



华南农业大学

## 本科毕业论文

基于 3D 卷积神经网络的深度学习人体行为识别

程航驰

201825010104

指导教师 李康顺 教授

学 院 名 称

数学与信息学院

专 业 名 称

计算机科学与技术

论文提交日期

2022 年 4 月 15 日

论文答辩日期

2022 年 5 月 8 日

## 摘 要

人体行为识别尤其是基于视频的人体行为识别是今年来的研究热点之一，由于它在安防系统、异常监控、智能家居等等方面的作用，人们的生产生活变得更安全、便捷，与此同时，它也带来了不少棘手的麻烦，常见的问题像比如如何在复杂的环境中仍然能提取到特定的特征、如何能在识别时候减少光线的影响、能识别的动作类别只有那么寥寥几种等。在以前，人体行为识别主要是通过人工提取特征的，也就是需要自己标特征点和贴标签再进行训练。这种人体行为识别的模型明显较为低效而且难以应用到大数据集。为了更加高效且精确地识别人体行为，有人提出了一种基于深度学习的人体行为识别方法，相较于以前的方法不再需要人工提取特征从而大大提高了效率。

而在基于深度学习的人体行为识别当中，查阅了近这些年来国内外的热点论文以及各类相关比赛活动之类的作品等等，可以得出这里边使用最广泛的应该是可以分为这三大类：基于双流卷积神经网络(two stream)的人体行为识别、基于时空网络(CNN-LSTM)的人体行为识别，还有基于 3D 卷积神经网络(C3D)的人体行为识别，这其中经过比较，本论文最终选择了使用识别最高效的 3D 卷积神经网络。

本论文在撰写代码的过程中使用了 Keras，它是包含在 Tensorflow 中并基于其中的 Python API 的高级神经网络 API，大大简化了神经网络的撰写过程，其中提供的一些工具比如通过 ImageDataGenerator(图片数据生成器)直接获得一个测试集或者让测试数据变得更加随机化从而使得结果更可靠，这些工具也带来极大的便利，另外在建立模型时选择默认参数，Keras 还会自动下载已经在 ImageNet 上训练好的参数得到一个预训练模型，使得后面的训练效率大大提升。

本论文系统整体的流程分为数据选取及其预处理、模型训练还有评估与实测，其中数据方面选取了 UCF101 这一经典数据集并在其处理中使用 FFmpeg 并对视频进行转换，模型方面选择了 InceptionV3 并将其顶层分类器(也就是全连接层)去除重写然后进行冻结训练，在评估与实测方面，进行评估时让该模型检测测试集中所有数据并得出准确率与 top-5 准确率，实测分为两种检测，一个实测是测试自己的数据集，从中任意挑选五个视频并输出自己判断可能性 top-5 的动作名及它们的可能性大小，另一个则是可以自己上传自己的任意视频，让模型识别出它可能性最大的动作并输出的动作名，若有多个动作并希望判别异常，还能识别多个，并判断其中较为异常的行为。

**关键词：**C3D Keras InceptionV3 深度学习 行为识别

# Deep Learning Human Action Recognition Classification Based on C3D

Cheng Hangchi

(College of Mathematics and Informatics, South China Agricultural University, Guangzhou  
510642, China)

**Abstract:** Human Action Recognition, especially video-based human body recognition, is one of the research hotspots recently. Due to its role in security systems, abnormal monitor, smart homes, etc., people's production and life have become safer and more convenient. , it also brings a lot of troubles, common problems such as how to still extract specific features in complex environments, how to reduce the influence of light during recognition, and only a few action categories that can be recognized species etc. In the past, human behavior recognition was mainly performed by manually extracting features, that is, you needed to mark feature points and label yourself before training. This model of human action recognition is obviously inefficient and difficult to apply to large datasets. In order to recognize human behavior more efficiently and accurately, a human behavior recognition method based on deep learning is proposed. Compared with the previous methods, the method based on deep learning does not need to manually extract features, which greatly improves the efficiency.

In the human body recognition based on deep learning, according to the hot papers at home and abroad this year and some competitions, etc., the most widely used here should be divided into these three categories: based on two-stream convolutional neural network ( two streams), human body recognition based on spatiotemporal network (CNN-LSTM), and human body recognition based on 3D convolutional neural network (C3D). After comparison, I finally chose to use 3D convolutional neural network ( For the comparison process, see Chapter 3 of this paper).

In the process of writing the code, I used Keras, a high-level neural network API that is now included in Tensorflow and based on the Python API, which greatly simplifies the writing process of neural networks. ) directly obtain a test set or make the test data more random to make the results more reliable. These tools also bring me great convenience. In addition, if the default parameters are selected when building the model, Keras will automatically download The parameters that have been trained on ImageNet get a pre-trained model, which will greatly

improve the efficiency of subsequent training.

The overall process is divided into data selection and preprocessing, model training, evaluation and actual measurement. In terms of data, I selected the classic data set UCF101 and used FFmpeg to convert the video in its processing. In terms of model, we chose InceptionV3 Remove and rewrite its top-level classifier (that is, the fully connected layer) and then freeze the training. In terms of evaluation and actual measurement, we let the model detect all the data in the test set by itself and get its own accuracy rate and actual measurement. The top-5 accuracy rate, the actual measurement, I divided into two types of tests, one is to test my own data set, randomly select five videos from it and output the action names and their probability of the top-5 possibility, and the other You can upload your own arbitrary video, and let the model identify its most likely action and output the action name. If there are multiple actions and you want to identify the abnormality, you can also identify more than one, and judge the more abnormal behavior.

**Key words:**C3D Keras InceptionV3 Deep Learning Action Recognition

# 目 录

1 引言.....	1
1.1 研究背景、目的与意义.....	1
1.2 研究现状及主要研究方法.....	1
1.2.1 基于 3D 卷积的方法.....	2
1.2.2 基于双流网络的方法.....	3
1.2.3 基于循环神经网络的方法.....	4
1.3 本论文的结构安排.....	5
2 开发平台以及技术介绍.....	6
2.1 PyCharm.....	6
2.1.1 PyCharm 简介.....	6
2.1.2 PyCharm 特性.....	6
2.2 Tensorflow.....	6
2.2.1 Tensorflow 简介.....	6
2.2.2 Tensorflow 特性.....	6
2.3 Keras.....	7
2.3.1 Keras 简介.....	7
2.3.2 Keras 特性.....	7
2.4 CNN .....	7
2.4.1 CNN 简介.....	7
2.4.2 CNN 特性.....	7
2.5 FFmpeg.....	7
2.5.1 FFmpeg 简介.....	7
2.5.2 FFmpeg 特性.....	8
2.6 Numpy.....	8
2.6.1 Numpy 简介.....	8
2.6.2 Numpy 特性.....	8
2.7 UCF101.....	9
2.7.1 UCF101 简介.....	9

2.7.2 UCF101 特性.....	9
2.8 Anaconda.....	9
2.8.1 Anaconda 简介.....	9
2.8.2 Anaconda 特性.....	9
2.9 PyQt6.....	10
2.9.1 PyQt6 简介.....	10
2.9.2 PyQt6 特性.....	10
2.10 本章小结.....	11
3 算法设计.....	12
3.1 基于深度学习的人体行为识别.....	12
3.2 3D 卷积神经网络.....	12
3.3 InceptionV3 模型.....	14
3.4 本章小结.....	18
4 实验设计及实验结果分析.....	19
4.1 实验设计.....	19
4.1.1 实验数据集选择.....	19
4.1.2 数据集划分与预处理.....	19
4.1.3 模型初始化.....	19
4.1.4 模型预训练与训练.....	20
4.1.5 模型评估与测试.....	20
4.2 实验过程及结果分析.....	21
4.2.1 训练模型.....	21
4.2.2 模型评估.....	31
4.2.3 模型测试.....	32
4.3 本章小结.....	36
5 总结与展望.....	37
5.1 总结.....	37
5.2 展望.....	37
参考文献.....	39
致谢.....	41

# 1 引言

## 1.1 研究背景、目的与意义

近些年人工智能发展飞速，各种关于它的传闻报道逐渐出现在大众视野面前，其中基于视频智能分析的人体行为识别虽然可能大多数人听着陌生，但是它却已经在很多方面开始应用并服务于人们了。在现代生活中，视频变得越来越重要。与图像和音频相比，视频可以表达更丰富的情感和信息，这已成为人类生活不可或缺的一部分。除了观看外，视频还可以通过计算机、智能手机、平板电脑和相机等设备、通过微博、微信、Facebook 等等社交平台共享许多用户内容。因此，在可预见的未来，视频将成为人与人之间沟通的重要方式，人作为视频内容的绝对主体，如何让机器了解视频中人的行为具有重要意义。（李智敏，2018）

在公共安全领域，通过行为识别技术可以检测打架斗殴、持刀抢劫等违反治安法的暴力行为，尽可能减少因此所造成的人员伤害和财产损失；在智慧交通领域，通过行为识别技术可以自动判别如行人/车辆 闯红灯、驾驶员不安全驾驶等交通违法行为，保障人们出行安全；在医疗监护领域，通过该技术可实现对患者的实时监控和意外跌倒检测等，确保患者能够得到及时治疗和帮助；在安全生产领域，可以实现对生产作业全过程的实时监测，对作业生产过程中出现的可能导致安全隐患的行为及时报警，确保作业生产在安全可控范围内进行，保障人员的人身安全和财产安全。（陈煜平，2019）

由此可见，基于视频分析的人体行为识别技术与人们的生产生活安全息息相关，对其研究具有深远的意义。

## 1.2 研究现状及主要研究方法

基于视频分析的行为识别任务需建立动作、姿态样本库，并对所设计模型进行训练，以实现视频行为的分类。行为识别的划分，根据其提取特征手段不同可以分为传统方法和基于深度学习的方法。其中，传统方法依赖手工对特征提取，由于早期样本库数据量小，场景简单，动作单一，它可以满足某些简单场景下的应用并识别部分动作，但是随着视频监控技术的普及，各种应用场景变得越来越复杂，用传统方式提取的视频特征在识别准确度以及复杂度上逐渐难以达标实际需要，其价值难以体现。卷积神经网络 (Convolutional Neural Network) 的出现则很好地解决了传统方式的局限性，取得了较好的效果。（曾如平，2019）

在 CNN 中有很多种网络模型，其中不少可以用于图像识别，经过加上时间维度或者

将输入流改为光流等等之类的方法，可以很好地实现基于视频的识别。这其中在当下最流行的是基于 3D 卷积的方法、基于双流网络的方法、基于循环神经网络的方法。（盖刚勇，2021）

1.2.1 基于 3D 卷积的方法

3D 卷积无疑是最为简单直接的方法，也是应用历史最久的一种方法，在这方面已经有了大量研究成果，3D 卷积的输入不是直接的视频而是需要先经过预处理的视频的连续帧，这些视频帧的连续便是一种时间维度，换句话说，这些连续的视频帧之中含有着关于该视频的短期的时间消息，3D 卷积相比于其它方法的特点是可以同时分析对这些连续的视频帧，在像一般卷积那样处理 2D 信息也就是空间信息时也同时对加入了时间维对时间也一起进行分析，3D 卷积输出的矩阵是视频的时空特征，最后通过分类层实现行为识别。Ji 提出的 C3D 卷积神经网络在人体行为识别领域具有开创性意义，其使用 Hardwire 层将每一个视频帧都处理成为灰度通道、x 梯度、y 梯度、x 光流、y 光流等 5 个通道之后，使用 3D 卷积核进行卷积操作，在经过池化后使用 softmax 实现行为分类；Carreira J 等将 Inception 网络中的 2D 卷积扩张成为 3D 卷积得到 I3D 网络，用于实现行为识别，该网络同时融合双流网络的思想，不仅在准确率上有明显的提升，而且相较于初代的 C3D 网络具有更少的参数；Qiu 等提出 P3D 网络，将  $3\times3\times3$  的 3D 卷积分为空间维的  $1\times3\times3$  的 2D 卷积和时间维的  $3\times1\times1$  的 1D 卷积分别提取空间和时间特征，通过对时空处理模块进行不同的组合来进行视频分析。卷积过程如下图 1 所示。

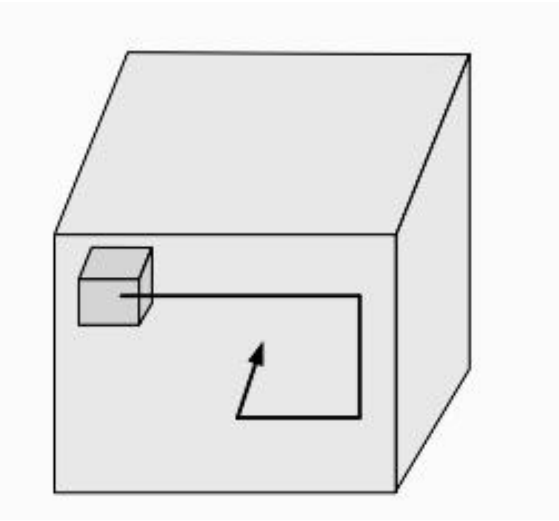


图 1 3D 卷积



### 1.2.2 基于双流网络的方法

双流网络相比于另外两种方法它最大的特点是将视频的空间和时间分成了两个分支网络，也就是使用两个分支网络，一个记载空间信息，一个记载时间信息。各种双流网络的差异之处就在于空间流和时间流网络的不同。两个分支网络一般都采用迁移学习的思想，将预训练好的网络直接用来提取时空特征，最终将提取到的计算机应用 信息技术与信息化时空特征融合之后，通过分类层实现行为的分类。Simonyan 等提出将空间特征和时间特征分开提取的双流网络，空间流网络利用单帧视频图像作为输入，时间维利用光流信息作为输入，二者 softmax 之后进行融合；Wang 等提出 TSN(temporal segment network) 网络，将视频切分为 K 个 segment，同样采用双流网络的结构，得出每一个 segment 的 class scores，最后进行融合得出最终分类结果，提高了识别的准确率；C. Feichtenhofer 等将 ResNet 的 2D 卷积扩展成 3D 卷积作为双流网络的基础网络进行行为识别。双流网络模型图如下图 2 所示。

