北京信息科技大学

毕业设计（论文）

题　　目：基于情感机器人的人脸表情识别算法研究

学　　院：计算机学院

专　　业：计算机科学与技术

学生姓名：康路 **班级/学号** 计科 1602/2016011230

指导老师/督导老师：乔文豹

起止时间：2020 年 1 月 6 日 **至** 2020 年 6 月 14 日

摘　要

本课题针对人脸表情识别问题，对表情识别的通用基本步骤和卷积神经网络原理做出阐述和说明，在基于卷积神经网络的 mini_Xception 表情识别算法的基础上，通过添加局部连接层等操作改进算法，并通过 Keras 深度学习库进行模型训练并进行评估。实验结果表明，改进的算法对比 mini_Xception 算法在测试数据集上的准确率从 66.8%提升到了 69.0%，可轻易的识别出高兴和中立两种表情，也可识别出夸张表现的生气，恐惧和惊讶三种表情，整体识别效果要优于 mini_Xception 算法。最终，本课题通过 Python 编程使 NAO 情感机器人可以利用改进的表情识别算法进行人脸表情识别，并做出相应语言和动作上的反应。

关键词： 人脸表情识别 ； 深度学习 ； 情感机器人 ； 卷积神经网络 ；

Abstract

This topic is aimed at the problem of facial expression recognition. In this paper, the general basic steps of expression recognition and the principle of convolutional neural network are described and explained. On the basis of the mini_Xception expression recognition algorithm based on convolutional neural network, the algorithm is improved by adding local connection layers and other operations, and the Keras deep learning library is used for model training and evaluating. Experimental results show that the accuracy of the improved algorithm compared to the mini_Xception algorithm on the test data set has been increased from 66.8% to 69.0%, which can easily identify two expressions of happiness and neutrality, and can also identify exaggerated expressions of anger, fear and surprise. The overall recognition effect is better than the mini_Xception algorithm. In the end, this subject uses Python programming to enable NAO emotional robots to use improved algorithms for facial expression recognition, and to act corresponding language and action responses accordingly.

Keywords: facial expression recognition; deep learning; emotional robot; convolutional neural network;

目　　录

第一章　绪论

本章首先介绍人脸表情识别算法研究的研究背景和意义，其次分别简述国外和国内表情识别研究的发展历程，然后概述本课题的研究内容，最后陈述本文的结构安排。

1.1　研究背景和意义

随着人工智能技术的不断发展，它的研究内容也在逐步扩展和延伸，越来越多人类拥有的能力得以在计算机上实现。其中对人的情感认知的研究作为人工智能的较高级阶段，逐渐走进了人们的视野之中，情感计算随之成为一个新兴研究领域。情感计算是指与情感相关，来源于情感或能够对情感施加影响的计算。其目的是通过赋予计算机类似于人一样的识别、理解、表达和适应情感的能力，来建立更加和谐便捷的人机环境，最终使计算机能和人进行自然、亲切和生动的交互，从而使计算机具有更高、更全面的智能[1]。

在情感计算的应用领域中，人脸的表情识别较为基础和易于实现，已成为热门的研究话题，并取得了一定成绩。人脸表情识别，即计算机从获取的静态图像或动态视频中分离出人脸表情状态并识别，从而确定被识别对象的心理情绪。在当前表情识别研究中，可识别的表情基本分为高兴、悲伤、生气、惊讶、恐惧、厌恶和中立七种表情[2]。这些表情较为常见和明显，一定程度上能够反映出真实的心理情绪，从而使计算机理解人类的情感。

人脸表情识别可以广泛地应用在人机交互、安全、智能机器人、医疗、通信和汽车驾驶等各个领域，具有很高的研究价值和发展潜力。其中人机交互现阶段主要是通过鼠标和键盘点击图形界面或输入命令语言来进行的，虽然可以满足基本需求和大部分应用场景，但在一些不方便使用手来操作或需要快速、远程操作等特殊场景下，传统人机交互就显示出不足的地方。而利用表情、语音、手势等方式来进行交互，则能解决以上提到的问题，达到快速、便捷的人机交互。又如在医疗领域，医务人员由于人数和精力限制，无法随时观察病人的情况。利用人脸表情识别系统，可随时获取病人脸部特征，从而反映病人的疼痛情况，及时发现紧急情况。再如汽车驾驶领域，车载表情识别系统可实时观察驾驶员的表情变化，及时发现疲劳、酒后驾驶等危险情况并给出警告，从而避免这类交通事故的发生。

除了上述列举的一些例子，情感机器人在未来也有着巨大的发展需求。随着科技的发展和人口老龄化的转变，情感机器人无论在照顾老人还是教育儿童，甚至是工作和生活的各个方面，都有着重要的作用和价值。而表情识别作为情感机器人所必须具备的能力，同样需要深入研究和探索，最终实现接近甚至匹敌人类的表情识别能力。综上所述，人脸表情识别的研究具有广阔的应用前景和深远的影响。

1.2　国内外研究发展现状

人脸表情识别技术是近三十年才逐渐发展起来的，并取得了一些成果。国内外有许多研究机构及学者致力于表情识别的研究，下面简单介绍国内外表情识别技术的发展及现状。

1.2.1 国外研究发展现状

上世纪 90 年代，随着计算机模式识别和图像处理技术的发展，使得人脸表情识别的实现逐渐成为可能。Mase[3]作为其中的先驱者，提出了一种使用光流来估计面部肌肉动作的方法，从而构建了基于光流数据的面部表情识别系统。该系统可以识别快乐、愤怒、厌恶和惊讶 4 种表情，识别率接近 80%。2007 年，Kotsia 等人[4]利用网格跟踪和变形系统提取人脸网格的最大几何变形，并选择多类支持向量机（SVM，Support Vector Machine）进行分类，取得了显著的识别性能，在 CK(Cohn-Kanade)数据库上的识别率为 99.7%。Almaev 等人[5]根据先前的经验改进了局部 Gabor 二值模型算法，在 MMI 面部表情数据库和 CK 数据库上取得了不错的识别效果。

2012 年后，随着深度学习和卷积神经网络（CNN，Convolutional Neural Networks）的兴起，深度神经网络逐渐应用于表情识别领域。Tran 等人[6]提出了一种简单有效的时空特征学习方法，即在大规模监督视频数据集上使用深度 3 维卷积网络（3D ConvNets），它们在概念上非常简单，且易于训练和使用。Lopes 等人[7]在 2016 年提出了一个简单的面部表情识别解决方案，它结合了卷积神经网络和特定的图像预处理步骤。该方案与其他面部表情识别方法相比具有竞争优势（CK+数据库中准确率达到 96.76％），训练速度快，并且可以使用标准计算机进行实时面部表情识别。Jeong 等人[8]在今年提出了基于深度外观和几何神经网络的有效深度联合时空特征，用于面部表情识别。该方案在 CK+，MMI 和 FERA 数据集的识别准确率分别为 99.21％，87.88％和 91.83％，通过比较分析表明，至少能够将识别精度提高 4％。

关于国外的表情识别研究还有很多，涉及表情识别的特征提取及分类等种种方法。特别是近几年，表情识别的发展迅猛，新方法多样，这里仅简单列举了几项。

1.2.2 国内研究发展现状

国内的表情识别研究相对于国外较晚，但近几年同样发展迅猛，越来越多的研究机构和学者投入到表情识别的研究，论文层出不穷，有后来居上的趋势。早在 1997 年，哈尔滨工业大学的高文教授和金辉博士[9]就对表情识别进行了研究，通过利用模板匹配方法提取目标特征，得到人脸表情的表征向量，由模式分类方法实现表情的识别。2004 年，东南大学郑文明博士[10]将基于核函数的机器学习方法应用于人脸的表情识别实验中，并且在 JAFFE 国际人脸表情数据库中取得了良好的实验效果。2007 年，鹿麟，吴伟国等人[11]设计并研制了具有面部表情和对人类表情识别性能的仿人头像机器人系统。

近些年，东南大学的唐传高等人[12]在 FG2017 表情识别与分析竞赛荣获冠军。北京大学陈颖婕等人[13]比较了使用传统机器学习模型或深度学习模型的不同面部表情识别方法，并为情感智能机器人提出了一种快速，准确的多模型面部表情识别方法，以完成实时和高精度的面部表情识别任务。张桐等人[14]提出了一种新颖的深度学习框架，称为时空递归神经网络（STRNN，spatial-temporal recurrent neural network），从而将对两个不同信号源的学习统一为时空依赖模型。所提出的 STRNN 方法在脑电图和面部表情的公共情感数据集上比其他最新方法更具竞争力。

1.3 本文研究内容概述

本课题对基于深度学习的人脸表情识别算法做出研究和改进，并对改进的表情识别算法进行模型训练及性能评估，最终在 NAO 情感机器人上实现。研究内容具体体现在两个方面：

1. 基于深度学习的人脸表情识别算法的调研和改进。通过研究人脸表情识别所需的基本步骤及其实现方法，确定使用基于卷积神经网络的算法来实现人脸表情识别。接着对 CNN 的基本原理及其发展进行研究和学习，确定 mini_Xception 算法可较好完成表情识别任务。并在此算法的基础上做出改进，使表情识别率在数据集上有所提升，并取得较好的实际识别效果。

2. 改进的人脸表情识别算法在智能机器人上实施和实现。本次研究通过 NAO 机器人官方网站了解 NAO 机器人的基本信息和功能，并学习 NAO 机器人编程，最终实现 NAO 机器人应用本课题改进的算法进行人脸表情识别。

1.4 本文结构安排

本论文一共分为五章，各章内容安排如下：

第一章：绪论。绪论作为论文的第一章，首先介绍本论文的研究背景和意义。接着简述了国内外关于表情识别研究的发展历程和最新进展，分国内外两小节分别陈述。绪论的第 3 节概述了本论文的研究内容，即本人在此次毕业设计期间所做的主要工作。最后罗列了本文的结构及内容安排，方便读者查阅。

第二章：基于深度学习的表情识别算法理论。本章第 1 节介绍表情识别在计算机实现的具体步骤，在介绍步骤的同时列举了传统表情识别算法。第 2 节重点详细讲述了基于深度学习的表情识别算法即卷积神经网络的原理，并对 CNN 实现图像分类和识别的方法做了简要说明。最后介绍了卷积神经网络的发展，对具有标志意义的 CNN 算法进行了展示和分析。

第三章：改进的基于深度学习的表情识别算法。第 1 节介绍 Xception 算法和 mini_Xception 算法，本课题改进的表情识别算法便基于此。第 2 节展示改进的人脸表情识别算法的框架。最后详细分析和展现了该算法的训练过程，性能及实际识别效果。

第四章：算法在情感机器人上的实现。本章首先介绍实验所用的 NAO 情感智能机器人；接着对算法在机器人上实现的方式做出说明；最后描述 NAO 机器人进行表情识别的具体状态和效果。

第五章：总结与展望。第五章分为两部分呈现。第一部分为全文总结与反思，既总结全文内容又分析了本次实验的不足与需要改进之处。第二部分是未来研究的展望，针对实验的不足提出未来的研究方向。

论文结尾为结束语及参考文献。

第二章　基于深度学习的表情识别算法理论

表情识别的具体实现算法有许多，但基本可以将是否用到深度学习方法作为分类标准，学术界通常将各种非深度学习方法归为传统表情识别算法。由于本课题选取更加新颖的深度学习方法作为表情识别算法进行研究，本章主要介绍深度学习表情识别算法即卷积神经网络的基本理论和发展。但在介绍卷积神经网络之前，本章的第 1 节首先介绍表情识别的基本步骤，顺带也会列举一些传统表情识别算法。

