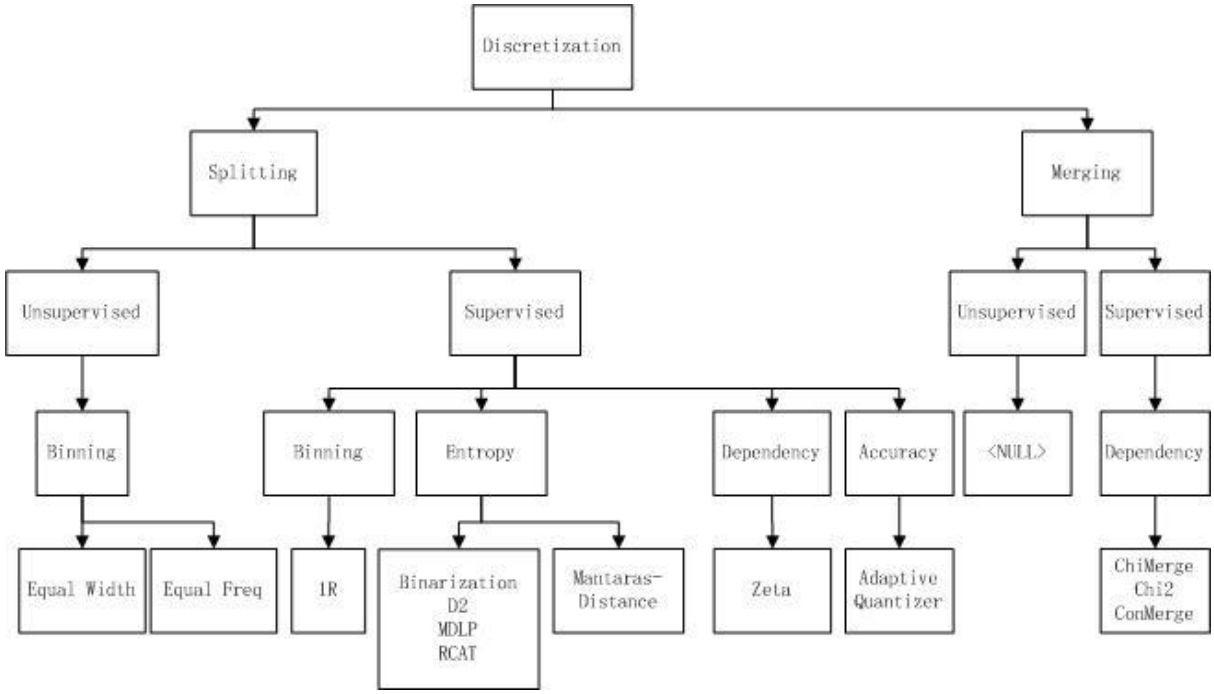


图 2.4 连续属性离散化方法归类

按照“分割/合并”,“监督/非监督”，划分依据“信息熵/属性相关性/精确程度”等来决策使用哪种方式对连续属性进行离散化。

2.4.4 决策树剪枝

在基本构造算法中，决策树的生长过程就是不断地对训练样本集进行划分的过程，每次划分生成一个新的内部结点。由于决策树采用自上而下、分而治之的学习策略，随着迭代深度的增加，算法所考虑的训练样本不断减少，其对总体数据的代表性也就不断降低。当行程决策树的叶结点时，分枝准则所处理的很可能是训练样本集中极少量的数据，该叶结点所呈现的也仅是极少数具有特定特征的数据特点，因而在很大程度上失去了一半代表性，并且对噪声数据十分敏感。这样决策树虽然能够“足够精确地”反映训练样本集中数据的特征，但它并未提取到训练数据中所包含的一般特性，无法对新数据进行合理的分析预测，即产生了过度拟合的现象。

决策树的繁华能力和可理解性和衡量决策树优劣的最为重要的两大属性。如果生成的决策树存在过度拟合问题，一方面决策树的泛化能力差，对新样本进行类别预测时，预测结果常常不准确；另一方面生成的决策树规模庞大，不易理解。

决策树剪枝就是利用某种评价标注你，对决策树进行简化，减小决策树的规模，提高决策树的泛化能力，削弱训练样本集中噪声数据的影响。根据剪枝的实际，可将决策树剪枝技术分为预剪枝和后剪枝两大类。预剪枝法利用某些准则在决策树生成过程中限制决策树的充分生长。后剪枝法在决策树完全生成后再对其进行简化。

（1）预剪枝技术

预剪枝法与决策树的构造过程融合在一起，每次对决策树的某一结点进行划分之前，根据一定的剪枝准则判定是否对当前样本子集进行进一步的划分。如限定深度剪枝法设定了决策树的最大深度，达到深度

的决策树将不再进行分枝；最少样本剪枝法设定了每个结点的最小样本数目，当分枝后某样本子集中的样本数目小于等于最小样本数目，则停止决策树在该结点处的生长。

（2）后剪枝技术

后剪枝算法输入一棵决策树 T，输出一棵剪枝后的决策树 T’，T’是修建了 T 中的一个或多个子树后获得的树。剪枝过程基于一些准则，将一些子树减去，用叶结点代替，叶结点的类别设定为样本集中样本最多的类别。常用的后剪枝算法有如下 5 种：降低错误剪枝法（reduced error pruning）、悲观剪枝法（pessimistic pruning）、基于错误的剪枝法（error-based pruning）、最小错误剪枝法（minimum error pruing）、代价复杂度剪枝法（cost-complexity pruning）。

实验中没有使用剪枝技术来优化决策树的生长，作为进一步的工作展望。剪枝技术在此不做详述。

2.5 决策树的经典算法

决策树方法的起源是概念学习系统 CLS，然后发展到 ID3 方法而为高潮，最后又演化为能处理连续属性的 C4.5。有名的决策树方法还有 J48（暂缺），CART 和 Assistant（暂缺）。

（1）ID3

Example 样本集进行分类，所有可用的测试条件集合为 Attributes，则创建决策树的过程 ID3(Example, Attributes)如下：

创建树的 Root 结点；
如果样本集 Example 都为同一分类，那么返回标记为此分类的单节点 Root；
否则如果 Attributes 为空，则返回单节点树 Root，标记为 Examples 中最普遍的分类类型；
否则开始执行
    A←Attributes 中信息熵增益最大的属性；
    Root 的决策属性 ←A ；
    对于决策属性 A 的每个可能值
        在 Root 下加一个新的分支对应测试条件 $A=v_i$；
        令 Example-v_i 为 Examples 中满足 A 属性值为 v_i 的子集；
        如果 Examples-v_i 为空
            在这个新分支下加一个叶结点，节点标记为 Examples 中最普遍的分类类型；
            否则在这个新分支下加一个子树 ID3(Example-v_i, Attributes-A)；
结束；
返回 Root；

（2）C4.5[11]