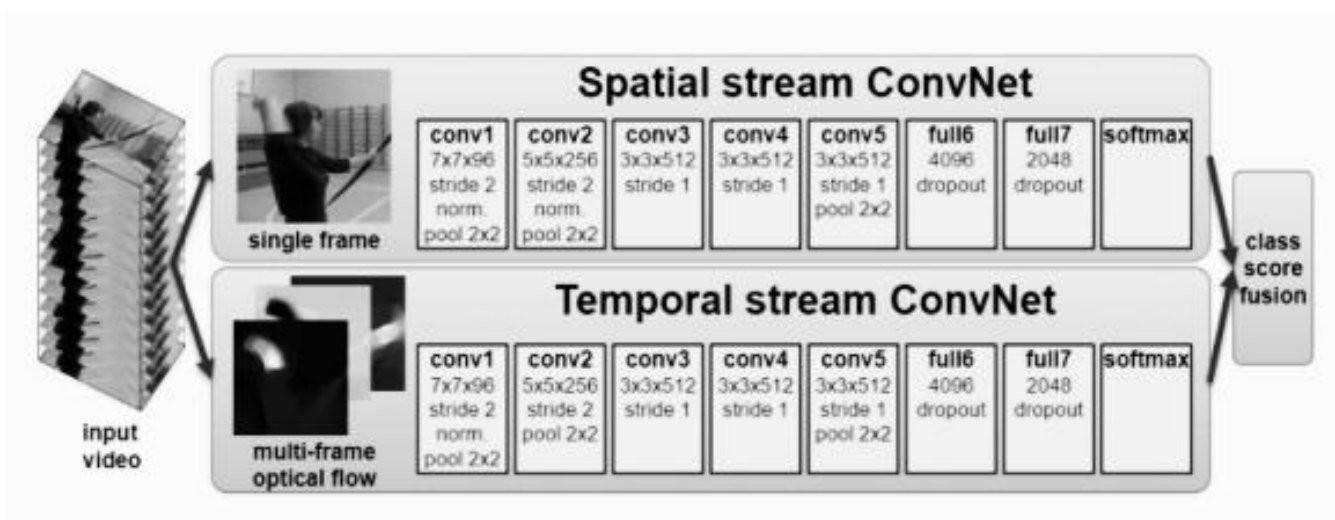


图 2 双流网络

由于在双流网络中单独使用了一个分支来分析时间，所以能够挖掘较长时间内的特征，这是它最大的优点，而且它的结构也有着很多可以进步的地方，如果网络设计合理，双流网络能够取得非常好的识别准确率；但是它也有一些不可避免的缺点，无法准确确定在何时融合两个分支网络提取到的特征，时空特征割裂之后，难以准确的融合，会对识别的准确率造成很大影响。

### 1.2.3 基于循环神经网络的方法

循环神经网络由于其强大的时序处理能力，最开始广泛使用在自然语言处理领域，尤其以 LSTM 的应用最为广泛，因为其能够有效解决循环神经网络梯度消失的问题。考虑到视频同样具有很强的时间相关性，因此将 LSTM 应用到视频处理中，如图 3 所示。LSTM 的输入是一个时间序列，因此，在 LSTM 的输入处需要进行预处理，将视频处理成一个时间序列，这个时间序列包含了视频的空间特征，然后依次输入到 LSTM 中，利用 LSTM 内部的门机制，可以有效挖掘视频中的长期的时间特征，最后将 LSTM 的输出送到分类器中，实现行为分类基于 LSTM 的方法的优点在于，LSTM 能够处理长期的时间信息，在时间信息的处理上更加高效，缺点在于识别速度较慢，其主要原因在于其需要先将视频处理成 LSTM 能够识别的时间序列，这在一定程度上影响了识别速度。

以上所介绍的三大类方法很多人都在使用它们来完成自己的人体行为识别，并尽力改善自己的算法以达到更高的准确率或效率，个人经过对比后，认为双流网络难以准确融合将影响到准确率，而循环神经网络则多了一步处理视频的步骤，效率低，识别速度慢，3D 卷积最为直接，效率最高，也有较好的准确率，最后决定研究基于 3D 卷积的方法，并且兼顾效率、准确率以及方便性。

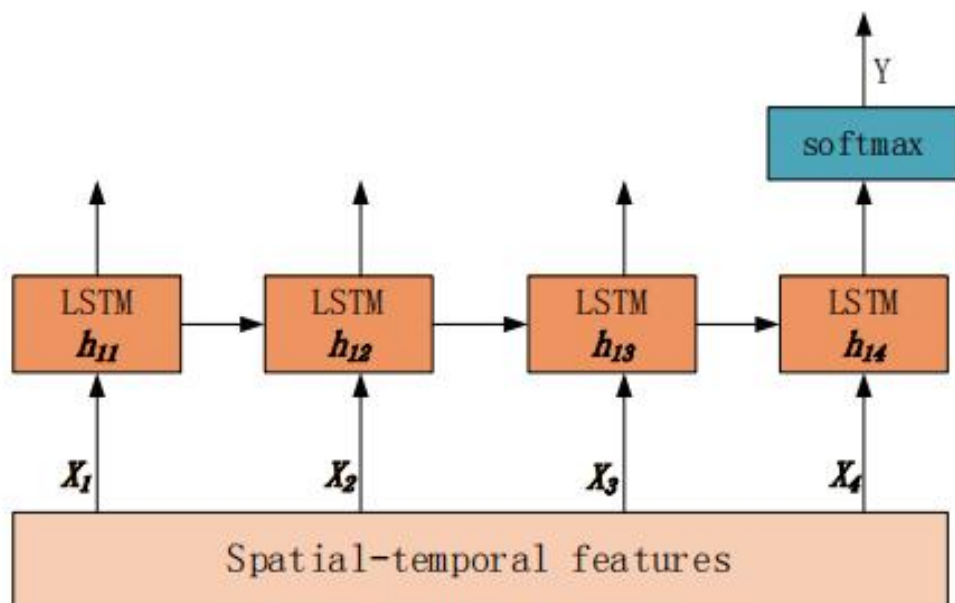


图 3 LSTM 网络模型

### 1.3 本论文的结构安排

此部分给大家介绍一下本篇论文的结构安排，具体情况如下所述：

第 2 章主要阐述了基于 3D 卷积神经网络的深度学习人体行为识别系统开发过程中使用到的开发平台和技术。基于深度学习的人体行为识别系统开发平台选择的是 PyCharm Community Edition 2021.3.2，数据集选择了用经典数据集 UCF101，开发语言选用了 Python，使用 CNN（卷积神经网络）进行训练，其中选用的模型是 inception-V3，数据预处理时使用了 FFmpeg 将视频转化为易处理的图片，为了更好使用系统性能，使用了 TensorFlow 对 GPU 进行调用使训练速度加快，使用了现封装于 TensorFlow 中的高级 API，Keras 使得整个神经网络编写过程大大简化。

第 3 章是算法的总体设计，从关于整个人体行为识别的经典算法大类说起，由上而下地说明了为何选取了基于深度学习的方法，又为何选择了 InceptionV3 这一模型，以及如何对它进行使用与改写。

第 4 章是实验流程的设计和结果分析，实验流程首先是关于数据集的选择，然后是关于数据集的预处理与划分，再其次是关于模型的预训练与训练，最后是用模型进行对模型进行评估以及用它进行实测。实验结果则是按照实验实验流程的步骤，自上而下地进行测试与分析还有同步改进。

第 5 章是个人的总结与展望。在这一章会对此论文的内容做一个总结，然后指出自己部分不足之处，最后阐述一些对未来的展望。

## 2 开发平台以及技术介绍

### 2.1 PyCharm

#### 2.1.1 PyCharm 简介

PyCharm 是一种 Python IDE, 这个 IDE 对于 Python 的开发非常友好, 里面有许多功能之类的大大地方便了开发大大提高了编写效率, 比如说可以进行调试检验、左栏目中可以直接管理工程、代码中进行单或多查找替换、自动标红及智能提醒修改方法、工具包软件包的下载于管理、还能直接在其中打开一个终端, 这样子就可以很方便地在编写的同时使用终端功能辅助编写。

#### 2.1.2 PyCharm 特性

PyCharm 具有如下特性:

(1)PyCharm 拥有一般 IDE 具备的功能, 比如说可以进行调试检验、左栏目中可以直接管理工程、代码中进行单或多查找替换、自动标红及智能提醒修改方法、工具包软件包的下载于管理、还能直接在其中打开一个终端, 这样子就可以很方便地在编写的同时使用终端功能辅助编写。

(2)除此之外, PyCharm 还提供了一些较高级的功能用于其它一些专业开发, 当然那些在本文中暂时还用不上

### 2.2 Tensorflow

#### 2.2.1 Tensorflow 简介

TensorFlow 表达了高层次的机器学习计算, 大幅简化了第一代系统, 并且具备更好的灵活性和可延展性. TensorFlow, 把这个单词拆分开来就成了 Tensor 和 Flow, 可以这样理解 Tensor 和 Flow 是组成 TensorFlow 的基础, 两者的融合就是 Tensorflow; Tensor 意味着 data, Flow 意味着流动, 意味着运算, 意味着映照, 即数据的流动, 数据的计算, 数据的映射, 同时也体现数据是有向的流动、计算和映射.

#### 2.2.2 Tensorflow 特性

Tensorflow 具有如下特性:

(1)易用性: TensorFlow 的工作流, 即使是编程新手也能很快理解, 同时还和 NumPy 很好地融合了在一起, 对于了解 Python 数据学的老手来说非常容易上手, 另外它的 API 始终保持相似性, 大大地方便了程序员。

(2)运算性能：可以选择 CPU 或 GPU（或者同时使用它们俩个）进行训练且能看见训练过程。

(3)多种环境支持：TensorFlow 支持多种环境，比如说它可以在电脑上的 CPU 或 GPU 环境下运行，再比如说在 windows 或 Linux 下都能运行。

## 2.3 Keras

### 2.3.1 Keras 简介

Keras 是一个基于 TensorFlow 的高阶神经网络 API，同时可以支持 TensorFlow、Theano 和 Microsoft-CNTK。这个高级 API 在 Tensorflow 中常常用来构建和训练模型，其操作简单易懂，大大简化了神经网络的编写，对于学习使用 Tensorflow 的新手非常友好。Keras 以 TensorFlow 的 Python API 为基础提供了一个将神经网络的编写、训练、配置、评估、测试等等这些各项操作进行封装的神经网络，这个封装了基础操作的神经网络不但使得神经网络的编写过程变得非常简单而且还让它的拓展也变得简单可行。除此之外，在 Keras 里面还封装着很多实用工具，比如 ImageDataGenerator(图片数据生成器)，通过它可以通过设定旋转、拉伸、倾斜等等随机化的程度区间，来随机获得大量检验数据，用于测试数据非常方便。

### 2.3.2 Keras 特性

Keras 的特性如下：

- (1)允许简单快速的原型设计（用户友好性，模块化和可扩展性）
- (2)支持卷积神经网络和循环神经网络，以及两者的组合
- (3)可以在 CPU 和 GPU 上无缝运行和切换

## 2.4 CNN

### 2.4.1 CNN 简介

CNN 的全称是"Convolutional Neural Network"(卷积神经网络)。神经网络是一种模仿生物神经网络（动物的中枢神经系统，特别是大脑）结构和功能的数学模型或计算模型 CNN 由三个部分构成：卷积层、池化层、全连接层，卷积层(Convolutional Layer)：卷积就是多个函数的叠加，应用在图像上则可以理解为拿一个滤镜放在图像上，找出图像中的某些特征，而需要找到很多特征才能区分某一物体比如要区别出猫和狗，它们的特征有相同之处都是四足、都是两个眼睛等

等，因此需要足够多特征保证更精确的区分，所以需要有很多滤镜，通过这些滤镜的组合，可以得出很多的特征，这一层的功能就是通过多个函数的叠加找出图像的多个特征；池化层(Max Pooling Layer)，经过卷积层处理的特征并不能直接拿来分类，道理很简单，即使是一张仅仅 100 x 100 的图片经过 10 个 Filter 的卷积层之后得到的结果也有 100 x 100 x 50 这么大，因此需要在不影响识别效果的情况下减少数据大小，因此就有了池化层，它的作用是负责下采样(downsampling)，比如可以将一个 16 x 16 的矩阵划分为 16 个的 4 x 4 的矩阵块，然后在每个 4 x 4 的矩阵里，都取其中最大的值保存下来，这样子取舍数据并不会影响到后来的识别，因为每次保存的都是最显著的特征丢掉其它无用信息，池化层的引入还保证了平移不变性，即同样的图像经过翻转变形之后，通过池化层，可以得到相似的结果，这样子就可以在不影响识别效果的情况下大幅降低参数量级；全连接层(Fully Connected Layer)则是最后的顶层，它主要是用于分类，因此也可以叫顶层分类器，前面在经过了卷积层的提取和池化层的降维后得出的特征，在全连接层会对这些已经总结好的特征再进行分类，每个神经元不一定是相同的重要性，它会根据不同神经元返回比重的不同，最后自己调节权重并反馈回网络从而得到分类的结果。