2.1　表情识别的基本步骤

大多数面部表情识别方法通常遵循如图 2-1 所示的基本过程，即人脸图像的获取与预处理，表情特征提取与选择，人脸表情分类。对于传统使用机器学习的表情识别方法，作为模式识别问题，每一步都可以选择不同的算法，特别是在特征提取和分类器的选择上，每一个算法都有自己的闪光点[15]。而对于深度学习的表情识别算法，主要是通过卷积神经网络来完成表情特征的提取与分类。

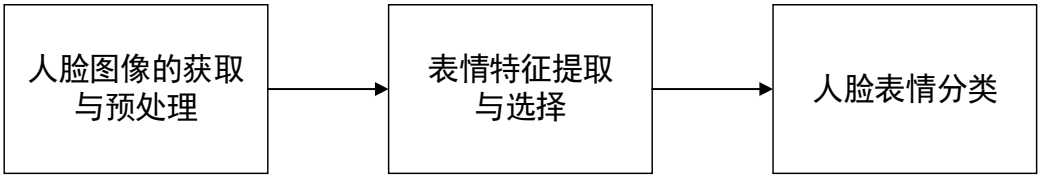


图 2-1 人脸表情识别的基本过程

2.1.1 人脸图像的获取与预处理

由于真实场景中采集到的面部表情受光照变化、头部倾角变化、非平面头部旋转、局部遮挡等多种因素的影响，对面部表情图像进行适当的预处理可以提高识别的稳定性和识别精度。常用的预处理方法包括以下步骤：人脸检测、人脸分割、人脸对齐、图像去噪、图像增强和人脸归一化[16]。

2.1.2 表情特征提取与选择

预处理后的图像具有较少与类别无关的特征，有利于特征提取。除了直接使用原始图像作为分类器输入的算法（如基于卷积神经网络的算法）外，人脸图像常用的特征可以分为两类：几何特征和纹理特征。

几何特征是指物体在图像中的位置、方向、周长和面积特征。人脸图像的几何特征主要包括人脸特征点的位置、移动速度和相互距离。人脸图像的几何特征直观、简单，在人脸图像分析中起着非常重要的作用。最常用的几何特征提取方法有：基于人脸动画参数定义（FAPs，facial animation parameters）定位标志点[17]；基于活动形状模型定位特征点[18]；使用 Candide 节点提取特征[4]；使用变形的 Candide 人脸网格提取特征[19]等。

纹理特征是反映图像均匀性的视觉特征。它反映了物体表面的组织和排列特性，而这些特性是缓慢或周期性变化的。纹理特征具有局部序列重复和非随机排列的特点，纹理区域基本上是均匀统一的。人脸图像的纹理特征主要是图像纹理的变化，如皮肤的皱纹和隆起。例如，当微笑时，鼻子两侧的判定图案会加深。常见的纹理特征有：Gabor 滤波器输出、像素强度、离散余弦变换特征和肤色信息。大多数纹理特征的提取基于：Gabor 小波[20]；尺度不变特征变换（SIFT，scale invariant feature

transform）[21]；局部二值模式（LBP，local binary pattern）特征[22]；类 Haar 特征[23]；光流[24]；判别非负矩阵分解[19]。

表情特征提取之后，特征选择对传统表情识别算法同样重要。合理的特征选择方法不仅可以滤除由非表达因素引起的“假特征”，而且可以消除特征之间的相关性，降低特征维数和计算复杂度。特征选择方法包括：运用粗糙集理论对面部特征选择[17]；基于线性规划（FSLP，feature selection by linear programming）的特征选择方法[25]等。

2.1.3 人脸表情分类

在人脸表情识别过程中，核心步骤是对给定的人脸图像或视频序列进行分类，即根据提取的特征将其映射到给定的类空间。所选特征向量可以用作所选分类器的输入，分类器将所选分类向量划分为基本表情类，如第一章第 1 节中提到的高兴、悲伤、生气、惊讶、恐惧、厌恶和中立七种基本表情。

常用的几何特征分类方法有：神经网络[16]、经验分类规则[26]、隐马尔可夫模型（HMM，Hidden Markov Model）[24]、Adaboost 算法[23]、支持向量机（SVM）[17]等。常用的纹理特征分类方法相对较少，主要包括神经网络[16]、经验分类规则[26]、Adaboost 算法[23]、支持向量机[27]等。

2.2 卷积神经网络的基本原理

2012 年，当基于卷积神经网络的 AlexNet 算法[28]在 ImageNet 大规模视觉识别挑战赛（ILSVRC，ImageNet Large Scale Visual Recognition Challenge）[29]中以领先第二名接近 10 个百分点的绝对优势夺冠时，似乎宣告着图像识别技术新时代的来临。此后的每一年，ILSVRC 的冠军都由深度学习领域的卷积神经网络算法取得。卷积神经网络已逐渐成为计算机视觉中的主要算法，开发性能更好的卷积神经网络算法框架也一直是计算机视觉领域的研究热点。

卷积神经网络基本可分为输入层，卷积层，激活函数，池化层，全连接层和输出层。下面简要介绍各层的基本原理，以及卷积神经网络是如何实现图像分类和识别的。

2.2.1 输入层

输入层代表输入到 CNN 中的原始图像。例如使用 RGB 图像作为输入，则输入层具有三个通道，分别对应于该层中显示的红色，绿色和蓝色通道。如图 2-2 左侧可表示为 32×32×3 的输入图像，其中 32×32 为图像的长乘宽，即像素值。3 为通道数，也可称为维度或深度。

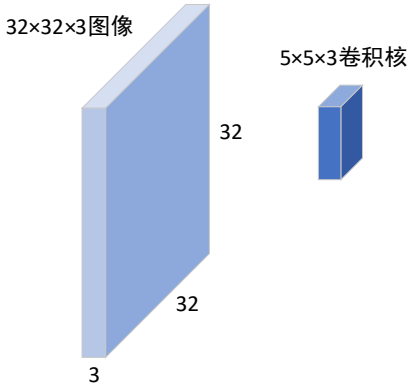


图 2-2 输入图像与卷积核示例

2.2.2 卷积层

卷积层是 CNN 的基础，它包含多个卷积核（filter，也称滤波器），这些内核用于提取出可将不同图像彼此区分开的特征，如图 2-2 右侧所示，其中卷积核的深度须与输入图像的深度相同。特征提取由卷积操作完成，每个卷积核都是独立且不同的，分别提取图像的不同特征。

卷积操作是通过卷积核在输入图像上滑动，计算出每个空间定位时的点积结果，形成输出特征图像，图 2-3 大致展现了卷积核在输入图像的滑动过程。具体计算原理如图 2-4 和图 2-5 所示，卷积核与图像的对应位置像素值相乘后求和，然后卷积核依次滑动，最终求出卷积结果。这里只演示了单通道输入图像的卷积计算过程，对于多通道的输入图像，则将各通道分别卷积操作后相加即可。其中图 2-5 中绿色为输入图像像素矩阵，黄色为卷积核，粉色矩阵的数值为对应卷积操作的计算结果。

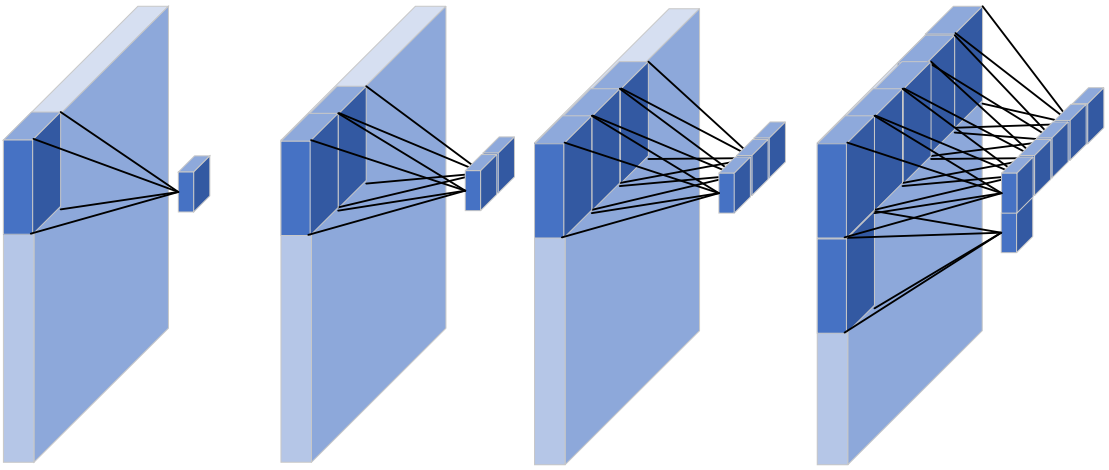


图 2-3 卷积操作中卷积核在输入图像的滑动过程示意

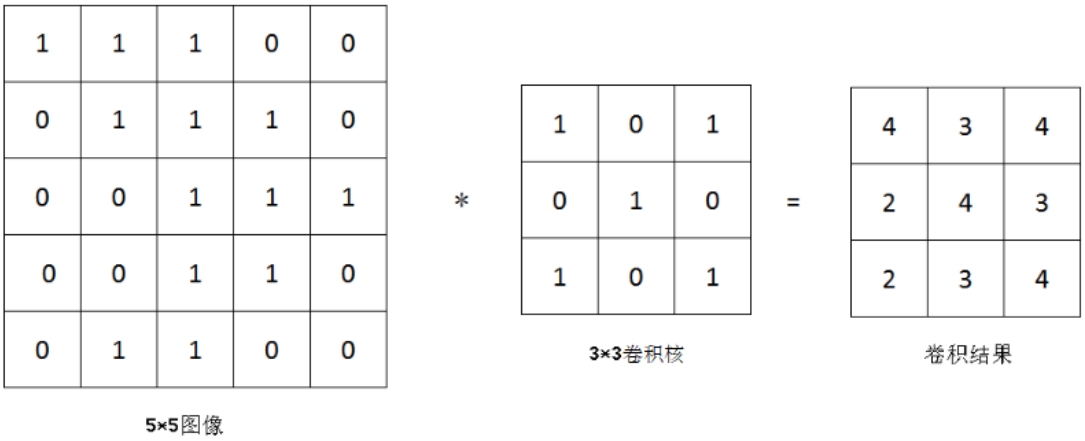


图 2-4 卷积操作的计算示例

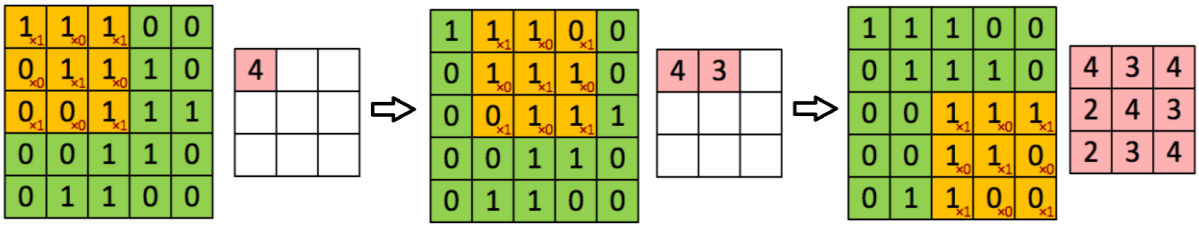


图 2-5 卷积操作的计算过程演示

实际应用卷积操作时通常还有 4 个超参数（Hyperparameters）需要设置：

1. 卷积核个数（filters）：卷积核的个数也就是输出图像的维度，因为一个卷积核对应一个卷积操作输出图像的维度，每个卷积核都是不同。例如前面提到 32×32×3 的输入图像，由于它是 3 通道的，则得到一个完整的输出图像需要 3 个不同的卷积核，即每个卷积核分别在各自通道进行卷积操作后相加。为了多提取不同的图像特征，若想要生成 10 个完整的输出图像，则需要 3×10=30 个不同的卷积核个数。

