



浙江工业大学

本科毕业设计（论文、创作）

题目：程序员的疲劳状态检测方法

作者姓名 孙悦铖

指导教师 赵云波教授

专业班级 自动化 1604

学 院 信息工程学院

提交日期 2020年6月20日

浙江工业大学本科毕业设计论文

程序员的疲劳状态检测方法

作者姓名：孙悦铖

指导教师：赵云波教授

浙江工业大学信息工程学院

2020年6月

**Dissertation Submitted to Zhejiang University of Technology
for the Degree of Bachelor**

Fatigue state detection method of programmers

Student: Sun Yuecheng

Advisor: Professor Zhao Yunbo

College of Information Engineering

Zhejiang University of Technology

June 2020

浙江工业大学

本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所提交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

2020 年 6 月 20 日

浙江工业大学

本科生毕业设计（论文、创作）任务书

专业 自动化 班级 自动化1604 学生姓名/学号 孙悦铖/201607760119

一、设计（论文、创作）题目：

程序员的疲劳状态检测方法

二、主要任务与目标：

1. 阅读相关文献，了解本领域研究现状；
2. 完成程序员疲劳状态的检测方法
3. 实现对程序员疲劳状态的检测；
4. 撰写毕业论文。

三、主要内容与基本要求：

本课题旨在研究程序员的疲劳状态的检测方法，将对程序员的疲劳状态进行分类检测和确定的方法进行研究。资料：将提供若干的相关文献以及相关书籍，同时培养学生查询相关文献的能力。本课题将主要基于文献阅读整理，问题模型化和算法研究等，学生要求对心理学有一定兴趣，有一定的算法基础和计算机编程能力。具有一定的英文阅读水平，能够较为独立的研读英文科技文献。具有一定的总结提炼能力。

四、计划进度：

- 2020 开学前 收集相关资料文献，学习相关知识，完成外文翻译、文献综述；熟悉课题，做好开题准备 - 第 1-3 周 完成开题报告，参加开题交流 - 第 4-8 周 完成程序员疲劳状态的检测方法，接受中期检查 - 第 9-14 周 完成程序员疲劳状态的检测方法并实现，撰写毕业论文 - 第 15 周 修改毕业论文，参加毕业答辩，提交相关文档资料

五、主要参考文献：

- [1] Chi-Pang Lam, S. Shankar Sastry, "A POMDP framework for human-in-the-loop system", 53rd IEEE Conference on Decision and Control [2] Jie Fu, Ufuk Topcu, "Synthesis of Shared Autonomy Policies With Temporal Logic Specifications" IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL.13, NO.1, JANUARY 2016 [3] Sharon Zhou, Tong Mu, Karan Goel, Michael Bernstein, Emma Brunskill, "Shared Autonomy for an Interactive AI System", UIST'18 Adjunct, October 14 - 17, 2018, Berlin, Germany [4] 王天本, 王海鹏等, "基于键鼠事件的疲劳感知" 计算机应用, 2014, 34(1): 227 - 231

任务书下发日期 2019 年 12 月 20 日

设计（论文、创作）工作自 2019 年 12 月 20 日至 2020 年 6 月 20 日

设计（论文、创作）指导教师

赵云波

系主任（专业负责人）

孙有兵

主管院长

孙有兵

程序员的疲劳状态检测方法

摘 要

随着计算机的发明和广泛应用，它早就成为了人类社会中的重要一环。而程序员作为开发以及维护计算机程序的专业人员，该人群的数量庞大并仍在逐年增加。但程序员的工作状态以及工作强度导致他们极易进入疲劳状态，长期的疲劳会引起工作效率下降、危害身心健康。准确检测其疲劳状态，进行适当的提醒可对程序员的健康起到一定的保护作用。

传统的疲劳检测技术并不非常契合程序员的疲劳状态检测情况。本文主要通过神经网络技术对判别程序员的疲劳状态方法进行探究和尝试，并实现具有检测功能的程序。

论文的主要工作如下：

- 1、简述课题研究背景和研究现状，简要介绍了疲劳检测产生并发展至今的各种主要技术，分析其对本课题的适用性。
- 2、概述神经网络结构原理，探究一些经典的图像分类神经网络。确立使用神经网络技术的疲劳检测方案。
- 3、自建脸部状态的疲劳检测数据集，对嘴部和眼部的疲劳特征有较好的反应。提出并实验了预先对所需特征进行切割凸显来保证识别准确的方法。
- 4、编写疲劳检测程序，基本实现了应用神经网络对人疲劳检测的功能。可通过对眨眼和哈欠等脸部状态识别，判读是否疲劳，并在检测到疲劳状态后进行提醒

本文预期目标为尝试检测方法，在程序中实现对程序员的疲劳状态检测。通过毕设期间的努力达成了该目标，但仍有不足之处：自建数据集的丰富度不足；检测程序的精度尚有提升空间。这些都应该是在未来进行改善的点。

关键词： 疲劳检测，卷积网络，数据集，PERCLOS

FATIGUE STATE DETECTION METHOD OF PROGRAMMERS

ABSTRACT

With the invention and wide application of computer, it has long been an important part of human society. As a professional who develops and maintains computer programs, the number of programmers is huge and increasing year by year. But programmers' working state and working intensity make them easily enter into fatigue state. Long-term fatigue will lead to the decline of working efficiency and harm physical and mental health. It can protect the programmer's health to detect the fatigue state accurately and remind them properly.

The traditional fatigue detection technology is not very suitable for the fatigue state detection of programmers. In this paper, neural network technology is mainly used to explore and try to identify the fatigue state of programmers, and to realize the program with detection function.

The main work of the thesis is as follows:

1. This paper briefly introduces the research background and current situation of the subject, briefly introduces the main technologies of fatigue detection, and analyzes their applicability to the subject.

2. This paper summarizes the structure and principle of neural network, and explores some classic image classification neural networks. Establish the fatigue detection scheme using neural network technology.

3. The fatigue detection data set of face state is built, which has a better response to the fatigue characteristics of mouth and eyes. This paper proposes and experiments the method of cutting and highlighting the required features in advance to ensure the accuracy of recognition.

4. The program of fatigue detection is compiled, and the function of human fatigue detection by neural network is basically realized. It can recognize the face state such as

blink and yawn, judge whether it is tired, and remind when it is detected.

The expected goal of this paper is to try to detect the fatigue state of programmers in the program. This goal has been achieved through the efforts during the completion period, but there are still shortcomings: The richness of self-built data set is insufficient; There is still room to improve the accuracy of the test program. These should be the points for improvement in the future.

Key Words: Fatigue detection, Convolutional network, Data set, PERCLOS

目 录

摘要	I
ABSTRACT.....	II
第 1 章 绪论	1
1.1 课题研究意义	1
1.2 课题研究背景综述	1
1.2.1 基于生理信号的检测方法.....	2
1.2.2 基于面部特征的检测方法.....	3
1.3 论文组织结构	6
1.4 本章小结	6
第 2 章 卷积神经网络	7
2.1 卷积神经网络结构简述.....	7
2.1.1 卷积层.....	7
2.1.2 池化层.....	8
2.1.3 全连接层.....	9
2.1.4 损失函数.....	9
2.2 图像分类卷积神经网络.....	10
2.2.1 LeNet-5	10
2.2.2 AlexNet	11
2.2.3 VGGNet	11
2.2.4 GoogLeNet.....	12
2.2.5 ResNet.....	14
2.2.6 DenseNet.....	16
2.3 本章小结	16
第 3 章 程序员疲劳检测实验	17
3.1 数据集构建和完善	17
3.1.1 数据集采集和简单优化.....	17
3.1.2 初步人脸裁剪对数据集优化.....	20
3.2 疲劳检测实现	22
3.2.1 单张图片的状态分类.....	22
3.2.2 疲劳检测原理.....	23
3.2.3 视频的疲劳状态检测.....	24
3.3 本章小结	26
第 4 章 总结与展望	27
4.1 工作总结	27

4.2 未来展望	27
参考文献	29
附录：部分主要程序	32
致谢	34

第 1 章 绪 论

1.1 课题研究意义

自上世纪 40 年代计算机的发明开始, 计算机技术就以一种不可阻挡的态势发展着。现如今, 计算机和互联网已经深入到了人民生活的方方面面。在我国推行的“互联网+”也让我们看到了这个领域的活力。如此重要的领域自然引领了一个非常庞大的行业, 其互联网从业者中的程序员人群就是本文的研究对象。

毫无疑问, 程序员人群在现在的社会结构中已经有了比较庞大的数量。这个庞大群体的身心健康问题也越来越频繁地牵动着人们的神经。2019 年 1 月的一份简单统计表明, 中国已有约 500 万的程序员。而根据埃文斯数据公司(Evans Data Corporation) 2019 最新的统计数据显示, 2019 年底全球共有 2640 万软件开发人员, 预计到 2023 年这个数字将达到 2770 万。

通过查阅医学类期刊杂志得知, 疲劳的工作状态非常容易造成健康危害。在程序员的工作中, 又不可避免地长时间坐着持续使用电脑。结合这样的工作状况我们发现, 当程序员长期处于疲劳的状态时: 在生理上会有视疲劳疾病, 会造成视力损害, 也容易引起手指和腕部关节不适, 腰部及颈部疲劳损伤等等; 在心理上, 长时间的疲劳会造成注意力涣散, 焦虑, 浮躁易怒的情况。如果疲劳状态得不到有效改善, 甚至容易导致抑郁等严重心理疾病, 更有甚者会产生轻生的念头。

进一步查找文献发现, 人的疲劳程度与时间并非呈简单的线性关系, 不同的工作内容以及工作环境都会对疲劳产生造成影响。现在也没有较好的模型可以归纳各种变量和疲劳状况的关系, 并得出进入疲劳状态的规律

那么从实际的生理情况出发, 对程序员的疲劳检测方法进行研究, 寻找合理合适的检测方法并实验, 可对程序员的疲劳情况进行实时反馈, 帮助其改善健康状况、提升工作效率。也可以为其余对于程序员群体的研究提供参考。

1.2 课题研究背景综述

基础的疲劳检测技术出现在上世纪 30 年代就并不断发展。时至今日, 疲劳检测技术也在实际应用中发挥出了自己的力量。现在疲劳检测广泛应用的领域还

是在生产安全要求较高的高风险系统中。这些环境要求操作人员在工作时具有较高的注意力，比如大型器械操纵。一旦处于疲劳状态，极有可能引起生产安全事故。其中疲劳检测应用最广，研究最深的领域就是驾驶员疲劳检测。