#### 2.4.2 CNN 特性

CNN 的特性如下：

- (1)能够将较大的图片通过提取特征和降维后转为较小的图片；
- (2)能够有效地保留图片特征，符合图片处理的原则；
- (3)跟神经网络模型中其它各大其他模型相比(比如 BP 神经网络，RNN 神经网络等)，卷积神经网络最大的区别就是它运用了卷积运算操作 (Convolutional operators)。

#### 2.5 FFmpeg

FFmpeg 是一个能将视频包括各类音频类文件比如 avi 文件、mp4 文件、gif 文件等等按照一定帧率转变为多帧图片的流数据的开源计算机程序。此处使用该软件处理数据，也就是用该软件将视频转化为电脑可以处理的连续视频帧，同时用一定规范的命名更加大大简化了数据的处理，由于该程序属于外部程序，因此在使用时需首先从外部下载，在 python 中使用 call 函数来调用外部程序。

## 2.6 Numpy

### 2.6.1 Numpy 简介

NumPy(Numerical Python) 是基于 Python 语言的一个扩展程序库，它不但支持着大量的维度数组与矩阵运算，而且针对数组运算提供了大量的数学函数库。

### 2.6.2 Numpy 特性

Numpy 的特性如下：

(1)运算速度非常快，效率高；

(2)支持大量的维度数组与矩阵运算，也针对数组运算提供大量的数学函数库。

## 2.7 UCF101

### 2.7.1 UCF101 简介

UCF101 是一个动作识别数据集，其中大部分动作是来自于 YouTube 中的真实的随机动作，具有较强随机性同时也说明更有价值意义，这些动作被分为了 101 种，总共有 13320 个视频每个视频都被分类到相对应动作名的文件夹下，另外还将它们分为了 25 个组，同组的动作会有一些类似的特征，比如差不多的动作或者背景等等，类型非常丰富，且数据具备一些随机性，挑战难度较大的同时也让训练之后的模型变得非常可靠，可以在一些杂乱环境下较为精准地识别出相应的或者类似的动作

### 2.7.2 UCF101 特性

UCF101 的特性如下：

(1)在动作方面提供了较大的多样性；

(2)会出现一些随机因素比如镜头乱晃，背景嘈杂；

(3)在光线、视角、动作幅度、目标大小方面会有很大的随机性。

## 2.8 Anaconda

### 2.8.1 Anaconda 简介

Anaconda 是一个对环境进行统一管理的发行版本，而且通过(里面的 pip 工具)可以方便快捷地安装和管理自己需要或不需要的包。在 Anaconda 包含了 conda、Python 在内的超过 180 个科学包及其依赖项。Python 是一种面向对象的解释型计算机程序设计语言，它具有跨平台的特点，可以在多种系统中比如像

Linux、Mac 又或是 Windows 中搭建环境并使用，而在任一平台编写的 Python 代码放到其它各种不同平台上时，一般不用多少改动就能够运行了，它的可移植性方便了不同平台程序员之间的协作或者需要改换环境的程序员。除此之外，Python 的使用得非常地广，在游戏研发、科学研究、WEB 开发、大数据、经济类、人工智能等等各种热门领域中均发挥着关键作用。而要实现如此多而复杂的功能，得益于 Python 中各种各样的库。由于这些库的数量庞大导致它们注定会非常繁杂，如何管理它们成为了一个非常头疼的问题，而 Anaconda 的出现很好解决了这一问题。Anaconda 可以通过 pip 工具来方便快捷地对它们进行统一管理，有点类似于在 PyCharm 中的软件包管理。

### 2.8.2 Anaconda 特性

Anaconda 的特性如下：

- (1)开源，免费的社区支持；
- (2)安装过程简单，而且高效应用了 Python；
- (3)使用 pip 工具来管理库。

## 2.9 PyQt6

### 2.9.1 PyQt6 简介

在以前制作 Python 的界面常常需要用 Tkinter 一行行敲和调整，有点类似于前端使用 html 底层代码开发界面的感觉，总之是非常不方便的，而 PyQt 可以让仅仅通过鼠标点击拖动就能完成界面设计，不但非常方便而且还让能更精确地设计出想要的界面，对于的界面设计给予了极大便利。它的前身其实就是 Qt，或者说，它的原型就是 Qt，但是是通过 C++ 重新结构编写。在使用 PyQt6 时需要导入外部工具 QtDesigner 和 PyUIC，其中前者用于编写界面，后者则是将 ui 文件转化为可以直接运行调用的 Python 类，这两个工具又使得 PyQt 更加地便捷了，可以在 Python 中直接调用类，并改写其中的按钮、容器、文字，设置其点击事件与改变内容等等。

### 2.9.2 PyQt6 特性

PyQt6 的特性如下：

- (1)支持多种环境支持，在主流的操作系统上都可以运行；
- (2)原型是 Qt，编写语言为 C++。



## 2.10 本章小结

本章阐述了基于 Web 的农产品销售系统开发过程中用到的平台和技术。介绍了开发平台 PyCharm Community Edition 2021.3.2，介绍了所使用的经典数据集 UCF101 的来源以及其特性，先简单对 CNN（卷积神经网络）进行了总结介绍，介绍了预处理时使用的 FFmpeg 是用于处理视频的程序，以及简单描述介绍了 TensorFlow，以及为何使用 TensorFlow 和 TensorFlow 为何可以对 GPU 进行调用使训练速度加快，介绍了 Keras 这一可以使得代码简化、神经网络编写难度大大降低的第三方高阶神经网络 API，介绍了对库进行统一管理的工具 Anaconda，最后介绍了用于设计界面的 PyQt6。

### 3 算法设计

#### 3.1 基于深度学习的人体行为识别

当前人体行为识别主题是针对基于视频的人体行为识别,它相比于仅仅基于图片的识别明显更为实用、适用面更广同时当然地难度也大了很多,相比于图像来说,视频所不同的是它在多帧图片的基础上还增加了一个时间维度,而对于视频中的人体行为识别中,需要另外处理的地方就是在于如何解决视频中时间维信息,这不但是该题目的重中之重也是至今仍然困扰大家的难点。根据国内外对于该题目的研究现状,现在基于深度学习的人体行为识别方法虽然有着很难多其中步伐一些非主流却取得奇效的方法,但是要总的概括起来的话,主流还是这三种:基于 3D 卷积(C3D)的方法、基于双流网络(two stream)的方法、基于时空网络(CNN-LSTM)的方法。其中基于 3D 卷积的方法是将传统的 2D 卷积上多附加了时间维度之后变为了 3D 卷积,也就是卷积过程中除了二维特征还直接处理了视频中的时间特征;基于双流网络的方法则是以单帧 RGB 作为输入的 CNN 来处理空间维度的信息,使用以多帧密度光流场作为输入的 CNN 来处理时间维度的信息,并通过多任务训练的方法将两个行为分类的数据集联合起来(UCF101 与 HMDB),去除过拟合进而获得更好效果,所谓双流指的是空间流卷积网络(Spatial stream ConvNet,输入为单张 RGB 图像,经过一系列卷积、全连接层后接一个 softmax 输出概率分布值,识别静态图片)和时间流卷积网络(Temporal stream ConvNet,多帧图像间的光流(optical flow),同样经过一系列卷积、全连接层后接一个 softmax 输出概率分布值,识别视频动作);基于时空网络的方法将 LSTM 与 CNN 相结合能够将空间特征与时间特征更完整的进行学习,从而实现"deep in time"(LSTM 是 RNN 中的一种),它借助光流、光流+LSTM 能捕捉到更准确的动作特征。

#### 3.2 3D 卷积神经网络

对于以上所提的三大类算法,国内外已经有不少人对它们的速度都进行了比较分析,在查找最近的国内外论文比较之后最后可以得出,最高精度者仍然是基于 3D 卷积神经网络的方法,即使某些论文中对它评价不高,但也不可否认它是现今最成熟的技术且简单直接,因此这里采取了基于 3D 卷积神经网络的方法,某论文中比较靠谱的各种算法比较如表 1。

表 1 基于深度学习的不同算法精度对比

算法	UCF-101 上的准确率 (%)	HMDB-51 上的准确率 (%)
3D 卷积	96.80	75.90
双流网络	94.20	69.40
时空网络	92.80	61.30
算法	UCF-101 上的准确率 (%)	HMDB-51 上的准确率 (%)

3D 卷积是一种最直接的分析视频的方法，其输入是连续的若干个视频帧，这些连续的视频帧包含视频的短期的时间信息，3D 卷积可以对这些连续的视频帧同时进行分析，其卷积过程如图 4 所示(得自于网络)，在处理空间信息的同时也对时间信息进行分析，其输出的矩阵即是视频的时空特征，最后通过分类层实现行为识别。

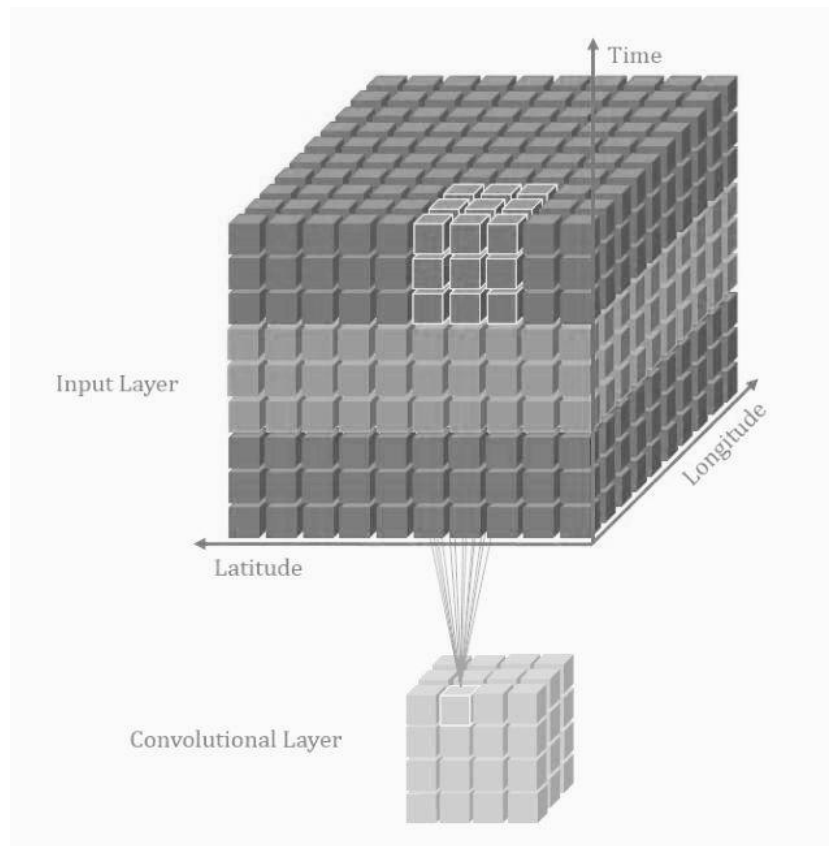


图 4 3D 卷积过程图

### 3.3 InceptionV3 模型

InceptionV3 模型是谷歌 Inception 系列里面的第三代模型，此次文中选用的模型正是 InceptionV3。InceptionV3 是一种比较经典的神经网络模型，相比于其它神经网络模型，Inception 网络最大的特点就在于 InceptionV3 将神经网络层与层之间的卷积运算进行了拓展，在 Inception 网络中采用不同大小的卷积核，使得存在不同大小的感受野，最后实现拼接达到不同尺度特征的融合。Inception-v3 架构的主要思想是分解卷积（factorized convolutions）和积极正则化（aggressive regularization），其中分解卷积的主要目的是减少参数量，分解卷积的方法有：大卷积分解成小卷积和分解为非对称卷积。

