“英特尔杯”第一届中国研究生人工智能创新大赛

[传统课堂的教学效果数字化探索与应用]

项目文档

[V2.0]

[2019.7.18]

[今晚打老虎]

[应用创意]

目录

记录更改历史

序号	更改原因	版本	作者	更改日期	备　注
1	撰写项目概况	V0.1	秘博闻	2019-7-9	
2	撰写技术方案	V0.1	刘佳阳	2019-7-11	
3	修改背景	V0.2	秘博闻	2019-7-11	
4	撰写项目规划	V0.2	秘博闻	2019-7-12	
5	整体格式调整	V1.0	刘佳阳	2019-7-15	
6	整理参考文献	V1.1	刘佳阳	2019-7-18	
7	终版修改	V2.0	刘佳阳	2019-7-19	

1 项目概况

1.1 背景和基础

1.1.1 研究背景

人类进入信息时代，期间最伟大的创举莫过于将真实世界与虚拟世界构建起联系。关于信息来源，实验心理学家 Treicher 曾做过一个关于人类获取信息来源的实验，即人类获取信息主要通过哪些途径。他通过大量的实验证实：人类获取的信息 83%来自视觉。视觉，作为人类获取信息的主要途径，其在人类感知能力中占有着举足轻重的地位。因此，计算机视觉一直都是在计算机领域的一大热点。随着信息技术的快速发展，包含视觉信息的图像和视频数据呈“爆炸式”增长。如此规模庞大的数据，若仅靠人的肉眼来处理，无疑是一个十分漫长繁琐的过程。如何从海量的图像和视频数据中获取目标可能包含的语义信息具有十分重要的意义。

近年来，以卷积神经网络为代表的一系列深度学习技术的发展为计算机视觉及其语义分析注入了新的生机。从图像拼接，如图 1 所示、目标追踪[1]，如图 2 所示、三维重构[2]，如图 3 所示、三维运动捕捉[3]，如图 4 所示，到图像语义理解[4]，如图 5 所示、人脸识别[6]及超大规模的人脸检测[6]，如图 6 图 7 所示、裂纹检测，如图 7 所示，……。在深度学习技术的推动下，计算机视觉在很多领域都已经明显超越了常人所能达到的视觉极限。

图 1 图像拼接

图 2 目标追踪

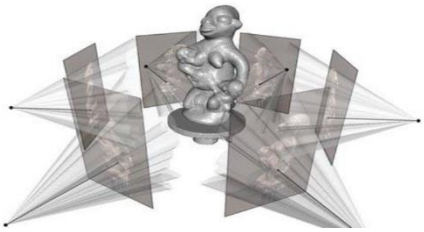

图 3 三维重构

图 4 动作捕捉

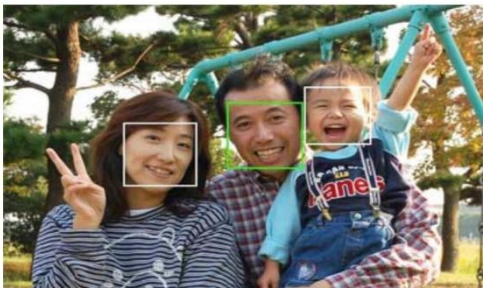

图 5 人脸识别

图 6 图像语义理解

图 7 大规模人脸检测

图 8 特征检测

然而，随着计算机视觉对准确度和精度追求不断提升，深度神经网络规模也随之不断扩大，如表 1 所示。2012 年，Hinton 等设计的新的卷积神经网络结构 AlexNet[7]在 ILSVRC 竞赛中的图像分类任务中获得第一名，Top-5 的错误率为 16.4%，参数量占 160MB 存储空间。VGG16 的网络模型中，网络层数为 16，已具有 13834 万个参数，占用了超过 500MB 的存储空间，并且处理一张图片需要 309.4 亿次浮点运算(FLOPs)，根本无法在资源有限的嵌入式及移动设备中部署。然而究竟这些冗余是否必要呢？经过研究人员发现，VGG[8]网络经过一定比例的稀疏后，网络模型变小，速度提升，精度反而有所上升。

表 1 深度神经网络规模对比

Year	Model	Layers	Parameter	FLOPs	ImageNet Top-5 error
2012	AlexNet	5+3	160M	725M	16.4%
2013	Clarifai	5+3	60M	—	11.7%
2014	MSRA	5+3	200M	—	8.06%
2014	VGG-19	16+3	143M	19.6B	7.32%
2014	GoogleNet	22	6.8M	1.566B	6.67%
2015	ResNet	152	19.4M	11.3B	3.57%

在计算机视觉中，如何快速准确地定位目标、识别物体，是当前亟需解决的问题。近年来，各种目标检测算法层出不穷，表 2 中给出了部分算法在 COCO 数据集上的神经网络规模及性能表现。

表 2 目标检测神经网络规模及性能表现对比

Year	Model	Train	FLOPs	FPS	mAP
2016	SSD300	COCO	—	46	41.2
2016	SSD500	COCO	—	19	46.5
2017	YOLOv2	COCO	62.94 Bn	40	48.1
2017	FPN FRCN	COCO	—	6	59.1
2018	YOLOv3-320	COCO	38.97 Bn	45	51.5
2018	YOLOv3-tiny	COCO	5.56 Bn	220	33.1