疲劳驾驶预警系统最初于 20 世纪 80 年代开始研究，当时由美国交通部开展对机动车驾驶和交通安全的研究。在 2000 年以后美国、英国、日本等国家和许多私人企业相继投入了大量资金进行进一步研究^[1]。

不过目前针对程序员的疲劳状况检测基本上还是空白，这是由检测技术决定的。大部分的检测方法都具有一定的侵入性，或是由于检测设备成本过高无法普及。最初的检测研究都是从医学方面入手，根据生理信号对进行疲劳状态检测，不过碍于各种技术的不发达，进度非常缓慢。得益于最近几十年计算机领域各种技术的飞速发展，计算力大量增加，并且催生出了机器视觉、图像处理等新兴技术。把它们应用到疲劳检测领域后又促进了检测研究新的进展。

1.2.1 基于生理信号的检测方法

人处于疲劳状态时，生理指标会发生较大变化，早期的疲劳检测就是通过一些医疗设备测量人体的生理参数，再基于参数分析疲劳状况。不过大部分生理参数的获得都需要借助专业设备，成本高，侵入性大，会对正常操作产生影响，不适合用于程序员疲劳检测。

(1) 脑电信号检测 EEG

这是相对比较精准的疲劳检测指标，客观并且实时性很强，因此也被广泛认为是疲劳识别中的“金标准”。脑电信号检测时，不同的波段体现了不同的疲劳状况，所以现在很多研究将重心放在波段的分类和特征提取上，包括一些研究人员用小波包分析与核学习方法实现疲劳的自动识别^[2]，也有小波分析提取脑电波数据特征^[3]。Kecklund 等发现了在从清醒状态过渡到瞌睡状态时脑电波的明显变化^[4]有学者将正则化流形信息极端学习机（GELM）与脑电图结合来检测疲劳^[5]部分学者通过支持向量机（SVM）搭建疲劳识别模型，最终达到了 86% 的准确识别率^[6]。有些学者把脑电图数据划分为四类，再经傅里叶变换转到频率空间研究，用幅度变化的不同来衡量疲劳程度^[7]。

(2) 心电信号检测 ECG

通过心率均值（Heart Rate, HR）和心率变异性来体现。疲劳时，心电信号中的高频、低频分量之比有显著的变化。有些学者再次实际实验证明了 LF、HF 和 LF/HF 三个指标的变化能反映疲劳情况，并且 LF/HF 的变化体现了疲劳产生特性。董占勋等人运用驾驶仿真平台，做了心率变异性（HRV）检测和眼动跟踪实验，探寻 HRV 和 PERCLOS 参数的联系，并且再一次证明了用心电指标反映疲劳情况的可行性^[8]。

（3）肌电信号检测 EMG

一些学者进行研究，得出了肌电信号可以有效体现疲劳状态的结论^[9]。部分学者通过检测肌肉表面的肌电信号，得出肌电信号反映疲劳状态的结论，并且发现疲劳产生时，肌电图频率下降。产生疲劳后，肌电图的幅度变大^[10]。Hostens 等人也进行了相关实验得出了类似的结论。疲劳发生时，频率下降，疲劳加剧后，幅值变大^[11]。

（4）其他生理信号

其他生理信号包括眼电信号 EOG，还有脉搏、呼吸等，也经常被用来作为疲劳检测的数据来源。眼电信号值的是在眼睛上下都放置电极，可以获得对于眼睛垂直方向上的眼电信号数据。部分学者根据基本的原理，分析眼电信号波形，波形的尖峰代表的是闭眼时电极值达到最大。闭眼时电极增大波形上升，睁眼时，波形下降，因此波形的上升时间就是闭眼的持续时间，下降时间是睁眼持续时间。他们根据不同的波形和参数进行聚类操作，准确地得出不同精神状态下的眼电信号波形图^[12]。也有部分学者综合地对皮电信号、心电信号和呼吸状态进行采集分析，来对被测者的疲劳状态进行衡量^[13]。

基于各项生理信号的疲劳检测研究是客观准确可信的，各种生理都信号忠实地反映了当前身体的真实状况，信号与疲劳状态的逻辑联系也都经过了严格的证明。但这些数据的采集大部分都具有极高的侵入性和对专业设备极高的依赖性，可用在高危险环境对操作员的疲劳检测和警示，以防止出现重大的安全事故。但不适合作为轻量的疲劳检测手段。

1.2.2 基于面部特征的检测方法

基于面部生理特征的疲劳检测，即通过采集面部头部的各项数据如，头部姿

态，嘴部状态，眨眼频率、视线变化等来进行分析，从而得出数据和疲劳状态的明确对应关系并应用于具体的检测。基于面部特征的疲劳检测无接触式反应，侵入性相对较低，检测系统装备简便。并且近年来计算力的不断发展和图像处理技术的成熟，以及从 2012 年开始，卷积神经网络技术的大量研究和应用，这类疲劳检测拥有了其得天独厚的优势。

（1）眼部信号检测

1944 年，Wali Wierwille 就分析了眼睛闭合时间与疲劳程度的关系^[14]之后卡内基梅隆研究所的 Dingus 等人对脑电信号，头部位置和眨眼参数相对于疲劳状态的对应关系进行了研究，反复实验后得出眼睑闭合时间（Percentage of eye closure, PERCLOS）有着非常高的准确率。PERCLOS 将眼睛闭合 80%以上认为闭眼，并通过单位时间内闭眼的时间来量化疲劳程度^[15]。自从机器视觉兴起以来，根据 PERCLOS 标准判定疲劳的研究也越发的火热，并且取得了不错的成果。

有些学者利用了深度学习和神经网络技术，对人眼的疲劳特征进行提取，使用贝叶斯网络对分类器进行训练，从而提高辨识准确度，获得更有效的疲劳特征^[16]。也有利用了视网膜对波长差异的红外光具有不同种反射性的生理特点，对瞳孔状况各种数据量化并计算，从而进一步验证了 PERCLOS 的正确性，完成了 PERCLOS 的具体计算^[17]。

也有很多单位根据算法的研究对检测系统进行实现并应用。美国明尼苏达大学实现了对眼睛的实时跟踪监控系统，有效定位和判断疲劳状态。国内南京远驱有限公司生产了 gogo805，在普通 PERCLOS 睁开闭眼的疲劳度量之外，还对眼睛睁开时瞳孔的收缩度进行测量，也取得了疲劳检测不错的效果^[18]。

还有很多研究人员拓展思路，开发出了基于不同生理理论的眼部疲劳检测方法。有些学者用小波提取出人脸中眼睛的部分，并用神经网络分类器进行疲劳识别^[19]部分学者用运动图像中的眨眼频率来进行疲劳检测^[20]也有通过眼睑运动来量化疲劳数据的成果案例^[21]

（2）嘴部信号检测

除了理论技术较为成熟的眼部检测，对于嘴部的研究也是基于面部生理特征疲劳检测的重要研究途径。在处于疲劳状态时，人容易产生打哈欠的现象，这种

情况会造成嘴部在短时间内形状变化并且面积增大。可以利用此类或相似的生理原理通过图像识别来进行嘴部状态分类并达到疲劳检测的目的。

有些学者明确分类了正常状态, 对话状态和疲劳状态下嘴巴的开度不同, 以几何特征为特征值, 令特征结合为特征向量后使用三级 BP 网络, 界定了三种不同精神状态^[22]。也有采用其他神经网络算法实现打哈欠等嘴部疲劳状态检测, 采取卷积神经网络提取出嘴部疲劳时的特征表现, 经过分类器训练, 来识别嘴部疲劳情况^[23]目前已经过验证的打哈欠状态判别方式还有计算嘴巴宽高比, 提取嘴巴轮廓获得所需宽高数据^[24]。

(3) 头部信号检测

头部的运动模式也可反映一部分疲劳状况, 不过针对头部的疲劳检测相对没有眼部和嘴部检测应用的广泛。相较于眨眼闭眼时间频率和嘴部打哈欠的直观性, 头部位移运动更加复杂, 因此也较难直接阐述它和疲劳之间的关系。不过还是有研究人员对该方向展开了研究。ASCI 等学者对点头动作和轻微疲劳之间的关系进行研究, 并用传感器和软件获得了头部运动和疲劳状态下驾驶的可信关系^[25]。

运用图像识别、机器视觉、以及神经网络等技术的面部生理特征疲劳检测, 已经在现在疲劳检测领域占有了很大的研究比重。比起传统的基于生理信号开展的疲劳识别, 它不需要太多笨重的设备, 对检测对象也不会造成过多的干扰, 对算法的持续研究也保证了它不输于生理信号的识别准确度。低成本, 低侵入性, 易于实现等种种优点让它为疲劳检测领域带来了更好的发展方向。

除了主流的疲劳检测方法, 也有学者针对计算机领域构想出了不同的疲劳检测方法。例如文献中提到的根据键鼠事件检测操作者的疲劳状态^[26]。根据医学研究, 肌肉的疲劳程度与无意识情况下肌肉抽搐延迟时间相关^[27]这也为疲劳检测提供了新的思路——通过检测操作人员工作时, 键盘和鼠标事件的延时情况来量化疲劳。

从生物信号疲劳检测开始, 到借用计算机的面部生理特征疲劳检测, 疲劳检测的方法一直都在变化着, 也不断有研究人员提出新的思路, 拓展疲劳检测的方法, 并且与原有体系碰撞融合从而诞生出更简便准确高效的检测方法。

1.3 论文组织结构

本文的研究内容主要是对桌面环境的程序员做比较轻量化、侵入性小但又能保证精确度的疲劳状态检测方法探究和简单实现。主要章节内容如下：

第一章 绪论。分析了对程序员的疲劳状态检测方法研究的意义。并且总结了研究现状，简要阐释了当前疲劳检测的应用领域和方法，分析各种疲劳检测方法对本毕业设计的适合度。最后介绍了，文章的主要研究内容和组织结构。

第二章 卷积神经网络。按照绪论的分析，需要通过面部特征进行疲劳检测。神经网络作为工具是较好的选择。这章主要介绍了卷积神经网络的结构知识，并罗列分析了几种适用于图像分类的卷积神经网络。