其中大卷积分解成小卷积的例子有，使用 2 个  $3 \times 3$  卷积代替一个  $5 \times 5$  卷积（关于为何能代替：对于单  $5 \times 5$  卷积计算其感受野（feature map，即使经过卷积核卷积之后的 map 上集中前一个图像或者 feature map 的特征，简单来说可以理解为某个点在前一张图像所对应的区域就是它的感受野）为  $(n + 2 * 0 - 5) / 1 + 1 = n - 4$ ，对于 2 个  $3 \times 3$  卷积则第一层  $3 \times 3$  卷积为  $(n + 2 * 0 - 3) / 1 + 1 = n - 2$ ，第二层为  $(n - 2 + 2 * 0 - 3) / 1 + 1 = n - 4$ ，两者感受野相同），其分解示意如图 5 所示，其具体结构如图 6 所示（后面称之为 Module A），这样子的话参数只需要  $2 \times 3 \times 3 = 18$  个，相比于原本的  $5 \times 5 = 25$  个，这个方法减少 28% 的参数量，而且分解后还额外多加了一个激活函数（因为每个卷积层后面跟着激活函数，以前只有一个  $5 \times 5$  卷积，所以就只有一个激活函数，现在有 2 个  $3 \times 3$  卷积了，也就有了 2 个激活函数，如果用图像解释会更加好理解，如图 5 所示，第一个  $3 \times 3$  卷积之后映射到的是一个  $3 \times 3$  图像，然后由于步长为 1，所以最后左右移动为  $3 + 3 - 1 = 5$ ，同理上下移动也是  $3 + 3 - 1 = 5$ ，所以最后映射的也是一个  $5 \times 5$  图像），也就是在减少参数量的同时增加了非线性表达的能力，甚者，这种做法还减轻了过拟合，而且在下面非对称的卷积结构拆分其结果还会比这种对称地拆分为几个相同小卷积核效果更加明显，可以处理更多、更丰富的空间特征，增加了特征的多样性。

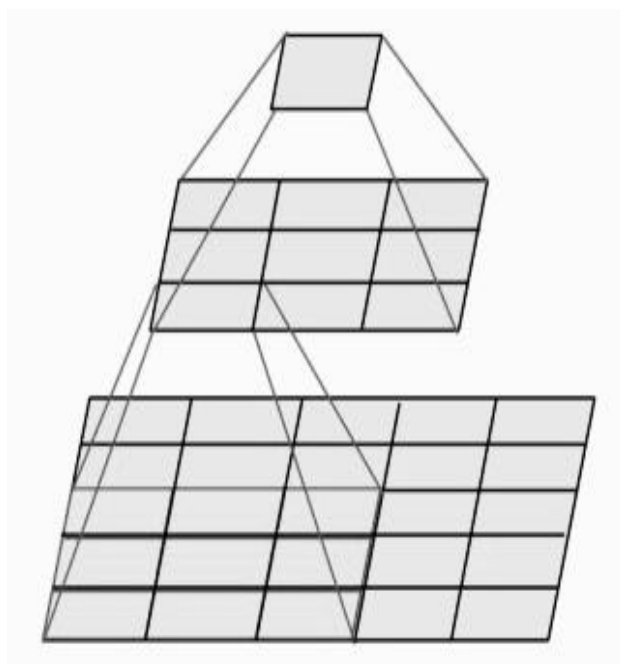


图 5  $3 \times 3$  卷积代替  $5 \times 5$  卷积

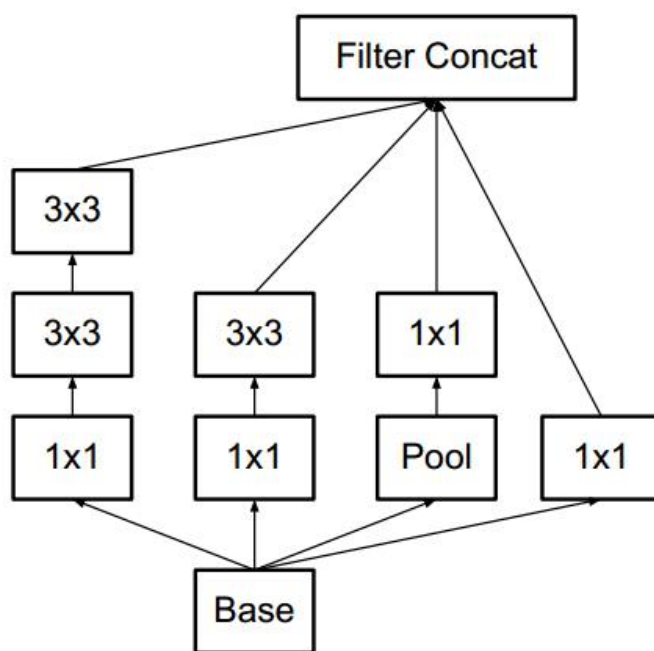


图 6 Module A，此处  $1 \times 1$  卷积核是为了降维减少计算成本，后 Module B 和 Module C 同此

分解为非对称卷积的例子则有，用 1 个  $1 \times 3$  卷积和 1 个  $3 \times 1$  卷积替换  $3 \times 3$  卷积，这样的话，输入参数量仅为  $3+3=6$ ，相比于原本  $3 \times 3=9$  的参数量，这样可以减少 33% 的参数量，所有  $n \times n$  卷积都可以这样子分为  $1 \times n$  和  $n \times 1$  的

两个卷积核，分解图如图 7 所示，模块具体结构为图 8，下文称之为 **Module B**，其它的非对称分解卷积解构如下图 9，在下文中称为 **Module C**。

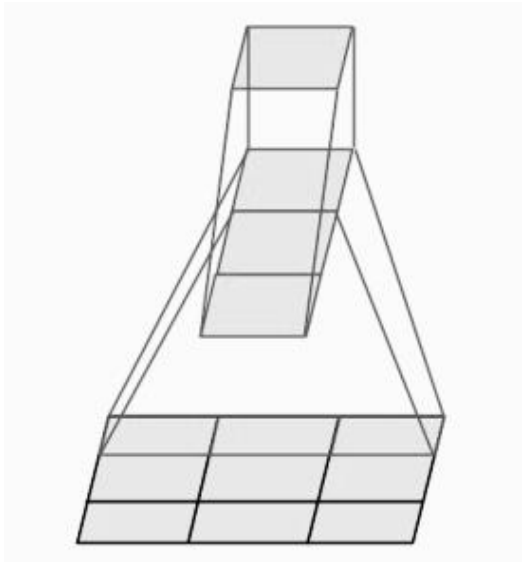


图 7  $3 \times 1$  卷积和  $1 \times 3$  卷积代替  $5 \times 5$  卷积

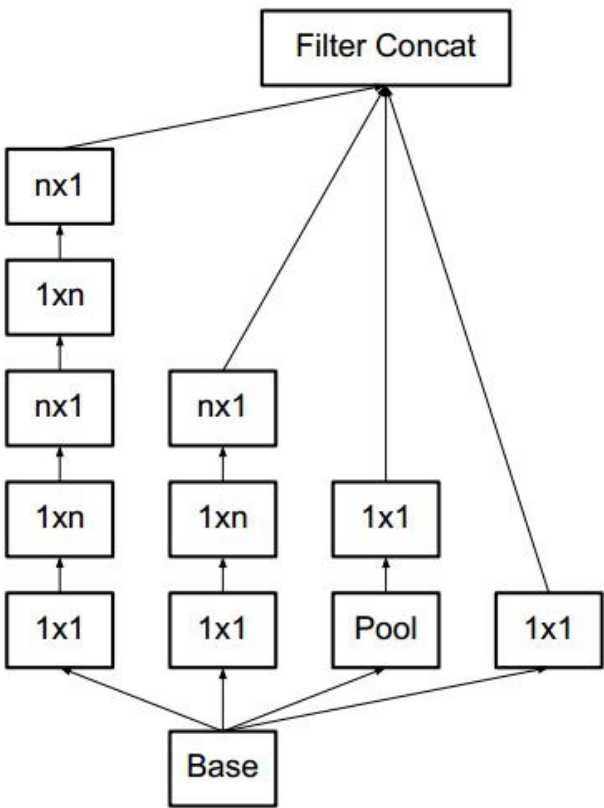


图 8 Module B

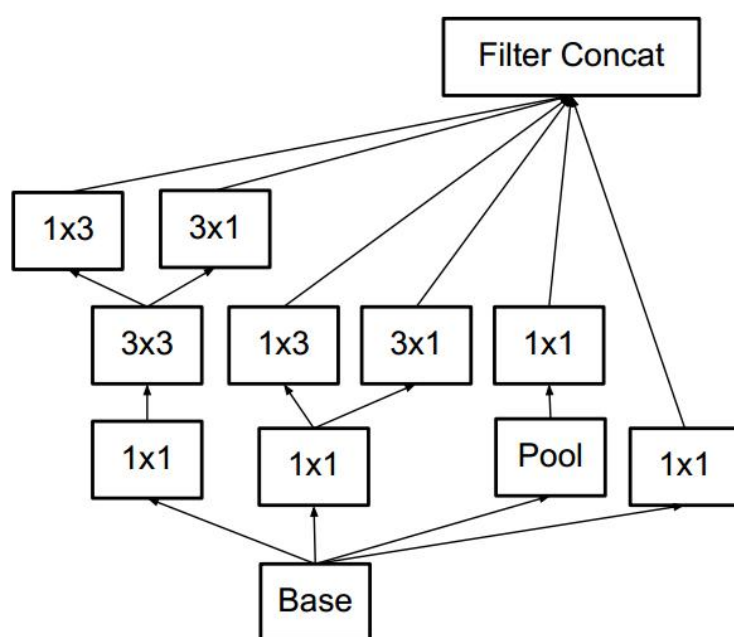


图 9 Module C

InceptionV3 模型的整体架构及其具体参数如表 2 所示，共有 42 层，其中模块 Module A，Module B，Module C 在上文中均已有说明及图示。

表 2 InceptionV3 整体架构表

类型	块大小/步长 (patch size/stride)	输入尺寸 (input size)
卷积层(conv)	3 x 2 / 2	299 x 299 x 3
卷积层(conv)	3 x 2 / 1	149 x 149 x 32
卷积填充层 (conv padded)	3 x 2 / 1	147 x 147 x 32
池化层(pool)	3 x 2 / 2	147 x 147 x 64
卷积层(conv)	3 x 2 / 1	73 x 73 x 64
3 x Inception	Module A	35 x 35 x 288
3 x Inception	Module B	17 x 17 x 768
3 x Inception	Module C	8 x 8 x 1280
池化层(pool)	8 x 8	8 x 8 x 2048
全连接层(linear)	(logits)	1 x 1 x 2048

### 3.4 本章小结

本章首先是对基于深度学习的身体行为识别进行了总结评价,讲述了当前基于深度学习方法的大三类框架,然后对各种算法进行简单介绍,其中双流网络难以准确融合将影响到准确率,而循环神经网络也就是光流方法则多了一步处理视频的步聚,效率低,识别速度慢,3D 卷积神经网络方法能够达到较高的准确率,且简单直接,将时间维加入卷积过程中即可捕获视频的时间维信息实现对视频的处理,所以本文最后选定-使用的是基于 3D 卷积神经网络的方法,其中选用的模型是 InceptionV3,它是由 Google 开发的一个非常深的卷积网络,通过分解卷积和 aggressive 正则化这两个核心思想使得神经网络变得非常高效,且用于可以处理更多、更丰富的空间特征,本章中对其主要思想中的分解卷积及其整体架构进行了介绍,其中着重介绍了对称分解的 Inception 模块 A 和非对称分解的 Inception 模块 B 和模块 C,这种分解不但使得参数量减少而且还多加了一个激活函数,使得它的非线性能力也变强了。



## 4 实验设计及实验结果分析

### 4.1 实验设计

#### 4.1.1 实验数据集选择

关于人体行为识别的数据集有不少，比较经典的有 Kinetics-[400/600/700] 数据集、Something-SomethingV1/V2 数据集、Charades 数据集、MomentsinTime 数据集、HVV 大规模视频理解数据集、Jester 手势数据集、FineGym 数据集、ActivityNet 数据集、UCF101 数据集、HMDB51 数据集，它们各有特色，都包含了大量的各类动作视频数据供大家使用，这里最后选择的是 UCF101 数据集，因为经过比较之后，可以得出，这个数据集不但包含着足够大量的行为视频数据，而且配有标签甚至连训练与测试集都给提前分好了，既方便又满足需求，所以这里选用它作为训练的数据集。各数据集比较如下表 3 所示。

表 3 各经典数据集比较表

数据集	动作数	视频数	发行年份
HMDB51	51	7000	2011
UCF101	101	13000	2012
EPIC-KITCHEN	149	9600	2014
AVA	80	5760	2015
SOA	148	16200	2018
HACS	200	20000	2019

#### 4.1.2 数据集划分与预处理

UCF101 数据集的官网中给预先准备了原作者所建议的训练集与测试集的划分，所以直接根据它们给的标签将整个数据集划分即可，划分之后，还需要进行一些预处理，需要将所有视频进行转换，通过 FFmpeg 将视频按一定帧率转换为多张图片，总体划分比例约为 3:7，前者为测试集，后者为训练集，所有视频按 30 的帧率都被分解为上百张连续帧。

#### 4.1.3 模型初始化

对于 InceptionV3 模型的初始化，是首先去除 InceptionV3 模型的顶层分类器（也就是全连接层），重写一个新的顶层分类器然后将初始模型冻结其所有层（这

样就可以正确获得 `bottleneck` 特征），冻结之后才重新配置该模型的训练方法，其中优化器选择的是 `rmsprop`，损失函数选用的是最经典的 `categorical_crossentropy`，指标的话这里暂时只需要精度指标，在初始模型的权重中选择了默认的 `imagenet`，这样子 Keras 会自动下载已经在 ImageNet（一个大数据集，参考价值高）上训练好的参数，这样的预训练模型将会让后面的训练更高效。