由于 ID3 算法在实际应用中存在一些问题，于是 Quinlan 提出了 C4.5 算法，严格上说 C4.5 只能是 ID3

的一个改进算法。在以下方面进行了改动：

1、代替信息熵增益，使用信息熵增益比来选择测试条件，克服了前者偏向选择取值多的属性的不足。在 2.4.2（2）增益比章节有详细介绍；

2、使用了决策树剪枝技术，减小决策树的规模，提高决策树的泛化能力；

3、能够完成对连续属性的离散化处理，C4.5 将连续属性排序后分纳入不同的区间使其离散化，随后当做离散属性按照 ID3 算法进行处理；

4、能够对属性不完整的数据进行处理。

C4.5 的优点有：产生的规则易于理解，准确率高。

缺点为：在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效；同样因为需要进行排序的原因，C4.5 只适合于能够驻留于内存的数据集，这使得能够使用的训练集的规模十分有限。

（3）CART[12]

与 ID3 相比，CART 主要在度量参数方面有所不同。 CART 生成决策树的特点有：使用 Gini 系数作为分枝划分方法，详见 2.4.2（4）Gini 系数分枝划分标准；CART 是一种产生二叉决策树，每次分枝产生两个子结点；每次分裂，使用 $Gini_{split}(T, \varphi)$ 来确定最佳分裂，测试条件φ进行划分的 Gini 系数越小越好。

2.6 本章小结

本章主要介绍了决策树方法的步骤，方法和发展历史。对分枝划分方法和剪枝算法中的几种常用数学模型进行了详细的描述。在最后给出了决策树的几种经典决策树算法的概要性描述，对决策树分类方法的优缺点进行了评价。

第三章 FER 分析与设计

3.1 人脸表情识别系统概述

3.1.1 系统应用背景

人机交互过程中，计算机可以根据表情，肢体动作，语音语调等有效地识别使用者的情绪，并做出智能的反馈。这就需要一套有效地人脸表情识别解决方案。

人脸表情识别系统可以根据图片或截取的视频来识别画面中人脸的表情，反馈识别的结构。

3.1.2 系统需求

用户参与的用例如图 3.1 所示：

1、注册用户账户；

2、根据用户名和密码登录系统；

3、根据和系统的交互来捕捉自己人脸的图像；

4、以各种表情所占的百分比形式来获知识别的结果；

5、以科学的方法给予识别结果评价。

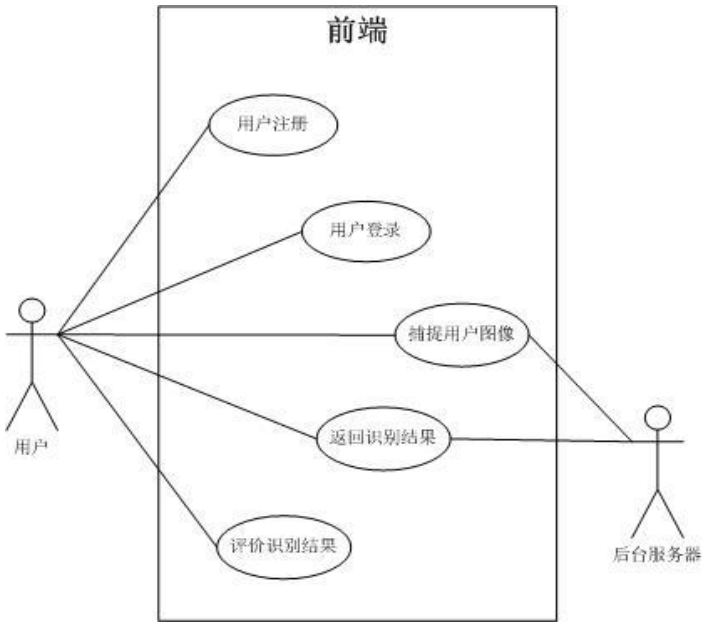


图 3.1 用户用例图

管理员参与的用例如图 3.2 所示：

1、管理员登录；

2、查看系统日志：包括什么时候、给什么用户做了识别，采用了什么特征提取算法，提取的特征是什么，识别算法是什么，识别结果是什么，用户的评价是什么；

3、配置系统的特征提取算法和表情识别算法；

4、管理用户帐户：包括查看，修改和删除账户；给帐户分配权限。

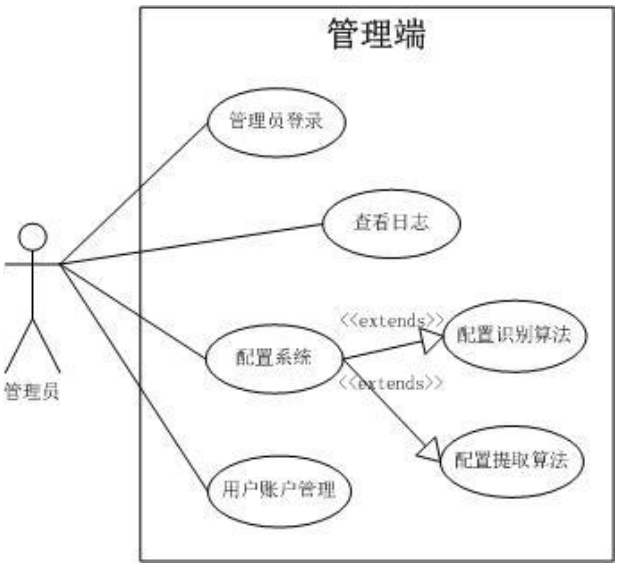


图 3.2 管理员用例图

进一步提出系统应用场景

1、普通用户通过浏览器载入 RIA 程序，输入账户验证信息，进入人脸定位界面，选择是上传图片还是摄像头拍照。如果选择上传图片，则直接在打开的窗口中选中本地图片，否则根据摄像头的反馈调整姿态，点击拍照，捕捉人脸图像。用户可以对选择对拍摄的图片进行一些编辑操作，最后把图像信息提交给 RIA 服务器；

RIA 服务器记录用户的操作日志，同时将用户的请求转发给调度中间件，调度中间件根据系统的配置或者设置的调度算法来分配特征提取算法模块和表情识别算法模块。等调度中间件将识别结果返回给 RIA 服务器后，再返回给前端的浏览器界面，由用户进行评价，之后把评价结果返回给 RIA 服务器的日志管理模块。