第三章 程序员疲劳检测实验。对文中疲劳检测方法的猜想进行探究性实验。主要进行了数据集的构建，并用 Python 语言编写了简单的疲劳检测程序。

第四章 总结与展望。对文章工作做出总结，分析不足之处并对未来可以改进的部分提出建议。

1.4 本章小结

本章为全文绪论部分。主要讲述了课题的意义以及现有研究背景，综述了较为常见的疲劳检测方法，并对全文的组织结构做了简要展示。

第 2 章 卷积神经网络

本次毕业设计期望通过神经网络技术对人脸的具体状态进行分类,加以区分后通过合理的程序设计大致实现对人的疲劳检测功能。因此,实现目的的首要一步是选择合适高效,素质好的图像分类卷积神经网络。对于这个研究,神经网络的知识是必不可少的,理解深入之后,才能对后期实验有更大助益。所以我们对神经网络做了一个简要的了解。

2.1 卷积神经网络结构简述

卷积神经网络是脱胎于视觉认知模型的人工神经网络。上世纪 60 年代, David Hubel 等人研究了猫脑内视觉皮层细胞,并由此提出了感受野的概念。之后的进一步研究表明,类似的细胞同样存在于人脑内,这是人脑拥有视觉功能的基础。到了 80 年代,日本学者福岛提出了神经认知机 (Neocognitron),这种结构对之后卷积神经网络的产生有着启发式的作用。

传统的人工设计采用浅层识别网络,难以有效刻画对象的变化,一旦分布变得复杂,就需要更深层次的网络结构。从低层次到高层次特征抽象,利用深层结构实现复杂的非线性拟合。而自 2006 年深度学习提出和兴起以来,它也确实表现出了各种优势:充分挖掘数据内部规律、特征提取更加简便、泛用型极佳,迁移学习适应不同领域能力强。此外,它对图像的各种变形例如:位移、旋转、缩放有很好的鲁棒性。

2.1.1 卷积层

卷积层是卷积神经网络得名缘由也是其中最重要的部分。它主要完成了特征提取和特征映射。卷积层的每个节点的输入都是前一层的一小块区域。低层次可以完成边缘检测,高层次则将神经网络中提取到的特征深入分析从而获得抽象度更高的特征。每一层中都有卷积核,不同的卷积核能够对不同的特征进行提取。卷积核公式如(2-1)所示。

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \quad (2-1)$$

其中 f 就是非线性激活函数，对输出进行处理，加入非线性因素，提高网络性能。常用非线性激活函数有 sigmoid、tanh、ReLU、PReLU 等。

通常的卷积层都会采用多个滤波器对特征进行提取，将它们的输出级联起来，便可以将生成数据视为三维矩阵。它的宽和高由滤波器的大小以及步长决定，深度则由步长决定。

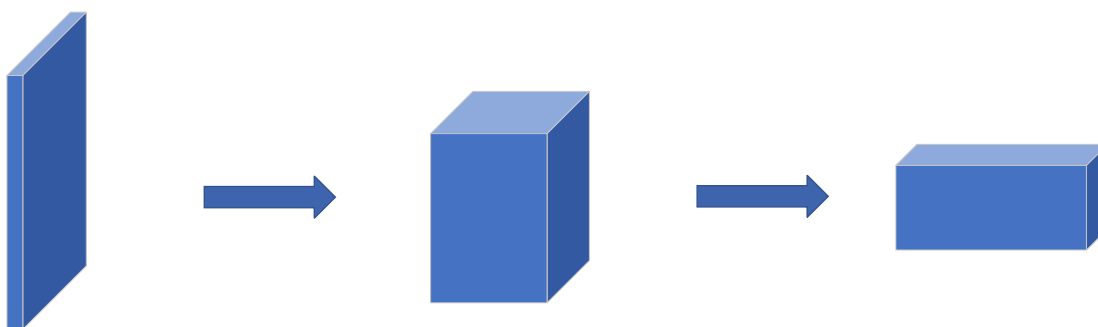


图 2-1 卷积层输出变化

从图 2-1 我们可以看出，在输出的某个位置沿着深度方向汇聚着不同滤波器提取的特征变化。网络层次深入之后，卷积层抽取的特征逐渐由底层细化的信息向着高层抽象的语义信息过渡。

2.1.2 池化层

滤波器移动计算的时候，不可避免地会重复计算很多信息。那么，池化就是去除此类信息，并且使计算速度加快的一种操作。池化层可以对特征图实现再采样，即把图像划分为一般 $2 * 2$ 像素大小的区域，再在小区域内进行采样，减少数据量的同时让采集到的特征更加有效，改善过拟合情况。除此之外，分成小区域降低特征分辨率使特征对例如平移的一些形变敏感度降低，提升了鲁棒性。

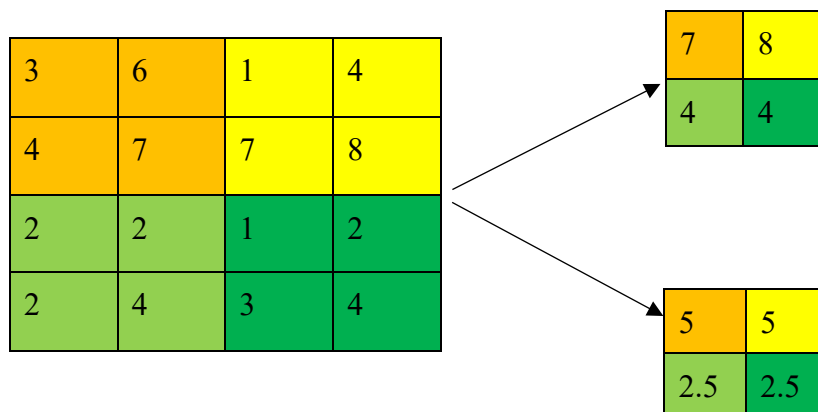


图 2-2 两种池化操作

基本的池化操作有两种，图 2-2 显示了这两种变化，从上到下依次是：最大池化操作（Max Pooling）和平均池化操作（Mean Pooling）。最大池化操作即在小区域内选出最大的数值作为输出，平均池化操作则是对小区域内所有数值进行算数平均再输出。

2.1.3 全连接层

全连接层一般位于顶部，对结果进行预测，是比较独特的卷积层。它的内部也有很多卷积层，不过全连接层内的卷积核大小与特征图大小正好一致，因此它可以把卷积层的输出转化为向量，完成了对抽象特征的组合。它使每个输出神经元和输入神经元连接。

2.1.4 损失函数

损失函数也是网络中的重要的一环，利用反向传导算法，一步步修正参数，对网络结构进行优化，让神经网络的输出与既定目标之间的距离缩小。

线性模型和神经网络有非常大的区别，其中一处就是神经网络的非线性导致大多数我们感兴趣的损失函数都变得非凸。这意味着神经网络的训练通常只要使用迭代的、基于梯度的优化，仅仅使得损失函数达到一个非常小的值；而不是像用于训练线性回归模型的线性方程求解器，或者用于训练逻辑回归或 SVM 的凸优化算法那样保证全局收敛。

和其他的机器学习模型一样,为了使用基于梯度的学习方法我们必须选择一个损失函数,并且我们必须选择如何表示模型的输出。深度神经网络设计中的一个重要方面是损失函数的选择。

常用的损失函数有 SoftMax 等。

2.2 图像分类卷积神经网络

给出一组已经标记好的图像训练集,其中每张图像都确定了所属类别。之后有一组新的图像测试集,对其标签分类,测试准确性即为图像分类。

图像分类步骤:

- 1.输入图像训练数据集
- 2.用这个训练集训练出一个分类器,它学习了各个类别的特征,可用来分类。
- 3.使用分类器对测试数据集进行分类,将预测出的结果和真实类别比较,评估出分类器性能。

接下来简要探究一下经典的一些图像分类神经网络情况

2.2.1 LeNet-5

1998年,Yann LeCun 提出的卷积神经 LeNet,首次提出了多层级联的卷积结构,可对手写数字进行有效识别。这是最初的卷积神经网络,也是最早的用于图像分类的经典网络。虽然网络结构尚且比较浅,但是它的内部结构已经包括了卷积层、池化层、全连接层非常完整。

·第一层:输入层,图片是大小 $32 * 32$ 的黑白图像,因为设计是作为识别手写数字体的网络,所以包含有0~9共十种类别的图片。

·第二层:卷积层,filter大小 $5 * 5$,个数6,padding为0,卷积步长 $s = 1$,输出矩阵大小为 $28 * 28 * 6$ 。共有 $28 * 28 * 6$ 个神经元,参数数量为156($5 * 5 * 6 + 6 = 156$,6为偏置项参数),每一个单元与输入层的25个单元连接。

·第三层:池化下采样层,采用平均池化方法。filter大小为 $2 * 2$,个数16,步长 $s=2$,padding为0,输出矩阵大小 $14 * 14 * 6$ 。

·第四层:卷积层,filter大小 $5 * 5$,个数6,padding为0,卷积步长 $s=1$,输出矩阵大小 $10 * 10 * 16$ 。

·第五层：池化下采样层，采用平均池化方法。filter 大小为 $2 * 2$ ，步长 $s = 2$ ，padding 为 0，输出矩阵大小 $5 * 5 * 6$ 。

·第六层：卷积层，120 个特征图，每个特征图有一个神经元，filter 大小为 $5 * 5$ ，步长 $s = 1$ ，padding 为 0。

·第七层：全连接层：84 个神经元，与第五层构成全连接关系，激活函数为 sigmoid。

·第八层：输出层，因为是十种分类的任务，所以有十个神经元。

我们发现随着网络越来越深，图像的宽度和高度逐渐变小，通道数量慢慢增加。

2.2.2 AlexNet

这个神经网络出现在 2012 年当时的 ImageNet 竞赛上。也正是它让卷积神经网络的研究焕发出巨大的活力。AlexNet 成功应用了 ReLU、Dropout、最大池化、LRN (Local Response Normalization, 局部响应归一化)、GPU 加速等新技术，启发了后续更多的技术创新，加速了卷积神经网络和深度学习的研究。从此，深度学习研究进入蓬勃发展的新阶段。