2. 卷积核尺寸（kernel size）：即卷积核大小。如图 2-4 和 2-5 展示的卷积核尺寸为 3×3。卷积核尺寸会对图像分类任务产生重大影响。例如，较小的卷积核尺寸能够从输入中提取大量包含局部特征的信息。同样，较小的核尺寸也导致较大的输出尺寸，这允许更深的网络体系结构。相反，较大的内核会提取较少的信息，这会导致输出图像尺寸减小得更快，从而常常导致性能下降。大内核更适合提取更大的特征。通常，更多图层的堆叠可以学习到更复杂的图像特征。

3. 步长（strides）：指卷积核沿宽度和高度方向一次应移动多少像素。其中图 2-4 和 2-5 展示的步长为 1。若步长为 2，则输出图像大小为 2×2。步长对 CNN 的影响类似于卷积核大小。随着步长的减小，由于提取了更多的数据，因此可以学习更多的特征，同样也导致了更大的输出图像。相反，随着步长的增加，将导致特征提取更加受限，输出图像尺寸更小。

4. 填充（padding）：当卷积核扩展到输入图像之外时，通常需要填充。填充可以在输入图像的边界处保存数据，保证输入图像的完整，从而获得更好的性能。填充方式存在许多，但是最常用的方法是零填充，因为它操作简单，计算效率高，性能好。零填充即在输入的边缘周围对称地添加零像素。

2.2.3 激活函数

卷积操作完成后，往往会使用激活函数对操作结果作非线性运算。非线性运算对表情分类至关重要，是产生非线性决策边界（decision boundaries）所必需的，如果不存在非线性激活函数，那么深度 CNN 架构将演变为等效的单一卷积层，其性能则完全不一样。激活函数有 Sigmoid 函数[30]，Tanh 函数，ReLU 函数（Rectified Linear Unit）[31]，Softmax 函数等。

由于 Sigmoid 函数和 Tanh 函数在卷积神经网络中已逐渐被淘汰，本小节只介绍主流激活函数 ReLU 函数，而 Softmax 函数主要用于输出层，随后将在输出层小节介绍。

ReLU 激活函数是一对一的数学运算，公式为：

$$\mathrm{ReLU}(x) = \max(0, x)$$

函数图像如图 2-6 所示。此激活函数逐个应用于图像像素矩阵中的每个值。此函数很简单，即正数保持不变，负数取 0。

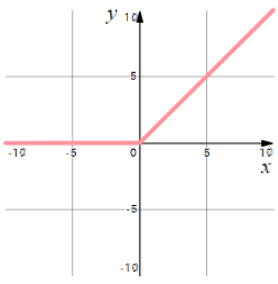


图 2-6 ReLU 激活函数的函数图像

2.2.4 池化层

在不同的 CNN 架构中，池化层的类型很多，但是它们的目的都是要逐渐减小网络的空间范围，从而减少网络的参数和总体计算。常见的池化类型有最大池化（Max-Pooling）和平均池化（Average-Pooling）。

最大池操作需要在网络体系结构设计期间选择内核大小和步幅。一旦选定，该操作将以指定的步幅在输入图像上滑动内核，同时仅从输入中选择每个内核切片上的最大值以产生输出值。如图 2-7 所示，池化层使用 2×2 内核，步幅为 2。通过池化操作丢弃一些值，使计算效率更高，并且避免了过拟合。

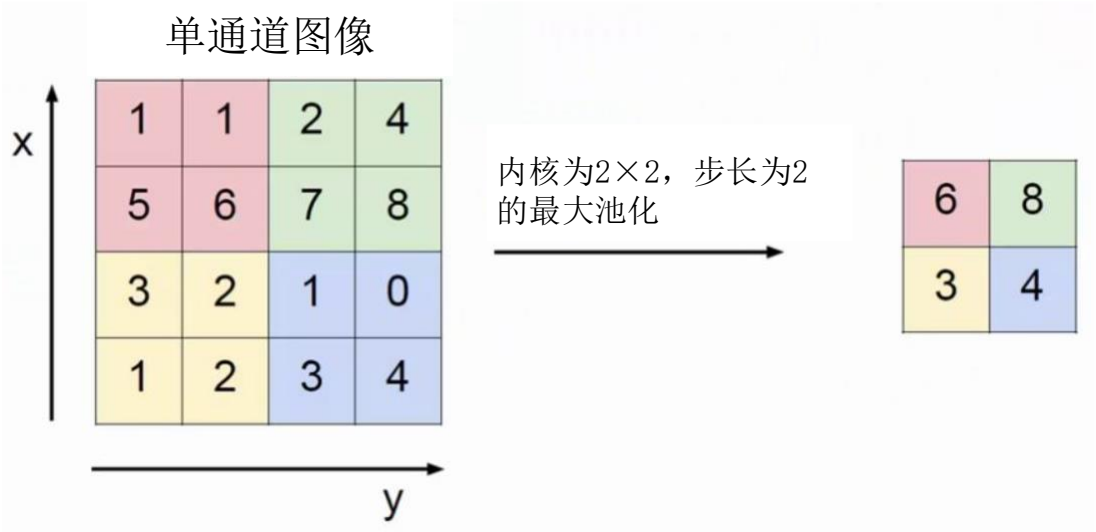


图 2-7 池化操作示意图

平均池化操作与最大池化类似，只是选择每个内核切片上的最大值改为平均值，其他不变。

2.2.5 全连接层

全连接层（fully connected layers，FC）在卷积神经网络中一般处在输出层前，它可以整合卷积层和池化层中具有类别区分性的局部信息。全连接层相当于通过卷积操作将输出特征图的大小变为 1×1，从而便于分类。例如 10 个 5×5 的特征图像，通过全连接想要得到 100 个 1×1 的特征点用于分类的话，需要 10×100 个 5×5 的卷积核，由此可见全连接层所需参数很多，不利于快速计算。现在许多 CNN 算法用全局平均池化（global average pooling，GAP）取代了全连接层。

全局平均池化将特征图每个通道的像素值直接求平均值，这样可以大大减少运算参数。10 个 5×5 的特征图像通过全局平均池化输出 10 个 1×1 的特征点，将这些特征点组成一个 1×10 的向量的话，就成为一个特征向量，可以送入到输出层通过 Softmax 函数进行分类了。

2.2.6 输出层

输出层作为卷积神经网络的最后一层，一般在全连接层后，通常采用 Softmax 函数用于分类。Softmax 函数公式如下：

$$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Softmax 操作的主要目的是：确保 CNN 输出的总和为 1。因此，Softmax 操作可用于将 CNN 模型输出缩放为概率。输出数值表示输入的图像属于某一分类的概率，输出最大的概率值对应的类别便为最后的分类结果。

2.2.7 CNN 实现图像分类和识别的方法

一个完整的卷积神经网络框架包含输入层，隐含层和输出层。隐含层就是卷积层、激活函数、池化层和全连接层的各种组合堆叠。CNN 模型的不同也主要体现在隐含层的多种结构设计的不同，几种主要的 CNN 模型框架图在下节有所展示。这里先讲述 CNN 是如何进行图像分类和识别的。

CNN 要想进行图像分类和识别，必须经过反复的模型训练，以得到最佳的识别率。简单来说就将大量的数据图像（这些图像已标有正确的类别标签）从输入层进入 CNN 模型，经过隐含层输出各类别的概率，将这些概率值与实际标签通过损失函数计算出差异值（也称损失值，loss）。为了使预测概率值与真实值接近即差异值尽可能小，需通过优化算法（反向传播算法及其变体）确定哪些权重（卷积核的数值）对损失值影响最大，从而进行权重更新。以上从图片输入到权重更新的过程构成一轮训练迭代。CNN 训练会经过多次迭代，直到损失函数收敛于一个较小的值。

训练完的模型就可以进行图像识别了，将待识别的图片输入到训练好模型中，会得到每个类别的概率，概率最大的类别就是 CNN 模型识别出的类别。既然 CNN 可以识别不同种类的图片，把每种表情看成不同的类别的话，自然可以进行表情识别。

2.3 卷积神经网络的发展

卷积神经网络设计的历史始于 1995 年的 LeNet 样式的模型[32]，如图 2-8 所示，该模型是用于特征提取的卷积和用于空间下采样的最大池化操作的简单堆叠，用于解决手写数字识别任务。自那时起，以卷积层、池化层、全连接层构成的 CNN 最基本的架构就定下来了。虽然 LeNet 看起来很简单，但以当时计算机的计算能力来说，卷积神经网络实现起来仍然相当困难，再加上在一般的实际任务中表现不如 SVM、Boosting 等算法好，所以 CNN 算法一直处于学术界边缘的地位，发展也暂时停滞了。

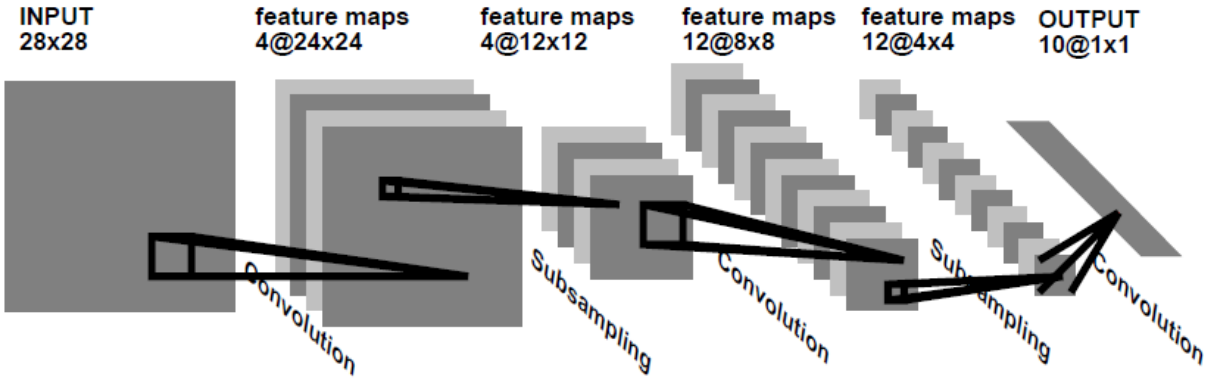


图 2-8 LeNet 样式模型示意图[32]

到了 2012 年，LeNet 思想被改进为 AlexNet 体系结构[28]，在最大池化操作之间多次重复卷积操作，使网络能够在每个空间尺度上学习更丰富的特征。如图 2-9 所示，AlexNet 体系结构可以简单理解为 LeNet 结构的加深，计算机计算能力的不断提升使更深层卷积神经网络的实现成为可能。

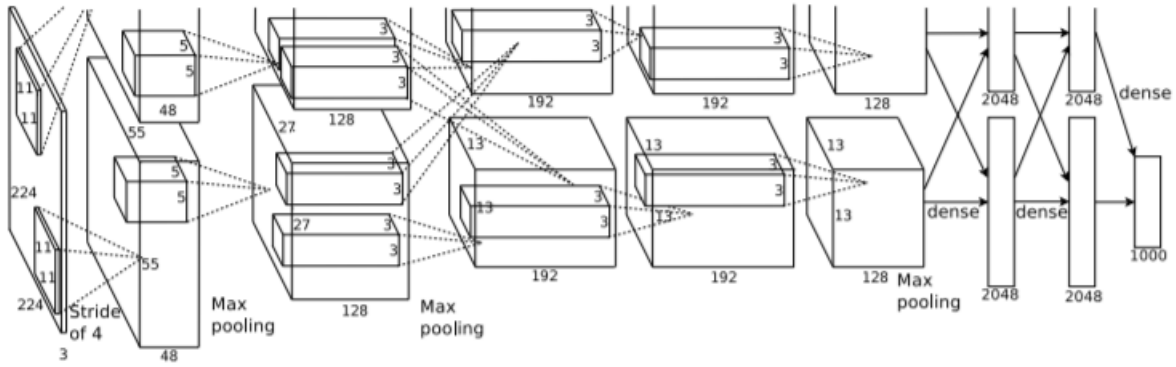


图 2-9 AlexNet 结构框架[28]

