



察颜观色——基于Jetson Nano B01开发板的 表情识别监测系统

详细设计说明书

项目负责人

许福权

项目成员

李杨

指导老师

徐曼

2024年12月20日

目录

第 1 章 作品概述	1
1.1 项目背景	1
1.2 项目创新点	2
第 2 章 问题分析	3
2.1 问题来源	3
2.2 现有解决方案	3
2.3 本作品要解决的痛点问题	4
2.4 解决问题的思路	4
第 3 章 技术方案	6
3.1 总体介绍	6
3.2 详细介绍	6
3.2.1 界面构造	6
3.2.2 人脸识别技术	9
3.2.3 表情识别技术	10
3.2.4 疲惫检测	12
第 4 章 系统实现	13
4.1 软件设计实现	13
4.1.1 系统架构	13
4.1.2 技术选型	13
4.1.3 模块设计	13
4.2 用户界面	14
4.3 数据来源	14
4.4 数据训练	15
4.4.1 数据预处理	15
4.4.2 模型训练	15
4.4.3 模型优化	16
4.5 困难及解决方法	16
4.5.1 实时性能问题	16
4.5.2 准确性问题	16
4.5.3 表情时间记录问题	17
4.5.4 摄像头异常卡顿问题	17
4.6 系统部署	17
第 5 章 测试分析	19
5.1 数据来源	19
5.2 功能测试	19
5.2.1 人脸识别功能测试	19
5.2.2 表情识别功能测试	21
5.2.3 提醒设置验证测试	25
6.1 作品特色	28
6.2 应用推广	28

6.3 作品展望	29
----------------	----

第 1 章 作品概述

1.1 项目背景

随着信息技术的快速发展和互联网的普及，人们对电脑的依赖程度越来越高，无论是在工作中处理文件，还是在学习中获取知识，甚至是在娱乐中放松身心，电脑都扮演着重要角色，已经成为人们日常工作、学习和娱乐的重要工具。然而，长时间使用电脑不仅会带来身体不适，如眼睛疲劳、颈椎疼痛等，还会对电脑用户的心理健康产生影响，如焦虑、压力、疲惫等负面情绪。因此，对电脑用户情绪状态的监测和提醒显得尤为迫切和重要。

当前，市面上已有一些针对电脑用户健康的监测系统，但大多集中在身体健康方面，对于电脑用户情绪状态方面的监测和提醒相对较少。基于此背景，我们提出了“察‘颜’观色——基于 Jetson Nano B01 开发板的表情识别监测系统”。该系统结合了 Jetson Nano B01 开发板的边缘计算特性，利用其强大的计算能力和高效的 GPU 加速，结合卷积神经网络（CNN）等人工智能技术，实现对电脑用户面部情绪的实时识别和分析。

Jetson Nano B01 开发板不仅具备低功耗和高性能的优势，还能够支持高效的视频处理和深度学习模型的推理，适合进行情绪识别任务。此外，系统运行在 Ubuntu 18.04 LTS 操作系统上，Ubuntu 操作系统提供了稳定、高效的开发环境，特别适合深度学习和计算机视觉等任务的执行。Ubuntu 18.04 LTS 作为长期支持版本，提供了广泛的社区支持和开发工具，包括 Python、TensorFlow、PyTorch 等常用的机器学习和图像处理框架。在 Jetson Nano B01 开发板上运行 Ubuntu 18.04 LTS 操作系统，可以充分利用其强大的图形处理能力和深度学习加速功能，同时确保系统的稳定性和长期可用性。

系统通过利用开发板上的 USB 摄像头实时捕捉用户的面部数据，并借助 Jetson Nano 的强大处理能力进行图像分析，使得系统能够快速、精准地识别用户的情绪状态。通过该系统，用户能够实时查看自己的情绪变化，及时调节使用电脑的状态，从而改善学习、工作效率，促进心理健康，提供更加全面的情绪监测服务。

1.2 项目创新点

本项目的核心创新点在于 Jetson Nano B01 开发板的应用,这一平台为系统提供了强大的计算能力和低功耗特性。Jetson Nano 配备了高效的 GPU 加速,使得卷积神经网络(CNN)模型能够在本地实时推理,避免了数据传输延迟和对云计算资源的依赖,大大提高了情绪识别的实时性和精确度。此外,Jetson Nano B01 的灵活性和高效性能使得系统可以在边缘设备上进行情绪识别和反馈,尤其适合需要实时反馈的场景,如学习和工作中对情绪状态的管理。

系统通过卷积神经网络技术实现了对电脑用户面部情绪的实时监测和识别,使用户能够在使用电脑的过程中随时了解自己的情绪状态。通过 Jetson Nano 的 GPU 加速,系统能够高效处理摄像头实时传输的视频流,确保情绪识别的低延迟和高准确性。其次,系统可以通过分析情绪变化,生成用户的情绪报告,让用户深入了解自己的情绪波动,帮助其调整工作或学习状态。

本系统还具备休息提醒和疲惫提醒功能,能够在长时间使用电脑的过程中,根据用户的情绪状态及时提出休息建议。得益于 Jetson Nano 的强大处理能力,系统能够精准地判断用户的疲劳状态,避免用户因长时间坐在电脑前而忽视身体的疲劳信号,从而提高工作和学习效率,减少电脑使用带来的健康风险。

本项目通过结合 Jetson Nano B01 开发板的强大计算能力和卷积神经网络技术,为电脑用户提供了一种全新的情绪管理方式,具有极高的应用价值。无论是在学生、职场人士还是在其他长时间使用电脑的群体中,本系统都可以通过情绪监测与管理,帮助用户更高效、健康地工作和生活,展现出广阔的应用前景和市場价值。