它的网络中包含了 5 个卷积层，3 个应用了最大池化操作的池化层，和 3 个全连接层，其中最后一个使用了 softmax 激活函数的 1000 维输出层。

模型中一共有大约六百万个参数。又使用了 ReLU 作为激活函数，缓解了使用 Sigmoid 作为激活函数时的梯度弥散问题。即靠近输出层梯度大，参数更新快，收敛迅速；靠近输入层梯度小，参数更新慢，几乎随机分布的问题。

这个模型中也使用了重叠池化方法，保证特征丰富性，减轻过拟合但是在两个测试集中分别只降低了 0.3%和 0.4%的错误率，效果不明显。

2.2.3 VGGNet

VGGNet 由牛津大学的团队提出，研究了神经网络中，网络深度和整体性能的关系，构建了 16~19 层的卷积神经网络。根据实际表现我们发现，VGGNet 在迁移学习时有着非常不错的表现。从图像中提取特征时，它也是非常推荐的算法。

VGGNet 是在 AlexNet 基础上修改而来的。比较常用的是 VGGNet-16 和

VGGNet-19。VGGNet 的结构是把所有的网络分成了 5 段，多个 3×3 卷积网络串联起来为一段，并且每一段都后接一个最大池化操作的池化层。在串联的五段结构之后是三个全连接层，以及一个 softmax 层。从下图 2-3，我们可以比较清晰地看出它的结构。

从内部网络组成和排列上看 VGGNet 和 AlexNet 其实非常相似，都有着卷积层和池化层相互交替排布的结构，并在网络最后安置了全连接层，两者比较大的差别就在卷积核大小，步长大小，网络深度不同。

VGGNet 网络中使用了小卷积核，其中的大多数卷积核都是大小为 3×3 ，步长为 1 的。虽然大的卷积核看起来能够提取到更大范围里的信息，但是大卷积核是可以用小卷积核串联替换的。例如 5×5 卷积核可以用两个 3×3 卷积核串联替换， 7×7 卷积核可用三个 3×3 卷积核串联替换。这样的替换我们能够看出好处。首先它减少了参数的个数 $5 \times 5 = 25$ ，而替换后只需要 $3 \times 3 \times 2 = 18$ ； $7 \times 7 = 49$ ，而替换后 $3 \times 3 \times 3 = 27$ ，即 27 个参数。其次它带来了更多的非线性。一个大的卷积核总共只有一次激活过程，但是串联的多个小卷积核则意味着有多个激活的过程，也就意味着引入了更多的线性变化，模型表达能力可能由此提升。

2.2.4 GoogLeNet

这个网络和 VGGNet 都参加了 2014 年的 ImageNet 挑战赛，GoogLeNet 获得了冠军而 VGGNet 是亚军，从性能上来看 GoogLeNet 确实优于 VGGNet。22 层的深度，但是参数个数仅有 VGGNet 的 $1/36$ 。所以是内存或者计算资源不足时，较好的选择。

想要减少参数，可以利用稀疏网络结构，但是硬件大都是对密集矩阵有着计算优化，稀疏矩阵即使数据量少，它的计算消耗时间却不能简单地降低下来。当时 GoogLeNet 提出了一种 Inception 网络结构，如下图所示。

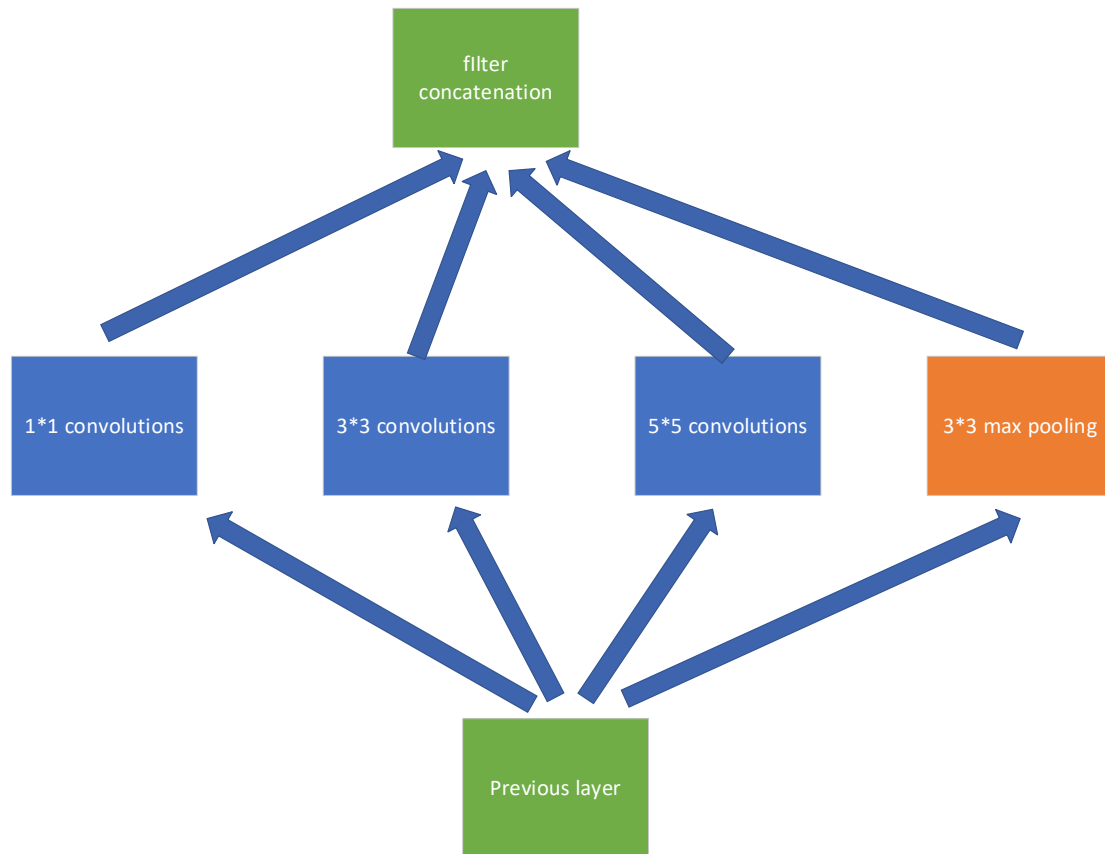


图 2-3 最原始 Inception 的基本结构

它将神经网络中常用的卷积和池化堆叠，增加了网络的宽度，也提升了网络对尺度的适应性。输入的细节由卷积层中的网络提取， $5 * 5$ 滤波器也能较好适应大部分输入。池化操作后，也可以减少空间，降低过拟合。并且每一个卷积层之后进行 ReLU 操作，还可以增加非线性特征。

所有卷积核在上一层输出上作用，卷积核所需计算量过大，为了降低特征图厚度，开发了 Inception v1 的网络结构，如下图所示。

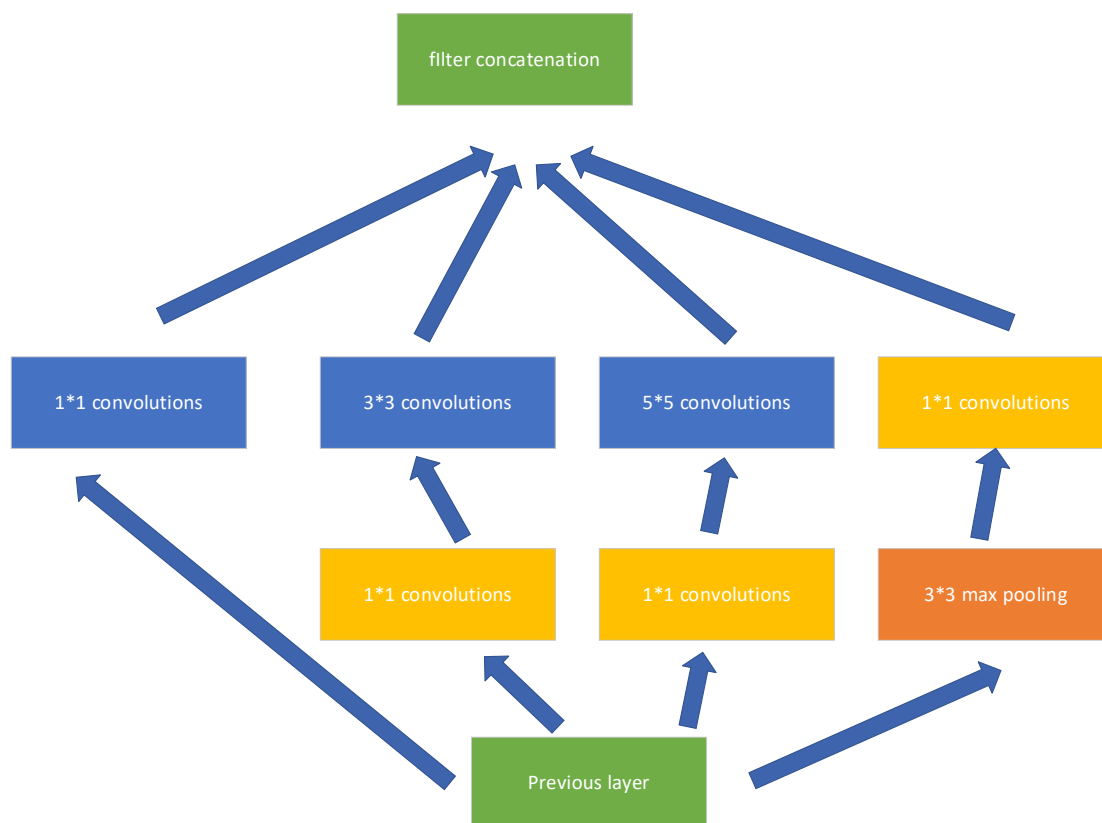


图 2-4 Inception v1 的网络结构

借助稀疏结构, GoogLeNet 可以更少的参数构建更深的网络, 其在 ImageNet 上的分类精度比 VGG 更高, 且训练速度更快。但是, 当网络继续变深的时候, 就会变得过于复杂, 并且会出现梯度弥散现象。

2.2.5 ResNet

直观来看, 一般网络越深, 面对复杂问题时就能够有更好的性能, 但是在实际测试的时候却经常发现, 网络深度增加的时候, 准确率却有所下降。随着网络层数慢慢增加, 一开始模型的精度确实得到了很好的提升, 但当层级到达了一定值后, 测试和训练的精度发生了迅速的下滑。网络过深, 神经网络难以训练, 训练集本身的误差也会加大, 这是之前也提到过问题。