2、管理员通过浏览器载入 RIA 程序，输入账户验证信息，进入管理界面，选择进入：

（1）日志管理界面：查看日志

（2）配置界面：配置调度器，添加各种算法

（3）用户账户管理界面：管理账户权限

以上操作都是由浏览器将请求发给 RIA 服务器执行。

3.2 系统设计

本节展示了系统的总体架构及设计，对各个模块的功能和关键技术分别进行了简要的介绍。最后详细介绍了表情分类模块的设计和实现。

3.2.1 总体架构及设计

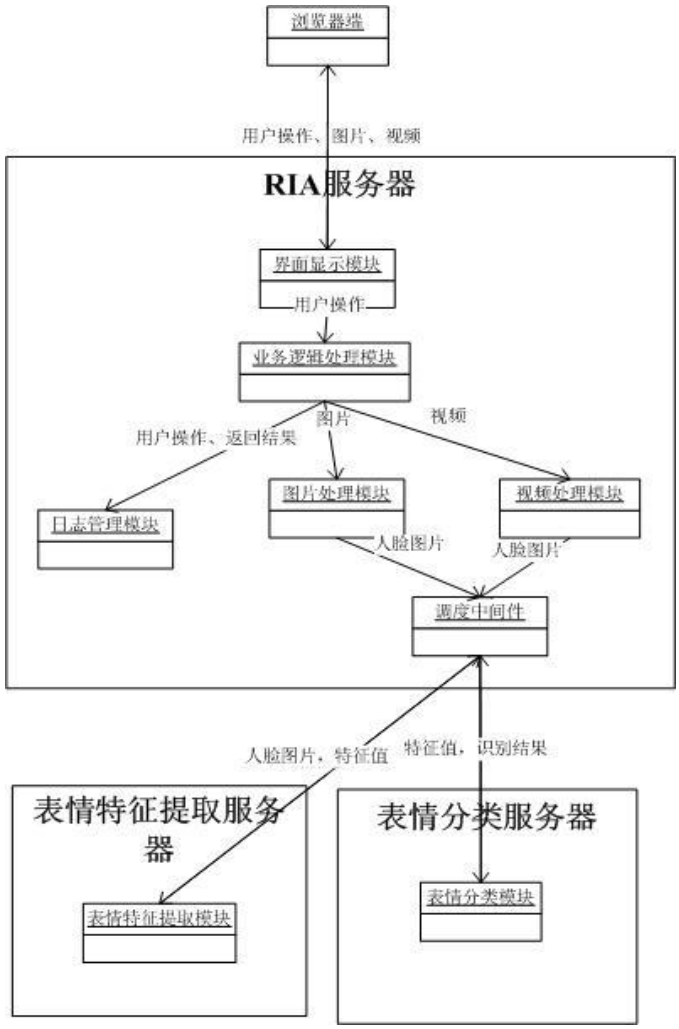


图 3.3 系统模块设计图

系统模块设计图如图 3.3 所示。

部署在 RIA 服务器上的模块有：界面显示模块，业务逻辑处理模块，日志管理模块，图片处理模块，视频处理模块，调度中间件；

部署在表情特征提取服务器上的模块有：表情特征提取模块；

部署在表情分类服务器上的模块有：表情分类模块。

3.2.2 模块说明

人脸表情识别系统主要分成前端的基于 Flex 的 Web 显示层，中间处理 Web 应用的逻辑层，以及后端接收前端传过来的人脸图片并且进行识别的服务器层。

在服务器层，我们又将系统分为三大块：接受前端请求并且协调另外两个模块的 RIA 服务器，接收并处理 RIA 服务器的中间件发过来的提取人脸图片的表情特征的请求的特征提取模块，以及接收并处理 RIA 服务器的中间件发过来的对表情特征进行识别的请求的特征识别模块。

界面显示模块：RIA 实现的 WEB 显示前端，接收用户操作输入，接收后台反馈的操作及识别反馈，将后台反馈的操作及识别结果展示给用户，将用户输入的操作及图像数据传递给业务处理模块。

业务逻辑处理模块：封装了业务逻辑；

日志管理模块：封装日志数据库操作；

图片处理模块：对图片进行预处理，输出一张人脸图片给调度中间件；

视频处理模块：解析视频（流），在接收用户操作输入后提取一帧图像，进行预处理后输出给调度中间件；

调度中间件：接收人脸图像，输出给表情特征提取模块进行表情特征提取，接收特征点信息输入并转发给表情分类模块，接收分类结果；

表情特征提取模块：接收人脸图像，进行表情特征提取，输出特征点信息；

表情分类模块：接收特征点信息，进行表情分类，将分类结果返回。这篇论文主要研究表情分类模块中决策树的使用，以提供一个良好的表情分类方法。

3.2.3 系统关键技术

(1)Adobe Flex

涵盖了支持RIA（Rich Internet Applications）的开发和部署的一系列技术组合。RIA(Rich Internet Applications)丰富互联网应用程序的强交互性与传统Web应用的灵活性结合起来，为用户带来全新的体验。RIA的富客户端采用异步方式和服务端通信，这是一种安全、具备良好适应性的服务器运行模式。[13]

(2)图像预处理

表情特征提取是人脸表情识别中最重要的一步，而这一步输入的人脸图片往往受到许多外界因素的干扰。比如光照，角度，复杂背景。对输入图像进行预处理可以有效地减少这些干扰。根据数字图像处理的知识，通常在识别前进行以下三项预处理工作：几何变换，灰度化，灰度直方图均衡化。

1、几何变换

几何变换可以适当纠正脸部倾斜，侧转的问题。

整个工程在 Flex 前端设置了可供用户自行调整输入表情的功能模块。

在个人的部分实验中，定义了以内眼角和鼻尖为基准，对所有特征点的坐标进行旋转和拉伸的方式来对图像进行几何变换。这样可以统一图像尺寸。

2、灰度化

色彩信息也可以影响人脸表情识别的正确性。

然而在我们还不知道如何利用色彩的时候，通过对图像进行灰度化操作，不仅可以减少光影对人脸特征提取造成的影响，还可以减少特征提取的运算量，并在特征提取步骤使用简单的方式定义纹理。

3、灰度直方图均衡化

直方图均衡化的思想是把原图的直方图变换为均匀的分布方式，这样就增加了像素灰度值的动态范围，从而达到增强图像整体对比度的效果。

增强图像对比度有助于区分边界和纹理。[14]

（3）人脸检测

人脸检测在实际中主要用于人脸表情识别的预处理，即在图像中准确标定出人脸的位置和大小。早期的人脸检测主要使用的方法有模板匹配、子空间方法，变形模板匹配等。近期人脸检测的研究主要集中在基于数据驱动的学习方法，如统计模型方法，神经网络学习方法，统计知识理论和支持向量机方法，基于马尔可夫随机域的方法，以及基于肤色的人脸检测。目前在实际中应用的人脸检测方法多为基于 Adaboost 学习算法的方法。

Adaboost 算法是一种用来分类的方法，它把一些弱分类器合并在一起，组合出新的强分类器。人脸检测的目的就是从图片中找出所有包含人脸的子窗口，将人脸的子窗口与非人脸的子窗口分开。

大致步骤如下：

在一个 20*20 的图片提取一些简单的特征（称为 Haar 特征），如图 3.4 所示：

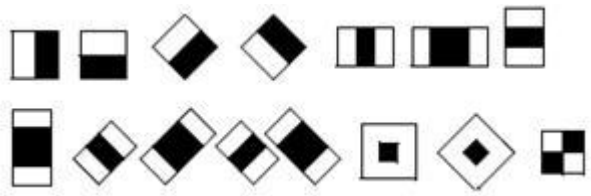


图 3.4 Harr 特征

它的计算方法就是将白色区域内的像素和减去黑色区域，因此在人脸与非人脸图片的相同位置上，值的大小是不一样的，这些特征可以用来区分人脸和分人脸。

目前的方法是使用数千张切割好的人脸图片，和上万张背景图片作为训练样本。训练图片一般归一化到 20*20 的大小。在这样大小的图片中，可供使用的 Haar 特征数在 1 万个左右，然后通过机器学习算法——Adaboost 算法挑选数千个有效的 Haar 特征来组成人脸检测器。