第 2 章 问题分析

2.1 问题来源

《第 53 次中国互联网网络发展状况统计报告》指出截至 2023 年 12 月，我国网民使用台式电脑、笔记本电脑、电视和平板电脑上网的比例分别为 33.9%、30.3%、22.5%和 26.6%。随着信息时代的深入发展，电脑等电子产品已经深入人们的生活，并给我们每一个人带来不可替代的深远影响。不可否认，电脑已经成为大多数人工作、学习、娱乐的必不可少的工具，我们享受着其强大功能给我们带来的快捷和方便，但同时也承受着电脑使用所带来的一些不利影响，比如长时间的使用电脑不仅会带来身体不适，如眼睛疲劳、颈椎疼痛等，还会对电脑用户的心理健康产生影响，如焦虑、压力、疲惫等负面情绪。常见的电脑带来的症状，例如“鼠标手”、“电脑视觉综合征”、“晕屏症”等。另外，还有一些报道指出，使用电脑上网可能会给网民带来心理负面影响，因为他们可能极易接触到负面新闻。

因此，对电脑用户情绪状态的监测和提醒显得尤为迫切和重要，而市面并没有相应的产品。

2.2 现有解决方案

目前市面上有类似的产品，比如可以定时提醒电脑使用者工作或者休息的软件——番茄钟，这类软件简单的以时间间隔作为提醒，比如每 40 分钟提醒一次使用者需要休息。相比之下，我们的产品具有如下特点是其他类似的产品所不具备的：

（1）实时性、动态变化

我们的产品通过实时监测使用者的面部表情，直接得出当前使用者的心情和状态，可以做到实时反馈，提醒使用者等功能。

（2）实现个性化监测

每个人的精力、时间安排都是不同的，甚至每个人每一天的状态都是不同的，简单的以时间间隔为标准来提醒用户，显然是不够科学、准确、有效的，我们的产品则可以实现对每个用户个性化服务，根据他们的实际情况进行监测并提醒。

（3）可拓展性强、适用范围广

我们的产品，是基于识别用户面部表情的，可以据此拓展出许多额外的功能，另外我们这一产品的应用场景也是相当广泛的，也可以应用在许多不同的生产、生活场景，比如应用于学校，可以监测学生的状态，并不仅仅局限于电脑使用者。

2.3 本作品要解决的痛点问题

我们的产品让电脑用户能够可视化自己的情绪变化、调节自身用机状态、改善学习效率，对电脑用户进行多方面辅助，提供更全面的健康监测服务。

（1）解决用户由于不能合理安排电脑使用时间，导致使用效率低下的问题。

（2）减缓用户使用电脑，带来的身心困扰，帮助其健康使用电脑。

2.4 解决问题的思路

我们的产品通过对用户表情的识别，进而记录并分析用户的状态，可以帮助用户动态调整对电脑的使用，提高用户的效率。另外，也可以在发现用户有不好的情绪时，及时给与提醒和帮助，以达到避免或减少因使用电脑而对身心造成的负面影响。（如表 2-1 所示）

表2-1

数据集	人脸识别数据集
格式	图片与标签映射形成的.csv 文件
来源	从网上获取专门的数据集，另搜集一些表情图片加以补充
获取方式	购买网上的数据集文件
特点	数据量庞大，方便训练出更好的模型
规模	约 55742 张图片

第 3 章 技术方案

3.1 总体介绍

本系统核心技术包括人脸识别技术、表情识别技术，产品通过对电脑用户面部状态以及电脑使用时间的记录，构建一个可视化电脑用户状态监测系统。通过交互可视化界面达到信息反馈的效果，本系统基于统计图和提示文本框反应电脑用户一段时间的状态变化情况。总体技术设计路线如图 3-1 所示。

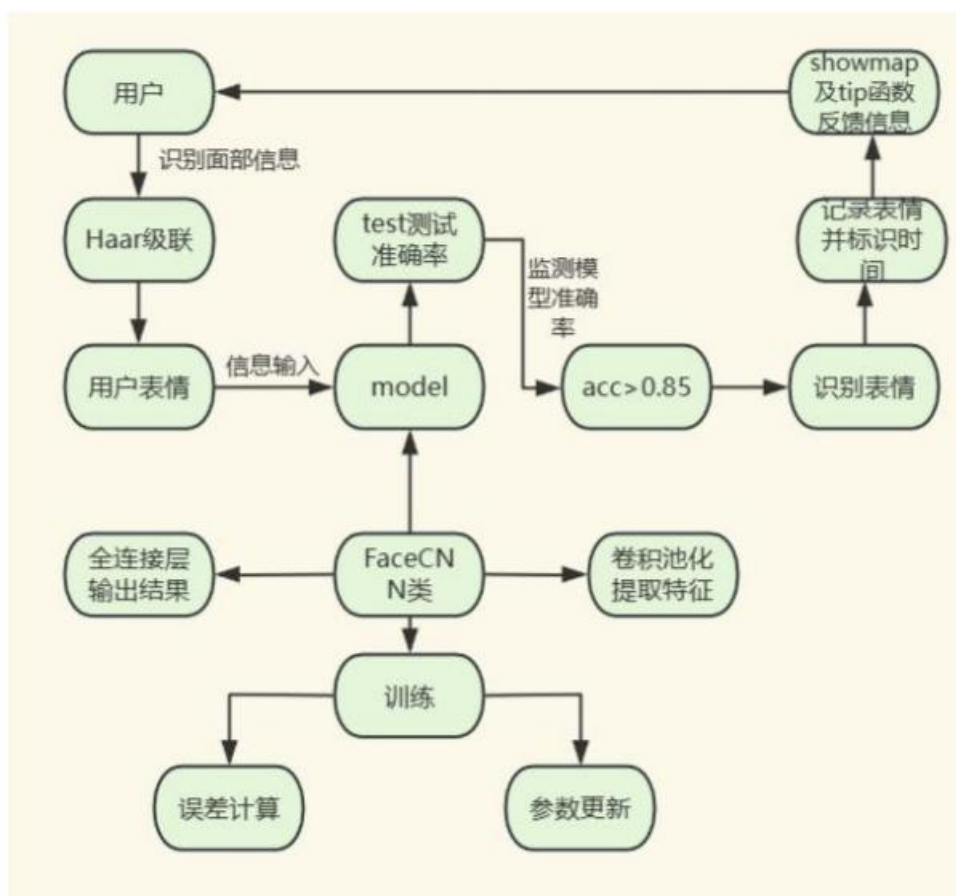


图 3-1 技术框架路线图

3.2 详细介绍

3.2.1 界面构造

在本系统的设计中，前端用户界面的开发采用了 PyQt5 技术框架。PyQt5 是一个跨平台的 GUI 开发库，它基于 Qt 库，可以用于开发各种桌面应用程序。PyQt5 提供了丰富的界面控件和图形功能，能够满足我们的实时情绪识别系统的