近年来，国内外基于深度网络模型压缩技术的研究已经取得了一系列进展，大致可以分为低秩分解、模型量化、网络剪枝、教师-学生网络和设计精简的网络模型五个方向。然而，通过前期初步研究，我们发现，虽然从算法层面对深度神经网络进行压缩“似乎”能从理论上得到更为精简的神经网络，然而脱离了硬件系统架构研究模型压缩成效甚微、甚至根本无法真正落地应用。神经网络“天生”就对硬件资源有着根深蒂固地需求，从其诞生开始，研究人员一直在寻求能提供更强大计算能力的平台，从早期的多核并行到 GPU，神经网络几乎毫无障碍地迅速跨越进大规模数据并行处理的时代。然而，基于深度学习的应用千差万别，仅靠并行无法解决所有问题，如何更好地运用新型计算平台，从软硬件协同的角度实现对特定计算机视觉应用的高精度、快速处理是当前亟待探索的课题。

传统课堂中，教师凭借个人感知能力获取课堂上的信息，部分经验丰富的教师能边讲课边观察学生的反馈调整教学进度，但受场地和个人能力所限，远不能达到实时根据学生能力调整教学进度的程度。同时，那些优秀教师的讲课经验由于不便于量化度量，因此也很难于广泛推广。鉴于上述原因，本团队将基于卷积神经网络“获取”传统线下授课环境中“教师”和“学生”的典型行为特征，借助于 Intel 提供的 OpenVINO 平台，构建起智能化、自动化教学场景，以提升线下教学效果。通过提升感知教室的能力，构建起自动、数字化、实时获取课堂信息的途径，在量化度量教学效果的前提下，有针对性提升教学效果，打造出“超级智能教师”，以提升线下教学效果。此外，该项目还可以直接应用于智能监控场景、智能会议录制等场景，通过上文提到的计算机视觉感知和信息处理技术，可以有效获取实际场景蕴含的信息。还可以通过此案例，探索基于“Intel 的

OpenVINO 平台”实现“云+端”模式的深度神经网络典型应用的开发范例。因此本项目有深远的社会价值和应用价值

1.1.2 研究基础

本课题的主要研究难点在于：1）基于深度学习技术（特别是卷积神经网络）的计算机视觉在捕获人体特征识别方面的研究；2）针对卷积神经网络的深度模型压缩方面的研究。以下将从这两个方面分别展开介绍。

深度学习在计算机视觉中的应用可以追溯到上世纪 60 年代，Hubel 等人首次提出感受野的概念，直到 2012 年 Krizhevsky 等人提出的 AlexNet[7]，在 ILSVRC 2012 图像分类任务中以 16.45%的 Top-5 错误率取得了第一名，准确率远高于第二名，成功掀起了卷积神经网络的研究热潮。计算机视觉与深度学习的大致发展路程如表 3 所示。正因为计算机视觉与深度学习的飞速发展，基于深度学习技术在计算机视觉领域的应用，学界和业界均展开了大规模的研究。

表 3 基于深度学习技术的计算机视觉发展历程

年份	事件
2009	ImageNet 建立[10]
2012	AlexNet 在 ImageNet 竞赛中获胜[13]
2014	Christian Szegedy 等人提出了对抗样本（Adversarial Examples）这个概念[11]
2015	提出 ResNet[12]
2016	提出将目标检测转化为回归问题求解的目标检测算法[13]
2017	CMU 开源了 openpose[9]
2018	Joseph Redmon 提出了 yolo-v3[14]

本项目采用 OpenPose[9]姿态估计方法捕获人体关键特征，此方法由卡耐基梅隆大学 CMU 实验室提出，是一种经典的自下而上的人体关键点检测算法，采用人体关键点亲和域场(Part Affinity Fields, PAFs)描述关键点间的关系，可以用于估计人体的动作、面部表情以及手指运动等姿态，适用于单人和多人，鲁棒性极好。

拥有大规模参数的卷积神经网络模型往往存在冗余，有相当一部分的神经元和连接权重在模型中的作用微弱，但这部分神经元和连接权重却显著增加了网络模型的参数量和计算量，增加了网络模型部署在资源有限的边缘节点中的难度。

本项目基于滤波器级和连接级剪枝思想，提出了一种多粒度卷积神经网络剪枝框架，设计并实现了相应的训练策略，实现了两类剪枝方法的有机结合，在有效降低网络模型的存储和计算开销的同时，也保证了最优压缩率及准确率。

鉴于课题组承接的“精品课程录制系统”项目的研究基础，课题组探索将深度神经网络应用于师生特征动作识别的研究中，即本项目的灵感来源于实验室和某相机厂商合作的智能录课系统这一项目，在此项目中实验室使用了传统的模式识别方法并将其移植在嵌入式设备上识别教师以及学生的动作从而使得录制设备在不需要摄像师的情况下，根据老师以及学生的动作，智能的捕捉拍摄教室的位置。并且在移植过程对于识别算法进行了优化，使得算法可以在嵌入式系统中不必消耗服务器中高额算力也可以对教师以及学生进行行为以及动作的捕捉。本项目具备充足的学术理论基础，已经发表数篇相关论文，如表 4 所示。