#### 4.1.4 模型预训练与训练

在开始对模型训练之前，先要进步行一次预训练，这次训练仅十轮，主要是为了获得一些初始参数这样子模型参数就不再是随机初始化了，然后开始正式训练，为了更高效的训练，将会进行冻结训练，已有部分预训练权重，这部分预训练权重所应用的那部分网络是通用的，如骨干网络，那么可以先冻结这部分权重的训练，将更多的资源放在训练后面部分的网络参数，这样使得时间和资源利用都能得到很大改善，在对部分块冻结后同时对于需要的块进行解冻之后，重新配置一下训练方法，其中优化器这里选择的是 `SGD` 即随机梯度下降优化算法，设置的学习率是 `0.0001`，学习率是非常重要的，学习率太小则收敛得慢，学习率太大则损失会震荡甚至变大，这里的学习率是经过几番实验后确定的，是比较符合算法的实际情况的，损失函数同上选用的是 `categorical_crossentropy`，指标则相比上面新增了一个精度，然后开始正式训练，训练过程中设置了每个 `epoch` 包含的步数为 `100`，迭代次数选择了 `1000` 次，同时设置了提前停止器，用 `EarlyStopping` 设置 `patience` 为 `10`，即是说连续十轮损失没有减少的话就会停止，开始训练并在每个 `epoch` 保存最优模型（即最小损失的模型）。

#### 4.1.5 模型评估与测试

在以上训练之后，手动选出最优模型，在数据生成器上对其进行评估，对测试集中的所有视频（这里的视频实际上都已经处理为多张连续帧图片）进行辨别，并计算准确率，确定该模型的真实准确率，然后再直接进行实测，加以验证，这里设置为随机选择五个动作进行识别测试，并得出各自的可能性前五的动作，并在它们后面以小数点的形式标出该动作的可能性，在后面我还会尝试现场实拍视频并识别动作。

## 4.2 实验过程及结果分析

### 4.2.1 训练模型

在各个参数调好之后开始训练，首先是要进行前十轮的预训练得到一些非随机生成的初始参数，如图 10 所示，在前面预先的十轮测试中，已经可以达到六十左右的准确率，这时候已经可以得出一些比较适合此次训练的参数了，将对后面的学习效率大有帮助。

```
Epoch 1/10
2022-04-10 03:13:13.940858: I tensorflow/stream_executor/cuda/cuda_dnn.cc:368] Loaded cuDNN version 8100
2022-04-10 03:13:16.937288: I tensorflow/stream_executor/cuda/cuda_blas.cc:1786] TensorFloat-32 will be used for the matrix multiplication. This will
100/100 [=====] - 26s 201ms/step - loss: 4.4638 - accuracy: 0.1425 - val_loss: 4.0123 - val_accuracy: 0.0000e+00
Epoch 2/10
100/100 [=====] - 19s 188ms/step - loss: 2.9016 - accuracy: 0.3325 - val_loss: 1.4143 - val_accuracy: 0.6625
Epoch 3/10
100/100 [=====] - 19s 193ms/step - loss: 2.2675 - accuracy: 0.4356 - val_loss: 3.0422 - val_accuracy: 0.0000e+00
Epoch 4/10
100/100 [=====] - 19s 191ms/step - loss: 2.0573 - accuracy: 0.4850 - val_loss: 2.4982 - val_accuracy: 0.0000e+00
Epoch 5/10
100/100 [=====] - 19s 192ms/step - loss: 1.8929 - accuracy: 0.5138 - val_loss: 0.4096 - val_accuracy: 0.9937
Epoch 6/10
100/100 [=====] - 19s 191ms/step - loss: 1.7534 - accuracy: 0.5481 - val_loss: 4.7117 - val_accuracy: 0.0000e+00
Epoch 7/10
100/100 [=====] - 20s 197ms/step - loss: 1.6351 - accuracy: 0.5656 - val_loss: 2.5932 - val_accuracy: 0.0000e+00
Epoch 8/10
100/100 [=====] - 19s 194ms/step - loss: 1.5385 - accuracy: 0.5869 - val_loss: 0.4858 - val_accuracy: 0.9875
Epoch 9/10
100/100 [=====] - 19s 194ms/step - loss: 1.5519 - accuracy: 0.5888 - val_loss: 1.1287 - val_accuracy: 0.4812
Epoch 10/10
100/100 [=====] - 19s 192ms/step - loss: 1.4844 - accuracy: 0.6056 - val_loss: 3.2677 - val_accuracy: 0.0000e+00
D:\Python\lib\site-packages\keras\optimizer_v2\gradient_descent.py:102: UserWarning: The `lr` argument is deprecated, use `learning_rate` instead.
  super(SGD, self)._init__(name, **kwargs)
Epoch 1/1000
```

图 10 预训练过程

在将部分层解冻剩余其它层冻结之后开始冻结训练，这种冻结部分层之后的训练将会比较高效，另外虽然设计了有 1000 个 epoch，但是也设置了提前停止器，设置了 EarlyStopping 的 patience 为 10，这样子它就会在损失值连续十次没有发生变化的情况下自动提前停止训练了，如图 11 所示，在进行了仅 29 个 epoch 之后便停止了训练，这个收敛速度还是蛮快的，说明这个模型还是比较适合用于行为识别的，在训练过程设置了 callback 保存了训练过程(日志)，并借助 tensorboard 这一工具来绘制了这次训练的几个曲线图，比如图 12 所示的是精确度变化曲线，如图 13 所示的是损失值变化曲线。此次训练参数中，learning rate 设置为 0.0005，batch size 设置为 32。

```
val_top_k_categorical_accuracy: 0.9156
Epoch 26/1000
100/100 [=====] - ETA: 0s - loss: 0.4408 - accuracy: 0.8844 - top_k_categorical_accuracy: 0.9800
Epoch 26: val_loss did not improve from 1.02401
100/100 [=====] - 196s 2s/step - loss: 0.4408 - accuracy: 0.8844 - top_k_categorical_accuracy: 0.9800 - val_loss: 1.1790 - val_accuracy: 0.6969 -
val_top_k_categorical_accuracy: 0.8938
Epoch 27/1000
100/100 [=====] - ETA: 0s - loss: 0.4462 - accuracy: 0.8866 - top_k_categorical_accuracy: 0.9750
Epoch 27: val_loss did not improve from 1.02401
100/100 [=====] - 196s 2s/step - loss: 0.4462 - accuracy: 0.8866 - top_k_categorical_accuracy: 0.9750 - val_loss: 1.2012 - val_accuracy: 0.6969 -
val_top_k_categorical_accuracy: 0.8750
Epoch 28/1000
100/100 [=====] - ETA: 0s - loss: 0.4684 - accuracy: 0.8697 - top_k_categorical_accuracy: 0.9772
Epoch 28: val_loss did not improve from 1.02401
100/100 [=====] - 196s 2s/step - loss: 0.4684 - accuracy: 0.8697 - top_k_categorical_accuracy: 0.9772 - val_loss: 1.2349 - val_accuracy: 0.6812 -
val_top_k_categorical_accuracy: 0.8813
Epoch 29/1000
100/100 [=====] - ETA: 0s - loss: 0.4514 - accuracy: 0.8778 - top_k_categorical_accuracy: 0.9806
Epoch 29: val_loss did not improve from 1.02401
100/100 [=====] - 197s 2s/step - loss: 0.4514 - accuracy: 0.8778 - top_k_categorical_accuracy: 0.9806 - val_loss: 1.2064 - val_accuracy: 0.6562 -
val_top_k_categorical_accuracy: 0.8938

进程已结束,退出代码0
```

图 11 正式训练过程

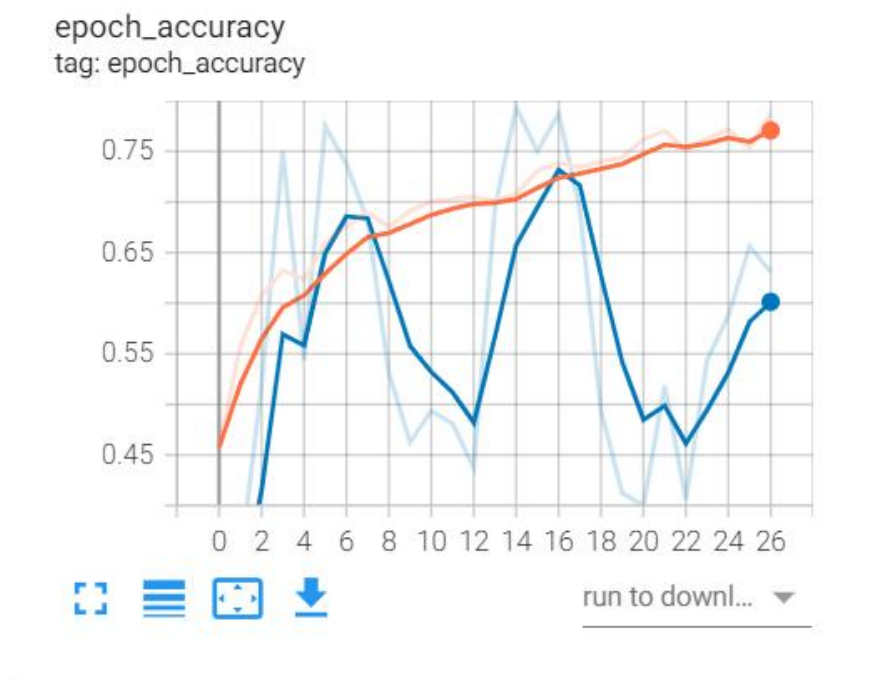


图 12 第一次训练的准确度曲线（红线为训练集，蓝线为测试集）



图 13 第一次训练过程的损失曲线

这是第一次训练模型时所保存的截图，后面还有多次训练，最多也是在 100 多个 epoch 的时候便停下来，说明这个模型收敛速度还是不错的，精度也能看到，能达到百分之 87 左右，top-5 准确率更是达到了百分之 98，这是在第一次训练时没有做太多调整的情况下，在后面进行多次实验调参之后准确率已经可以达到百分之 92，top-5 准确率为百分之 99。但是看上面的曲线也可以很明显看到一些问题，两个曲线中测试集的精确度或者损失值都出现了比较大的震荡，研究了其中原因，经过各方排查之后，最后确定了主要是 batch size 设置过大的缘故，于是将 batch size 从 32 降低至 16，结果如以下图 14 与图 15 所示曲线。

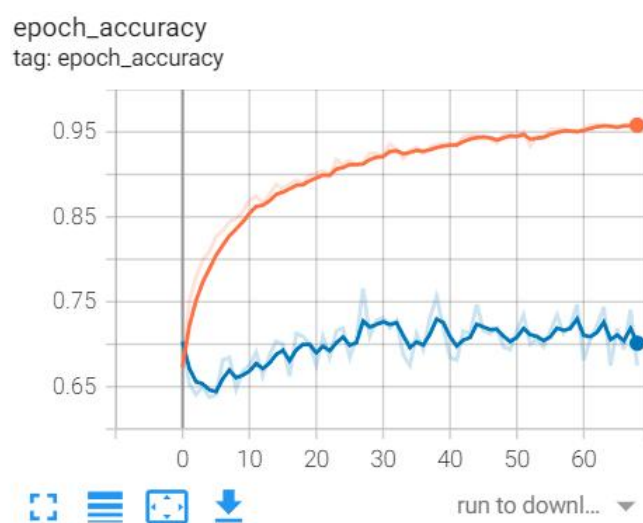


图 14 改进 batch\_size 为 16 后的准确率曲线

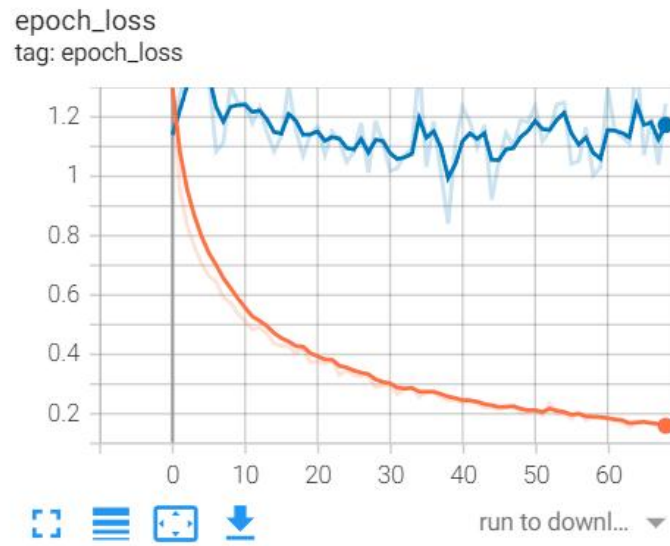


图 15 改进 batch\_size 为 16 后的损失曲线

以上是 learning rate 设置为 0.0001，batch size 设置为 16 的训练曲线，可以看出来，会比之前合理了一些，震荡现象大大减小了，但是还是存在明显问题，精确度和损失值方面在测试集和训练集上的差异很大，训练集上的曲线比较正常，但是测试集上的指标却总是保持平稳，变化很少，也就是出现了过拟合，于是我进行了以下多次实验，以求得出更为合理的训练曲线。首先是改进学习率（lr，learning rate），它是指在梯度下降的过程中更新权重时的超参数，而梯度下降的公式如下所示：

$$\theta = \theta - \alpha * \partial / \partial \theta * J(\theta)$$

如公式中的 $\alpha$ ，一般来说，学习率越低，损失函数的变化速度就越慢，容易过拟合，但是太高的话容易发生梯度爆炸的现象，从曲线上来看就是很强烈的震荡，而上面训练出现的问题可以确定学习率的问题是主要问题需要优先改进，原设定为 0.0005，有点太大了，如上所言，学习率太大容易震荡，太小又收敛太慢，因此我后面尝试逐渐改小，batch size 先暂定为 16。在上面学习率设置为 0.005，后面我尝试了逐渐减小学习率，从 0.0005 到 0.0001，每次下降 0.0001，在下降过程中可以看到曲线是逐渐趋近于合理的，如图 16 和图 17 所示为学习率设置为 0.001 时的训练曲线。