随后出现的趋势是这种类型的网络越来越深，这主要是由每年的 ILSVRC 竞争推动的。首先是 2013 年的 Zeiler 和 Fergus（ZF）框架[33]，然后是 2014 年的 VGG（Visual Geometry Group）架构[34]。ZFNet 是 2013 年 ILSVRC 的冠军，其网络结构相对于 AlexNet 没什么改进，只是调整了一些参数，使性能较 AlexNet 提升了不少。VGG 可以看成是加深版本的 AlexNet，其层数从 AlexNet 的 8 层提升到了 19 层。

此时，出现了一种新的网络样式，即由 Szegedy 等人发明的 Inception 体系结构，在 2014 年被命名为 GoogLeNet（Inception V1）[35]，后来又被更名为 Inception V2 [36]，Inception V3 [37]，以及 Inception V4[38]。Inception 本身受到早期 Network-In-Network 体系结构的启发[39]。Inception 样式模型的基本构建模块是 Inception 模块，其中存在几种不同的版本。在图 2-10 中展示了 Inception V3 体系结构中的 Inception 模块的规范形式。一个 Inception 模型可以理解为此类模块的堆叠，这与早期的 VGG 样式网络不同，不再是简单的卷积层的堆叠。Inception 模块的想法是通过一组 1×1 卷积降低通道深度，使输入数据映射到比原始输入空间小的 3 个或 4 个独立空间，然后通过常规的 3×3 或 5×5 卷积来在这些较小的 3D 空间中进行卷积操作，从而减少了运算参数。

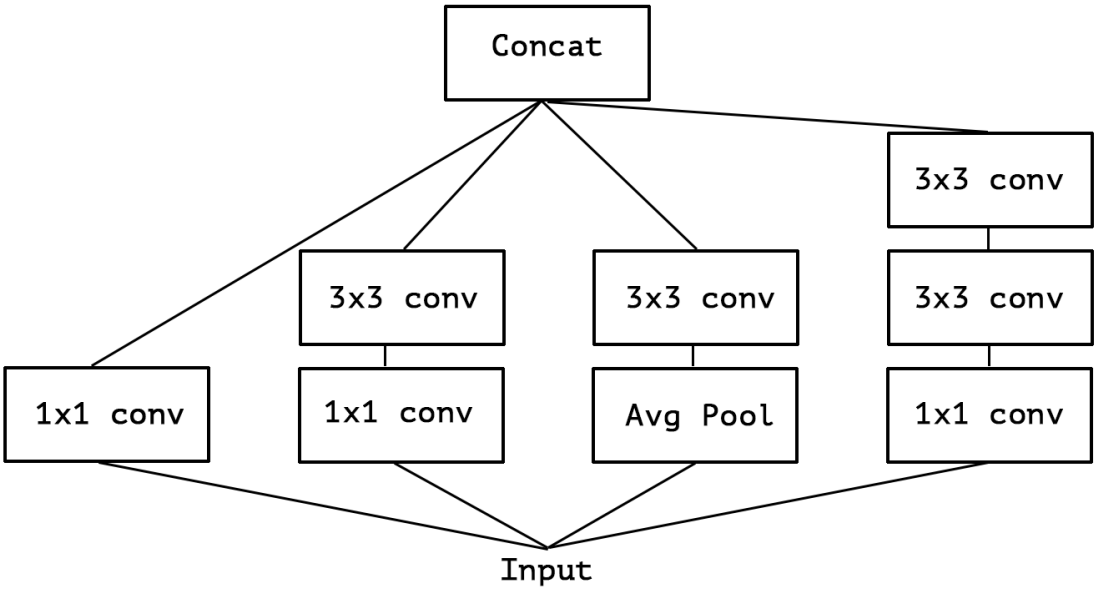


图 2-10 一个规范的 Inception 模块（Inception V3）[37]

到了 2015 年，何恺明推出的 ResNet[40]在 ISLVRC 上横扫其它所有算法，获得冠军。ResNet 在

网络结构上做出了大创新，这一新思路绝对可以称得上深度学习发展历程上里程碑式的事件。ResNet 的关键创新就是残差模块，如图 2-11 所示。左侧是常规残差模块，右侧是用于更深网络的残差模块。残差模块解决了随着网络深度增加导致的梯度消失和误差增大的问题，使网络层数从 GoogLeNet 的 22 层一下增加到 152 层，图像分类和识别的能力也大大提升，在 ILSVRC 的 Top-5 错误率仅为 3.57%，甚至超过了人类的识别能力。

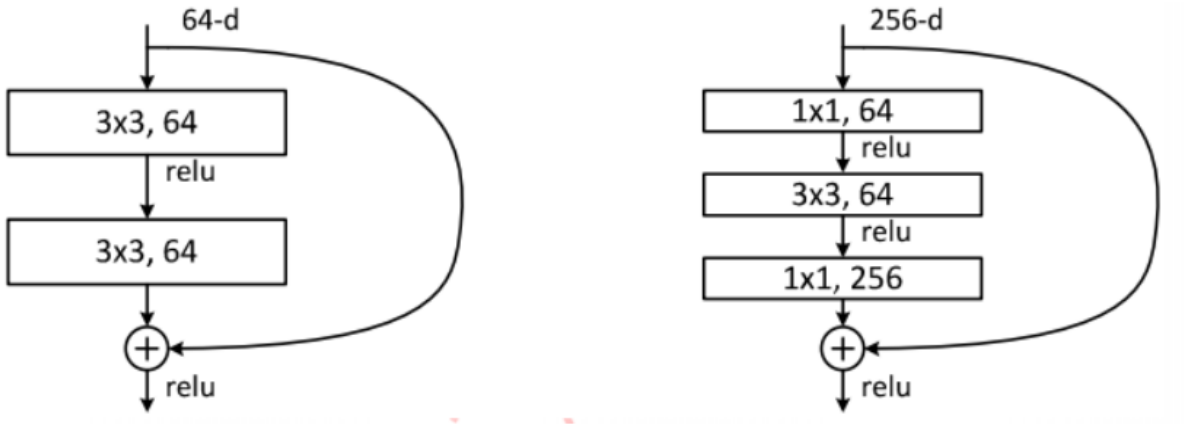


图 2-11 两种残差模块[40]

接下来两年的 CNN 算法架构基本在 Inception 模块和残差模块的基础上不断更新改进，在 ILSVRC 的识别错误率也不断降低，在 2017 年 Top-5 错误率低至 2.25%。随着 2018 年 ILSVRC 被更具挑战性的竞赛取代后，CNN 的发展仍在继续，相信更出色的 CNN 算法架构会不断问世。

第三章　改进的基于深度学习的表情识别算法

在上一章介绍完卷积神经网络的基本原理及发展历程后，本章介绍本课题所实现的表情识别算法。该算法在 Xception 算法[41]和 mini_Xception 算法[42]的基础上做出了一些改进。本章的第 1 节首先介绍 Xception 算法和 mini_Xception 算法的相关理论，之后提出本论文所改进的表情识别算法，并对算法做出性能分析和评估。

3.1　Xception 算法和 mini_Xception 算法

Xception 算法是由 François Chollet 在 2017 年提出的一种受 Inception 启发的新型深度卷积神经网络体系结构，ImageNet 数据集（Inception V3 是专为该数据库设计的）上略优于 Inception V3，在包含 3.5 亿个图像和 1.7 万个类的更大图像分类数据集 JFT[43]上显著优于 Inception V3。由于 Xception 体系结构具有与 Inception V3 相同数量的参数，因此性能的提高不是由于容量的增加，而是由于模型参数的使用更有效。Xception 体系结构的革新在于用深度可分离卷积[44]代替了 Inception 模块，即通过构建带有残差连接（residual connections）的深度可分离卷积堆叠的模型，改进 Inception 系列体系结构。

深度可分离卷积通过将卷积操作分为深度卷积和点卷积两个方向执行，来达到减少参数数量和运算成本的目的，从而使加深层数的网络更容易训练。首先进行深度方向卷积，即在输入的每个通道上独立执行空间卷积；然后是点卷积，即 1×1 卷积，将通过深度卷积输出的通道图像投影到新的通道空间上。例如图 3-1 所示，要将拥有 3 个通道的 5×5 输入图像通过卷积核大小为 3×3 的深度可分离卷积操作输出成 5 个通道的输出图像，首先在 3 个通道各自进行常规卷积操作，得到 3 个通道的中间特征图，此操作需要 3×3×3=27 个参数；然后通过 5 个 3 通道的 1×1 卷积将中间特征图扩展成 5 通道输出图，需 1×1×3×5=15 个参数，则一共需要 27+15=42 个参数。若要完整实现上述操作，常规卷积层需要 3×3×3×5=135 个参数，由此可知深度可分离卷积大大减少了训练参数，同时在特征提取效果上几乎与常规卷积操作一样。

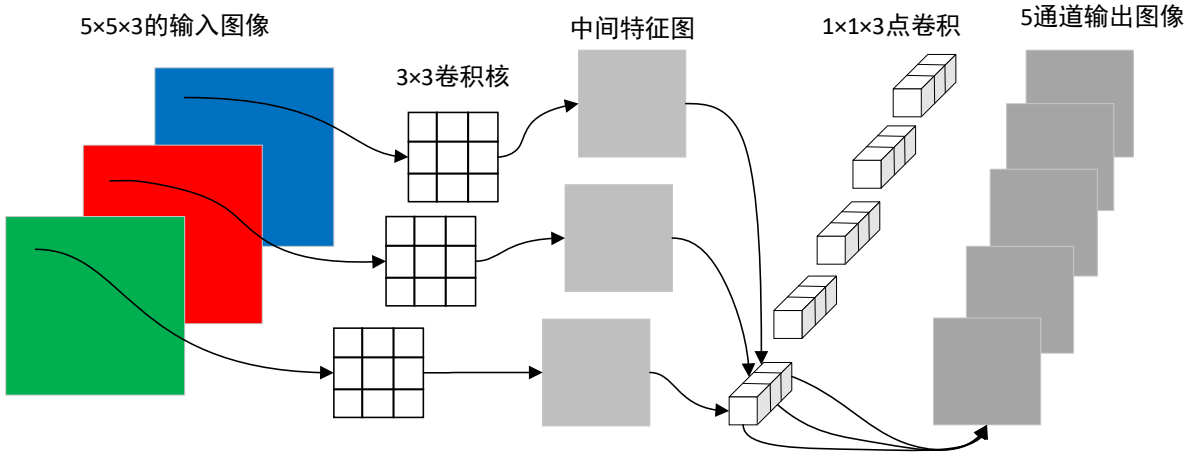


图 3-1　深度可分离卷积操作示意图

Xception 的完整体系结构如图 3-2 所示，具有 36 个卷积层，构成了网络的特征提取基础。数据首先通过入口流，然后通过中间流，该中间流重复八次，最后通过出口流。其中 Conv 代表卷积层，SeparateConv 代表深度可分离卷积层，MaxPooling 代表最大池化操作，后面的数字（如 32，

3×3，stride=2×2）分别代表内核数量，核尺寸大小和步长。所有卷积和深度可分离卷积层后面都进行了批量归范化（Batch normalization）[36]（图中未包括），所有深度可分离卷积层均使用深度倍数为 1（无深度扩展）。