表 4 已发表相关论文

会议类别	论文题目	作者
International Coference of Pioneering Computer Scientists, Engineers and Educators	Multi-grained Pruning Method of Convolutional Neural Network	周晚晴
KSII Transactions on Internet and Information Systems	The Design and optimization of teacher actions recognition algorithm in intelligent recording system	周晚晴
International Conference on Machine Learning and Machine Intelligence	Using Distillation to improve network performance after Pruning and Quantization	刘佳阳

论文《Multi-grained Pruning Method of Convolutional Neural Network》阐述了先使用粗粒度剪枝再使用细粒度剪枝的联合方法，并分别使用 L1，L2 正则化来优化训练过程，在保证模型精度不受损的情况下对模型进行压缩；论文《The Design and optimization of teacher actions recognition algorithm in intelligent recording system》阐述了教师动作识别算法在智能录播系统中的设计与优化；论文《Using Distillation to improve network performance after Pruning and Quantization》阐述了使用蒸馏的方法提高剪枝与量化后网络模型的精度，为神经网络移植到嵌入式设备打下了理论基础。

本项目的技术研究基础包括这一套与相机厂商合作的智能教室录播原型系

统，以及神经网络在嵌入式设备上的移植工作。此外本实验室长期从事与计算机体系结构相关的研究，曾发表了《基于 HSA 异构平台的任务调度研究》等论文，并且在最近的研究中也涵盖深度神经网络压缩以及 FPGA 平台方面的研究。

本项目的团队构成由北京工业大学信息学部计算机学院体系结构系四名研究生组成，本团队既有工程经验，又具备学术能力，希望在此次大赛中可以获得好的成绩。

1.2 场景和价值

智慧课堂是一种未来有望在学校中推广普及的新型智能化系统。智能课堂是指利用校园内的计算机技术、网络技术、通信技术以及科学规范的管理对课堂内学习、教学、科研管理等信息资源进行整合、集成和全面数字化，以构成统一的用户的资源管理和统一的权限管理控制的系统。智慧课堂的建立将能对于传统教学产生了变革性的影响。智慧课堂将能有利于构建理想的学习环境，利用大数据、云计算、互联网等新一代信息技术打造的智能高效的课堂环境。智慧课堂的提出与发展既是信息技术在教学领域的应用的产物，同时也是课堂教学不断变革的结果。

1.2.1 直接应用价值

我们的智慧课堂系统直接价值体现在教师以及学生机位的课堂捕捉与录制是利用计算机以及网络技术对我国教育信息化的一次全面并且重要的升级，是教育改革事业的发展结果。整个系统顺应了教育公平，实现了优质教育资源的共享，推动了教育信息化的前进；另一方面，本智能课堂系统打破了时间与地点的限制，提供了一条学生进行自主学习的渠道，解决了传统教育模式存在的痛点问题，并且促进了学生的全面发展。

本系统还提供了专业的录制平台，可以高质量全面的记录教学活动的细节以及全过程，并且提供方便的授课场景在线功能。授课场景的再现有利于促进学生对课堂信息的回顾、促进教师之间的学习和借鉴、促进教学水平的提高，同时有利于远程教学、在线教学活动的推广。

对比性分析：本项目相对于市面上其他的智能课堂系统而言成本更加低廉，相对于百度公司推出的智能课堂系统为例，百度公司在每个学生的课桌上都布置了一个学生机位来观察学生的上课情况，而对于本系统而言，只分为教师机位和学生机位两个机位，学生机位分为两个摄像装置，一个为角度较大的全景相机，可拍摄到教室内所有学生的听讲情况，另外一台为可变焦的摄像装置，当某个学生发生特定举手或者示范动作时可将摄像头推近从而捕捉其动作。双机位有如下

优点：1.这样的智能教室布置会相对容易，相对于每个学生桌上一台设备的情况教室走线方面不会很复杂，影响老师和学生的正常活动。2.众多的拍摄器材会导致云端处理数据负荷巨大从而使得数据处理传输十分缓慢，整个系统的实时性会受到比较大的干扰。3.学生也应该拥有自己的私人空间，如果每个人都在摄像头下学习和工作则会对学生造成一定伤害，本系统学生机位只拍摄对于教学有关的学生的行为并不侵占学生的隐私。4.本系统考虑了经济性，在成本更低的情况下保证了拍摄以及录制的质量。

1.2.2 潜在社会价值

本系统的原型系统不仅仅可以用作智慧课堂的录制，还可以稍加改动在其他的社会场景中加以应用。

- **智能会议录制：**本系统的主要功能还可以应用到智能会议系统当中，可将配置的广角摄像装置更换为一个 360 度全景相机，这样即可对于会议的整个场景进行全方位的捕捉，分析参会者的行为又可以通过另外配置的可变焦摄像装置捕捉正在演讲者的特写镜头。并且可以使得在国外或者不在会议现场的职工可以实时参加会议。
- **智能监控系统：**本系统可以推广到更加广泛的使用空间，需要监控的一些其他场景，例如闹市区的路口就可以使用本项目中可变焦摄像装置加上广角摄像装置的模式，识别行人的一些特征动作，例如斗殴，剽窃等等，如果识别出之后使用变焦相机将图像拉近并进行拍照捕捉，上传到云端，可以对于警方追捕犯人起到较大的帮助。