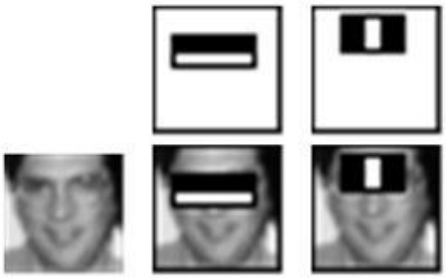


图 3.5 使用 Harr 特征组成人脸检测器

学习算法训练出一个人脸检测器后，便可以在各个场合使用了。使用时，将图像按比例依次缩放，然后在缩放后的图片的 20*20 的子窗口依次判别是人脸还是非人脸。

（4）面部特征标定

面部特征标定即提取人脸形状的过程。按照区分度从粗到细，面部特征标定的方法通常可以分为三个

层次：（1）仿射变换；（2）稀疏的特征对应；（3）基于像素的密集的特征对应。

仿射变换进行人脸配准就是通过少量特征点的对应来标准化人脸图像，通常要求至少 3 个以上，但在假设人脸位置满足一定条件时可以只用两个眼睛作为对应点。目前的很多识别系统都是基于这一层次的对应。它只需要定位少量的几个特征点，并且多数有明确物理意义，具有统一的亮度变化规律，易于定位。但是，这样粗糙的人脸配准对于复杂的人脸分析任务来说有些过于简单。

稀疏的特征对应目前在计算机视觉领域里有较为广泛的应用，ASM（Active Shape Model）和 AAM(Active Appearance Model)是目前解决稀疏特征配准问题的主流方法,它们都是采用形状的点分布模型（Point Distribution Model，简称 PDM）。

ASM 将局部纹理匹配和全局形状子空间约束融合起来，通过局部搜索和全局形状约束的交替迭代，以期收敛到一个最优的结果。AAM 与 ASM 相似，建立了一个融合形状和纹理于一体的外观（appearance）模型。通过优化外观模型的参数实现特征配准,最终目标是期望合成的图像纹理能够最佳的匹配输入图像纹理。

3.3 表情识别模块

表情识别模块的类图如图 3.6 所示：

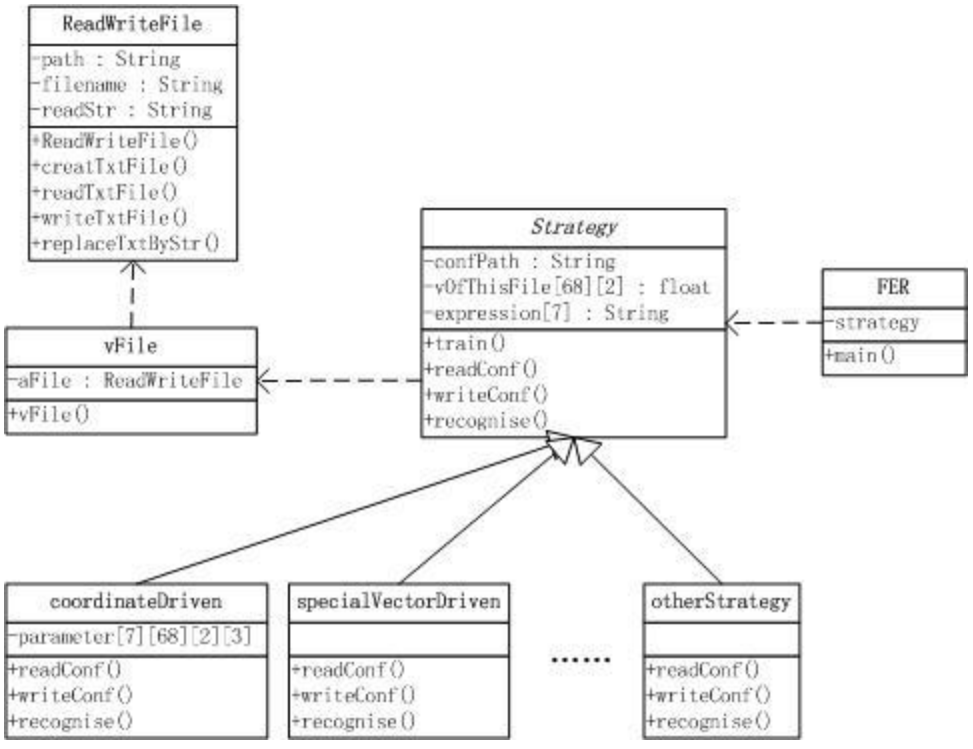


图 3.6 表情识别模块类图

采用工厂模式，以应对不同的训练和识别策略，如不同的测试条件，不同的样本结构等等，保证了以后代码的可扩展性。

3.3.1 类功能说明

ReadWriteFile：读写 txt 文件的类。实验中，用来训练的样本为表情特征提取模块生成的 txt 文件，里面存储了 68 个特征点的坐标信息。

vFile：封装了特征点坐标信息的文件类。读入文件内容后通过构造 Scanner，将文本中的所有特征点信息提取出来存放到数组中。

Strategy：工厂类，提供了可供调用的策略的模板。

coordinateDriven：使用特征点位置作为测试条件的决策树配置文件控制类。

specialVectorDriven：使用特殊向量作为测试条件的决策树配置文件控制类。

FER：包含调用决策树策略，进行分类的主函数。

3.3.2 主要函数说明

ReadWriteFile 类：

readTxtFile()：调用 FileReader()来读取文件，调用 BufferedReader()将文本读入缓冲区，随后调用 readLine()来逐行读取文本。将文本内容添加到一个字符串变量末尾，形成 Java 可以处理的格式。

vFile 类：

vFile()：构造函数，使用调用 readTxtFile()获得的字符串构造一个 Scanner 对象，扫描依次出现的浮点数，并将其值写入坐标点向量数组。

Strategy 类：

train()：规定了训练行为的顺序。倘若没有不完全训练的配置文件，则初始化一个配置文件。随后调用 readConf()读入配置文件。进行训练。然后调用 writeConf()将训练后的参数写入配置文件。使用 XML 作为配置文件存储格式，import 了第三方库 jdom。

readConf()，writeConf()：abstract 方法，因为配置文件结构不同，所以由不同的策略来具体实现。

recognize()：abstract 方法，因为识别策略不同，所以在子类中具体实现。

3.3.3 配置文件结构

配置文件需要方便存储，易于传输，文件结构明了使人容易理解。

因为在实验过程中，往往需要多次代入样本进行训练，因此配置文件需要记录每次实验的结果，用于识别或继续训练。

考虑到文件结构和技术支持，使用 XML 作为参数存储标准。

实验中使用第三方库 JDOM 解析和存储 XML 配置文件。

coordinateDriven 策略的配置文件格式为：

```
−<para>
    -<expression class="{angry|disgust|fear|happy|neutral|sad|surprise}">
        -<dot id="{0-67}" >
            -<x>
                <max>xMax</max>
                <min>xMin</min>
                <value>xValue</value>
              </x>
            -<y>
                <max>yMax</max>
                <min>yMin</min>
                <value>yValue</value>
              </y>
          </dot >
      </expression>
  </para>
```