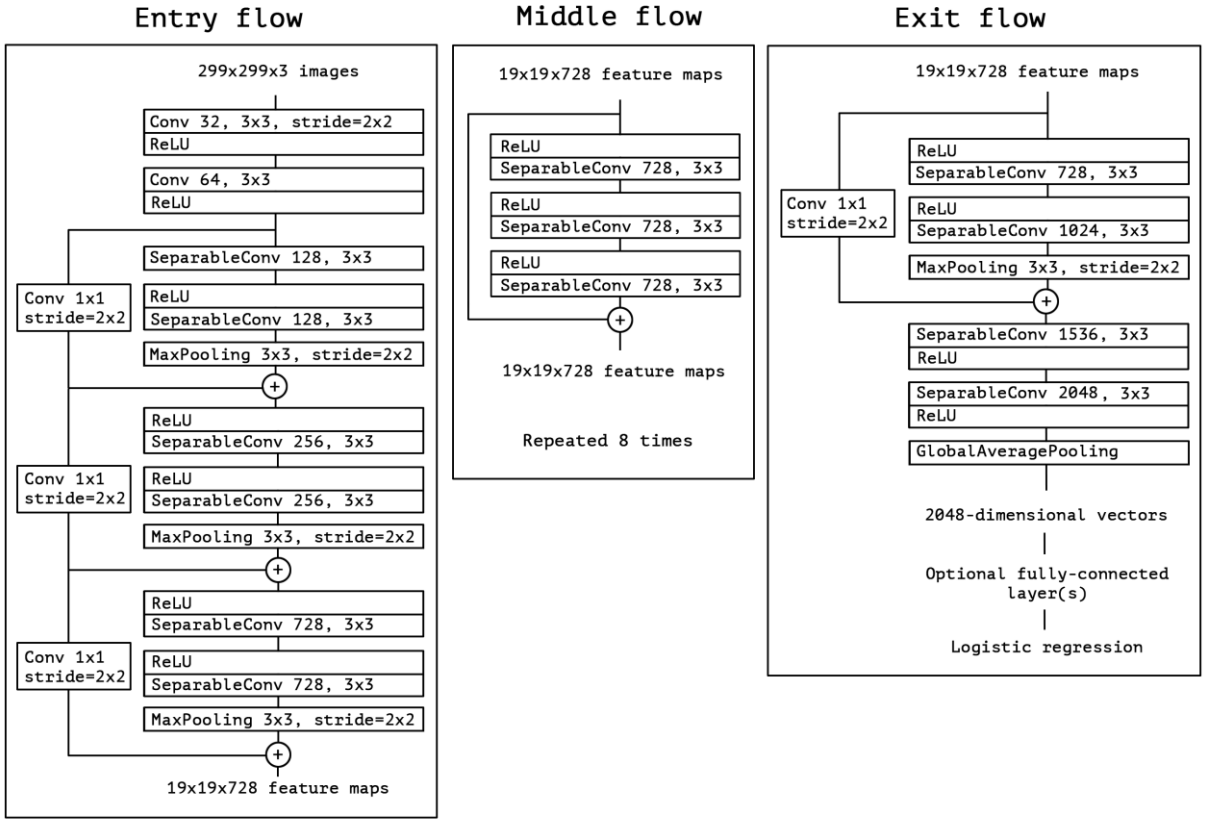


图 3-2 Xception 体系结构描述图[41]

mini_Xception 算法是 Octavio Arriaga 等人在 Xception 框架的启发下专门为表情识别所设计的，它包含 4 个带有残差连接的深度可分离卷积模块，每个卷积后面跟着一个批量归范化操作（Batch normalization）和一个 ReLU 激活函数。最后一层应用全局平均池化和 Softmax 激活函数来生成预测。mini_Xception 算法架构如图 3-3 所示，该算法在 FER-2013 数据集上达到了 66%的准确率，且大约只有 60000 个参数，可以存储在 855 千字节（KB）的文件中，在 i5-4210M CPU 上进行一次表情识别仅耗时 0.22±0.0003ms，可在 Care-O-bot 3 机器人上实现实时表情识别。

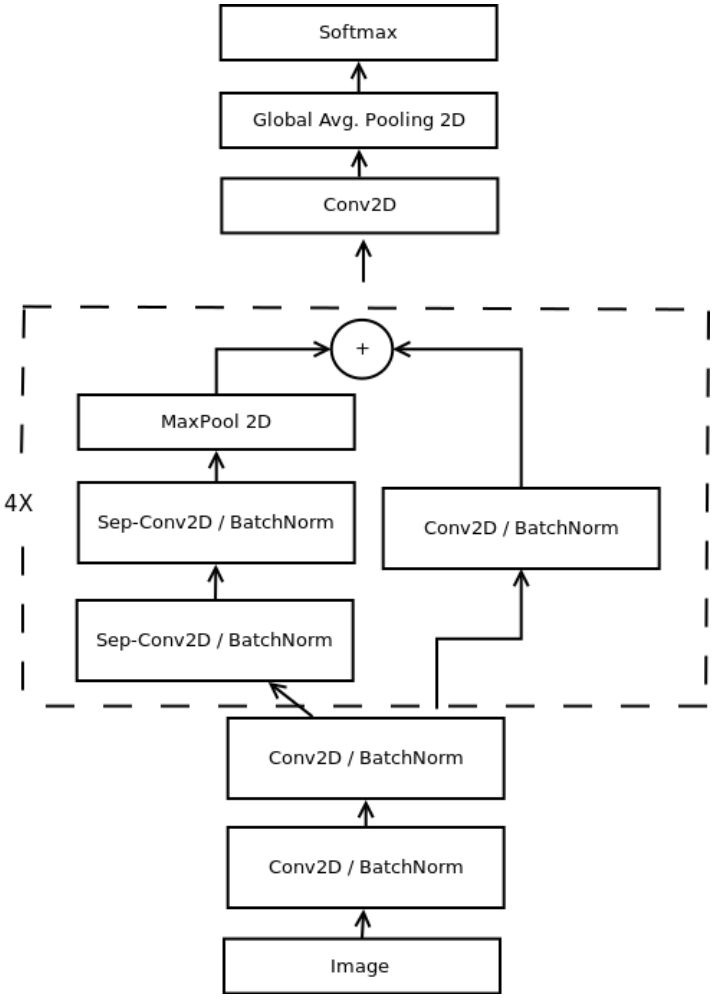


图 3-3 mini_Xception 算法架构图

3.2 改进的表情识别算法

本课题所实现的人脸表情识别算法是在 mini_Xception 算法的基础上进行改进的。改进主要表现在两个方面：

1. 将 mini_Xception 的倒数第三层卷积层改为局部连接层（LocallyConnected2D）。局部连接层与卷积层的工作方式相同，唯一的区别是局部连接层的权值不共享，也就是说，过滤器的权重在输入图像滑动时是会变的，这样更容易提取局部的特征，如眼，鼻子，嘴等。实验结果表明，这一改动可将表情识别率提升 1%左右。

2. 调整参数。原算法的参数是在输入图片为 64×64 像素的基础上设置的，而本次实验所选用的 FER-2013 数据集和 CK+数据集的原始图片像素为 48×48，为了防止增大图片像素而可能导致的特征丢失，本次实验最终选择原始图片作为输入，并对算法的参数进行调整以适应 48×48 像素的图片输入。例如调整了池化层和局部连接层的内核大小，将大小从 3×3 改为 2×2，以防止内核过大而可能导致的特征提取的丢失。此外，还适当增加了卷积核的数量，以保证可提取出更丰富的特征。具体参数设置见附录源代码。

本课题最终改进的算法框架如图 3-4 所示。中间经过了 4 遍带有残差连接的深度可分离卷积模块（用虚线框出）。另外所有卷积和深度可分离卷积层后面都进行了批量归范化，并在中间穿插了适量的 ReLU 激活函数（图中未包括）。

值得一提的是，本课题还尝试了在第 1 和第 2 个虚线内模块之间加入两个如图 3-5 所示的残差模块，可将算法在数据集的识别率进一步提升，但这一改动在实际识别效果上并没有体现出令人满意的效果，具体分析见下节。

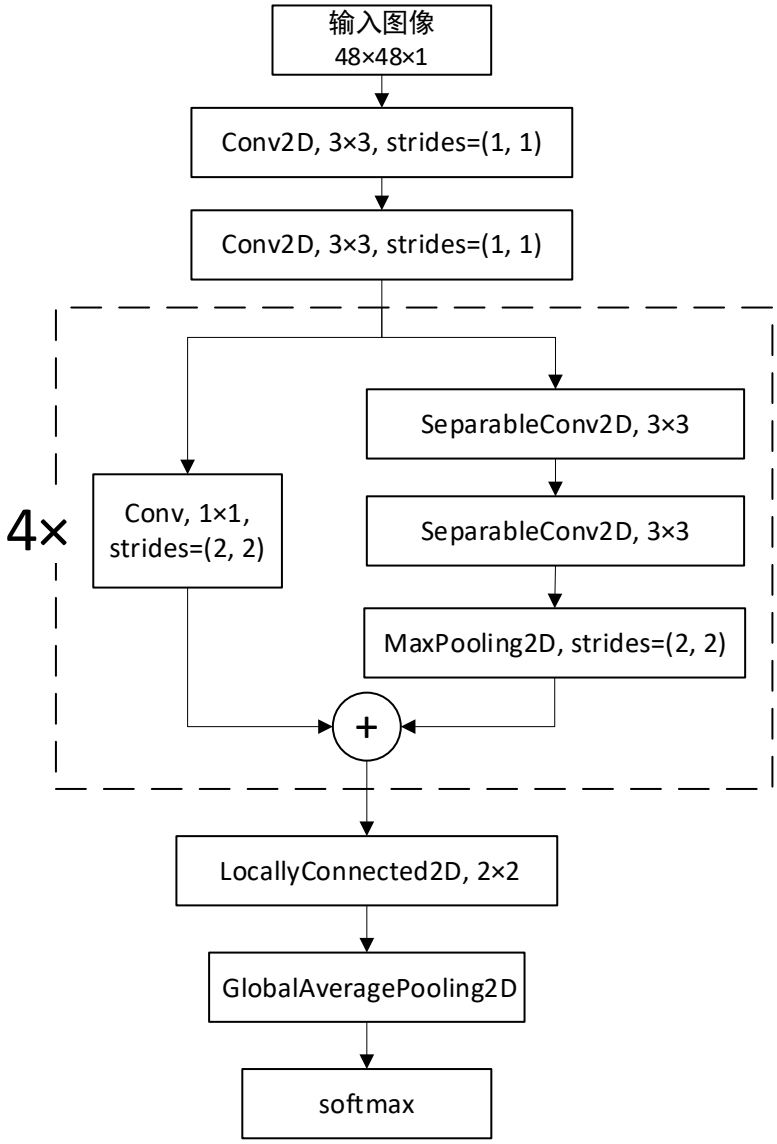


图 3-4 改进的表情识别算法框架

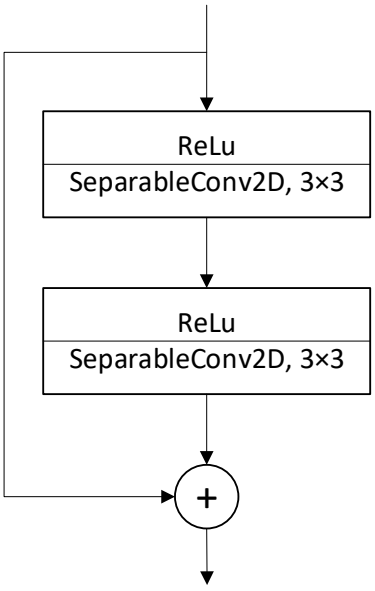


图 3-5 额外添加的残差模块

3.3 算法实现及性能分析

本课题所改进的算法基于 Keras 深度学习库在本人笔记本电脑上进行训练。Keras 提供了一致和简单的 API，易于学习和使用，与其他众多深度学习框架相比，Keras 在行业和研究领域的应用率更高。同时，由于 Keras API 是 TensorFlow 的官方前端，本次实验使用 TensorFlow 作为后端。在人脸检测方面，实验使用 Opencv 的 Viola-Jones 算法[45]实现。实验还应用了一些其它的软件库，具体代码见附录源代码。