1.3 所需支持

算力：TB 级别工作平台用于网络训练

目前硬件平台：推断平台 NVIDIA Jetson TX2 8GB，训练平台 GTX1080Ti

需要使用的硬件平台：

相关培训：需要 intel 至强云计算平台资源的培训以及 intel Movidius 神经计算平台的培训

2 项目规划

2.1 整体目标

在参加比赛中的整体目标在于可以在嵌入式智能课堂平台上可以清晰、快速、

准确的捕获到教师以及学生的行为动作，并将动作存储下来进行进一步的分析，如图 9 所示。

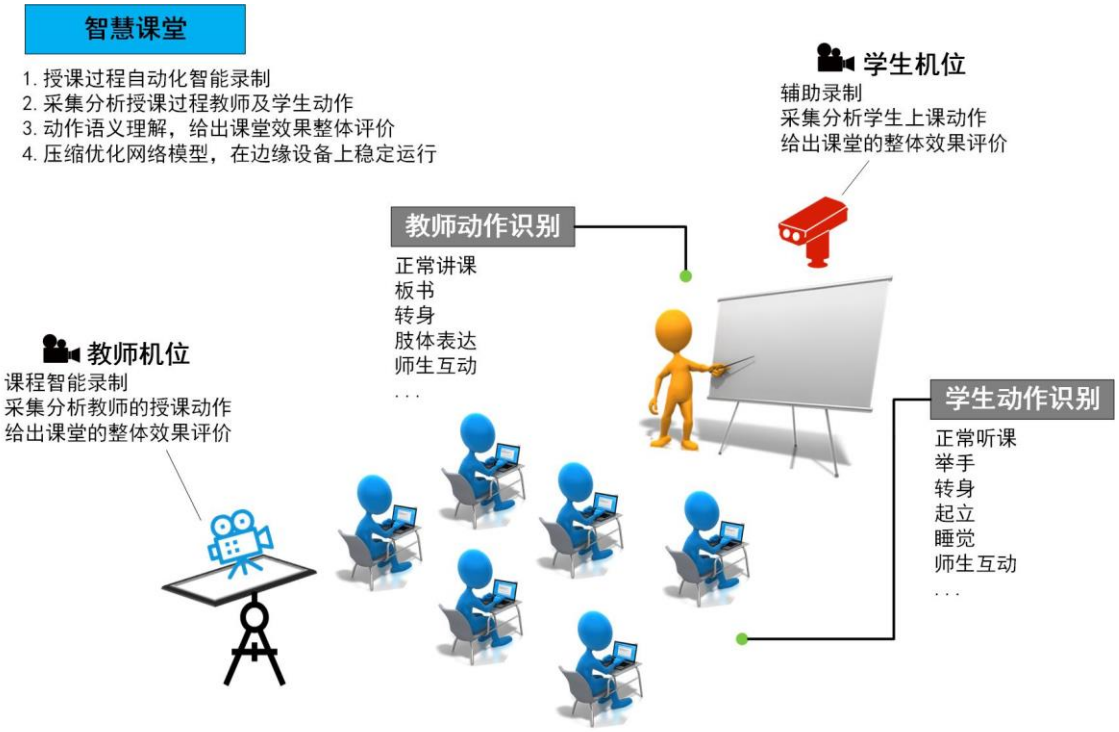

图 9 智慧课堂场景效果

2.2 技术创新点

技术创新点在于深度神经网络的压缩应用于录播平台之上，识别老师以及学生的动作使得在不需要摄像师的情况下进行录制设备的变焦跟随以及切换。在课程录制完成之后对于整堂课的教师行为以及学生行为进行分析，给出结果以便评价者或者想要学习此课的人能有相应的了解。