界面需求。

A. 界面布局设计

在界面布局方面，我们使用了 `QWidget` 作为基础窗口，并通过布局管理器（如 `QVBoxLayout` 和 `QHBoxLayout`）来安排窗口中的各个控件。界面的主要功能包括以下几个部分：

a. 视频显示区：用于显示通过摄像头捕获的实时图像流。在这个区域，用户可以看到自己的面部图像，以及系统实时捕捉的情绪状态。

b. 情绪识别结果显示区：该区域用来展示当前识别到的情绪类别，并实时更新。当系统检测到新的情绪时，显示区会刷新，告诉用户自己当前的情绪状态。

c. 操作按钮：通过设计多个按钮（如开始、停止等），用户可以方便地控制摄像头捕获、表情识别的开始和停止。这些按钮通过 `PyQt5` 的 `QPushButton` 控件实现，并与后台的操作逻辑通过事件绑定进行交互。

d. 图形化数据展示：为了帮助用户更直观地了解自己的情绪变化趋势，我们设计了图形化的数据展示区域。通过 `Matplotlib` 和 `PyQt5` 的集成，我们在界面中嵌入了绘图区域，用于展示用户的情绪变化曲线，如情绪识别的准确率、情绪类别变化等。

B. 控件与交互设计

为了提升用户体验和操作的便捷性，我们在界面中加入了一些常见的控件和交互方式：

a. `QPushButton`（按钮）：用于启动和停止摄像头捕捉、开始和停止表情识别等操作。每个按钮都与一个特定的事件关联，比如点击“开始”按钮时，系统会启动摄像头并开始进行实时的表情识别；点击“停止”按钮时，系统则停止识别。

b. `QLabel`（标签）：用于显示静态文本或动态更新的内容，如情绪识别结果、提示信息等。通过 `QLabel` 控件，用户能够实时看到自己的情绪状态，并且可以根据提示采取适当的操作。

c. `QComboBox`（下拉框）：用于选择不同的操作模式或调整摄像头的参数等。例如，用户可以通过下拉框选择不同的情绪分析模式，或者调整图像显示的分辨率、帧率等参数。

d.QProgressBar（进度条）：用于显示系统处理的进度，例如情绪识别的过程或图像处理的状态，帮助用户更好地了解当前系统的工作状态。

C.功能实现与事件绑定

通过 PyQt5 的信号和槽机制，我们将界面控件与具体功能逻辑进行了绑定。例如，当用户点击“开始”按钮时，PyQt5 会触发一个信号，后端系统接收到该信号后开始运行摄像头获取图像并进行表情识别，识别结果随即通过 QLabel 更新到界面上。系统中的图像显示区域会实时显示摄像头捕捉到的图像，情绪识别结果也会实时反馈给用户。

此外，我们还实现了事件的异步处理。摄像头捕捉和情绪识别过程是在独立的线程中进行的，这样可以避免界面在处理图像时被阻塞，从而确保用户界面的流畅性。QThread 被用来处理摄像头图像捕捉和表情识别的任务，而界面的显示部分则由主线程负责更新。

D.数据可视化与反馈

为了帮助用户更直观地了解自己的情绪变化，我们使用 Matplotlib 与 PyQt5 结合实现了情绪变化的可视化图表。通过动态绘制用户在不同时间点的情绪状态变化，我们能够生成情绪识别的趋势图，用户可以清晰地看到自己在一段时间内的情绪波动情况。

此外，情绪识别的准确率和情绪类别变化也通过图表形式展示，让用户更加直观地理解系统的工作状态和自己的情绪变化。这不仅提升了系统的用户体验，也帮助用户更好地调整使用电脑的状态，避免过度疲劳或产生不良情绪。

E.界面样式与用户体验

在界面设计上，我们注重了样式的简洁和直观，使用了浅色调的背景色和清晰的按钮布局，避免界面过于复杂和冗余。按钮和标签的设计符合用户操作习惯，易于理解和使用，确保用户能够轻松上手。

总的来说，界面设计的目标是提供一个简单易用、直观友好的操作平台，使用户能够快速启动系统并实时监测自己的情绪状态。通过合理的控件布局、事件绑定和数据可视化，我们成功构建了一个高效且易于使用的前端界面。

3.2.2 人脸识别技术

本系统的人脸识别技术采用了 OpenCV 中的 Haar 级联技术，该技术能够高效地检测图像中的人脸区域，包括面部、眼睛和嘴部等特征。Haar 级联分类器基于 Haar 特征进行人脸检测，其核心思想是通过计算图像中的多个特征差异，快速判断是否包含人脸信息。具体来说，Haar 特征主要包括边缘特征、线性特征、中心特征和对角线特征。这些特征通过图像中相邻区域的像素值差异来计算，构成了每个 Haar 特征的基础。每个 Haar 特征是由两个或多个矩形区域组成，通过计算这些矩形区域的像素和差值，形成一个特征值。当这个特征值超过预设的阈值时，说明图像中存在特定的人脸特征。