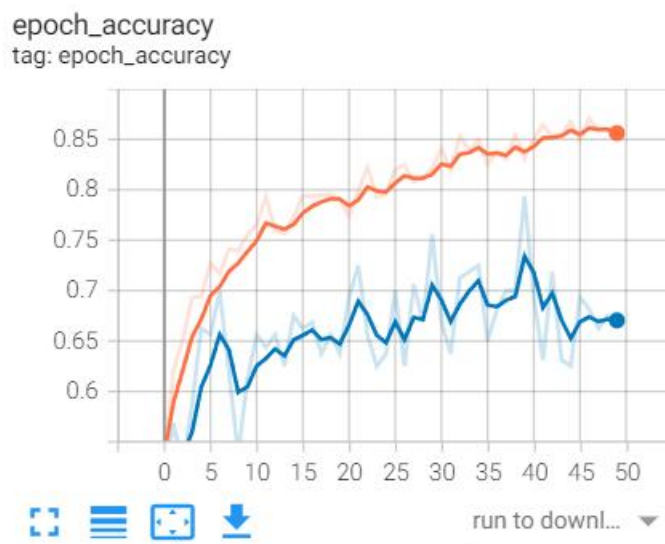


图 16 改进 learning rate 之后的准确率曲线

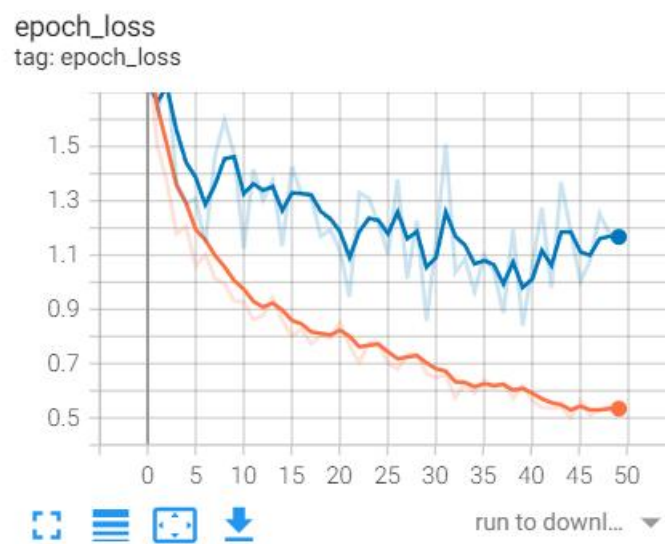


图 17 改进 learning rate 之后的损失曲线

以上的参数设置为学习率(learning rate)为 0.0001，批大小(batch size)设置，可以看到此时的曲线是比较合理的，没有明显的震荡现象，也没有过拟合的现象出现，曲线的收敛也是比较快的，而且在第 39 个 epoch 达到最佳准确率和最低损失值，由于前面在学习率逐渐下降的过程中曲线也逐渐趋向于合理，因此又尝试了将学习率设置为更小的 0.00005，看看是否能得到更加好的训练曲线，它的训练曲线如下图 18 和图 19 所示。



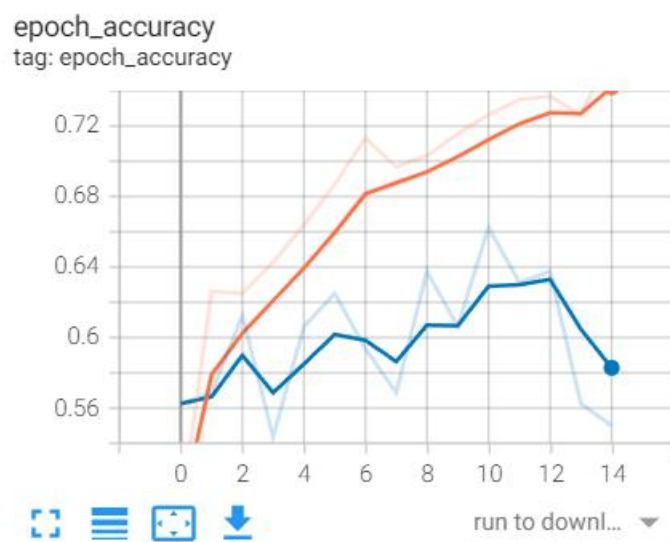


图 18 learning rate = 0.00005 时的准确率曲线

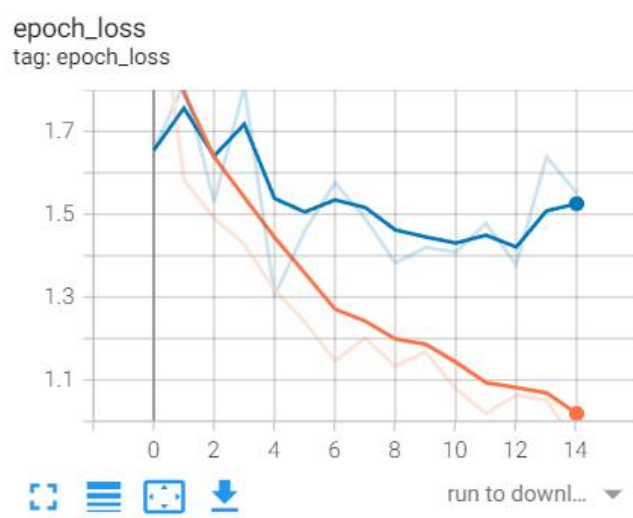


图 19 learning rate = 0.00005 时的损失曲线

可以看到曲线变化很大，收敛速度明显不如学习率设置为 0.0001 时的训练，而且有过拟合的趋势，因此学习率暂时可以确定，是 0.0001 时更为合理，学习率确定之后，对于 batch size，前面是得出了 16 比 32 更合理，但是还没有得出是否更小点会更合理，由于电脑一般擅长处理的是二进制数据，一般 batch size 也是设置为 2 的倍数会明显提高效率，因此我这里是尝试了将 batch size 依次改为 8、4、2、1，其中将 batch size 设置为 8 和 4 时的训练曲线分别如下图 20、21、22、23 所所示。



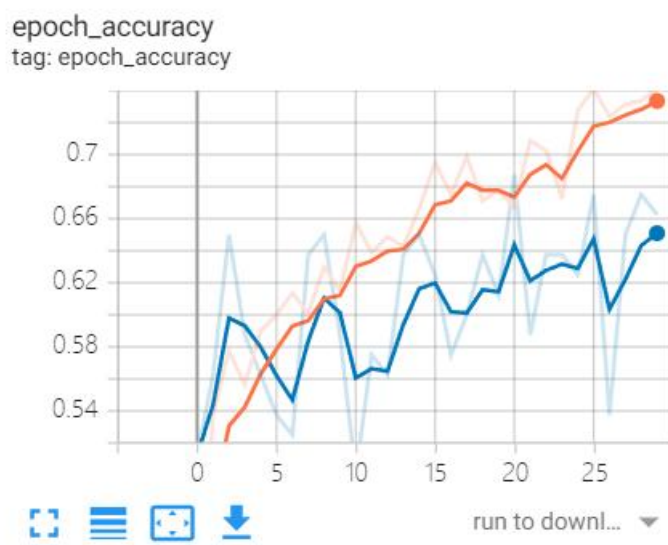


图 20 batch size 为 8 时的准确率曲线



图 21 batch size 为 8 时的损失曲线

可以看到，如果将 batch size 改为 8，那么训练曲线中，对训练集的准确度曲线几乎呈现为一条直线，而在测试集上的准确率则是震荡且明显低于训练集，为了更加严谨，我还是将 batch size 设置为 4、2、1 之后进行训练并保存了训练日志，其中 batch size 设置为 4 时的训练曲线如下图 22 和图 23 所示。

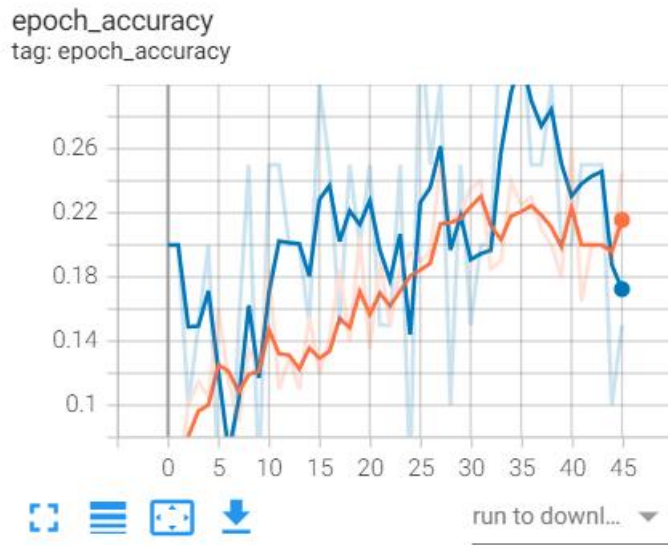


图 22 batch size 为 4 时的准确率曲线

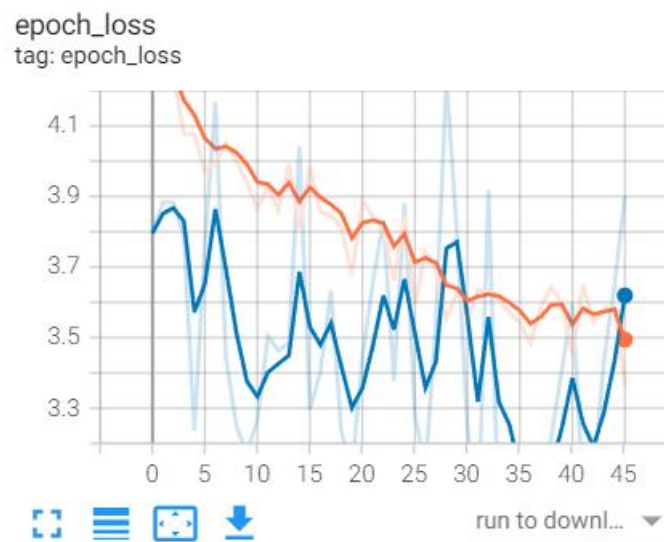


图 23 batch size 为 4 时的损失曲线

当 batch size 设置为 4 时已经可以看到,震荡现象非常严重,后面将 batch size 设置为 2 和 1 时,会比这个更加严重,因此很明显, batch size 的设置是 16 时为最佳,综上所述,可以得出的是,比较合理的训练参数应该是学习率(learning rate)设置为 0.0001,批大小( batch size)设置为 16,在这个参数下所得的结果如下图 24 和图 25 所示。

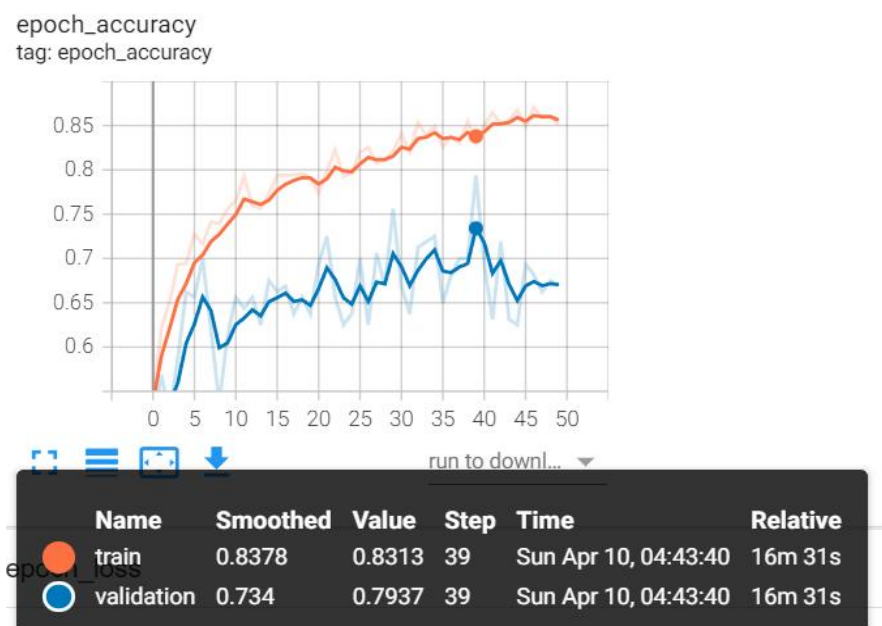


图 24 learning rate 为 0.0001，batch size 为 16 时的准确率

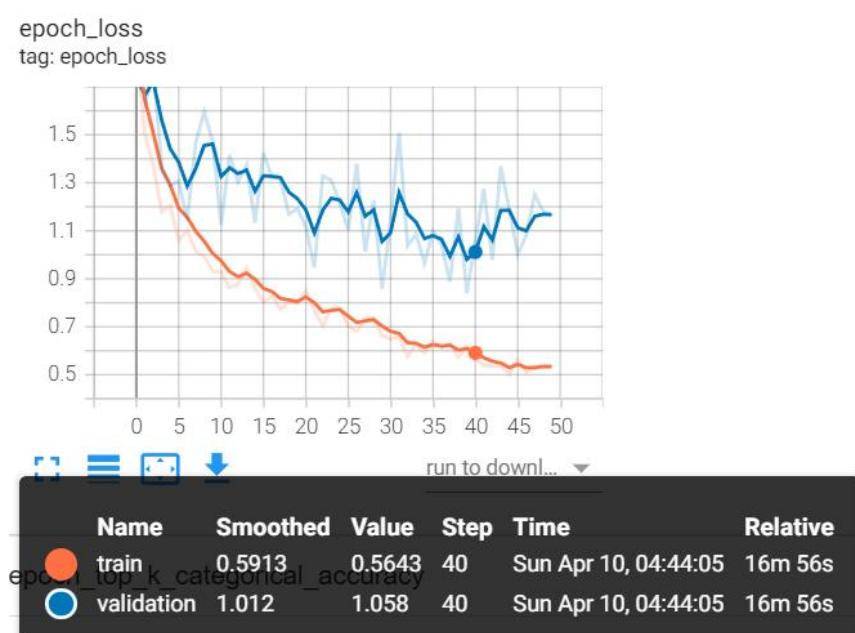


图 25 learning rate 为 0.0001，batch size 为 16 时的损失值

经过以上改进之后曲线变得合理很多，由于设定的自停器是连续十次损失没出现下降就停止训练，而这里在第 39 个 epoch 之后后十个 epoch 都没出现损失减少，因此他在第 49 个 epoch 的时候就停下来了，并记录保存了第 39 个 epoch 所训练的模型，它在训练集上的准确率为 0.8313，在测试集上也有 0.734，它在