3 实施方案

3.1 技术可行性分析

(1) 数据采集

在教室录制上课视频以及拍摄照片作为训练数据集，在教室后方录制拍摄老师的动作获得教师机位的数据集，在教室前方拍摄与捕获获得学生机位数据集。

(2) 行业知识获取

有机器学习、深度学习基础研究。

有神经网络模型压缩与加速相关研究基础。

有 caffe、pytorch 等相关框架使用经验。

有 openpose、yolov3 等模型使用经验。

有异构计算平台 HSA 性能调优相关研究。

(3)是否有足够的算力

目前使用 1080Ti 作为训练用平台，TX2 作为推断平台，处理起来有些吃力，还需要算法优化以及硬件平台升级。

(4)硬件支持

现有设备 GTX1080TI*2 、 GTX1060TI*1、I7-8700K、Jetson tx2*1、内存 16G。

3.2 技术细节

技术路线图，如图 10，图 11 所示

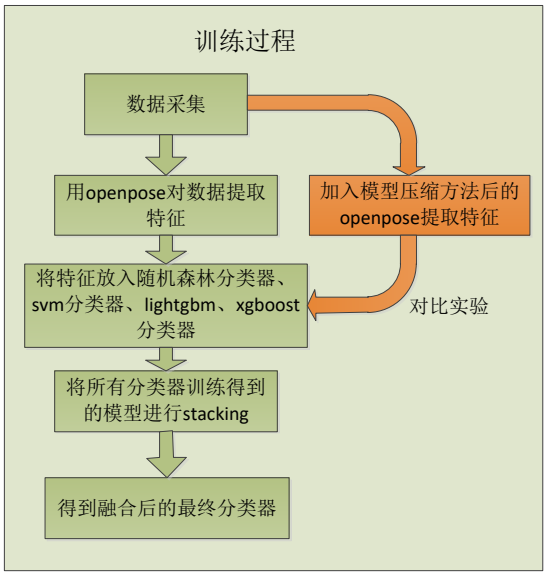

图 10 训练过程图

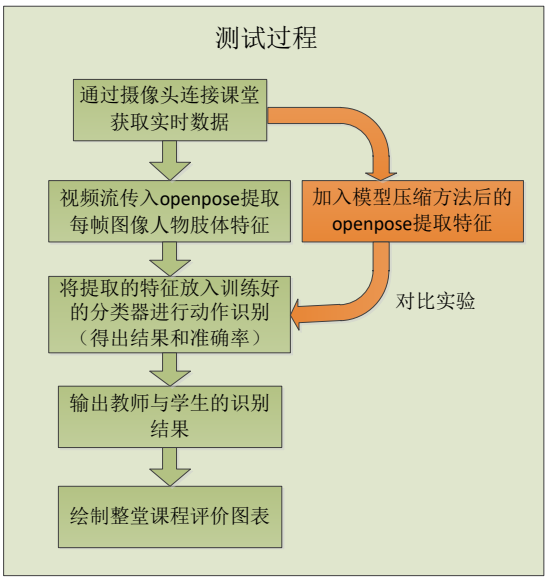

图 11 测试过程图

3.2.1 训练过程

(1)数据采集：

用相机拍摄教室内上课情景，分两个机位，一个是学生机位，一个是教师机位。其中教师机位的位置应放在教室后面中央上方，以俯视角度拍摄教室画面，拍到的学生都是背对相机，主要可以采集教师及讲台和黑板区域的信息。学生机位的位置应放在教室前面中央上方，同样以俯视角度拍摄教室画面，主要采集学生的正面信息，识别并分析学生的动作。

(2)实验环境搭建：

搭建硬件平台，使用 cpu：i7-8700K，显卡：GTX1080TI，内存：16G，硬盘：1T。软件平台，操作系统：ubuntu16.04，工具：python3.5、pytorch1.0、cuda9.0。openpose 环境使用：https://github.com/TreB1eN/Pytorch0.4.1_Openpose 此代码库为基础，对其进行修改。

(3)用 openpose 对数据提取特征：

首先将采集的视频数据进行同一格式处理，将视频输入 openpose 使用 inference 模式对视频检测，检测时是使用 opencv 将视频分解成每一帧画面，由于拍摄后相邻帧数之间画面差别很小，为避免无效的训练，我们不对视频的所有帧数进行识别，根据数据量的情况采用每隔 4/7/10 帧训练一次的方案，在现实课堂应用时，可采取每 1~5 秒采集一次画面，捕捉课堂人物动作，可大幅减少对设备性能的要求。原版的 openpose 对肢体的检测一共有 18 个关键部位，分别是：鼻、首、右肩 、右肘、右手、左肩、左肘、左手、右腰、右膝、右足、左腰、左膝、左足、右目、左目、右耳、左耳。取每个部位的（x，y）坐标作为特征，共有 2*18=36 个特征。

标签为检测到人物类别及动作，目前设置有七类，分别是：学生（student）、教师（teacher），学生转身（student-back）、教师转身（teacher-back）、教师板书（teacher-write）、学生起立（student-stand）、教师学生互动（tea-stu-interaction），如图 12-19 所示。可根据实际需求再添加或删除动作类别。