记录了已训练的样本中，某种表情下，特定编号的特征点的坐标值的出现区间与权值，区间使用出现的数值的最大最小值标记，权值为出现次数的累加。

specialVectorDriven 策略的配置文件格式为：

```
−<para>
    -< expression class="{angry|disgust|fear|happy|neutral|sad|surprise}">
        -<vector description="{ratioOfBrowsEyesAndMouseNoseInner|ratioOfBrowsEyesAndMouse-
NoseOutter|ratioOfPupilAndLipCorner|kLeft|kRight|kUpLeft|kUpRight }">
            <expectation>e</expectation>
            <num>n</num>
            <variance>v</variance>
          </vector>
      < /expression>
  </para>
```

记录了已训练的样本中，某种表情下，某种特征向量的统计量，包括期望，数量，方差。使用期望表示此特征向量的均值与存在范围，使用数量表示用来训练此测试条件的样本的数目，使用方差来表示此向量是否值得作为可信的测试条件。

3.4 本章小结

本章先介绍了人脸表情识别系统的系统背景，并依次逐步分析功能性需求和非功能性需求，提取出需求中影响项目设计的因素并提出了相应的设计策略。

随后本章介绍了整个系统的模块分布，介绍了各个功能模块的主要功能及接口。并给出了在各个模块中应用到的关键技术。

最后对表情分类模块的设计实现进行了介绍，给出了模块内的类图，介绍了各个类及主要函数。

第四章　基于决策树的人脸表情识别

4.1 应用场景概述

4.1.1 输入

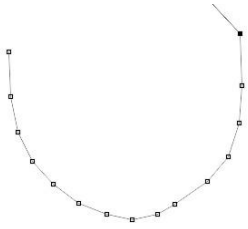
本试验中使用的训练集为日本 JAFFE 标准表情库，从中研究出亚洲人表情的一般特征。

图 4.1 日本 JAFFE 标准表情数据库

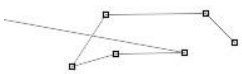
使用的测试集为 JAFFE 数据库及自行拍摄的一组照片。

由特征提取模块处理过后的图像标记出了 68 个特征点及其坐标，通过调度中间件转发给表情分类模块。标记的特征点如下：

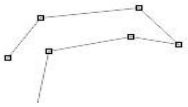
0-14 号特征点：脸廓



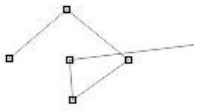
15-20 号特征点：左眉



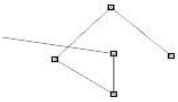
21-26 号特征点：右眉



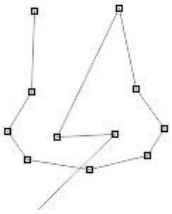
27-31 号特征点：右眼



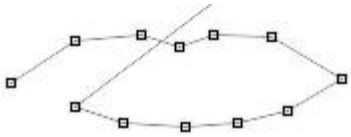
32-36 号特征点：左眼



37-47 号特征点：鼻



48-59 号特征点：唇外廓



60-66 号特征点：唇缝

67 号特征点：鼻尖

61,66,67 号特征点共同构成唇中线。

4.1.2 输出

表情分类模块的输出为七种表情分类之一：愤怒（angry），厌恶（disgust），恐惧（fear），开心（happy），中性（neutral），悲伤（sad），惊讶（surprise）。

4.1.3 分类策略选择

考虑到决策树具有以下几个特点：

（1）决策树易于理解和实现。人们在通过解释后都有能力去理解决策树测试条件所表达的意义。

（2）决策树具有处理离散属性和连续属性的能力。

（4）使用白盒模型。如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。

（5）易于通过静态测试来对模型进行评测。这意味着可以测量该模型的可信度。

（6）在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

因此可以运用决策树分类方法来构建一个具有一定可信度的分类模型。

4.2 测试条件选择

4.2.1 实验一——特征点坐标累积权值

思路：

人类对于表情的认知和脸部器官的位置、形状有关，首先的实验是使用坐标信息信息匹配，计算权值大小来进行分来。

训练过程：

（1）将各个特征点的坐标按照内眼角及鼻尖距离进行旋转和缩放操作，定义右眼内眼角为（-18, 0），定义左眼内眼角为（18，0），保证之后使用的坐标处在同一参考系中。

（2）从 0-67，统计每个特征点坐标的 x 分量和 y 分量，记录他们的上下界，并记录参与训练样本的数量作为权值。将参数记录在训练过的 XML 配置文件中。

测试过程：

（1）仍然使用 JAFFE 表情库进行测试。

（2）对每个测试样本，比对测试文件中的表情 E 的参数。如果 n 号特征点的坐标，处于该表情中这一编号的特征点的上下界中，则给这个样本的表情 E 累加上相应的权值 $value_n$。

（3）68 个点匹配过后，此样本将获得表情 E 的权值。比较七种表情的权值大小，判断此样本的表情分类。

实验数据：

Angry: 11 angry / 18 fear / 1 happy, accuracy: 36.67%

Disgust: 13 disgust / 1 angry / 12 fear / 1 surprise / 2 sad, accuracy: 44.83%

Fear: 32 fear, accuracy: 100%

Happy: 17 happy / 14 fear, accuracy: 54.84%

Neutral: 5 neutral / 3 sad / 20 fear / 2 happy, accuracy: 16.67%

Sad: 17 sad / 13 fear / 1 happy, accuracy: 54.84%

Surprise: 18 surprise / 12 fear, accuracy: 60.00%

实验结果：

所有测试样本最高的权值所占半分比均在 15%-20%左右，没有进行明显的区分，并且以此作为判断依据的精确程度很差。在愤怒，厌恶，中性表情上有明显的误判。

反思：

从实验数据中可以看出，愤怒，厌恶、悲伤、惊讶表情均有很大一部分被误判成为恐惧，观察表情后发现，恐惧表情局部特征变化跨度很大，不能使用局部特征来作为决策条件，必须使用组合条件判断。

4.2.2 实验二——特征点坐标范围匹配

思路：

在人的认知过程中，清晰定义了一个表情为“开心”的话，是不会因为后面对“愤怒”进行更多认知而把“开心”误判成为“愤怒”的。用于训练的样本必须被正确的分类成为这种表情。

训练过程：

使用与 4.2.1 同样的训练方式与同样的配置文件。

测试过程：

（1）对每个测试样本，比对测试文件中的表情 E 的参数。如果 n 号特征点的坐标，处于该表情中这一编号的特征点的上下界中，则给这个样本的表情 E 累加 1 的匹配度。