训练集上的最小损失值为 0.5913，在测试集上的损失值为 1.012，这是根据以上得出的比较合理的训练参数训练之后所得到的训练结果。多次训练过程都有保存了训练日志，并借助 **tensorboard** 绘制了多次训练中的训练集与测试集的精度曲线图与损失曲线图，分别如图 26 与图 27 所示。

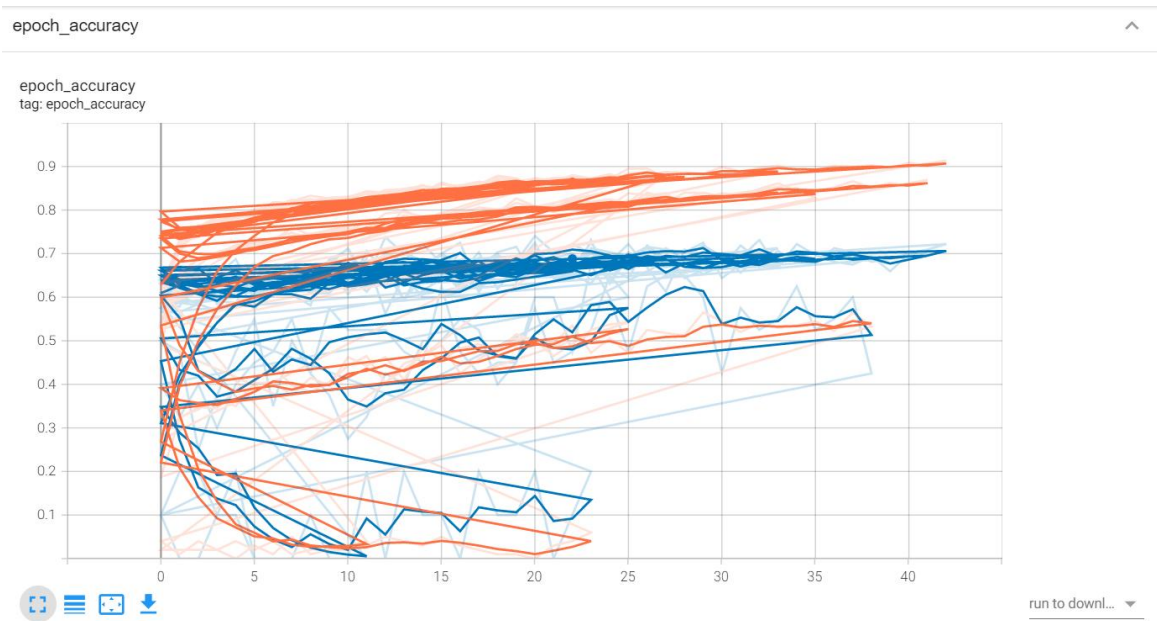


图 26 多次训练过程精度曲线图

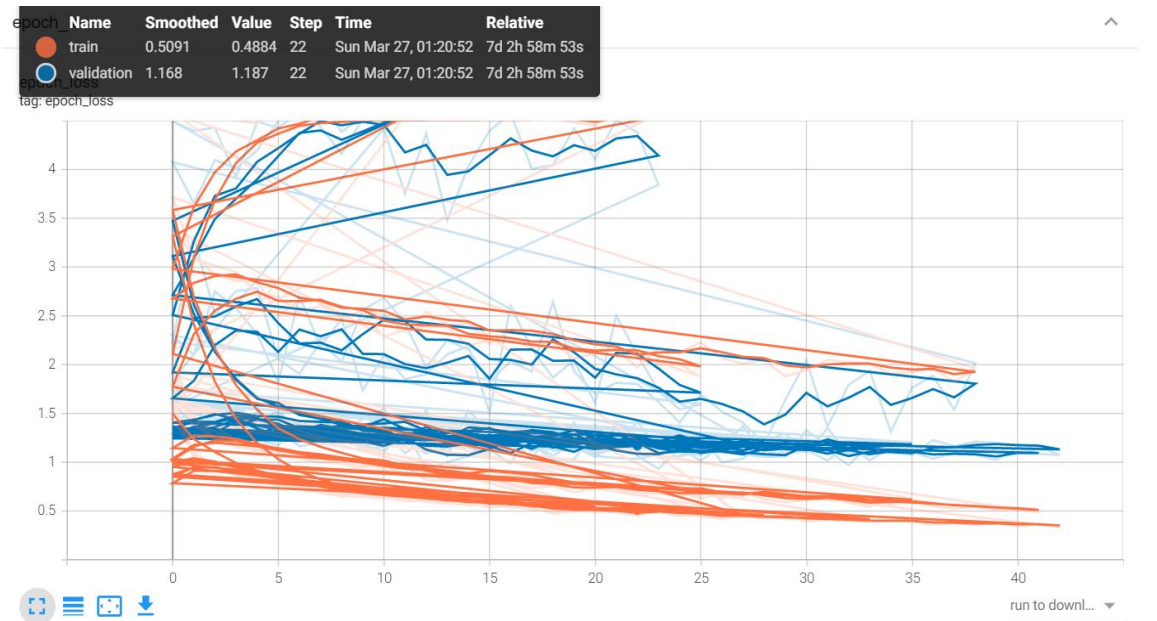


图 27 多次训练过程损失曲线图

### 4.2.2 模型评估

通过以上多次实验之后得出了损失函数值非常低、准确率非常高的模型，对他进行评估，即用它对真个训练集所有视频进行识别尝试，然后得出相应损失值，以及在测试集上的准确率和 top-5 准确率，以下如图 28 所示是用第一次训练得出的最佳模型来进行评估后得出的三个指标，从左到右依次是损失值、准确率和 top-5 准确率，可以看到这个第一次训练用的模型在训练集上的准确率能达到百分之 87 的同时在测试集上也达到了百分之 71，top-5 准确率更是接近百分之 90，说明这个模型可行。在后面经过多次实验之后，得出的最好的模型在验证集上甚至已经可以达到百分之 85 左右，top-5 准确率可以达到百分之 98，这个准确率可以说是相当不错了。

```
022-04-10 01:18:34.905282: I tensorflow/stream_executor/cuda/cuda_blas.cc:1786] Te
1.1205871105194092, 0.7120208144187927, 0.8995265364646912] ←
<keras.metrics.Mean object at 0x000001CE18EF7A90>, <keras.metrics.MeanMetricWrapp
0x000001CE18F23CA0>]
```

图 28 模型评估指标结果

### 4.2.3 模型测试

训练得出模型之后，经过评估，也已经达到了足够高的准确率了，然后就可以开始尝试一下，用训练好的模型进行实测，这里选用的是测试集里的随机五个动作视频，让系统分辨并输出关于它最可能会是哪个动作，以及可能性前五的五个动作，实测结果如下图 29 所示，可以看到上面几个动作它都能正确识别，并判定的可能性都在百分之 90 以上，第一个行为更是百分百确定了正确动作名，即使是最后一个动作的判别出现了一些失误，但是也只是对类似动作判定了百分之 40 多的可能性，对正确动作仍判定了百分之 40 多的正确性。基本可以达到识别的效果。



```

Billiards: 1.00
CricketShot: 0.00
JumpRope: 0.00
ApplyLipstick: 0.00
PlayingDhol: 0.00
-----
./data/test\Haircut\v_Haircut_g05_c02-0008.jpg
Haircut: 0.92
ApplyEyeMakeup: 0.05
BlowDryHair: 0.03
ShavingBeard: 0.00
HeadMassage: 0.00
-----
./data/test\PlayingSitar\v_PlayingSitar_g02_c06-0134.jpg
PlayingSitar: 0.99
PlayingGuitar: 0.01
PlayingFlute: 0.00
YoYo: 0.00
PlayingDhol: 0.00
-----
./data/test\SoccerJuggling\v_SoccerJuggling_g03_c01-0071.jpg
Lunges: 0.44
SoccerJuggling: 0.34
LongJump: 0.09
JavelinThrow: 0.03
HighJump: 0.03
-----

```

图 29 模型实测结果

到此，便可以尝试将它投入到平时生活中的运用了，于是使用 PyQt6 设计界面并实现相关功能得出以下程序，让这个算法更加简单易用使得它可以用于平时生活中，其中主界面如下图 30 所示，通过在这个界面的操作可以快速实现动作识别，动作识别的整体流程如下图 31、32、33 所示。



图 30 主界面

点击“打开文件”后会在当前目录打开文件资源管理器，可以在上面查找视频选择识别，为了方便快捷找到视频，在右下角加了筛选器，可以快速筛选出 avi、mp4 和 gif 文件，如下图 31。

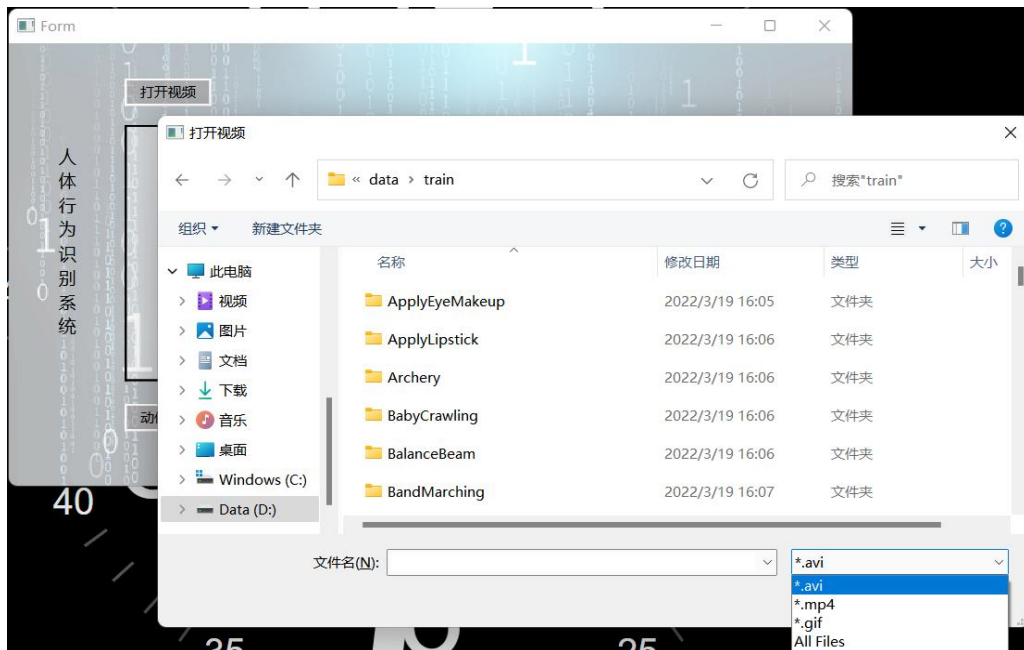


图 31 打开文件界面

在上面打开指定视频文件之后，会将该视频显示在左框中，且会在”打开视频”的右方显示这个视频的路径和全名，以便于更好比对，打开视频之后便可以点击左下角的动作识别，经过模型识别后，会在右边显示识别结果，这里设置了显示可能性最大的五种，每行写上可能的动作名以及它的可能性，如下图 32 所示，这个视频是在 BabyCrawling 文件夹下的示例动作，识别结果中动作 BabyCrawling 可能性为 0.99，是比较精确地识别了该动作。另外其它动作识别都是 0.01 或 0.00 的概率，说明虽然也会有误判但是即使误判，所判定的概率也是很小的，不会太影响识别结果。