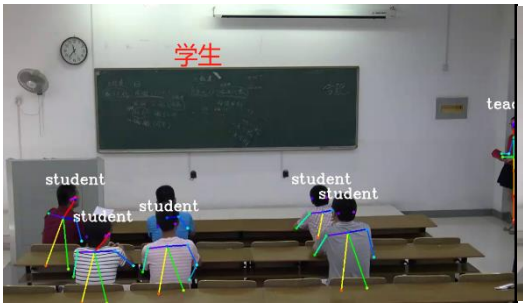

图 12 学生标签

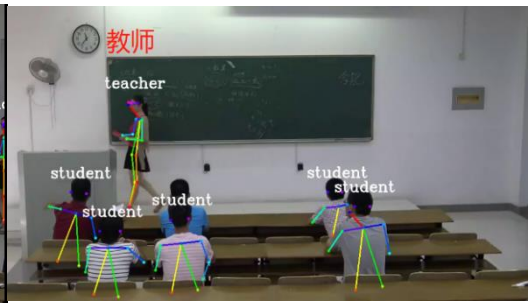

图 13 教师标签

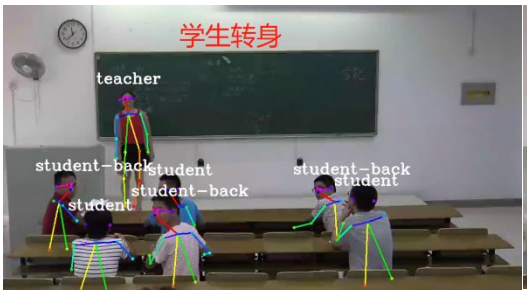

图 14 学生转身

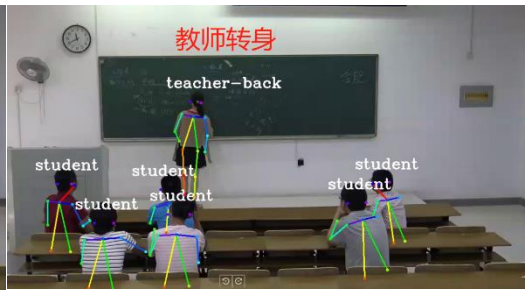

图 16 教师转身

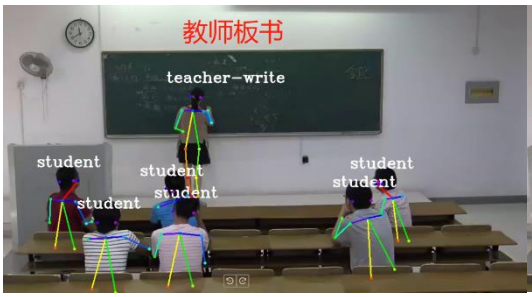

图 17 教师板书

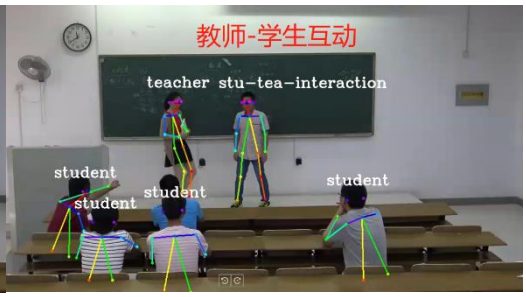

图 18 教师-学生互动

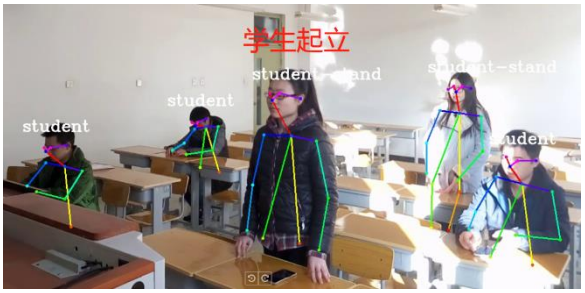

图 19 学生起立

提取特征的保存格式，如表 5 所示。

表 5 人体特征保存格式

特征 / 人物	鼻(x,y)	首(x,y)	右肩(x,y)	右肘(x,y)	• • •	左耳(x,y)	右耳(x,y)	标签
P1	( )	( )	( )	( )	• • •	( )	( )	动作*
P2	( )	( )	( )	( )	• • •	( )	( )	动作*
P3	( )	( )	( )	( )	• • •	( )	( )	动作*
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
Pn	( )	( )	( )	( )	• • •	( )	( )	动作*

图 20-21 为实际提取出的特征值：

C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Nose_x	Nose_y	Neck_x	Neck_y	RightShoul	RightShoul	RightElbow	RightElbow	RightHand	RightHand	LeftShould	LeftShould	LeftElbow_	LeftElbow_	LeftHand_x	LeftHand_y	RightWaist	RightWaist	RightKnee_	RightKnee_	RightFoot_	RightFoot_
838	209	836	245	811	250	816	304	849	338	856	243	862	295	867	333	822	338	822	403	816	479
993	333	969	358	960	358	936	400	987	423	980	358	980	400	996	425	947	443	0	0	0	0
1176	421	1218	477	1249	475	0	0	0	0	1189	484	1156	544	0	0	1247	601	0	0	0	0
382	394	384	414	364	418	347	443	0	0	409	412	440	423	409	412	382	454	0	0	0	0
0	0	1022	502	1073	511	0	0	0	0	973	493	0	0	0	0	0	0	0	0	0	0
1182	349	1196	389	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
807	236	804	266	787	268	787	313	816	342	822	266	831	304	836	335	798	349	789	407	789	461
936	335	916	362	900	360	889	398	927	405	929	364	931	410	0	0	898	439	0	0	0	0
0	0	958	477	1000	484	1009	520	0	0	913	468	0	0	0	0	0	0	0	0	0	0
1069	418	1118	464	1149	459	0	0	0	0	1082	472	1060	526	0	0	0	0	0	0	0	0
1091	349	1104	387	0	0	0	0	0	0	1129	385	0	0	0	0	0	0	0	0	0	0
824	223	822	256	802	256	804	306	833	340	844	254	851	302	851	335	813	349	804	418	804	479
967	335	944	362	931	362	916	400	956	416	960	364	949	394	0	0	920	441	0	0	0	0
1147	349	1167	385	1136	387	0	0	0	0	1184	387	1182	430	0	0	0	0	0	0	0	0
0	0	989	493	1036	500	1051	538	0	0	942	479	0	0	0	0	0	0	0	0	0	0
1116	423	1151	484	1204	484	0	0	0	0	1096	486	1073	542	0	0	0	0	0	0	0	0
833	218	827	254	807	254	807	304	838	338	849	252	853	297	858	333	818	346	809	416	807	479
978	333	951	360	942	360	922	398	962	418	962	362	0	0	0	0	933	446	0	0	0	0
1158	349	1182	385	1153	387	1138	416	0	0	1198	387	0	0	0	0	0	0	0	0	0	0
0	0	1164	495	1220	502	1224	565	0	0	1104	493	1078	544	0	0	0	0	0	0	0	0
0	0	998	495	1044	500	1060	538	0	0	953	486	0	0	0	0	0	0	0	0	0	0
836	216	831	254	807	256	807	306	838	338	851	252	856	299	858	333	818	349	804	421	804	479
978	333	951	362	938	360	920	400	964	418	964	367	962	400	0	0	933	448	0	0	0	0
1140	351	1167	385	1136	382	1120	416	0	0	1198	387	0	0	0	0	1149	452	0	0	0	0
0	0	1164	495	1220	500	1224	569	0	0	1104	490	1078	547	0	0	0	0	0	0	0	0
0	0	996	493	1047	502	1062	542	0	0	949	484	0	0	0	0	0	0	0	0	0	0
838	218	831	254	807	256	807	308	838	340	851	252	853	299	858	335	818	340	813	403	807	475
976	333	951	360	936	360	922	400	960	414	964	362	962	394	0	0	931	446	0	0	0	0
0	0	1164	495	1218	502	1222	569	0	0	1104	493	1084	547	0	0	1178	628	0	0	0	0
0	0	996	493	1044	502	1060	540	0	0	949	484	0	0	0	0	0	0	0	0	0	0
1138	344	1158	389	1136	385	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
836	216	827	254	807	256	807	306	838	340	851	252	853	302	858	335	818	349	804	416	804	475
976	333	949	362	938	360	920	400	962	416	964	364	0	0	0	0	933	448	0	0	0	0
0	0	1164	495	1218	502	1227	565	0	0	1104	493	1080	544	0	0	0	0	0	0	0	0
0	0	996	495	1044	502	1060	540	0	0	949	486	0	0	0	0	0	0	0	0	0	0
0	0	1171	387	1198	387	0	0	0	0	1151	387	0	0	0	0	0	0	0	0	0	0
833	214	827	254	804	254	807	306	838	338	849	252	853	299	856	335	818	349	807	418	807	477
976	333	949	362	936	360	922	400	962	416	964	364	960	396	0	0	929	446	0	0	0	0