为解决这种题, ResNet 被提出, 它准确, 并且参数量比 VGGNet 也低, 并且拥有很好的兼容适应推广性能, 直接使用到 InceptionNet 网络中也没问题。

简单思考, 即使较浅网络已经达到准确率的饱和, 在它之后加上输出等于输入的恒等映射层也不会造成误差的增加。使用恒等映射层在前后层之间传递就是

ResNet 的灵感来源。

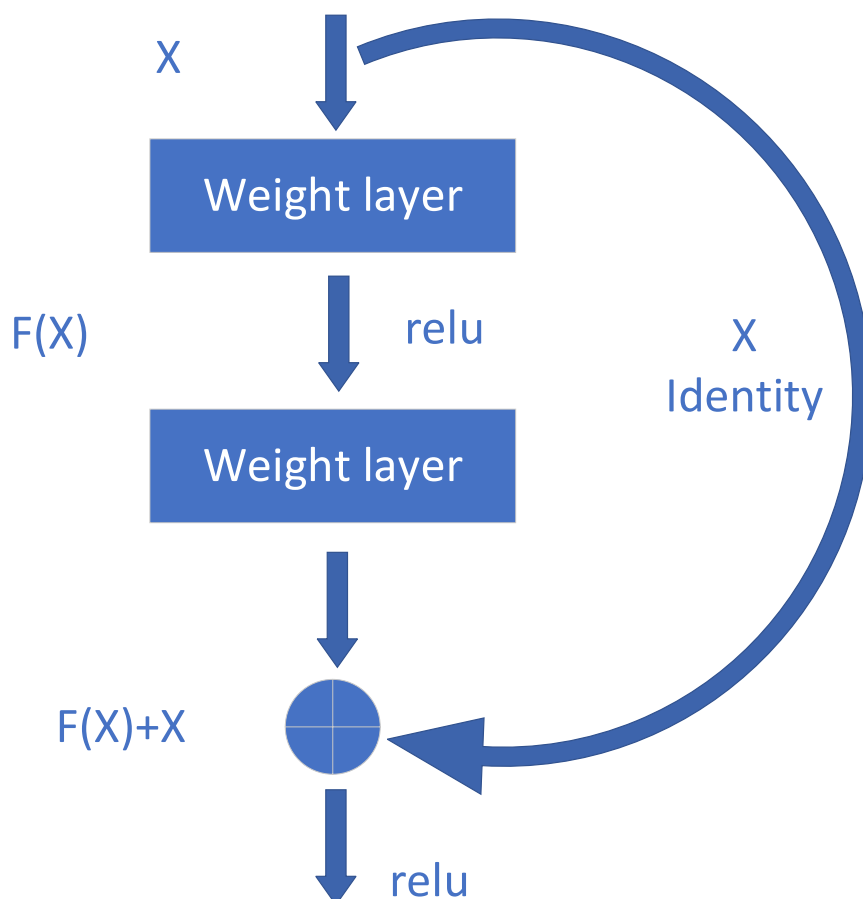


图 2-5 残差网络基本结构展示

由此 ResNet 引入了残差网络结构 (residual network)，通过这种残差网络结构，对提升网络深度有很好的作用，并且最终的分类效果也非常好，残差网络的基本结构如图 2-5 所示，很明显，这个设计的精髓是其中的跳跃结构。在 ResNet 中，相当于将学习目标改变了，不再是学习一个完整的输出，而是目标值 $H(X)$ 和 x 的差值，也就是所谓的残差 $F(x) = H(x) - x$ ，因此，后面的训练目标就是要将残差结果逼近于 0，这就使得网络不断加深，准确率依旧不下降。而最终的实验结果也证明了它的有效性。

它较为有效地缓解了传统卷积网络在信息传递时不可避免会出现的信息损

耗丢失问题，也能对过深网络产生梯度弥散和梯度爆炸的情况进行解决。将输入信息传递到输出的操作，对信息的完整性是一种确保，也将网络改为残差式，跟换了学习目标，简化了学习难度。

2.2.6 DenseNet

这是一种密集连接神经网络，它的独特之处在于，网络中任意两层之间都有直接连接，所以每一层的输入都是前面所有层的输出集合，这一层的结果也会直接传给接下来所有层作为输入。这种网络结构的高效也就在于此，减少每层计算量，并且重复利用特征。而且每层都包含所有之前输出信息，对特征图的需求相对来说减少了，因此参数量也降低了。并且每层直接连接的结构也可以缓解梯度弥散现象。这种网络结构，也可以提取到更丰富的特征，保护特征信息，令其多样化得以确保。

2.3 本章小结

这一章主要对之后需要用到的神经网络方面知识进行了简单的归纳和梳理。对通常神经网络中的各个模块结构进行阐释。之后简单介绍了用于图像分类的神经网络逐渐发展时，各个网络的演进方式，性能如何提升以及各自的长处。

第 3 章 程序员疲劳检测实验

本文实验主要步骤如下：(1) 获取内容数据丰富可用于实验的疲劳检测人脸数据集。(2) 对数据集进行神经网络训练，尝试几种图像分类神经网络之后，挑选准确度高的图像分类神经网络应用。(3) 实现单张图片的状态分类和标注。(4) 寻找人脸状态和疲劳之间的联系，将其以简单的计算方式量化。(5) 完成对实时视频的人脸疲劳状态检测。

3.1 数据集构建和完善

目前疲劳检测主要的成熟应用领域依旧是驾驶员疲劳检测等重机械操控环境。在查阅了一些资料后，经过搜寻并未发现能够很好契合本课题的公开数据集。大部分著名公开的人脸数据集有部分用于年龄肤色和名人识别。还有部分主要应用于对人脸位置检测。没有成熟的数据集，用于疲劳状态检测。于是我们尝试自行构建一个简单的数据集，来满足本次毕业设计实验要求，主要用于人的脸部的疲劳状态检测。因此，基本要求有 (1) 各个表情区分明确，特征明确不模糊。

(2) 采集图片丰富，在保证所需特征鲜明的情况下尽量包含不同状态，数据集多样化。

3.1.1 数据集采集和简单优化

最初进行拍摄任务时，我们邀请了 7 位志愿者进行采集。共选取了 5 个不同的背景，主要参考依据是为了模拟还原程序员办公时的场景。并且分三个时间段进行拍摄，采集到了不同光线条件下的数据。让他们进行眼睛的睁开、闭合和嘴部张开、闭合（模拟哈欠）等动作。为保证数据集多样，也让志愿者进行头部偏转，对不同角度的面部情况进行采集。前后几次采集，共收集到图片 5000 余张。

之后对原始图片进行简单整理，通过人工筛选明确剔除了分类模糊的图片，例如：(1) 志愿者眨眼过程中被采集，导致眼睛半睁半闭开度不易确定 (2) 志愿者嘴部动作时被采集，嘴部张合度难以确定 (3) 采集相机意外晃动，导致采集图片过于模糊 (4) 有其他物体遮挡面部情况。部分不合规范难以分类的图片

如图 7 所示。经过初步筛选，质量尚可的图片留存约 4700 余张，大致样式如图 8 所示。



(a)闭眼特征不明确

(b)特征有遮挡

图 3-1 不合规图片举例

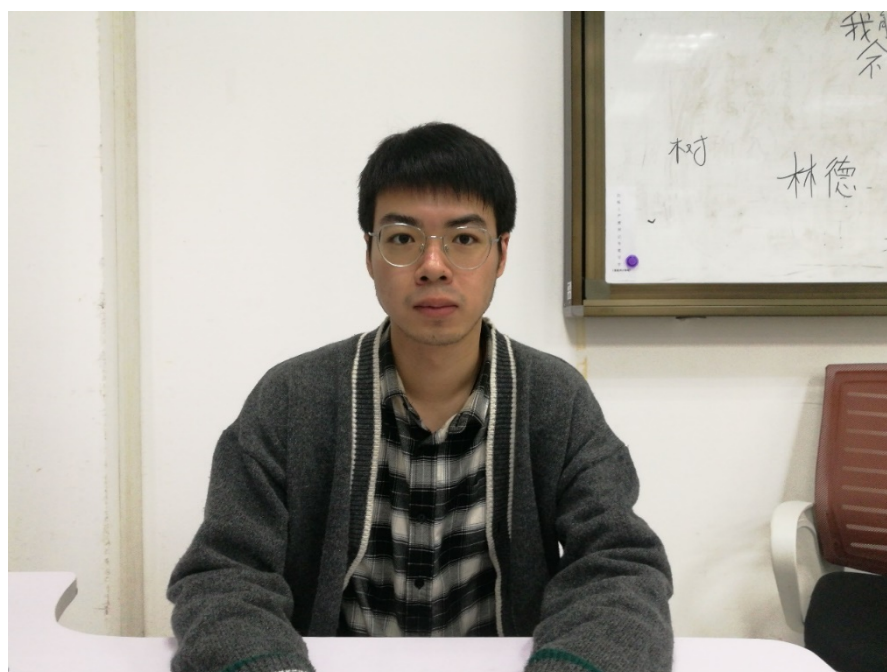


图 3-2 可采用图片举例

之后根据测试目的，为提取出脸部闭眼和哈欠的特征。我们手动将图片分为 11 类：（1）正视正常（2）正视闭眼（3）正视哈欠（4）左侧正常（5）左侧闭眼（6）左侧哈欠（7）右侧正常（8）右侧闭眼（9）右侧哈欠（10）低头（11）抬

头。

挑选出其中一位志愿者的采集图像为测试集，其余作为训练集。分类后基本每个类中含有 300 余张~500 余张图片。

此时我们仍对自建数据集性能存在疑惑，是否能够满足实验标准。所以尝试进行了一次训练，使用网络 resnet50。测试结果并不十分理想，测试准确率只有 63%左右，如下图所示。

```
Epoch 22/60 Batch 7/65 Loss 0.000321 (0.002594)
Epoch 22/60 Batch 14/65 Loss 0.002039 (0.001707)
Epoch 22/60 Batch 21/65 Loss 0.134354 (0.012636)
Epoch 22/60 Batch 28/65 Loss 0.081013 (0.014008)
Epoch 22/60 Batch 35/65 Loss 0.007094 (0.011855)
Epoch 22/60 Batch 42/65 Loss 0.003233 (0.010822)
Epoch 22/60 Batch 49/65 Loss 0.008214 (0.009784)
Epoch 22/60 Batch 56/65 Loss 0.001480 (0.010543)
Epoch 22/60 Batch 63/65 Loss 0.000837 (0.009753)
==> Test
acc:63.55
```