硬件环境方面：笔记本电脑的 CPU（中央处理器，central processing unit）为 Intel Core i5 5200U（主频 2.20GHz），GPU（图形处理器，Graphics Processing Unit）为 NVIDIA GeForce 910M（显存 2GB，DDR3），内存为 12GB（DDR3）。

实现语言为 Python 3.6，IDE（集成开发环境，Integrated Development Environment）为 PyCharm 2020.1.1(Community Edition)。

3.3.1 数据集

用于改进的表情识别算法模型训练所需的数据集融合了 FER-2013 数据集[46]和 CK+数据集[47]，具体操作及介绍如下：

FER-2013 数据集是 Kaggle 人脸表情分析比赛所提供的一个数据集。共有 35887 张图片，每张图片像素固定为 48×48 的灰度图像，分为生气（angry）、厌恶（disgust）、恐惧（fear）、高兴（happy）、悲伤（sad）、惊讶（surprise）和中立（neutral）七种类别的表情。图 3-6 展示了几张 FER-2013 数据集里的图片。FER-2013 数据集图片数量较多，且包含各年龄段各肤色的人脸表情，但该数据集也不是完美的，里面包括一些错误和人脸不清晰的图片，再加上本实验面向的表情识别为正脸表情识别，为保证实际识别效果能更好，本实验删除了如图 3-7 所示的一些侧脸、错误和人脸过小等会对训练及预测造成干扰的图片，共 2845 张。

图 3-6 FER-2013 数据集样本

图 3-7 FER-2013 数据集中错误、侧脸等不适合训练的图片

CK+数据集是在 Cohn-Kanade 数据集的基础上扩展来的，发布于 2010 年，共有 981 张 48×48 的灰度表情图像，如图 3-8 所示。该数据集共分为 7 种表情，即愤怒（anger）、轻视（contempt）、厌恶（disgust）、恐惧（fear）、开心（happy）、悲伤（sadness）及惊讶（surprise），除去 54 张轻视的表情图片不在本实验所设计识别的表情范围内，其余图片的质量都很高，表情清晰且完整。

图 3-8 CK+数据集

综上所述，本实验的数据集选用了 33042 张 FER-2013 数据集中的图片及 927 张 CK+数据集里的图片，共计 33969 张图片。

3.3.2 模型训练参数设置

模型训练所需的参数包括优化器及正则化配置，具体如下：

优化器：Adam[48]

初始学习率（Initial learning rate）：0.001

损失函数（Losses）：categorical_crossentropy

学习率调整（ReduceLROnPlateau）：连续 10 轮（epochs）损失值没有下降，则将学习率×0.1

正则化配置：l2_regularization=0.01

批量尺寸（batch_size）：32

为了将本实验改进的算法与 mini_Xception 算法做对比，以上参数均和 mini_Xception 算法所用参数保持一致，且这些参数基本为原始论文的默认良好参数，故不做更改。

此外，在训练算法模型时，我们还使用了 Keras 提供的 ImageDataGenerator 图片生成器来进行实时数据增强。ImageDataGenerator 可以通过将输入的图片进行随机旋转和水平翻转等操作，使模型不会训练到任何两张完全相同的图片，这有利于抑制过拟合，使得模型的泛化能力更好。

3.3.3 算法性能评估

本实验将数据集的 80%用于训练，剩下 20%用于验证。实验结果表明，本实验改进的算法（图 3-3 所示算法）在数据集的整体准确率达到 69.0%，高于 mini_Xception 算法在此数据集 66.8%的准确率。混淆矩阵如图 3-9 和 3-10 所示，对比可以发现，改进的算法除了对高兴表情的识别率没有提升外，其余表情的识别率均有不同程度的提升。

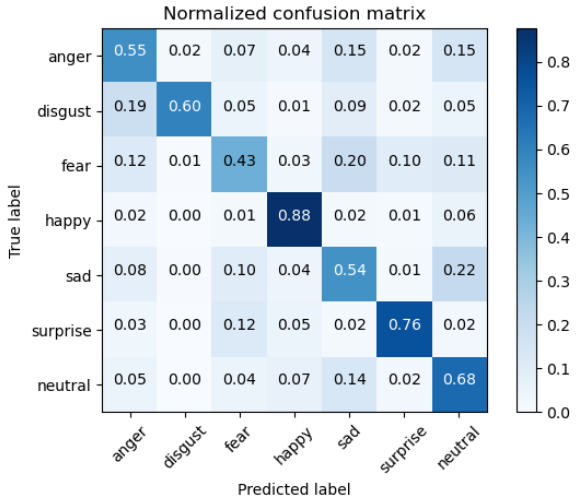


图 3-9 mini_Xception 算法在本实验数据集的混淆矩阵

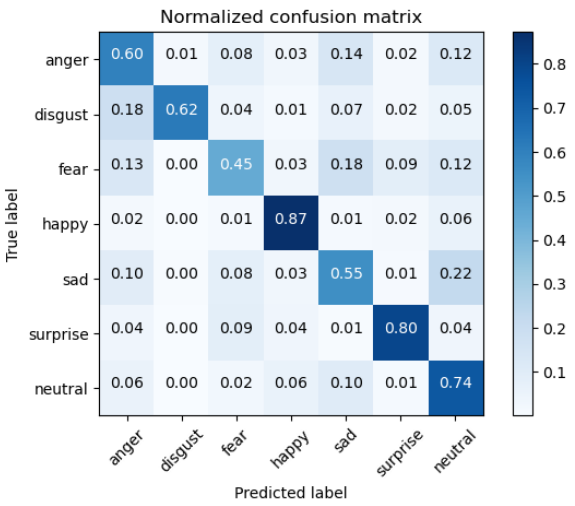


图 3-10 改进的算法在本实验数据集的混淆矩阵

上一节（3.2 节最后一段）提到的额外改进的算法最终达到了 70.0%的整体准确率，是本实验多次进行算法改进尝试得到的最高准确率，它的混淆矩阵如图 3-11 所示。可以看到，该算法在生气、厌恶、恐惧、悲伤的表情识别率进一步提升了，然而遗憾的是，该算法在实际表情识别效果中反而不如图 3-3 所示改进的算法表现的好。这里的原因还有待进一步研究。

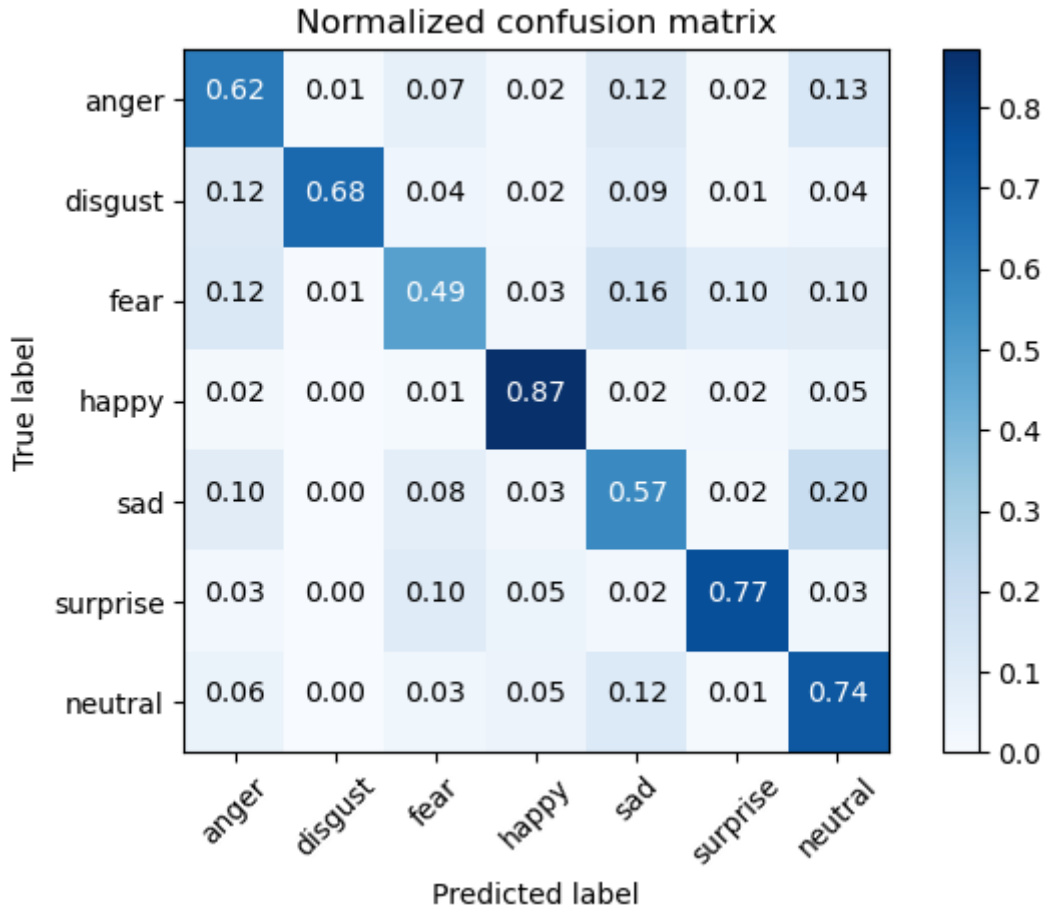


图 3-11 加入残差连接的额外改进算法的混淆矩阵图

最终，本课题选用图 3-3 所示改进的算法来进行实际表情识别，并应用在情感机器人上。该算法的实际识别效果如图 3-12 的（a）至（e）所示，每张图的左侧为本人做出的表情，右侧为对应的表情预测概率。由图中的预测概率和算法的混淆矩阵可以看出，该算法对开心和中性表情可以轻易的进行识别。例如开心的表情，只要被识别人员的嘴角轻轻上扬，开心表情的预测率就能很高，如果咧开嘴笑，预测率甚至能接近 100%。中立也是如此，只要不做任何表情，中立表情的预测率就能接近 90%。

对于生气，惊讶和恐惧这三个表情，改进的表情识别算法虽然可以识别出来，但不稳定。从图中预测率不高以及混淆矩阵中准确率的不高也能证明，其中惊讶表情虽然在混淆矩阵中达到了 80%，但实际识别效果远没有达到，猜想数据集里的表情和实际表情还存在一定出入。具体原因还需进一步研究。

至于剩下两种厌恶和伤心的表情，本改进的算法很难将其识别出来。对于厌恶的表情，猜想是由于数据集中数量过低而导致的（厌恶表情只要 704 张，其余表情皆 4000 张以上）。而对于伤心的表情，虽然在图 3-3 改进的算法中很难识别，但在加入残差连接的额外改进算法（3.2 节最后一段提到）中可以轻易的识别，如图 3-11 的（f）所示。发生这种现象的原因可能是此算法重点提取了伤心的特征，然而令人遗憾的是，此算法在生气，恐惧及惊讶三种表情的识别效果都不好。综上所述本课题最终选择了图 3-3 所示改进的算法并在机器人上实现表情识别，具体见下章说明。

图 3-12 算法实际识别效果图，其中（a）至（e）使用的是图 3-3 所示改进的算法，（f）为额外改进的算法

第四章　算法在情感机器人上的实现

本章介绍改进的人脸表情识别算法在情感机器人的实现过程。本次研究选用学校实验室提供的NAO 机器人实现表情识别功能，本章首先介绍 NAO 机器人的基本参数和功能，之后的第 2 节重点介绍算法在机器人上实现的方式，最后描述 NAO 机器人进行人脸表情识别的实际效果。

4.1　NAO 机器人介绍

NAO 是 SoftBank Robotics 公司创建的一个应用于全球教育市场的双足人形情感智能机器人。NAO 机器人身高 58 公分，拥有流畅的肢体语言，能够听、说、看，也能够与人互动，甚至可以 NAO 之间彼此进行互动。NAO 机器人还提供一个独立、可编程、功能强大且易用的操作应用环境。