然而，单一的 Haar 特征并不足以准确地识别复杂的图像模式，因为它是一个弱分类器。为了提高识别准确性，我们采用了 AdaBoost 算法（如图 3-2 所示）。AdaBoost 算法通过训练多个弱分类器，并将它们组合成一个强分类器，从而显著提升分类性能。每个弱分类器根据其在训练集中的表现分配不同的权重，系统会通过迭代不断更新这些权重，优化每个分类器的性能。通过这种方式，AdaBoost 将多个不太准确的分类器组合成一个更强的、准确的分类器，进而提高了人脸识别的精度。

为了进一步提高计算效率和加速识别过程，我们使用了积分图技术。积分图是一种加速图像特征计算的方法，通过预先计算图像中每个位置的像素和，能够在遍历图像时快速获得任何矩形区域的像素和，大幅提升了计算速度。通过这种方式，我们避免了重复计算每个像素点的值，提高了特征提取的速度。

最终，通过组合多个强分类器并利用积分图加速计算，OpenCV 的 Haar 级联分类器能够在不同光照和环境条件下，快速且准确地定位人脸特征区域。这使得本系统能够实时捕捉并识别用户的面部表情，为后续的情绪分析和监测提供了稳定可靠的基础。

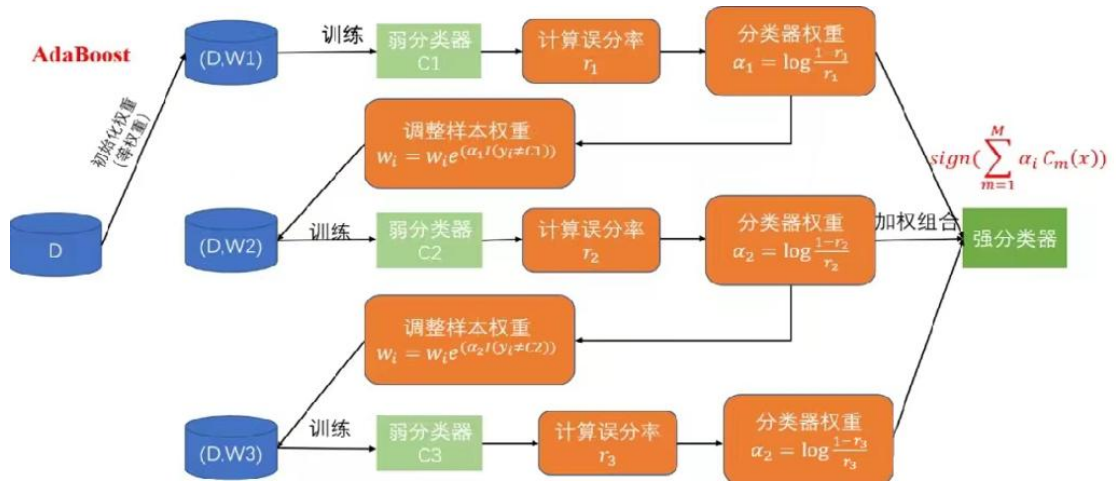


图 3-2 OpenCV Harr 级联过程

3.2.3 表情识别技术

在能够进行表情识别之前，本系统需要训练出一个准确率较高的模型来进行表情检测。在训练部分，我们将 55338 图片按照 8: 2 分为训练集和测试集，其中 98% 的图片为中国人脸，符合国人面部特征。在训练之前，需要采取一些预处理方法，我们使用高斯模糊等处理方法添加噪声，提高神经网络的鲁棒性和泛化能力，增强数据集。同时建立映射表为每一张图片都设置一个标签，总共有 8 类表情，如图 3-3 所示，包括生气、中立、开心、疲惫、厌恶、惊喜、伤心、害怕，故通过 0-7 分别表示每一种表情，具体映射关系如图 3-4 所示，方便每次训练过程读取图片数据。

表情识别技术实现的核心在于卷积神经网络（CNN）的构建，整体网络结构由卷积层、池化层以及全连接层组成。

卷积层的核心思想是通过卷积核在图像上滑动来提取局部特征，卷积核在图像上按步长进行滑动，通过与对应位置的像素进行加权求和等计算来提取局部特征。在整个图像上滑动卷积核的过程中，同一个卷积核的参数是固定不变的。大大减少了网络需要学习的参数数量，使得网络可以用较少的参数去学习图像中不同位置的相似特征模式。

池化层使用了最大池化，以一定大小的窗口（本文是 2*2）在特征图上滑动，选取每个窗口内的最大值作为输出值。在保留主要特征信息的同时减少数据量（特征图尺寸变小），起到下采样的作用，让后续的计算更加高效，也能在一定

程度上增强网络对图像平移、缩放等形变的鲁棒性。在不断压缩数据规模同时保留关键特征信息。

经过前面的卷积、池化等操作后，数据需要通过全连接层来进行最终的整合和分类决策。首先将前面卷积池化后得到的多维特征图数据进行扁平化处理（展平成一维向量），然后输入到全连接层中。全连接层里每个神经元与前一层的所有神经元都有连接，通过不断调整神经元之间连接的权重（在训练过程中学习得到），将前面提取到的特征综合起来，逐步映射到最终的分类类别空间中。本项目是进行 8 分类，因此模型最终输出的维度为 8。

模型具体实现是采用了三次卷积池化操作（如图 3-5 所示），设置好输入通道数、输出通道数、卷积核大小、步长等参数，提取特征的有效信息并且保证特征的鲁棒性。将提取到的特征输入至全连接层，利用特征信息（包括眼睛、鼻子、眉毛、嘴巴等）进行分类。通过多轮循环，计算交叉熵损失率，改变学习率，通过 SGD 优化器进行优化，不断调整相关参数，使模型的检测准确率不断提高，准确率接近 90%，大于 85%。使用误差反向传播不断减小误差，将当前的误差分摊给每个单元，通过每个单元的误差来修正相关权值。



图 3-3 数据集

图片名称	图片标签
angry\41. jpg	0
neutral\2638. jpg	1
happy\51667. jpg	2
tired\2140. jpg	3
disgust\12828. jpg	4
surprise\22439. jpg	5
sad\13226. jpg	6
fear\7609. jpg	7