（2）68 个点匹配过后，此样本表情 E 的匹配度为 68 的话，则分类为这种表情。可能对一个样本存在一种以上的分类。

实验数据：

Angry: 30 angry / 2 disgust / 1 fear

Disgust: 29 disgust / 1 angry / 2 fear / 1 sad, accuracy

Fear: 32 fear / 3 sad

Happy: 31 happy / 3 angry / 5 disgust / 5 fear / 3 sad / 4 surprise

Neutral: 30 neutral / 2 angry / 1 disgust / 8 fear / 8 sad / 10 surprise

Sad: 31 sad / 1 angry / 4 disgust / 5 fear / 4 happy / 2 surprise

Surprise: 30 surprise / 3 fear

实验结果：

虽然实验结果看上去较为理想。但是因为训练和测试使用了同一个样本集，当使用自行拍摄的照片进行测试的时候，分类结果很不理想，因此使用出现范围进行匹配的方法不具有预测能力。

反思：

高兴、中性、悲伤三种表情出现了很多其他的表情判断。向左撇嘴的厌恶表情和向右撇嘴的厌恶表情就有可能融合出一个可以识别笑脸的分类器。所以坐标出现范围并不能作为测试条件。

4.2.3 实验三——嘴角间距与瞳孔间距比寻找笑脸

思路：

脸部表情特征会随着拍摄角度，脸部倾斜旋转等因素而变化，使用特征点坐标直接对进行分类会产生很多的错误。需要寻找一些在脸部几何变形过程中不会产生变化的向量来标识表情。

素描中的人脸构建存在一个“五三原则”：人脸从上到下可以三等分，三段分别为发际线到眉尖、眉尖到鼻子下端、鼻子下端到下巴；人脸从左到右可以进行五等分：两眼眼角水平到脸部轮廓、两只眼睛、鼻翼。在特定表情中，五三原则中的某些单位就会变形，相互之间的比例关系就会变化，可以以此来区分表情。

测试过程：

（1）31、36 号点为两瞳孔，48、54 号点为两嘴角。读取样本四个特征点的坐标值，计算 31 号点到 36 号点的距离作为瞳孔间距，计算 48 号点到 54 号点距离作为嘴角间距。

（2）瞳孔间距除以嘴角间距，记录比值。

（3）对瞳孔间距与嘴角间距的比值进行统计，记录样本数量，期望值和方差。

实验数据：

Angry: maxRatio: 1.3614607 / minRatio: 1.1236231

Disgust: maxRatio: 1.2739617 / minRatio: 1.1126832

Fear: maxRatio: 1.2990239 / minRatio: 1.1218128

Happy: maxRatio: 1.1858386/1.1549578（去除样本 UY.HA1.137.bmp.txt，UY.HA2.138.bmp.txt 后的结果）/ minRatio: 0.9833541


图4.2 JAFFE表情库中的UY.HA1.137.bmp和UY.HA2.138.bmp

Neutral: maxRatio: 1.30347 / minRatio: 1.0868077

Sad: maxRatio: 1.3978069 / minRatio: 1.0114455/1.1241（去除样本 kr.sa3.79.bmp.txt 后的结果）

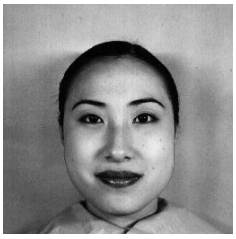


图4.3 JAFFE表情库中的KR.SA3.79.bmp

Surprise: maxRatio: 1.3989136 / minRatio: 0.9879441/1.1272032（去除样本NA.SU1.208.bmp.txt 和YM.SU2.59.bmp.txt后的取值）

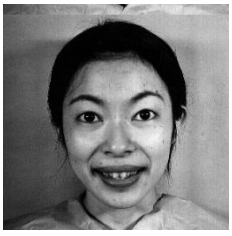
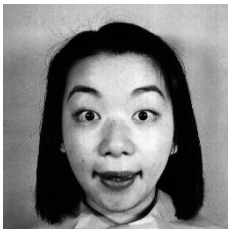



图4.4 JAFFE表情库中的NA.SU1.208.bmp和YM.SU2.59.bmp

实验结果：

瞳孔嘴角间距比能够明显的区分出 happy 表情，使得这个向量可以作为测试条件。

反思：

因为眼睛大小存在差异（这里通常表现为眼眶大小），使用瞳孔间距就可以减少这种差异带来的误差。在笑脸中，嘴角向两旁咧开，而眼睛间距是始终不变的，因此这个比值也有其实际的意义。另外，观察实验数据后发现，sad 和 surprise 的 ratioOfPupilAndLipCorner 值分布偏高，虽然与其他表情没有明确的分界点，但是给出了一个定性的参考。

4.2.4 实验四——眉毛上扬表现出的惊讶表情

思路：

观察 surprise 表情，明显特征是眉毛上扬，表现出惊讶的感觉。因此考虑到可以使用眉毛到眼角的距离作为判定。同样为了避免人脸器官差异性，给定一个比值，除以鼻尖到上齿的距离。

测试过程：

（1）17,23 号点为左右眉毛上端，29,34 为左右内眼角。67 号点为鼻尖，64 号点为上齿。17,34 号点为左眼左眉间距，23,29 号点为右眼右眉间距。为了减少误差，这儿使用两个距离。

（2）两个眉眼距离相加除以鼻齿距离，记录比值。

（3）对眉眼间距与鼻齿间距的比值进行统计，记录样本数量，期望值和方差。

实验数据：

Angry: maxRatioInner: 2.2383296 / minRatioInner: 1.1217914

Disgust: maxRatioInner: 2.2559814 / minRatioInner: 1.2878433

Fear: maxRatioInner: 2.2529743 / minRatioInner: 0.87313974

Happy: maxRatioInner: 2.5731797 / minRatioInner: 1.4116415

Neutral: maxRatioInner: 2.208942 / minRatioInner: 1.5571685

Sad: maxRatioInner: 2.2118814 / minRatioInner: 1.5464413

Surprise: maxRatioInner: 2.2166398 / minRatioInner: 0.85429484

实验结果：

很明显 fear 和 surprise 两个表情的这项数值较多的分布在低端范围内。可以用于分类。

反思：

在 1.5 以上的数值范围内依然有很多 fear 和 surprise 的表情样本。事实上，有些 fear 和 surprise 表情不用上扬眉毛，甚至下压眉毛也是可以实现的，尽管大多数人习惯性的动作是上扬眉毛。另外，尽管之前的四次实验里，fear 和 surprise 都是同时出现，在这些向量上具有类似的取值，但是很明显的，两个表情间存在很大的差异性，到底这个差异是什么还有待寻找。

4.2.5 实验五——上嘴唇平滑区分惊讶和恐惧

思路：

仔细观察了 JAFFE 标准表情库里德 fear 和 surprise 表情，明显的差异就在嘴部。具体表现为两者口型不同，尽管下唇线均会呈现很大的曲线和斜率，但是 surprise 表情会将嘴巴张大而使得上唇线也具有很大