本课题使用的是 2018 年发布的最新版第 6 版（NAO[6]），它的样子如图 4-1 所示。它一共拥有 25 个自由度，在头部，手和脚，声纳和惯性装置上带有 7 个触摸传感器，使他能够移动并感知周围环境。此外，NAO 还设置了 4 个定向麦克风和扬声器，使它可与人互动，最多支持 20 种语言的语音识别和对话。两台 2D 摄像机可以识别形状，物体甚至是人。最后，它开放和完全可编程的平台使改进的表情识别算法在机器人上实现成为可能。图 4-2 展示了 NAO 机器人关节与电机的分布。

图 4-1 NAO[6]机器人外貌图

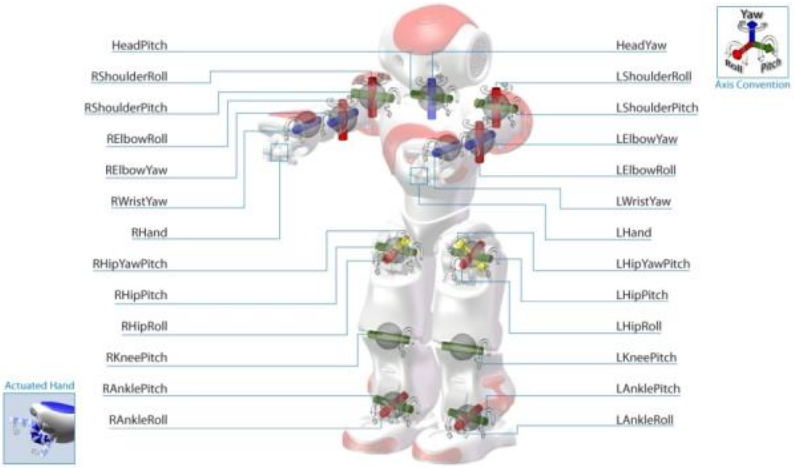


图 4-2 NAO 机器人关节和电机分布图

4.2　算法实现方式

改进的人脸表情识别算法在情感机器人上的实现大致经历了两个阶段。第一阶段为编程方式的选择，第二阶段为通过编程使 NAO 机器人利用第三章提出的改进算法进行人脸表情识别，并做出相应的反应。下面分别进行陈述。

4.2.1 编程方式的选择

NAO 机器人提供两种编程方式，分别是：

1. Choregraphe：专门为 NAO 机器人编程而设计的 IDE，可使用简单的拖放界面和 Python 进行编程。

2. SDK（软件开发工具包，Software Development Kit）：可访问 NAO 机器人的全部功能。适用于 Python 和 C++语言。

虽然 Choregraphe 编程比较简单，只需通过拖拽程序内的工具盒及修改参数就能完成编程，但它仅限于实现 NAO 机器人内部自带的功能，无法将本课题所改进的算法在机器人上实现，故本实验选择使用更灵活的 Python SDK 进行编程。

4.2.2 算法实现过程

算法实现的过程中遇到了两个难题。

一是无法将整个表情识别算法模型及程序完整的集成到 NAO 机器人上。即使阅读了官方编程文档，也做出了一些尝试，但最终没能找到可以将模型及程序输入到机器人的办法。

二是 NAO 机器人编程只支持 python2.7，而本次实验所训练的表情识别算法是基于 python3.6 实现的，两个版本不兼容。

基于这两个难题短时间内无法解决，本课题最终选择通过 Python Socket 网络编程来实现 NAO 机器人利用改进的表情识别算法进行表情识别。具体实现思路如下：

分为两个程序。一个程序编写 NAO 机器人相关行动，充当“客户端”；另一个程序充当“服务端”，负责人脸的表情识别。思路流程图如图 4-3 所示。

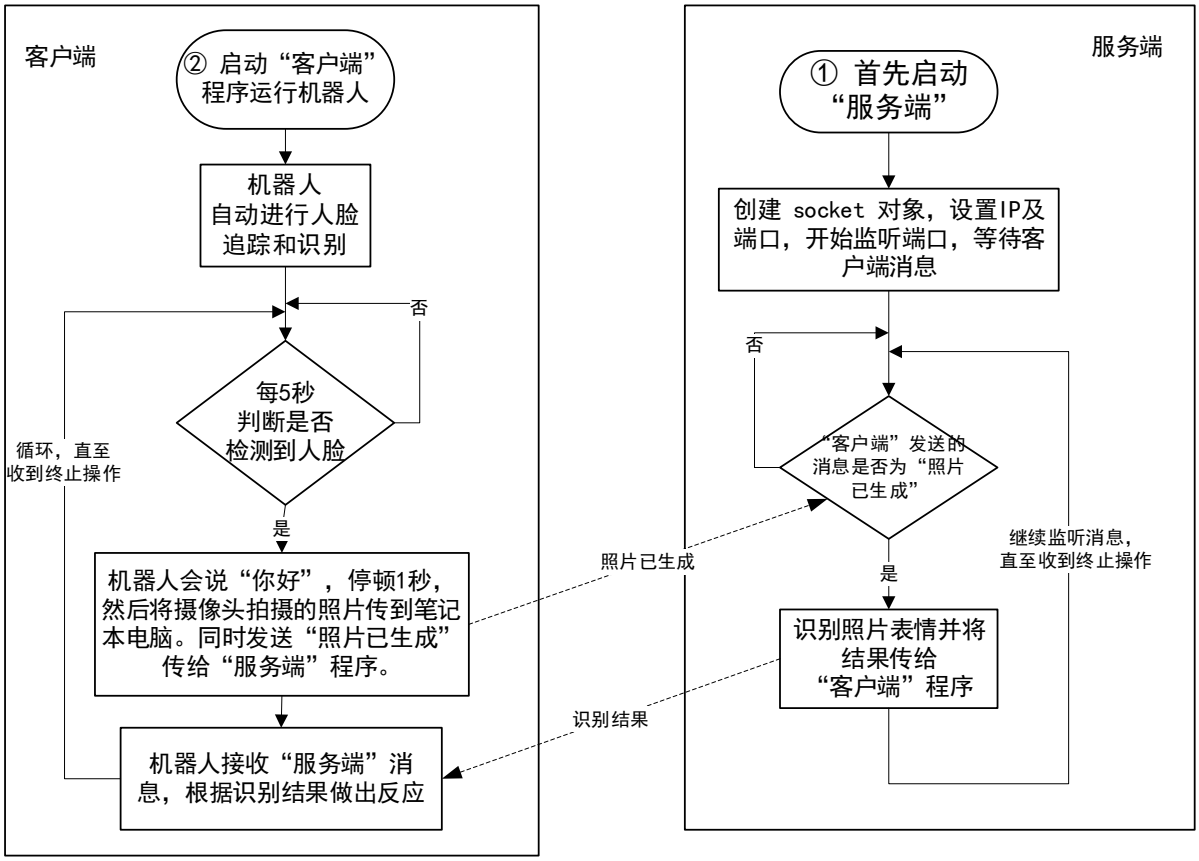


图 4-3 实现思路流程图

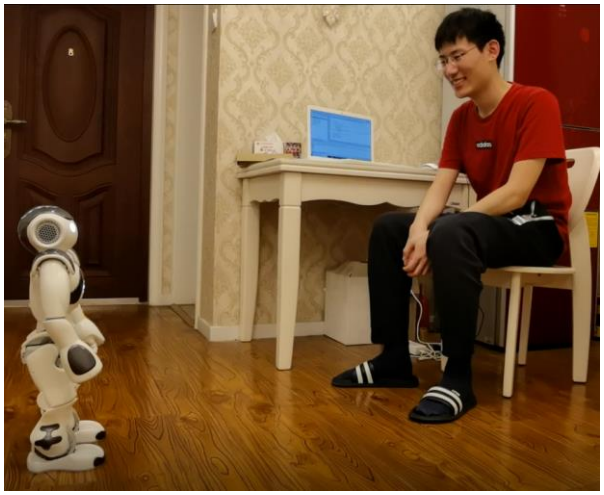
首先启动“服务端”，开始监听端口。然后运行“客户端”程序，启动机器人。机器人启动后，通过订阅 NAO 机器人官方提供的基础服务，会自动进行人脸追踪和识别。机器人每 5 秒进行一次判断，如果检测到人脸，机器人会说“你好”，并停顿 1 秒，给测试人员做出表情的时间，然后利用官网提供的摄像服务将机器人摄像头拍摄的照片传到笔记本电脑。同时“客户端”程序发送“照片已生成”信息传给“服务端”程序。“服务端”程序接收到消息后，将图片通过 Opencv 进行人脸检测，再将检测到的人脸图像载入训练好的算法（本课题改进的表情识别算法）模型，得出每种表情的预测概率。“服务端”取最大的预测概率作为表情识别的结果并将信息发送给“客户端”程序。“客户端”程序收到识别结果后便控制 NAO 机器人做出相应的语言及动作反应。如此循环直至停止“客户端”和“服务端”程序。

4.3 表情识别实际效果描述

启动程序后，NAO 机器人会站起来，头部进行人脸的跟踪。当机器人检测到人脸时，它会说“你好”，然后 1 秒后进行人脸表情识别。机器人根据识别不同的结果会做出不同的反应，具体如表 4-1 所示。图 4-4 展示了实际生活场景中 NAO 进行表情识别的情形，从图中可看出 NAO 在识别表情后做出了不同的动作。

表 4-1 NAO 机器人表情识别的效果描述

机器人识别出的表情	对应说的话	同时做出的动作
开心	“你看起来很开心，我也很开心”	抬起手臂，做出开心的动作
愤怒	“你看起来很生气，请冷静下来”	做出震惊和劝人冷静的动作
惊讶	“为什么你看起来这么惊讶？发生了什么”	半蹲并左右转头，做出惊讶的动作
恐惧	“你看起来很恐惧，怎么了”	做出困惑的动作
伤心	“别伤心，我可以让你高兴起来”	做出安抚和鼓舞人心的动作
厌恶	“你的表情看起来是厌恶，我说的对吗”	做出疑问的动作
中立	“你的表情是中立的”	做出肯定的动作
没有识别出表情	“对不起，我没能识别你的表情”	做出抱歉和遗憾的动作



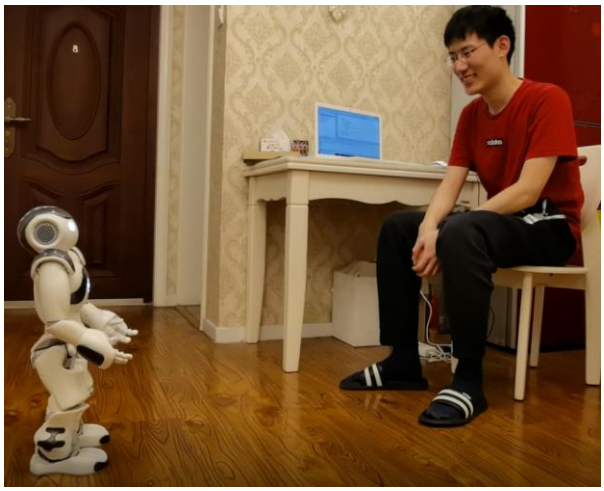


图 4-4 NAO 机器人进行表情识别的场景

第五章　总结与展望

本章对全文做出总结，概括和归纳出本论文的主要工作和研究内容，并对研究及实验中的不足做出反思，最后针对不足提出对未来研究的展望。

5.1　全文总结与反思

本课题主要进行的是基于深度学习的人脸表情识别算法的研究。全文总结如下：

1. 阐述了人脸表情识别算法研究的研究背景和意义，列举了国内外一些学者对表情识别算法的研究成果。

2. 介绍了实现表情识别的基本步骤，并列举一些传统表情识别算法。

3. 对卷积神经网络的基本原理做出讲解，分别说明输入层，卷积层，激活函数，池化层，全连接层和输出层各自的基本理论，以及卷积神经网络是如何实现图像分类和识别的。同时也对 CNN 的发展历程做了梳理。