图 20 特征提取部分一

R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
LeftHand_y	RightWaist	RightWaist	RightKnee_	RightKnee_	RightFoot_	RightFoot_	LeftWaist_y	LeftWaist_y	LeftKnee_x	LeftKnee_y	LeftFoot_x	LeftFoot_y	RightEye_x	RightEye_y	LeftEye_x	LeftEye_y	RightEar_x	RightEar_y	LeftEar_x	LeftEar_y	person_class
333	822	338	822	403	816	479	851	335	840	407	833	477	833	207	842	205	820	209	851	207	teacher
425	947	443	0	0	0	0	969	450	0	0	0	0	989	322	998	324	967	322	0	0	
0	1247	601	0	0	0	0	1198	605	0	0	0	0	0	0	1184	414	0	0	1213	418	student
412	382	454	0	0	0	0	409	452	0	0	0	0	0	0	0	0	369	392	393	387	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1053	459	1007	450	student
0	0	0	0	0	0	0	0	0	0	0	0	0	1182	338	1191	340	0	0	1209	346	
335	798	349	789	407	789	461	820	346	807	410	802	461	804	230	811	230	791	234	818	234	teacher
0	898	439	0	0	0	0	913	452	0	0	0	0	931	328	942	331	913	328	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	982	446	940	439	student
0	0	0	0	0	0	0	0	0	0	0	0	0	1067	414	1073	412	0	0	1104	414	student-back
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1102	344	0	0	1122	353	
335	813	349	804	418	804	479	836	349	824	423	818	479	820	218	831	216	809	223	838	220	teacher
0	920	441	0	0	0	0	944	452	0	0	0	0	962	326	973	331	942	326	0	0	
0	0	0	0	0	0	0	1169	457	0	0	0	0	1144	340	1156	342	0	0	1180	353	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1020	454	973	448	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1169	430	1122	428	student
333	818	346	809	416	807	479	842	344	829	416	820	479	824	209	836	209	816	218	842	212	teacher
0	933	446	0	0	0	0	951	450	0	0	0	0	969	322	980	324	949	324	0	0	
0	0	0	0	0	0	0	1187	446	0	0	0	0	1156	340	1169	342	0	0	1196	353	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1193	436	1138	430	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1029	457	980	452	student
333	818	349	804	421	804	479	842	346	827	421	820	479	829	209	840	209	818	216	844	214	teacher
0	933	448	0	0	0	0	949	464	0	0	0	0	969	322	980	324	949	322	0	0	
0	1149	452	0	0	0	0	1182	452	0	0	0	0	1142	342	1151	344	0	0	1180	353	
0	0	0	0	0	0	0	1120	621	0	0	0	0	0	0	0	0	1193	436	1138	430	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1029	457	980	448	student
335	818	340	813	403	807	475	840	340	831	405	818	472	831	209	840	209	818	218	844	214	teacher
0	931	446	0	0	0	0	956	448	0	0	0	0	967	322	980	324	947	322	0	0	
0	1178	628	0	0	0	0	1120	610	0	0	0	0	0	0	0	0	1189	436	1140	430	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1029	457	980	448	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1147	338	0	0	1167	353	
335	818	349	804	416	804	475	840	346	827	416	818	477	827	209	838	209	816	218	842	214	teacher
0	933	448	0	0	0	0	949	459	0	0	0	0	967	322	978	324	947	324	0	0	
0	0	0	0	0	0	0	1120	612	0	0	0	0	0	0	0	0	1191	439	1138	432	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1029	457	980	450	student
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1184	356	1151	356	
335	818	349	807	418	807	477	840	346	822	418	818	479	822	209	836	209	813	216	842	212	teacher
0	929	446	0	0	0	0	949	461	0	0	0	0	967	322	980	326	947	324	0	0	