的斜率。fear 表情则不是如此，大多数情况下，人物的上唇线是较为平滑的。

测试过程：

（1）48 号为右嘴角，54 号为左嘴角，51 号为上嘴唇顶点。计算 48,54 号点斜率与 51,54 号点斜率，斜率取值为正。

（2）调整，计算两个倾斜角平均的正切值，这样削弱图像倾斜造成的影响，记录比值。

（3）对上唇线斜率进行统计，记录样本数量，期望值和方差。

实验数据：

Angry: maxk: 0.6378685 / mink: 0.15124482

Disgust: maxk: 0.8282093 / mink: 0.31630206

Fear: maxk: 1.2269626 / mink: 0.32215005

Happy: maxk: 0.51408714 / mink: 0.21552324

Neutral: maxk: 0.68250126 / mink: 0.19145754

Sad: maxk: 1.0079103 / mink: 0.23353505

Surprise: maxk: 1.5606438 / mink: 0.37865376

实验结果：

surprise 表情明显分布在较高的数值上。这个向量可以用于分类。

反思：

使用几个简单的规则可以对部分表情进行区分，但是人类表情很多都是动态的，模糊的。还有更多的分类规则需要寻找。

4.3 创建决策树

4.3.1 连续属性的离散化

使用分割后两个子集中 happy 样本与非 happy 样本的数目来计算分割信息熵。

（1）瞳孔间距与嘴角间距比

分割点：1.10　分割信息熵：0.4763287547676137

分割点：1.11　分割信息熵：0.41245040696592533

分割点：1.16　分割信息熵：1.305344680495739

所以使用 1.11 作为瞳孔间距比的分割点。

（2）眉眼间距与鼻齿间距比

分割点：1.7　分割信息熵：3.475515645935861

分割点：1.9　分割信息熵：1.2953867272836632

分割点：2.1　分割信息熵：0.5808421623569474

所以使用 2.1 作为眉眼间距与鼻齿间距比的分割点。

（3）上唇线斜率

分割点：0.31　分割信息熵：1.347373847219949

分割点：0.37　分割信息熵：1.6104160815847037

分割点：0.44　分割信息熵：1.500562619926335

分割点：0.51　分割信息熵：1.2055330799405026

所以使用 0.51 作为上唇线斜率的分割点。

4.3.2 使用信息熵增益的分枝划分方式构建决策树

（1）首先计算 happy 表情所包含的信息量：

$$\mathrm{Info(T)} = \mathrm{I(29+,184-)} = -\left[\frac{29}{213} * \log_2 \frac{29}{213} + \frac{184}{213} * \log_2 \frac{184}{213}\right] = 3.087876289267092\ \mathrm{bit}$$

分割信息熵在 4.3.1 中计算过，记瞳孔间距与嘴角间距比（ratioOfPupilAndLipCorner）为测试条件φ_1，眉眼间距与鼻齿间距比(ratioOfBrowsEyesAndMouseNoseInner)为测试条件φ_2，上唇线斜率(k)为φ_3。则三条测试信息所获得的信息熵增益分别为：

由此可以看出所有条件当中瞳孔间距与嘴角间距比是最能区别 happy 与否的条件。

$$\mathrm{Gain}(\varphi_1) = \mathrm{Info(T)} - \mathrm{Info}_{\varphi_1}(\mathrm{T}) = 3.087876289267092 - 0.41245040696592533 = 2.675425882301167\ \mathrm{bit}$$

$$\mathrm{Gain}(\varphi_2) = \mathrm{Info(T)} - \mathrm{Info}_{\varphi_2}(\mathrm{T}) = 3.087876289267092 - 0.5808421623569474 = 2.5070341269101446\ \mathrm{bit}$$

$$\mathrm{Gain}(\varphi_3) = \mathrm{Info(T)} - \mathrm{Info}_{\varphi_3}(\mathrm{T}) = 3.087876289267092 - 1.2055330799405026 = 1.8823432093265895\ \mathrm{bit}$$

（2）接下来，创建一个树结点，并创建该结点的子链，每个子链代表所选属性的一个唯一值。使用子链的值进一步细化子类。当出现以下两种情形之一时可以停止分类：1、一个结点上的数据都是属于同一类别；2、没有属性可以再对样本集进行分割。

根据各个条件的信息熵增益度，应该选择φ_1作为所建决策树的根结点。由于其属性值离散成为小于 1.11 和大于等于 1.11，所以在φ_1下可以建立两个分枝。

经统计，φ_1<1.11 且为 happy 表情的样本数量为 21，其准确率为 21/25=84%，所以对φ_1<1.11 这个分枝停止分割。又经统计φ_1≥1.11 的 188 的样本中有 10 个 happy 样本，178 个不是其他表情样本。所以应该对φ_1≥1.11 这个分枝进行分割。按照分割信息熵增益度，应该选取φ_2眉眼间距与鼻齿间距比进行细化。

分割后经统计显示：φ_1≥1.11 且φ_2≥2.1 的 10 个样本中，有 8 个样本为 happy 表情分类，1 个是 sad 分类 1 个 surprise 分类，准确率为 8/10=80%；φ_1≥1.11 且φ_2＜2.1 的样本中，有 2 个 happy 分类，176 个非 happy 分类，准确率为 176/178=98.88%。由此可以构建出决策树，如图 4.5 所示：