4. 介绍了 Xception 和 mini_Xception 算法的基本框架，并在 mini_Xception 算法的基础上做出了改进，并对改进的算法进行训练和评估。实验结果表明，改进的算法对比 mini_Xception 算法在数据集上的准确率从 66.8%提升到了 69.0%。实际识别效果来看，改进的算法可很好的识别出高兴和中立两种表情，不稳定的识别出生气，恐惧和惊讶三种表情，较难识别出厌恶和悲伤两种表情，整体识别效果要优于 mini_Xception 算法。此外，还在改进的算法基础上加入两个残差模块，使数据集上的准确率进一步提升到 70%。但此额外改进的算法在实际识别效果上并不尽如人意。

5. 将改进的算法在 NAO 情感机器人上实现，使机器人实现人脸表情识别并做出相应反应。

同时，本次研究还存在许多的不足之处：

1. 对卷积神经网络的原理，深度学习领域及机器学习领域相关知识了解不深，导致无法提出一个创新的表情识别算法，只能在既有的算法基础上做出些许改动。

2. 改进的算法在实际表情识别中还远没有达到令人满意的效果。算法只能识别出较夸张的愤怒、恐惧和惊讶的表情，对这三种轻微的表情很难识别。对厌恶及伤心的表情则几乎不能识别出来。此外，识别过程中受光线影响较大。图 5-1 展示的是中立的表情在光线影响下被错误识别成开心。

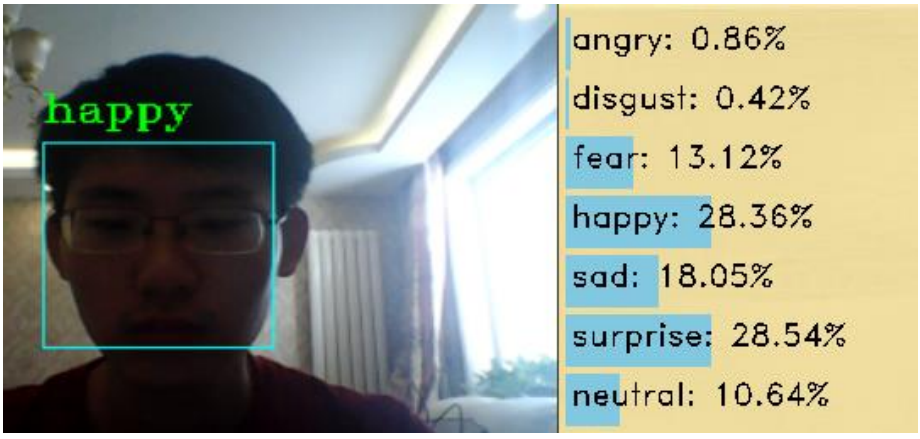


图 5-1 光线影响导致的算法表情识别错误

3. 算法没有完全集成到 NAO 机器人上。本课题没能将训练好的模型和全部程序融合进 NAO 机器人，NAO 机器人需要通过网络将图片和消息传到笔记本“服务端”，在笔记本“服务端”进行表情识别后再通过网络将识别结果传回机器人。NAO 机器人无法脱离笔记本“服务端”独立进行表情识别。

5.1 未来研究展望

针对本次研究存在的不足之处，本课题做出以下对未来研究的展望：

1. 表情数据集的改善。本次实验用到的 FER-2013 和 CK+数据集虽然在现有开源数据集里是较为出色的数据集，但在图片质量和数量上都有很大的改善空间。例如图片像素的大小可以提高，这样有利于提取到更丰富的表情特征；图片中人物的表情质量的更加规范，使之不存在错误、模糊、脸部过小或不完整的干扰训练的图片；数据集进一步扩充，多多益善。

2. 表情识别算法的进一步完善。当前实验所实现的表情识别算法在特征提取的针对性上还有很大提升空间，对微笑表情之外的其它表情变化不敏感。

3. 解决实际表情识别受光线等环境因素影响的问题。

4. 对 NAO 机器人编程还需进一步深入了解和实践。

结束语

本次毕业设计历时 5 个月完成。在此期间，我学习了计算机视觉相关课程，了解到卷积神经网络的原理，对图像识别等技术知识有了基本概念，并针对人脸表情识别进行了实践，初步踏入了深度学习领域的大门。越对深度学习领域进行研究，越感觉到该领域的广阔前景与复杂困难，深深意识到自己能力的欠缺与渺小。深度学习领域还有许多的知识和方向值得我去探索和研究。

我要特别感谢我的指导教师乔文豹老师，乔老师在毕设期间给了我许多教导和鼓励。乔老师不仅传授我专业的知识，仔细修改我的毕设论文，还教会我求知的精神与做人的准则，认真地帮助我修改写给教授的信。正是在乔老师悉心的帮助和耐心的督促下，我的毕业设计才能顺利完成。衷心祝愿乔老师在今后的工作和生活中一切顺利！

四年大学生活转瞬即逝，接受各位任课老师的传道受业解惑，在这里向所有教过我的老师表达感谢。我要还感谢舍友王鹏、李坤鹏、蒋煜楷、罗安平、徐英晨、毛勤峰、康岳一阳、巢文宇、蒲一民和李家根的陪伴，感谢他们在学习和生活中对我的种种帮助。

最后我要感谢父母在这大学四年期间对我的支持与关心。希望自己在今后的工作和学习中多多努力，早日报答父母的养育之恩。

参考文献

[1] 罗森林, 潘丽敏. 情感计算理论与技术[J]. 系统工程与电子技术, 2003, 25(7):905-909.
[2] Ekman P , Friesen W V . Constants across cultures in the face and emotion.[J]. Journal of Personality & Social Psychology, 1971, 17(2):124-129.
[3] Mase K . Recognition of facial expression from optical flow[J]. Ieice Transactions on Information & Systems, 1991, 74(10):3474-3483.
[4] Kotsia I , Pitas I . Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines[J]. IEEE Transactions on Image Processing, 2007, 16(1):172-187.
[5] Almaev T R , Valstar M F . Local Gabor Binary Patterns from Three Orthogonal Planes for Automatic Facial Expression Recognition[C]// Conference on Affective Computing & Intelligent Interaction. IEEE Computer Society, 2013.
[6] Tran D , Bourdev L , Fergus R , et al. Learning spatiotemporal features with 3D convolutional networks[C]// 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015.
[7] Lopes A T , Aguiar E D , Souza A F D , et al. Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order[J]. Pattern Recognition, 2016, 61:610-628.
[8] Jeong D, Kim BG, Dong SY. Deep Joint Spatiotemporal Network (DJSTN) for Efficient Facial Expression Recognition. Sensors (Basel). 2020;20(7):1936. Published 2020 Mar 30. doi:10.3390/s20071936
[9] 高文, 金辉. 面部表情图像的分析与识别[J]. 计算机学报, 1997(09):782-789.
[10] 郑文明. 基于核函数的判别分析研究[D]. 2004.
[11] 鹿麟, 吴伟国, 孟庆梅. 具有视觉及面部表情的仿人头像机器人系统设计与研制[J]. 机械设计, 2007, 024(007):20-24.
[12] Tang C , Zheng W , Yan J , et al. View-Independent Facial Action Unit Detection[C]// 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017.
[13] Chen Y , Wang T , Wu H , et al. A Fast and Accurate Multi-Model Facial Expression Recognition Method for Affective Intelligent Robots[C]// 2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR). IEEE, 2018.
[14] Zhang T , Zheng W , Cui Z , et al. Spatial-Temporal Recurrent Neural Network for Emotion Recognition[J]. IEEE Transactions on Cybernetics, 2018:1-9.
[15] Chen Y , Wu H , Wang T , et al. A Comparison of Methods of Facial Expression Recognition[C]// Wrc Symposium on Advanced Robotics & Automation. 2018.
[16] Chang C Y , Huang Y C , Yang C L . Personalized Facial Expression Recognition in Color Image[C]// Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on. IEEE Computer Society, 2010.
[17] Yang Y, Wang G, Kong H. Self-learning facial emotional feature selection based on rough set

theory[J]. Mathematical Problems in Engineering, 2009, 2009:1-16.
[18] Lyons M, Akamatsu S, Kamachi M, et al. Coding facial expressions with gabor wavelets[C]//Proceedings Third IEEE international conference on automatic face and gesture recognition. IEEE, 1998: 200-205.
[19] Kotsia I , Zafeiriou S , Pitas I . Texture and shape information fusion for facial expression and facial action unit recognition[J]. Pattern Recognition, 2008, 41(3):833-851.
[20] Donato G , Bartlett M S , Hager J C , et al. Classifying Facial Actions[J]. IEEE Transactions on Software Engineering, 1999, 21(10):974-989.
[21] Chu W S, De la Torre F, Cohn J F. Selective transfer machine for personalized facial action unit detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013: 3515-3522.
[22] Jiang B , Valstar M F , Pantic M . Action unit detection using sparse appearance descriptors in space-time video volumes[C]// Face and Gesture 2011. IEEE, 2011.
[23] Yang P . Boosting Coded Dynamic Features for Facial Action Units and Facial Expression Recognition[C]// 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA. IEEE, 2007.
[24] Uddin M Z , Kim T S , Song B C . AN OPTICAL FLOW FEATURE-BASED ROBUST FACIAL EXPRESSION RECOGNITION WITH HMM FROM VIDEO[J]. International Journal of Innovative Computing Information & Control, 2013, 9(4):1409-1421.
[25] Guo G, Dyer C R. Simultaneous feature selection and classifier training via linear programming: A case study for face expression recognition[C]//2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. IEEE, 2003, 03:1063-6919.
[26] Pantic M., Rothkrantz L.J.M.. Expert system for automatic analysis of facial expressions[J]. Image and Vision Computing, 2000, 2000(18):881–905.
[27] Hsu S C , Huang H H , Huang C L . Facial Expression Recognition for Human-Robot Interaction[J]. 2017 First IEEE International Conference on Robotic Computing (IRC), 2017:1-7.
[28] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).
[29] Russakovsky O , Deng J , Su H , et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3):211-252.
[30] Han J , Moraga C . The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning[C]// International Workshop on from Natural to Artificial Neural Computation. Springer-Verlag, 1995.
[31] Nair V , Hinton G E . Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair[C]// Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. Omnipress, 2010.
[32] LeCun Y, Jackel L D, Bottou L, et al. Learning algorithms for classification: A comparison on handwritten digit recognition[J]. Neural networks: the statistical mechanics perspective, 1995, 261: 276.

[33] Zeiler M D , Fergus R . Visualizing and Understanding Convolutional Networks[C]// European Conference on Computer Vision. Springer, Cham, 2014.
[34] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
[35] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.
[36] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.
[37] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.
[38] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//Thirty-first AAAI conference on artificial intelligence. 2017.
[39] Lin M, Chen Q, Yan S. Network in network[J]. arXiv preprint arXiv:1312.4400, 2013.
[40] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[41] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 1251-1258.
[42] Arriaga O, Valdenegro-Toro M, Plöger P. Real-time convolutional neural networks for emotion and gender classification[J]. arXiv preprint arXiv:1710.07557, 2017.
[43] Hinton G , Vinyals O , Dean J . Distilling the Knowledge in a Neural Network[J]. Computer Science, 2015, 14(7):38-39.
[44] Sifre L, Mallat S. Rigid-motion scattering for image classification[D]. Ph. D. thesis, 2014.
[45] Viola P, Jones M J. Robust real-time face detection[J]. International journal of computer vision, 2004, 57(2): 137-154.
[46] Goodfellow I J, Erhan D, Carrier P L, et al. Challenges in representation learning: A report on three machine learning contests[C]//International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, 2013: 117-124.
[47] Lucey P , Cohn J F , Kanade T , et al. The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression[C]// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. IEEE, 2010.
[48] Kingma D , Ba J . Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.