图 3-4 表情映射表

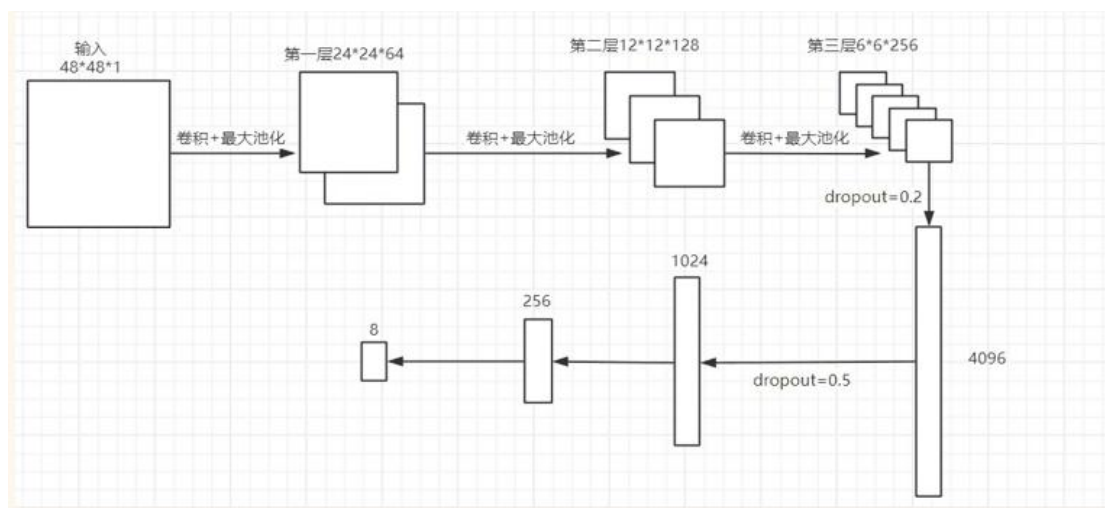


图 3-5 卷积池化过程

3.2.4 疲惫检测

虽然模型能够识别出 8 种不同的表情，但是毕竟对于每一帧而言，针对表情的细微变化，模型都有可能会识别出相关表情，如果仅仅根据一帧的表情来进行判断用户是否处于疲惫状态，显然是不合理的。针对此问题，本系统采用一个模拟队列的数组来判断是否处于疲惫，通过设置一个大小为 20 的固定数组。每次检测一帧后，就进入队列，队列充满后，就将队首元素移出，并统计疲惫状态的次数，如果疲惫的次数/20 大于设定的阈值（如 0.8），我们就认为当前状态是疲惫的。

本系统在实现了上述功能后，就能够面向用户进行使用了。通过图形界面的文本框及相关组件提示，让用户自由地选择相关功能。在打开摄像头后，可以实现对用户的状态进行检测，根据用户的选择进行相关的提示，完成信息反馈。

第 4 章 系统实现

4.1 软件设计实现

4.1.1 系统架构

系统采用单体架构设计，即前端界面与后端逻辑集成在一个应用程序中。这种设计简化了开发流程，便于维护和部署。

4.1.2 技术选型

开发语言：Python，具有丰富的库支持和简洁的语法，适合快速开发。

GUI 框架：PyQt5，用于构建跨平台的桌面应用程序界面。

图像处理库：OpenCV，用于图像捕获、面部检测和特征提取。

深度学习框架：PyTorch，用于构建和训练情绪识别的深度学习模型。

4.1.3 模块设计

系统主要分为以下几个模块：

数据采集模块：负责从连接的摄像头实时获取视频流，并对其进行预处理，如格式转换和大小调整。

面部检测与特征提取模块：利用 OpenCV 库进行面部检测，定位图像中的人脸区域，并从中提取关键特征。

情绪识别模块：采用 PyTorch 构建的深度学习模型，接收面部特征输入，输出情绪分类结果。

提醒模块：根据情绪识别结果和用户设定的参数，当检测到用户情绪异常或长时间处于某一状态时，触发休息提醒。

用户界面模块：提供用户交互界面，包括实时情绪展示、设置调整、历史数据查询等功能。

系统配置模块：允许用户或管理员配置系统参数，如开关信息、提醒设置等。

4.2 用户界面

用户界面采用 PyQt5 开发，提供直观的操作界面。界面主要包括：

监测页面：展示实时视频流，并显示情绪识别结果。

设置页面：允许用户配置休息提醒参数，如休息间隔时间、番茄工作法参数等。

报告页面：展示情绪识别的历史数据和统计图表。

4.3 数据来源

实时视频流：实时视频流数据通过客户端程序直接从用户电脑的摄像头获取。客户端使用 OpenCV 库来捕获视频流，并对其进行必要的图像处理，如灰度化、降噪等，以提高面部检测和情绪识别的准确性。

系统的训练数据集来源于公开的面部表情数据集，用于深度学习模型的训练。本项目在训练时使用了以下数据集：

FER-2013 数据集：一个广泛使用的面部表情识别数据集，包含多种情绪类别。

CK+数据集：提供了在不同情境下捕捉的面部表情，具有较高的多样性。

亚洲人脸数据集：特别针对亚洲人群的面部特征进行收集的数据集。

疲惫表情数据集：专门为识别用户疲劳状态收集的数据集，增强了模型识别疲惫表情的能力。

在这些数据集中，针对亚洲人脸数据集进行了重点训练，因为我们的目标用户主要是国内人群。通过细致的特征工程和模型调优，亚洲人脸数据集上训练得到的模型在验证集上达到了 90% 的准确度，是所有数据集中最高的。此外，加入疲惫表情数据集后，模型在识别用户疲劳状态方面也表现出色，进一步提升了系统的实用性和准确性。