图 32 识别界面

以上类似的识别我手动尝试了有几十次，基本都像此视频的识别一样，识别动作的精确率非常高，偶尔低一点的识别正确动作的概率也是在 80%以上的，但是上面这些都是在我本来所训练数据集中的例子，本来就是对这些例子进行识别训练有可能它其实过拟合了，所以说服力不够，于是我又尝试让人现场拍了一个视频来进行识别，主要是拍了一个弹吉他的动作，得出结果如图 33 所示，可以看到，判断为弹吉他的概率为 0.99，可以见得，生活中常见动作，即使是数据集以外，也可以基本做到正确的识别，当然这里这个图片中背景非常简单，对于该动作的识别也明显大大降低了，所以我又尝试了去户外杂乱场景取景，最后识别结果确实准确率不高但是依然能有 80%以上基本可以做到识别。





图 33 实例测试

另外,为了使得该系统的使用更为便捷,因此后面又对界面进行了一定改进,如图 34 所示,将所有步骤集合到了界面中,通过点击即可依次完成数据划分、数据处理、模型训练和模型评估一系列步骤,也可以在此界面直接利用训练好的模型直接如上进行实测。



图 34 改进后的主界面

### 4.3 本章小结

本章是对于整个实验过程的记录与结果分析，由训练模型时的精确度曲线和损失曲线可以知道自己的训练哪里存在问题并及时加以改正，最后得到较为合适的参数值并根据它得出较好的模型，训练之后比较好的模型还是需要评估一下，尝试一下对测试集中的所有动作进行识别，得出该模型的识别准确率和 top-5 识别准确率，尝试之后得知，基本上能在训练时拥有较高准确率与较低损失的模型都能评估得到一个较高的准确率与较低的损失，当然也有个别例外，在最后能得到一个准确率较高的模型后，就用它进行了实测，先是尝试在训练集中随机选取五个动作视频，让模型识别并输出最高可能性的五个动作的可能性，结果可以看到准确率还是非常不错的，然后就尝试了下用自己 and 身边人的一些视频让其进行识别，同样也是有着不错的准确率，即使有误也能有相似的结果输出。

## 5 总结与展望

### 5.1 总结

本课题设计并实现了基于 3D 卷积神经网络的人体行为识别，基本上实现了对于大多数动作的识别功能。本课题及论文完成过程中的主要工作可总结如下：

- (1) 查阅人体行为识别相关话题的论文，关注人体行为识别相关的活比赛等等，理解当前国内外对人类行为识别方面的研究现状及技术前沿，对于人体行为识别的研究现状有个大概了解；
- (2) 了解当前人体行为识别相关的技术以及它们当前的研究情况并确定自己所要使用的技术框架，了解当前可以用于人体行为识别的数据集以及尝试爬取数据集并对比做出选择；
- (3) 在大概确定好研究方向和技术框架后开始学习相关知识技术，代码编写方面的知识技术如 Python、Tensorflow 和 Keras，深度学习方面的知识技术比如 CNN、C3D、LSTM 等等的相关知识，数据处理方面的知识技术比如 ffmpeg、PyQt、Pyuid 等等；
- (4) 尝试使用所学习的知识与技术框架来完成人体行为识别的核心算法，并在完成过程中更加深入学习；
- (5) 查找所需数据集相关资源并做整理，然后使用外部下载的分解工具和官方所推荐的标签划分文件对所选数据集进行规范划分与预处理，并编写类来实现便捷的一键划分；
- (6) 使用 PyQt6 来编写界面并将界面中的部分控件如按钮、文本框等等绑定使用上面的接口，以此来快捷实现数据处理；
- (7) 尝试使默认的方法参数开展训练，并保存每次训练的日志，其中包含每次训练的准确度曲线与损失曲线；
- (8) 在研究过程中，对于训练过程中出现的过拟合或欠拟合现象进行调整，多次训练最后得出了识别准确率较为精准的训练方法配置参数包括如学习率、批大小等参数；
- (9) 在研究完毕之后，又进行了多次评估确定其正确性能够达到一个比较高的数值，并继续优化了界面使其更加美观便捷，可以在界面通过点击各功能按钮直接实现点击处理数据与训练以及动作识别。

## 5.2 展望

基于深度学习的人体行为识别的算法设计还存在一些瑕疵，在设计过程中，有些地方仍然是存在着低效的问题还需要花时间加以改进。特别以下几个方面：

- (1) 本算法中对于部分参数仍然没有得出最优解，只是得出了比较合适的值，比如对于 SGD 算法中的学习率（learning rate），通过多次训练对比，得出了一个相对合理的值，但在此 lr 下的训练曲线仍然有瑕疵；
- (2) 对项目文件没有做好管理，各类文件混杂在一起且整个项目大小达到了 56G 之大，项目里面掺杂了大量重复性的数据、无用数据以及一些用不上的软件包，让整个项目看起来相当臃肿；
- (3) 本算法的代码中有大量混乱注释及旧代码未好好整理，还有部分原本是用于调试所用代码也忘记了处理，整体观感相当糟糕。

相应的改进措施如下：

- (1) 在调参方面，会花更多时间对模型进行训练得出更多实验数据，由此得出更加合理的各个参数；
- (2) 在项目文件管理方面，会重新分类整理并剔除不必要的东西，让整个项目看起来更加分布有序且大小合理，在以后的工作中也会在研发的过程中注意对文件的分类整理与及时处理；
- (3) 在代码规范方面，会学习网上的优秀代码模板，养成更好的写代码的习惯。

## 参 考 文 献

- 陈煜平,邱卫根.基于 CNN/LSTM 和稀疏下采样的人体行为识别[J].计算机工程与设计,2019,40(5):1445-1450.
- 盖勇刚,王佳越,刘天明等.基于深度学习的人体行为识别技术综述[J].信息技术与信息化,2021,67(05):143-144.
- 赫磊,邵展鹏,张剑华,等.基于深度学习的行为识别算法综述[J].计算机科学,2020,47(S1):139-147.
- 黄凯奇,陈晓棠,康运锋等.人工智能视频监控技术综述[J].计算机学报,2015,38(6):1093-1118.[2]
- 黄友文,万超伦.基于深度学习的人体行为识别算法[J].电子技术应用,2018,44(10):1-5+10.
- 揭志浩,曾明如,周鑫恒,何强.结合 Attention-ConvLSTM 的双流卷积行为识别[J].小型微型计算机系统,2021,42(02):405-408.
- 李智敏,刘一鹏,郑海峰,等. LSTM 递归神经网络人体活动行为识别算法研究[J].电气技术,2018,19(11):26-30+36.
- 邓淼磊,高振东,李磊,陈斯.基于深度学习的人体行为识别综述[J/OL].计算机工程与应用,2022-05-10(21):1-15
- 叶青,杨航.基于深度学习的人体行为识别网络设计[J].中国科技信息,2020(10):91-94.
- 曾明如,郑子胜,罗顺.结合 LSTM 的双流卷积人体行为识别[J].现代电子技术,2019,42(19):37-40.DOI:10.16652/j.issn.1004-373x.2019.19.009.
- 周楠,陆卫忠,丁漪杰,吴宏杰,傅启明,张郁.基于深度学习的人体行为识别方法研究综述[J].工业控制计算机,2021,34(08):116-117+119.
- 朱煜,赵江坤,王逸宁,等.基于深度学习的人体行为识别算法综述[J].自动化学报,2016,42(6):848-857.
- 苏江毅,宋晓宁,吴小俊,於东军.多模态轻量级图卷积人体骨架行为识别方法[J].计算机科学与探索,2021,15(4):733-742.
- A review of convolutional-neural-networkbased action recognition. Yao G,Lei T,Zhong J. Pattern Recognition . 2019

- Latent Dirichlet Allocation[J] . David M. Blei,Andrew Y. Ng,Michael I. Jordan.  
Journal of machine learning research . 2003
- HATIRNZA E, SAH M, DIREKOGLU C. A novel framework and concept-based  
semantic search Interface for abnormal crowd behaviour analysis in surveillance  
videos[J]. 2020, 79(25/26):17579-17617.
- UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild[J] .  
Khurram Soomro,Amir Roshan Zamir,Mubarak Shah.CoRR . 2012

## 致 谢

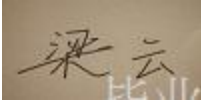
论文能够顺利的完成，非常地开心。在编写毕业论文的过程中遇到了一些困难，在老师、同学、朋友和家人的帮助和鼓励下，最终终于战胜困难完成了毕业论文。

首先要感谢的指导老师李康顺教授。在一开始的选题，老师就给了很多宝贵有效的建议，使得很快找到了做毕设设计的方向。老师在刚开始选题阶段就给开会为每个人指明了后面更具体的实现及创新点，为讲解开题报告和任务书的撰写规范督促完成了开题报告和任务书，在这个过程中多次与老师商讨，老师给出的建议非常中肯。在完成毕设设计的过程中，老师又给了一些修改建议，让的毕设更加完善。在编写论文的时候，老师也指出了的一些不足，帮助改正过来。知道老师很忙，但是也还是能在百忙中抽出时间来为做指导，督促尽快完成各阶段的工作，在这里非常感谢老师的一片苦心。

然后要感谢的同学和朋友们，在写毕设刚开始的时候对算法是一窍不通的状态，是他们带着从零开始学习，在算法出现各种奇奇怪怪的问题时不厌其烦地教帮助，在对某个地方百思不得其解时点醒了，即使在写论文的时候也有指正的论文格式。真的太感谢你们了。



最后要感谢的家人们，在因为毕设而心情低落、焦虑不堪的时候他们替排忧解难，陪走过这道难过的坎。寒假在家期间为营造了研究算法与撰写论文的良好环境，对的毕设完成有着莫大帮助。

# 华南农业大学 本科生毕业论文成绩评定表

学号	201825010104	姓名	程航驰	专业	计算机科学与技术	
毕业论文题目	基于深度学习的人体行为识别					
指导教师评语						
<p>该论文设计了一种基于深度学习的人体行为识别方法，该论文选择使用 3D 卷积神经网络对样本进行训练，使用 Keras 框架，并使用预训练模型提升训练效率，并改进了 InceptionV3 模型进行冻结训练，对比实验比较充足，不仅在测试集上检测效果不错，并且能够识别出自己上传的任意视频中的动作行为。论文选题具有较好的创新性和应用价值，同时参考了丰富的文献资料，撰写格式规范，达到了学士学位论文水平，同意参加答辩。</p>						
<p>成绩（百分制）： 91 指导教师签名：  2022 年 5 月 6 日</p>						
评阅人评语及成绩评定	成绩评定标准	评分项目			分值	得分
		选题质量 20%	1	专业培养目标	5	4
			2	课题难易度与工作量	10	9
			3	理论意义或生产实践意义	5	4
		能力水平 40%	4	查阅文献资料与综合运用知识能力	10	8
			5	研究方案的设计能力	10	8
			6	研究方法和手段的运用能力	10	8
			7	外文应用能力	10	8
		成果质量 40%	8	写作水平与写作规范	20	18
			9	研究结果的理论或实际应用价值	20	18
<p>评阅人评语：</p> <p>本论文针对人体行为识别展开研究，探索了基于 Tensorflow 的深度学习框架，利用图像生成器构建了大量训练数据和测试数据。为更好的得到训练参数，运用了 ImageNet 的预训练模型，迁移了已知的训练效果，优化了框架思路。本文算法在相应数据集上进行了测试，得到了初步验证。该研究前期已有大量成果，可适当对相关工作进行阐述；论文撰写可以进一步学术化，减少口语化表达。论文选题有意义，有一定工作量，同意参加论文答辩。</p>						
<p>成绩（百分制）： 85 评阅人签名：  2022 年 5 月 6 日</p>						



续上表:

评价项目	具体要求 (A 级标准)	最高分	评分				
			A	B	C	D	E
论文质量	论文 (设计) 结构严谨, 逻辑性强; 有一定的学术价值或实用价值; 文字表达准确流畅; 论文格式规范; 图表 (或图纸) 规范、符合要求。	60	55-60	49-54	43-48	37-42	≤36
					47		
论文报告、讲解	思路清晰; 概念清楚, 重点 (创新点) 突出; 语言表达准确; 报告时间、节奏掌握好。	20	19-20	17-18	15-16	13-14	≤12
					15		
答辩情况	答辩态度认真, 能准确回答问题	20	19-20	17-18	15-16	13-14	≤12
					16		
<p>答辩小组评语</p> <p>该生论文撰写基本符合规范, 结构基本合理, 语句基本通顺; 论文工作量适中, 内容质量一般, 表明该生具备一定的独立解决问题的能力; 论文报告过程表述较清晰, 答辩过程思路一般, 回答问题基本正确, 达到本科毕业要求。</p> <p>是否同意通过论文答辩 (打√)</p> <p>1. 同意 <input checked="" type="checkbox"/></p> <p>2. 不同意 <input type="checkbox"/></p> <p>成绩 (百分制): <u>78</u>      答辩小组成员 (签名):</p> <p>    </p> <p>2022 年 5 月 8 日</p>							
成绩总评	<p>论文总评分数: <u>85</u></p> <p>学院院长签名: </p> <p>学院盖章: </p> <p>2022 年 5 月 8 日</p>						