图 3-3 使用网络 resnet50 对数据集初步测试

简要分析，造成这种情况的可能有很多：建立的数据集中，正视、低头和抬头三种情况头部变化不大，区分过小导致不好区分。并且初次拍摄进行数据集采集时，为方便图片分类，让志愿者摆出要求姿势后，用间隔时间短的连拍方法进行采集。这样就导致数据集内的图片虽然数量大，但是有很多区别不明显的相近图片，导致实际丰富度很差。

为了改善上述情况，我们对数据集内的图片进行了再次筛选和删除。也进行了部分再拍摄，对数据集进行了扩充。并在采集过程中强调，让志愿者在保证需要区分的眼部和嘴部状态明确的情况下尽量丰富表情，头部进行适当运动。在采集设备上，我们也适当延长了间隔时间，保证图片间小区别增大，扩大数据集的整体丰富程度。并且尝试使用电脑摄像头进行一部分采样工作，模拟更真实的程序员工作环境。

对提升之后的数据集进行再次测试，此时准确率确实有了一部分明显的提升。

3.1.2 初步人脸裁剪对数据集优化

考虑到数据集要求特征明显以及提升丰富度的目的。我们尝试了对人脸进行简单的识别和分割，期望在数据集数量不扩展的情况下，继续提升它的丰富的和检测性能。如图 3-4 所示。

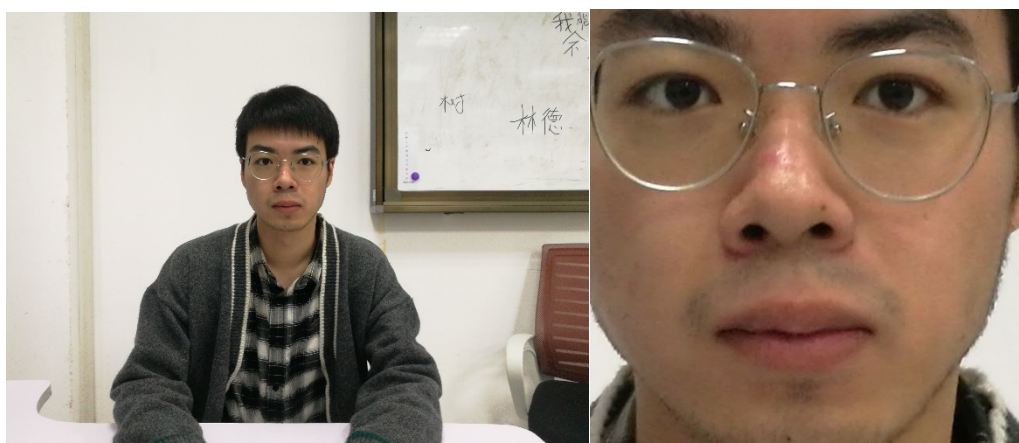


图 3-4 简单识别裁剪示例

此步骤主要应用了 `face-recognize` 函数，进行人脸识别并反馈坐标，再通过脚本剪切。不过数据集中存在一部分图片，无法通过这种简单方法进行识别，出于进一步保证所建数据集的图片数量和丰富度的目的，对于简单识别未能识别和成功操作的图像，也进行了手动裁剪分割并保留。

再次对数据集优化后，通过实验可以发现。不直接进行特征识别，通过人脸裁剪减少背景干扰后再进行识别分类，对该数据集的检测准确率提升确实有非常大的作用。这种方法思路对神经网络检测都有一定的适用性，即通过初步分析，先进行简单的识别和裁剪，突出特征信息。在从裁剪处理过后的数据集上进行训练，获得更好的分类精度。

之后再使用 `ResNet-50` 测试时得到了相对较为满意的结果，如下图所示。数据集构建完毕。

```
Epoch 1/60  Batch 7/49  Loss 1.241747 (1.741694)
Epoch 1/60  Batch 14/49   Loss 0.831318 (1.309217)
Epoch 1/60  Batch 21/49   Loss 0.364233 (1.051130)
Epoch 1/60  Batch 28/49   Loss 0.216105 (0.863353)
Epoch 1/60  Batch 35/49   Loss 0.190034 (0.729644)
Epoch 1/60  Batch 42/49   Loss 0.191282 (0.634509)
Epoch 1/60  Batch 49/49   Loss 0.076358 (0.557879)
==> Test
acc:79.75
```

图 3-5 裁剪后的数据集测试

前文介绍中的数据集建立和网络准确度训练，在实际实验中是同步进行的。我们在不断完善数据集的过程中，使用了各种神经网络对其进行测试，验证数据集效果的同时，也相对直观地看到了较为主流的几个图像分类神经网络的效果。主要选择的网络有 VGGNet 和 ResNet。在前期测试的时候就发现 VGGNet 效果不如 ResNet。于是主要选取了不同层数的 ResNet 测试。

测试的网络依照层数不同，有 ResNet-50，ResNet-34，ResNet-18。在对自建数据集的测试中，各网络的效果都有待加强。ResNet-50 和 ResNet-18 效果较为相似，ResNet-18 的部分训练结果如下图所示。从数据上看 ResNet-50 确实优于 ResNet-18 于是后续实验采用 ResNet-50 网络模型进行。

```
Epoch 56/60  Batch 7/49  Loss 0.000088 (0.000196)
Epoch 56/60  Batch 14/49   Loss 0.000478 (0.000371)
Epoch 56/60  Batch 21/49   Loss 0.000412 (0.000438)
Epoch 56/60  Batch 28/49   Loss 0.000255 (0.000422)
Epoch 56/60  Batch 35/49   Loss 0.000306 (0.000591)
Epoch 56/60  Batch 42/49   Loss 0.000144 (0.000790)
Epoch 56/60  Batch 49/49   Loss 0.000134 (0.000708)
==> Test
acc:77.57
```

图 3-6 ResNet-18 训练效果

3.2 疲劳检测实现

疲劳检测的实现方法主要通过神经网络学习，将各个特征准确分类。再通过疲劳识别原理，对区分出的特征进行归纳计算，得出疲劳状态。

3.2.1 单张图片的状态分类

在完善数据集的过程中进行了多次测验，发现 ResNet-50 对本文问题有着较好的识别准确率。因此在实际测试时，我们也选用了 ResNet-50 模型。对 resnet 模型的后两层进行修改，将它从千种分类转化为对 11 个特征进行区分。在数据集输入的时候也进行了 Random2Dtranslation（随机裁剪）、RandomHorizontalFlip（水平翻转）操作。并且使用了结合 AdaGrad 和 RMSProp 两种优化算法的优点的 Adam 优化器进行优化。优化了超参数的调整过程，计算更加高效。

单张图片的状态分类中也优先使用了 face-recognize 函数，先对人脸存在状态进行识别，若判断存在人脸则返回人脸坐标，根据训练好的模型判断人脸状态，并添加框格和文字进行显示说明。具体结果如下图所示。

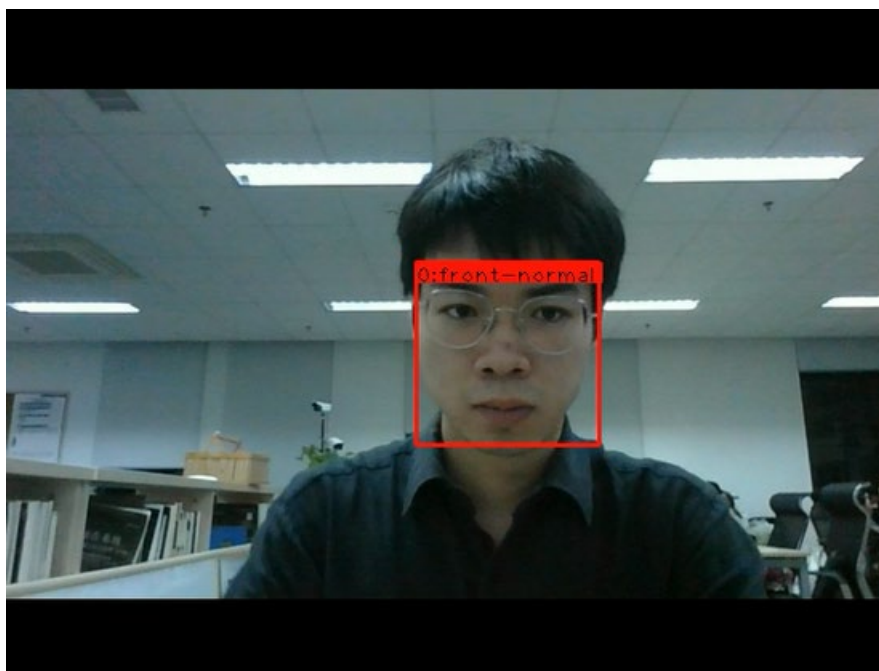


图 3-7 单张图片的分类标注

3.2.2 疲劳检测原理

实现疲劳检测的一个重要部分是如何判断一个人处于疲劳状态。有一个眼部检测标准主要参考的是 PERCLOS (percentage of eyelid closure over the pupil over time), 绪论中介绍基于面部特征的疲劳检测方法时也有提到。该参考标准主要判定单位时间内眼睛闭合到一定比例的时间所占的百分比。

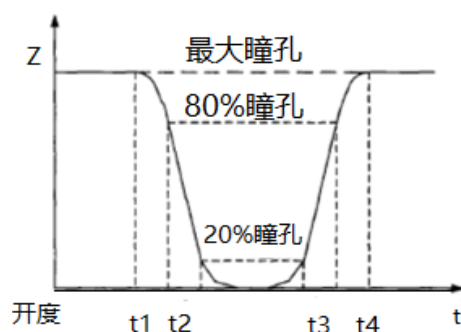


图 3-7 PERCLOS 分类的眼睛闭合过程

主要有三种测量标准 P80, P70, P50 (眼珠被眼皮遮挡区域大于 80%、70%、50%的时间占整个时间的比值)。美国交通局也对这项标准做过检测,发现 P80 最能准确反映当前疲劳程度^[28]。