综合考虑，项目最终选择使用基于亚洲人脸数据集和疲惫表情数据集共同训练得到的模型作为表情识别的核心模型。这一决策确保了系统在识别亚洲用户面部表情，尤其是在识别疲劳表情时的高准确性和可靠性，从而更好地满足用户需求并提供及时的休息提醒。

4.4 数据训练

4.4.1 数据预处理

在数据训练阶段之前，我们对数据集进行了必要的预处理以提升模型学习效率和性能。具体步骤包括：

归一化：将图像的像素值从[0, 255]线性映射到[0, 1]，以便于模型处理。

数据增强：通过对训练图像进行旋转、缩放、剪切和翻转等操作，增加数据多样性，提高模型的泛化能力。

面部特征点标注：使用面部特征点标注数据，辅助模型更准确地捕捉面部表情特征。

4.4.2 模型训练

我们采用了卷积神经网络（CNN）作为情绪分类任务的模型，具体实现细节如下：

模型架构：构建了一个包含多个卷积层、池化层、归一化层和全连接层的 CNN 模型。模型初始化中，我们使用了 `gaussian_weights_init` 函数对卷积层的权重进行高斯初始化。

激活函数：选用了 RReLU（随机 ReLU）作为激活函数，以加速模型训练并提高性能。

损失函数：使用交叉熵损失函数来衡量预测值与真实标签之间的差异。

优化器：使用 SGD（随机梯度下降）优化器，并添加了权重衰减（L2 正则化）来防止过拟合。

在训练模型的代码中，我们定义了 `FaceCNN` 类来构建模型，并实现了 `train` 函数来训练模型。训练过程中，我们使用了 `FaceDataset` 类来加载和处理数据，并通过 `DataLoader` 来批量加载数据。

4.4.3 模型优化

为进一步提升模型性能，我们采取了以下优化措施：

超参数调整：通过实验确定了学习率、批量大小、迭代轮数等超参数的最佳值。

模型保存：在训练完成后，我们将训练好的模型保存为 `pkl` 文件，以便后续的模式部署和应用。

验证集评估：在训练过程中，我们在验证集上评估模型性能，以监控模型的泛化能力。

在训练过程中，我们还记录了每个 `epoch` 后的损失率，并在特定 `epoch`（每 5 个 `epoch`）评估了训练集和验证集上的准确率，以此来监控模型训练进度和性能。

4.5 困难及解决方法

在系统开发过程中，遇到了以下困难及解决方法：

4.5.1 实时性能问题

问题描述：初始版本的系统在处理视频画面时，如果未能检测到人脸，会停止运行，导致视频流处理中断，程序直接停止。

解决方法：我们改进了人脸检测算法，确保即使在人脸不在标准位置或部分被遮挡的情况下也能稳定运行。此外，通过优化代码逻辑和降低模型的复杂度，提高了系统处理速度，确保了视频流的连续性和实时性。

4.5.2 准确性问题

问题描述：模型在某些表情上的识别准确率不高，特别是在表情微妙或不清晰的情况下。

解决方法：扩充了训练数据集，特别是那些难以识别的表情类别，以增加模型的多样性和鲁棒性。同时，采用了数据增强技术，如图像旋转、缩放、剪切和翻转等，来提高模型对不同表情变化的识别能力。

4.5.3 表情时间记录问题

问题描述：系统在记录表情持续时间时存在问题，导致无法准确反映用户的实时情绪状态。

解决方法：修正了时间记录的逻辑，确保系统能够准确跟踪并记录每种表情的持续时间。此外，增加了异常检测机制，当系统检测到时间记录异常时，会自动进行调整或提醒用户检查摄像头和面部定位设置。

4.5.4 摄像头异常卡顿问题

问题描述：在使用新的 USB 摄像头进行实时图像捕捉时，出现了图像延迟、卡顿和不稳定的问题。

解决办法：对摄像头的分辨率和帧率进行了优化调整。首先，通过调低分辨率和帧率（调整为 640x480 和 15 帧每秒），使得数据量减小，避免处理过载。其次，在 Jetson Nano 中配置了适合其硬件性能的摄像头参数，使用了 OpenCV 库中的内置函数对图像进行预处理，如去噪声、调整对比度等。

4.6 系统部署

系统的成功部署是确保其能够稳定运行并满足用户需求的关键环节。在本项目中，系统部署的过程主要涉及硬件配置、软件环境的搭建、功能测试以及维护和优化等方面。具体步骤如下：

1. 硬件平台配置

本系统部署在 Jetson Nano B01 开发板上，利用其强大的计算能力和高效的 GPU 加速进行面部情绪识别。首先，我们完成了 Jetson Nano B01 开发板的硬件组装和系统镜像的烧录，确保开发板的基本硬件和操作系统正常工作。接着，部署了与 Jetson Nano B01 兼容的 USB 摄像头，并确保摄像头能够被系统正确识别并开始工作。为了满足系统的实时性要求，还对摄像头进行了参数配置，优化了分辨率和帧率设置，确保图像数据流畅传输。

2. 软件环境配置

在 Jetson Nano 上，我们配置了 Python 环境，并安装了必需的深度学习框架与依赖库，如 PyTorch（用于情绪识别模型的训练与推理）、OpenCV（用于

图像处理与人脸识别）、NumPy、Matplotlib 等。这些库的安装确保了模型能够高效运行并处理来自摄像头的视频流。同时，由于 Jetson Nano 的计算能力相对较强，我们针对该平台进行了优化，利用其 GPU 加速能力提升了卷积神经网络（CNN）模型的推理速度，从而提高了实时性。