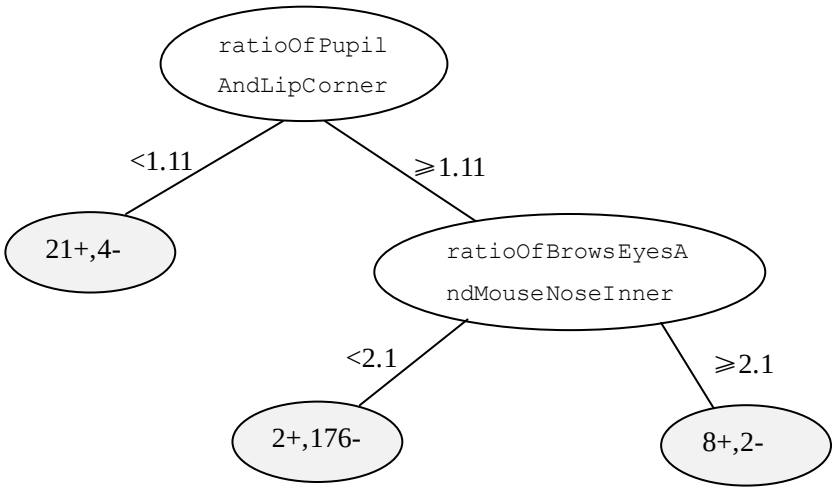


图 4.5 构建的决策树

（3）从决策树中得出如下规则：

IF ratioOfPupilAndLipCorner<1.11

THEN expression=happy

Accuracy=21/25=84%

CoverageRate=25/213=11.74%

IF ratioOfPupilAndLipCorner⩾1.11 AND ratioOfBrowsEyesAndMouseNoseInner＜2.1

THEN expression!=happy

Accuracy=176/178=98.88%

CoverageRate=178/213=83.57%

IF ratioOfPupilAndLipCorner⩾1.11 AND ratioOfBrowsEyesAndMouseNoseInner⩾2.1

THEN expression=happy

Accuracy=8/10=80%

CoverageRate=10/213=4.69%

（4）使用自行拍摄的照片进行测试：

angry 表情集：23 个对象，判定 isHappy，2 个判定为 true，21 个判定为 false，正确率 91.30%

disgust 表情集：18 个对象，判定 isHappy，1 个判定为 true，17 个判定为 false，正确率 94.44%

fear 表情集：暂缺

happy 表情集：44 个对象，判定 isHappy，37 个判定为 true，7 个判定为 false，正确率 84.09%

neutral 表情集：58 个对象，判定 isHappy，1 个判定为 true，57 个判定为 false，正确率 98.28%

sad 表情集：50 个对象，判定 isHappy，13 个判定为 true，37 个判定为 false，正确率 74.00%

surprise 表情集：12 个对象，判定 isHappy，1 个判定为 true，11 个判定为 false，正确率 91.67%

分类效果达到了理想的百分比。

4.4 本章小结

本章介绍了表情分类模块的应用场景，并分析了选用决策树作为分类方法的原因。通过实验选择了几个属性作为具有分类能力的测试条件，根据分割信息熵对这些连续属性进行了离散化。后根据信息熵增益比的分枝划分方式构建了决策树，并给出了每个分枝的覆盖度和正确率。最后使用自行拍摄的照片来测试决策树的分类效果。

第五章 总结与展望

5.1 论文工作总结

本文总结了决策树算法的概念和发展历史，描述了决策树的算法流程和通常需要考虑的算法特征，并详细叙述了几种经典决策树算法。

本文主要介绍了使用决策树进行表情分类的方法：

（1）实验选取了可作为测试条件的向量。使用分隔信息熵的准则对这些连续型的变量进行了离散化。

（2）使用信息熵增益来衡量各个测试条件的分类效益，从大到小进行分类。

（3）给出识别笑脸的决策树分类器，使用自行拍摄的图片作为测试集输入，观察实验结果。

5.2 进一步的工作展望

（1）自适应学习

在使用特殊向量作为测试条件的过程中，人为标定了几个分割点，来对向量的取值进行离散化。这样在新样本进入训练集后，分割点的带来的划分信息熵（公式 2.12）可能并不是最优取值。

需要在代码中加入自适应的调整分割点的部分，使得每次训练之后都可以找到最佳的连续属性离散化分割点。

（2）后剪枝

由于实验过程使用的样本数量较小，测试条件较为简单，人们容易理解，所以决策树本身规模较小。

考虑到后期可能发掘出更多更有分类价值的测试条件，对决策树进行更深层次的构造；以及更多泛化的样本加入训练集，可能带来决策树旁枝的错乱生长。因此在决策树学习规模扩大的同时，必须加入剪枝技术来限制决策树的过度生长，防止出现过度拟合现象，保证决策树的泛化能力。

（3）决策树规则化

剪枝过后的决策树会比原先简洁，但是仍然可能显得笨重、复杂、难以理解。要在保证对未知样本的预测精度的同时，更为便捷地理解决策树所表示的知识，需要进行决策树规则化。

规模较大的决策树所表示的知识很难理解，这是因为决策树的每个结点都有自己所处的上下文，即路径中所有测试条件的合取，这个上下文是存在语义可以理解的。但是决策树规模很大的时候，跟踪上下文的连续变化是一件很困难的事情，从而导致大规模的决策树难以理解。此外，决策树的结构可能导致一些子概念被分割的支离破碎，增加了理解决策树的难度。明显的表现就是在不同分支上出现了两颗同样的子树用于判断同样的测试条件取值。

决策树规则化就是利用规则表示法重新描述决策树中的知识，使得决策树易于理解。

参考文献

[1]ANIMETRICS, Inc. Biometrics and Facial Recognition[EB/OL].
http://www.animetrics.com/technology/frapplications.html, 2010-05-27.

[2]Dr John van Wyhe. The Complete Work of Charles Darwin Online[M/OL].
http://darwin-online.org.uk/, 2002-10

[3]Yang M, Kriegman D J, Ahuja N. Detecting faces in images: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002, 24(1): 34-58.

[4]Zhang Y, Ji Q．Active and dynamic information fusion for facial expression understanding from image sequences [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5)：699-714.

[5]Cohen I, Sebe N, Garg A, et al. Facial expression recognition from video sequences: Temporal and static modeling[J]. Computer Vision and Image Understanding, 2003, 91(1-2)：160-187.

[6]WIKIPEDIA. Support vector machine[DB/OL].
http://en.wikipedia.org/wiki/Support_vector_machine, 2010-05-27.

[7]leivo. 人脸表情识别综述[EB/OL].
http://www.cnblogs.com/leivo/archive/2008/08/07/1263176.html, 2008-08-07.

[8]Nirvana. 数据挖掘中分类算法小结[EB/OL].
http://www.chinabi.net/blog/user1/105/archives/2005/332.html, 2005-11-8

[9]WIKIPEDIA. Decision tree[DB/OL].
http://en.wikipedia.org/wiki/Decision_tree, 2010-05-21.

[10]Earl B Hunt. Experiments in Induction[M].

[11]aladdina. 数据挖掘十大经典算法（1）C4.5[EB/OL].
http://blog.csdn.net/aladdina/archive/2009/04/30/4141048.aspx, 2009-04-30.

[12]Valder. CART 算法的简单实现（1）[EB/OL].
http://www.cnblogs.com/valder/articles/1707609.html, 2010-04-10

[13]董龙飞，肖娜. Adobe Flex 大师之路[M]. 北京：电子工业出版社, 2009. 3-15.

[14]GIS 门户网. 直方图及其匹配和均衡化的种种[EB/OL].
http://www.ggiiss.com/gis/c2/1052.html, 2008-09-23.

致谢

感谢刘峰老师。在我面对决策树和神经网络难以作出抉择的时候，他给予了我理论上的帮助，并指导我如何一步一步展开对课题的思考。在撰写论文初稿期间，他仔细审阅，对论文的组织结构及内容取舍提出了很多宝贵的建议。

感谢王崇骏老师。在 IIP 研讨班上，了解很多人工智能的有趣话题，激发了我在实验中的一些思路和创意。

感谢顾士元和管铭驰。4 月底我刚完成代码中重要的一部分，做了几个重要的实验，这时候硬盘分区表损坏了，重要的文档，代码，数据都只有硬盘中一个拷贝。在一些维修点尚且没法修复的时候，他们应用专业知识和探索精神，利用 WinHEX，VI 等工具成功的帮我恢复了硬盘数据。

感谢父母。证实笔记本硬盘的损坏是因为南桥故障，维修需要太长的时间成本，影响工程的进度。在知道这点后，我想迅速购置一台新的笔记本，这比预算提前了一年。他们慷慨的提供了支持。

感谢陈芳源。在论文的格式方面她给予了专业性的指导,简化了论文的审阅和校对工作。

感谢那些协助拍摄表情图片的同学们。因为他们才使得这个表情分类算法有了实用性的依据。