如上图所示, $t_1 \sim t_2$ 为闭眼过程, $t_3 \sim t_4$ 为睁眼过程。假设 $t_1 \sim t_4$ 为一个检测周期, 那么 P80 即为

$$PERCLOS = \frac{t_3 - t_2}{t_4 - t_1} \times 100\% \quad (3-1)$$

本课题的检测并不能很好地做到如此高精度检测, 因此在检测过程中也将睁眼状态和闭眼状态做出了完全的区分。所以将上述过程按原理近似, 可以获得较为简单的检测方法。即统计一定时间内所有眼睛闭合的帧数, 算出它和总帧数的比值, 所得百分比即为近似 PERCLOS 值。

$$PERCLOS = \frac{n}{N} \times 100\% \quad (3-2)$$

式中 N 为此段时间内的图像总数, n 为同样时间闭眼的图像数目。

3.2.3 视频的疲劳状态检测

在能较好完成对单张图片的分类以及掌握如何判断疲劳的逻辑之后,就开始对完整的视频疲劳状态检测进行开发。

通过 VideoCapture 启动电脑摄像头,准备对当前坐在电脑前的人员进行检测。如摄像头前没有人,或由于各种干扰无法判别到完整的人脸,则在展示中输出"can't find face"如下图所示。



图 3-8 未检测到人脸状态

检测到人脸后,与单张图像分类一样先进行状态标注。完全正常状态下,程序不开始疲劳判定过程。此时程序在不断等待第一次闭眼或者第一次哈欠出现。当这些特征出现,意味着被测者可能发生疲劳状态,程序开始正式的疲劳检测过程。眼部的疲劳判别依据如 3.2.3 节中描述,在现有的研究中已经有了可靠的判别标准。在本文中参考文献的描述,并且根据原理近似之后变为:单位时间内闭眼的帧数总和达到阈值 50%是需要提醒的疲劳状态。当闭眼状态超过阈值时,页面上会显示报警信息,如下图 3-9 所示。



图 3-9 长时间闭眼提示

嘴部的状态较为复杂，一般说话和大笑时都有可能出现嘴部大张的情况，但上述情况的嘴部张开持续时间一般与打哈欠的持续时间有较大差异。简单研究发现，当嘴部持续张开时间到达 5 秒，则极有可能处于打哈欠的疲劳状态，应该予以提醒。报警如下图 3-10 所示。

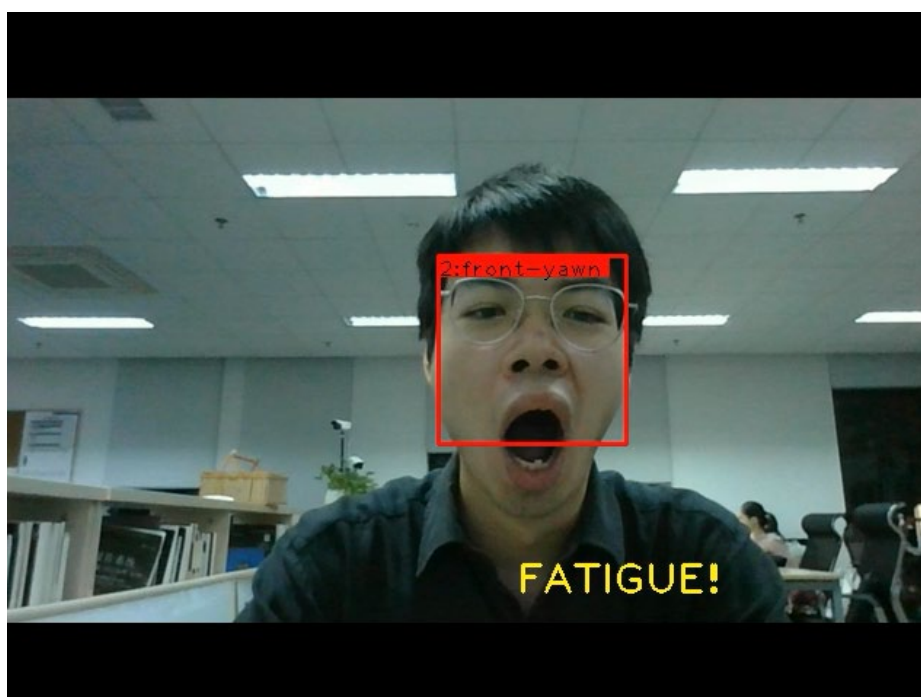


图 3-10 打哈欠状态提示

在实际运行程序的时候，我们发现了一种比较特殊的情况。当摄像头检测区域出现两个人，并且二者的脸部都可以识别。如下图所示。

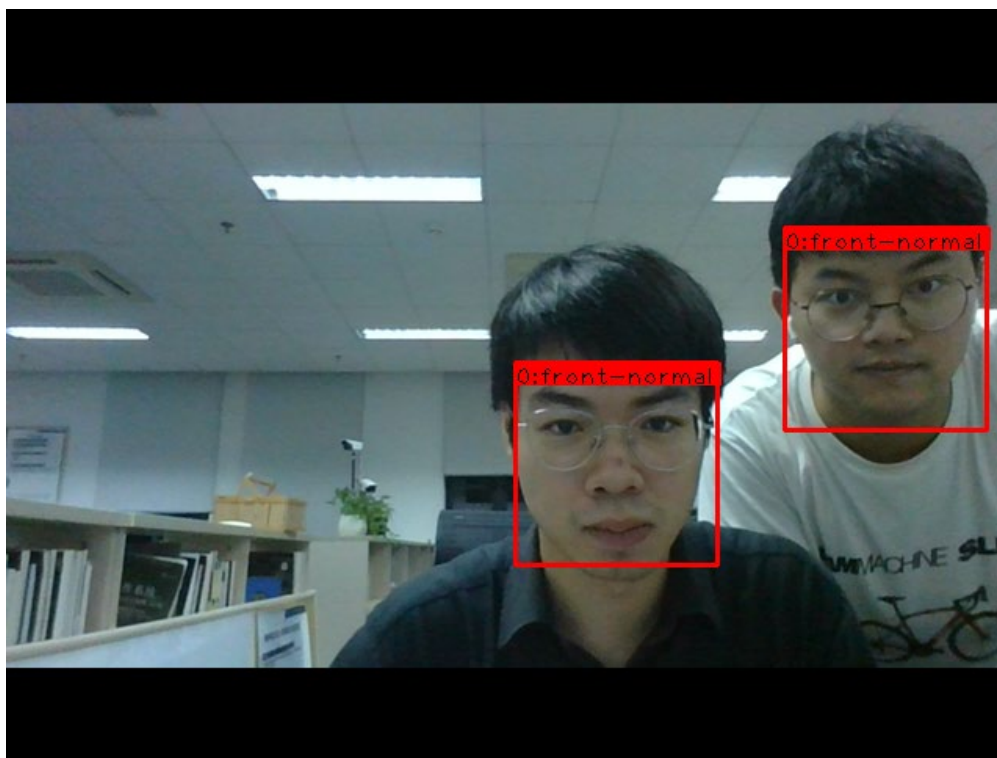


图 3-11 检测到两张人脸的情况

为防止在出现多个检测对象的情况下，程序出现错误，我们在程序判别人脸时增加了限定。状态分类可对所有识别出的人脸分类标注。疲劳检测和记录只对占画面主体的面部进行。

3.3 本章小结

第4章 总结与展望

4.1 工作总结

前苏联教授伯克曼提出了介于健康和疾病之间的“第三状态”，即“亚健康”的概念。简单的亚健康定义，是指身体检测没有疾病时，生理和心理均处于低质量的状况。亚健康人群在应激状态下，非常容易导致猝死。而世界卫生组织（WHO）的全国调查显示，75%的人均处于亚健康状态。很多从业人员尤其是程序员因为工作任务和工作状态处于疲劳而不自知。适当地告知疲劳情况反而是益于工作效率提升的。本文对程序员人群的疲劳检测方法做了简单探究。我将本文的毕业设计工作总结如下：

- 1、说明了对程序员人群进行疲劳检测的重要性和意义，简要介绍并分析了疲劳检测产生并发展至今的各种主要技术。
- 2、确立使用神经网络技术的疲劳检测方案。对神经网络结构原理进行概述，并探究了一些经典的图像分类神经网络。
- 3、自行构建并优化符合程序员实际工作环境的数据集，制作出适用于脸部疲劳状态检测的数据集，并提出预先对所需特征进行切割凸显来保证识别准确的方法。
- 4、编写了检测程序，基本实现了应用神经网络对人脸疲劳检测的功能。通过对眨眼和哈欠等脸部状态识别，判读是否疲劳，并在检测到疲劳状态后进行提醒。

4.2 未来展望

从生物信号疲劳检测开始，到借用计算机的面部生理特征疲劳检测，疲劳检测的方法一直都在变化着，也不断有研究人员提出新的思路，拓展疲劳检测的方法，并且与原有体系碰撞融合从而诞生出更简便准确高效的检测方法。

可以预见的，在疲劳检测技术接下来的发展当中，各种检测传感器也会不断更新迭代，将传统生理信号疲劳检测的难点慢慢克服。在不损失或少损失检测的可靠性与灵敏性的前提下，提高检测设备的非接触性，从而减小对被测者的直接影响。例如无粘贴电极的生理信号检测就是正在努力突破的研究方向。此外还要

提高检测设备性价比，降低成本，更易普及使用。测试设备更加智能，系统进一步优化，缩短检测时间达到最佳的时效性。算法会更加创新和融合，力求降低对被测试者和环境影响的敏感度。并且努力达到更加准确稳定的目标。并且信息融合也受到了广泛的认可，在前人的文献资料中，已经充分证明了各种检测方法各自的正确性，那么设法对它们进行综合，改善算法，并且在现有的一部分研究中已经验证了这种方法对疲劳检测准确度带来的提高。