3. 系统功能验证与性能测试

在硬件和软件配置完成后，我们对系统进行了全面的功能验证与性能测试。测试内容包括：

硬件兼容性测试：确保 Jetson Nano B01 开发板与 USB 摄像头的兼容性，并验证摄像头在不同光照条件下的图像捕获能力。

性能与稳定性测试：在不同的操作环境和硬件配置下，测试系统的稳定性和实时性，确保面部情绪识别过程中的数据流畅性和准确性。

算法优化测试：验证情绪识别模型在 Jetson Nano 平台上的推理速度和精度，确保系统能够在合理的时间内准确识别用户的情绪。

4. 部署后的维护与优化

在系统部署并投入使用后，我们将持续进行系统优化和维护，收集用户反馈并根据实际需求进行调整。例如，可能会针对特定环境下的性能问题进行进一步优化，或加入新的功能来提高系统的适用性和用户体验。定期的系统更新和维护确保系统在长期运行中保持良好的性能，并能及时适应新需求。

通过这些步骤，我们确保了系统在 Jetson Nano B01 开发板上的顺利运行，为用户提供稳定、可靠的情绪监测和健康管理服务。同时，本系统也为未来的扩展与优化奠定了坚实的基础。

第 5 章 测试分析

5.1 数据来源

本系统的数据主要来源于电脑用户的面部表情，通过摄像头实时捕捉并记录用户的面部状态信息。数据规模方面，我们在开发和测试阶段收集了大量的面部表情数据，这些数据包括各种情绪状态如高兴、惊讶、愤怒、厌恶、悲伤、害怕、疲惫等。为了确保数据的多样性和代表性，数据采集覆盖了不同性别、年龄的用户。

5.2 功能测试

功能测试是确保系统各项功能正常运行的关键步骤。我们的功能测试过程包括以下具体内容：

5.2.1 人脸识别功能测试

测试系统是否能够准确地捕捉并识别用户的面部轮廓。通过使用不同光照条件和背景环境，验证系统在各种条件下的识别能力。

A. 正常光照环境（如图 5-1 所示）

测试方法：在光线均匀、适中的环境中测试系统的识别能力。测试场景包括白天室内自然光照、均匀人工照明等。

系统反应：系统在正常光照下表现最佳，能够迅速准确地识别用户的面部。



图 5-1 正常光照环境

B. 强光环境（如图 5-2 所示）

测试方法：在强光下测试系统的识别能力。测试场景包括室内强光照明等。

系统反应：系统能够在这些条件下能够快速定位并识别出用户的面部轮廓。



图 5-2 强光环境

C. 弱光环境（如图 5-3 所示）

测试方法：在光线较弱的环境中测试系统的识别能力。测试场景包括傍晚室内光线不足、夜晚仅有微弱灯光等。

系统反应：系统在弱光条件下的识别能力略有下降，但仍能有效定位面部特征。



图 5-3 弱光环境

5.2.2 表情识别功能测试

表情识别功能测试的主要目的是验证系统在不同条件下是否能够准确地识别用户的表情状态。测试过程中，用户表现不同的情绪（如高兴、惊讶、愤怒等），系统将识别结果实时显示在用户界面上，包括识别出的表情类别和对应的置信度值。通过观察系统是否能够正确分类并显示相应的表情状态以验证系统表情识别的准确性。

A. 高兴（如图 5-4 所示）

用户表现：用户露出微笑或大笑表情。



图 5-4 高兴表情

系统反应：系统识别为“高兴”，置信度为 97.63%。

B. 惊讶（如图 5-5 所示）

用户表现：用户表现出瞪大眼睛、张大嘴巴的惊讶表情。

系统反应：系统识别为“惊讶”，置信度为 99.94%。



图 5-5 惊讶表情

C. 愤怒（如图 5-6 所示）

用户表现：用户皱眉、紧闭嘴巴或怒目而视。

系统反应：系统识别为“愤怒”，置信度为 97.35%。



图 5-6 愤怒表情

D. 悲伤（如图 5-7 所示）

用户表现：用户表现出垂下嘴角、眼神黯淡的表情。

系统反应：系统识别为“悲伤”，置信度为 65.99%。



图 5-7 悲伤表情

E. 害怕（如图 5-8 所示）

用户表现：用户表现出眉毛上扬、眼睛睁大的恐惧表情。

系统反应：系统识别为“害怕”，置信度为 69.04%。



图 5-8 害怕表情

F. 厌恶（如图 5-9 所示）

用户表现：用户表现出皱鼻、嘴巴微张的厌恶表情。

系统反应：系统识别为“厌恶”，置信度为 97.65%。



图 5-9 厌恶表情

G. 疲惫（如图 5-10 所示）

用户表现：用户表现出闭眼、打哈欠的疲惫表情。

系统反应：系统识别为“疲惫”，置信度为 95.33%。



图 5-10 疲惫表情

H. 中立（如图 5-11 所示）

用户表现：用户表现出正常平静的表情。

系统反应：系统识别为“中立”，置信度为 97.76%。



图 5-11 中立表情

5.2.3 提醒设置验证测试

提醒设置功能是系统的重要辅助功能之一，它能够帮助用户管理工作和休息时间，提高工作效率，并且有助于保护用户的身体健康。提醒设置功能包括休息提醒、疲惫提醒以及退出设置三大部分。在测试提醒设置功能时，将分别验证这三个部分的功能及其与设置文件的交互。

A. 休息提醒功能测试：

测试目标：确保系统能够在设定的休息时间结束时提醒用户休息。

测试方法：设置休息时间，启动休息提醒功能。模拟休息时间结束，验证系统是否能够准时发送休息提醒。

系统反应：能够在设定的休息时间结束时发送提醒信息，提示用户休息，能够在设定时间结束时提醒用户休息。（如图 5-12 所示）



图 5-12 休息提醒

开启工作提醒设置后，系统能够在设定时间结束时提醒用户结束休息时间开始工作。（如图 5-13 所示）



图 5-13 工作提醒

B. 疲惫提醒功能测试：（如图 5-14 所示）

测试目标：确保系统能够在用户疲惫时进行提醒。

测试方法：设置疲惫提醒功能开启。模拟用户疲惫情况，验证系统是否能够及时发送疲惫提醒。

系统反应：能够在用户出现疲惫情况时发送提醒信息，提醒用户注意休息。



图 5-14 疲惫提醒

3. 退出设置保存功能测试:

测试目标: 确保系统能够将用户的设置信息正确保存到配置文件中。

测试方法: 对各项提醒设置进行修改。保存设置并退出系统, 检查配置文件是否正确保存了用户的设置信息。

系统反应: 能够将用户对提醒设置的修改正确保存到配置文件中, 下次启动时能够读取到用户的设置。(如图 5-15, 5-16 所示)



图 5-15 提醒设置

第 6 章 作品总结

6.1 作品特色

“察‘颜’观色——基于 Jetson Nano B01 开发板的表情识别监测系统”是一项创新性的项目，旨在利用 Jetson Nano B01 开发板和人工智能技术，帮助电脑用户实时管理情绪、调节状态、提升学习效率。通过将先进的深度学习技术与边缘计算平台相结合，本系统能够对电脑用户的面部情绪进行高效、实时的识别和分析，为用户提供情绪报告、休息提醒以及疲惫提示等多方面辅助功能，从而实现对用户的全面关怀。

6.2 应用推广

作为本项目的应用拓展，我们通过将系统部署在 Jetson Nano B01 开发板上，实现了系统的高效运作和灵活部署。Jetson Nano B01 开发板为本系统提供了强大的计算能力和低功耗的优势，特别适合在边缘设备上实时面部情绪识别和分析。这一部署不仅使系统的应用范围得到拓展，而且通过与 Jetson Nano 的深度集成，我们能够为用户提供更加个性化和优化的体验。

在 Jetson Nano 平台上，我们完成了系统的适配、用户界面的优化以及功能集成，使得系统能够更好地运行于嵌入式平台，并保证高效的性能和流畅的操作体验。这一部署不仅展现了项目在硬件和软件结合上的创新，也为未来的功能扩展和商业化提供了坚实的技术基础。

通过将 Jetson Nano B01 开发板作为硬件平台支持，我们的系统可以在多个领域和行业中得到广泛应用，具体包括：

A. 学生成长监护系统：通过对学生情绪和疲劳状态的实时监测，可以帮助教师和家长关注学生在校期间的学习状态和心理健康。该系统能够识别学生的情绪变化，及时提醒教师进行个性化辅导，提升学生的学习效果和身心健康。

B. 疲劳驾驶监测系统：系统可以应用于汽车驾驶员的情绪监控，实时识别驾驶员的疲劳状态。当系统检测到驾驶员疲劳时，可以发出提醒，避免疲劳驾驶带来的安全隐患。通过 Jetson Nano 强大的实时处理能力，系统能够在车载环境中提供高效的情绪监测服务。

C. 工人情绪与健康监测系统：本系统还可以扩展应用于生产线工人的情绪

和身体健康监控，实时分析工人疲劳状态，防止过度劳累。通过面部表情的识别与分析，系统可以帮助企业实时了解员工的健康状态，避免因情绪不佳或疲劳引起的生产事故。

D. 养老院心理健康监测系统：该系统可以部署在养老院中，帮助护理人员及时了解老年人的情绪变化，识别潜在的抑郁或焦虑等心理问题。通过面部识别技术，系统能够监测老人情绪波动并提供个性化的心理健康管理，提升老年人的生活质量。

通过 Jetson Nano B01 开发板的硬件支持，本系统的应用场景不仅局限于电脑用户，还可以拓展到教育、交通、工业、养老等多个领域，为不同群体提供情绪监控和健康管理服务。随着系统功能的不断优化和技术的成熟，我们相信该项目将具有广阔的市场前景，并能够为更多行业提供创新性的解决方案。

6.3 作品展望

通过本项目的实施，我们成功地解决了电脑用户情绪管理和学习效率提升的问题，为用户提供了全面的情绪监控和健康管理服务。结合 Jetson Nano B01 开发板的强大计算能力和本系统的实时面部情绪识别功能，我们为电脑用户提供了一种全新的情绪管理方式。然而，项目仍有进一步提升的空间，未来将在技术、功能和市场拓展等方面进行多维度的优化。

在技术提升方面，我们将继续优化情绪识别算法，进一步提高系统的识别准确性和实时性。尤其是在 Jetson Nano B01 开发板的硬件加速下，我们将进一步优化卷积神经网络（CNN）模型的推理效率，以适应更多复杂的实时应用场景。同时，我们计划开发移动端应用和跨平台适配，使得该系统能够支持更多用户设备，如智能手机、平板电脑等，从而满足更多用户的需求，拓宽应用场景。

在功能拓展方面，我们计划在现有情绪识别功能的基础上增加个性化情绪调节建议，根据用户的情绪状态提供具体的放松或激励建议。通过与心理健康机构合作，结合情绪监测数据，我们还将提供在线心理咨询和支持服务，进一步帮助用户有效管理情绪和心理健康，提升用户的整体幸福感与工作效率。

在数据分析与挖掘方面，我们将借助大数据技术分析用户的情绪变化趋势，挖掘群体层面的情绪模式，从而为个性化情绪管理提供更多深度的建议。未来，

我们还将引入机器学习和人工智能技术，根据用户的长期使用数据进行深度优化，使得系统能根据历史数据进行智能推测和精准提醒，逐步提升用户体验和准确性。

在市场拓展方面，我们的目标是将系统推广到国际市场，并根据不同地区的文化差异和用户需求进行本地化调整。同时，我们将深度挖掘各个行业的应用场景，与更多行业领军企业合作，拓展系统的应用领域。例如，除了目前的教育、交通和健康领域，我们还将探索金融、医疗等新兴领域的应用，以提供更广泛的情绪监控和健康管理服务。

通过持续创新和优化，我们相信本项目将能在未来实现更广泛的应用，并为全球用户提供更加智能、个性化的情绪管理工具。随着技术的不断发展，系统的社会影响力将逐渐扩大，成为用户情绪管理和身心健康的重要助手，推动社会整体心理健康水平的提升。