经过查阅文献和分析总结，本文提出了对于程序员人群的疲劳检测方法，方法的有效性在简单的测试中也有所验证。但是存在以下几个方面的局限性：

第一，实验中所用到的数据集依旧不够丰富，即使对网络进行较为细致的调试后，这部分的缺失仍旧很难弥补。数据集的丰富程度直接影响了程序的检测精度。

第二，所研究的面部表现出的疲劳状态种类有限。只从较为明显的特征入手做了简要分析。

第三，从实际表现到疲劳状况的量化依然可以深入研究。本文只根据部分研究应用了较为简单的疲劳判定逻辑。

另外根据本文提出的方法所使用的数据以及所能够解决的问题类型，提出以下三点展望。

第一，疲劳检测应用更加广泛，出现更多特征丰富的疲劳状态数据集。

第二，神经网络的性能一直在提升，不断有更多优秀的网络结构被发掘出来。基于神经网络开发的疲劳检测方法有非常好的迁移性。之后的机器视觉将成为疲劳检测的很大助力。

第三，疲劳检测方法更加轻量化，降低检测设备成本。并且神经网络机器视觉兴起后，适用范围很广的疲劳检测方法也将越来越成熟，展现出更好的精准度，鲁棒性和实时性。

参 考 文 献

- [1] 田启川, 张润生生物特征识别综述[J]. 计算机应用研究, 2009, 26(12): 4401-4410.
- [2] ZHANG Chong, ZHENG Chong-xun, Y U Xiao-lin. Automatic recognition of cognitive fatigue from physiological indices by using wavelet packet transform and kernel learning algorithms[J]. Expert Systems with Applications, 2009 , 36 (3): 4664-4671.
- [3] MAO Ming-wang, DU Li-ping. Research on drive fatigue detection using wavelet transform[C]//Proc of IEEE International Conference on Vehicular Electronics and Safety. 2007: 1-40.
- [4] Kecklund G, A kerstedt T. Sleepiness in long distance truck driving: an ambulatory EEG study of night driving. Stockholm, Sweden: Department of Stress Research and National Institute for Psychosocial Factors and Karolinska Institute, 1998; 36 (9): 1007.
- [5] Huo X Q, Zheng W L, Lu B L. Driving fatigue detection with fusion of EEG and forehead EOG [C]. International Joint Conference on Neural Networks. IEEE, 2016:897-904.
- [6] HU S, ZHENG G, PETERS B. Driver fatigue detection from electroencephalogram spectrum after electrooculography artefact removal[J]. IET Intelligent Transport Systems, 2013, 7(1): 105-113.
- [7] GU Hai-song, JI Qiang. An automated face reader for fatigue detection [C]//Proc of the 6th IEEE International Conference on Automatic Face and Gesture Recognition. 2004: 111-116.
- [8] 董占勋, 孙守迁, 吴群等. 心率变异性与驾驶疲劳相关性研究[J]. 浙江大学学报(工学版), 2010, 44(1): 46-50.
- [9] Sahayadha A, Sundaraj K, Murugappan M. Electromyogram signal based hypovigilance detection[J]. Biomedical Research, 2014, 25(3): 281-288.

- [10] Katsis C D, Ntouvas N E, Bafas C G, et al. Assessment of muscle fatigue during driving using surface EMG[C]. Proceedings of the IASTED International Conference on Biomedical Engineering, 2004: 259-262.
- [11] Hostens I, Ramon H. Assessment of muscle fatigue in low level monotonous task performance during car driving[J]. Journal of Electromyography and Kinesiology, 2005,15(3): 266-274.
- [12] Ohsuga M, Kamakura Y, Inoue Y, et al. Classification of blink waveforms toward the assessment of driver's arousal levels: an EOG approach and the correlation with physiological measures[C]//Proc of the 7th International Conference On Engineering Psychology and Cognitive Ergonomics. Berlin: Springer, 2007: 787-795.
- [13] Watson T, Krause J, LE J, et al. Vehicle Integrated Non-Intrusive Monitoring of Driver Biological Signals, PB 2011-01-1095[R]. Warrendale: SAE international, 2011.
- [14] Wierwille W W, Wreggit S S, Kirn C L, et al. Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness. final report[R]. 1994.
- [15] Dingus T A, Hardee L H, Wierwille W W. Detection of drowsy and intoxicated drivers based on highway driving performance measures[J]. Vehicle Simulation Laboratory, Human Factor Group. Virginia Polytechnic Institute and State University, 1985.
- [16] Dwivedi K, Biswaranjan K, Sethi A. drowsy driver detection using representation learning[C]. International Advance Computing Conference. IEEE, 2014: 995-999.
- [17] Grace R, Byrne V E, Bierman D M, et al. A drowsy driver detection system for heavy vehicles[C]//17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No. 98CH36267). IEEE, 1998.
- [18] Wang Rong-ben, Guo Ke-you, Shi Shu-ming, et al. A monitoring method of driver fatigue behavior based on machine vision[C]//Proc of IEEE Intelligent Vehicles Symposium. 2003: 110-113.

- [19] Ito T, Mita S, Kozuka K, et al. Driver blink measurement by the motion picture processing and its application to drowsiness detection[C]//Proc of International Conference on Intelligent Transportation Systems. 2003: 168-173.
- [20] Boverie S, Lequellec J M, Hirl A. Intelligent systems for video monitoring of vehicle cockpit[C]//Proc of International Congress and Exposition ITS: Advanced Controls and Vehicle Navigation Systems. 1998: 1-5.
- [21] Chu Jiang-wei, Jin Li-sheng, Tong Bin-liang, et al. A monitoring method of driver mouth behavior based on machine vision[C]//Proc of IEEE Intelligent Vehicles Symposium. 2004: 351-356.
- [22] Zhang W, Yi L M, Wang T, et al. Driver yawning detection based deep convolutional neural learning and robust nose tracking[C]. International Joint Conference on Neural Networks. IEEE, 2015: 1-8.
- [23] 李绍文, 王江波. 驾驶员疲劳检测系统的研究[J]. 计算机工程与应用, 2013, 49(15): 253-258.
- [24] Kithil P W, Jones R D, Jone M. Development of driver alertness detection systems using overhead capacitive sensor array, SAE Technical Paper Series 982292 [R]. [S. l.]: Advanced Safety Concepts Inc, 1998.
- [25] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In Conference on Computer Vision and Pattern Recognition (CVPR), 2011.
- [26] 王天本, 王海鹏, 周兴社, 倪红波, 林强, 基于键鼠事件的疲劳感知[J]. 计算机应用, 2014, 34(1): 227-231.
- [27] Komandu R S, Johnson P W, Storch R L. Relation between mouse button click duration and muscle contraction time[C]// Proceedings of the 2008 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Piscataway, NJ: IEEE Press, 2008: 2299-2301.
- [28] Hong W B, Chen C Y, Chang Y, Fan C H. Driver fatigue detection based on eye tracking and dynamic template matching [C]. Networking Sensing and Control, 2004, 1:7~12.

附录：部分主要程序

```
import cv2
import torch
import time
from process import *
import models
import os
import argparse

parser = argparse.ArgumentParser(description='age recognition:video and image ')
parser.add_argument('-a', '--arch', type=str, default='resnet50',
                    choices=models.get_names())
parser.add_argument('--resume', type=str, default='log-FatigueFace/6-7-
resnet50/best_model.pth.tar', metavar='path')
parser.add_argument('--gpu-devices', default='0', type=str, help='gpu device ids for
cuda_visible_devices')
parser.add_argument('--threshold', default=0.5, type=int, help='gpu device ids for
cuda_visible_devices')
parser.add_argument('--det_time_length', default=10, type=int, help='gpu device ids
for cuda_visible_devices')
args = parser.parse_args()

if __name__ == '__main__':
    cap = cv2.VideoCapture(0) # 0 的是自带， 1 是外设摄像头
    # video_writer =
cv2.VideoWriter('./Test/VID/demo.avi',cv2.VideoWriter_fourcc('X','V','I','D'),5,(600,4
80))
    ret,img = cap.read()
    # img = cv2.resize(img, (600, 480), interpolation=cv2.INTER_AREA)
    model = Init_model(args.resume,args.arch)
    count = 0
    fatigue_count = 0
    mark = 0
    while ret:
        face_img, is_face, face_locations = face_det(img)
        print(face_locations)
        if (face_locations != None):
            # kind = recognition(face_img, args.resume, args.arch)
            kind = Fatigue_recognition(model, face_img)
```

```
result = int(kind.split(":")[0])
if(result in [1,4,7,2,5,8]):
    fatigue_count +=1
print('程序员状态: %s' % kind)
img = Information_show(img, face_locations, kind)
cv2.imshow('face', face_img)
count += 1
else:
    print("未检测到人脸")
    img = Fatigue_Information_show(img, "can't find face")

if(count==args.det_time_length):
    rate = float(fatigue_count/count)
    if(rate>args.threshold):
        print("系统检测到您存在疲劳状态，请注意休息！")
        mark = 1
    else:
        mark = 0
    count = 0
    fatigue_count = 0
if(mark):
    img = Fatigue_Information_show(img, "FATIGUE!",w=350)
cv2.imshow('image', img)
# video_writer.write(img)
cv2.waitKey(5)
ret, img = cap.read()
cap.release()
# video_writer.release()
cv2.destroyAllWindows()
```

致 谢

大学四年匆匆而过，入学时的场景似乎仍旧历历在目。从初中高中一路走来，大学生活或许是对一个人影响最大的四年。这是我们褪去稚气的四年，也是我们走向成熟学会担当的四年。这些日子里有欢笑，有泪水，有成功的喜悦，也有不肯止步的不甘。看似漫长的四年也一下就走到了尾声，在这四年中，有很多很多人鼓励了我，帮助了我。对我的父母，我的老师，我的朋友们有说不尽的感谢。

首先，我要谢谢母校浙江工业大学，她给了我一个安定和谐温馨的环境，也正是她，我才能认识这些给了莫大帮助的同学老师们。也让我度过了四年快乐的本科生活。

其次，我想谢谢毕业设计的指导老师，赵云波老师。疫情期间，老师定期为我们开展线上会议，很多毕业设计中自己难以解决的问题，在老师的教导下都得到了非常好的处理。老师的提点经常会让我有种茅塞顿开的感觉，可以发现自己疏忽的细节点。

我还想谢谢实验室中的学长学姐们，在我做毕业设计的过程中，经常拿我自己无法解决的问题麻烦他们。他们还都非常热心，引导着我，和我一起探讨解决方法。他们非常热情平和，就像普通朋友一样和我相处，为我排忧解难。

我也要谢谢我的父母，是他们一直支持着我，鼓励着我，引导着我走出更好的人生轨迹。在我失落的时候开导我，我成功的时候祝福我。他们是我人生中最重要的人，最亲近的人，最可爱的人，也是我最要感谢的人。

大学四年中有自豪，也有失意，未来路还很长，请不要轻言放弃。困难中满是磨砺，砥砺前行，才能见证更多姿多彩的人生，加油。