图 21 特征提取部分二

(4)加入压缩算法的 openpose：

在此步骤 3 中，可用压缩后的 openpose 提取特征，与原版 openpose 模型进行速度上（识别一帧画面所需时间）和准确率上的对比。我们研究了 Intel 研究员 Daniil Osokin 的 "Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose" [15] 这篇文章提到的方案。并根据 https://github.com/Daniil-Osokin/lightweight-human-pose-estimation.pytorch 代码库在 GTX1080TI 与 Jetson TX2 设备上分别进行测试，下图为对一帧（1270，720）图像检测所用时间对比，如图 22-23 所示。

```
time = 0.2877s
time = 0.3026s
time = 0.2895s
time = 0.3152s
time = 0.3060s
time = 0.2858s
time = 0.3116s
time = 0.2897s
time = 0.2996s
time = 0.2876s
time = 0.3083s
time = 0.2832s
time = 0.2902s
time = 0.2970s
time = 0.2950s
time = 0.3190s
time = 0.3160s
time = 0.3091s
time = 0.3129s
```

图 22 原版 openpose 检测时间

```
time = 0.0338s
time = 0.0238s
time = 0.0223s
time = 0.0227s
time = 0.0228s
time = 0.0219s
time = 0.0223s
time = 0.0225s
time = 0.0227s
time = 0.0220s
time = 0.0220s
time = 0.0220s
time = 0.0225s
time = 0.0226s
time = 0.0224s
time = 0.0219s
time = 0.0219s
time = 0.0217s
time = 0.0219s
```

图 23 lightweight-human-pose 检测时间

后续研究中我们将考虑将这两种版本的 openpose 与剪枝、量化、知识蒸馏等方案相结合，探索进一步的压缩方法。

模型压缩方案细节介绍，剪枝算法根据裁剪粒度的不同，可以划分为粗粒度剪枝和细粒度剪枝（又称滤波器级剪枝和连接级剪枝），我们使用粗粒度剪枝算法，通过删除模型中冗余参数来删除压缩和加速模型效果，重点在于对模型结构进行裁剪；量化利用了模型中参数所占用的硬件空间冗余，用更少的比特来表示参数，重点是压缩参数本身比特位的大小。知识蒸馏就是利用强大的教师网络来引导更浅更薄的学生网络进行学习，使学生网络的性能尽可能接近于教师网络的性能，其重点是利用教师网络中丰富的信息来提高学生网络的准确率。我们发现上述三种压缩算法是正交的。当它们结合在一起时，能从不同的角度压缩和加速模型。图 24 是拟采用的压缩框架路线图。

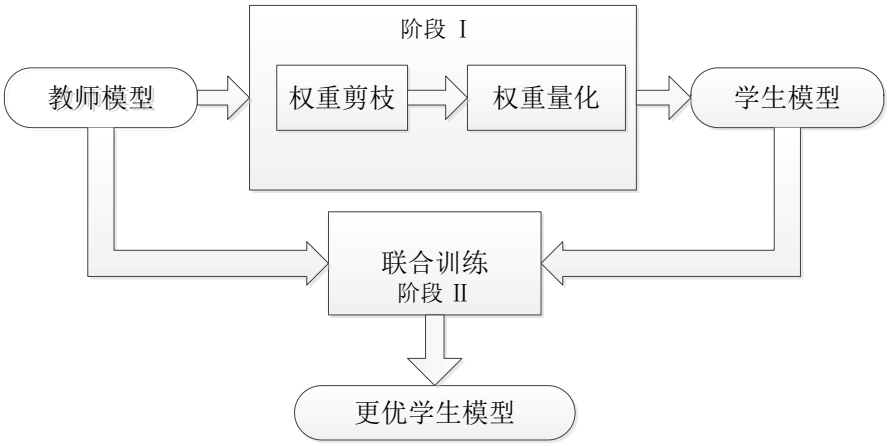

图 24 模型压缩技术框架图

首先，利用过滤级剪枝算法[16]对教师模型（这里的教师模型即 openpose 的 vgg 模块）进行剪枝。剪枝操作步骤:首先设定剪枝率，然后对整个网络中所有卷积核的重要性进行评估，找出最不重要的一个并将其剔除。然后判断此时的修枝

率是否高于设定的修枝率。如果没有，则重复修剪操作或结束修剪。剪枝后得到的模型非常不稳定。我们需要对其进行再培训，并使模型调整其参数。最后，我们得到最终修剪后的学生模型，如图 25 所示。

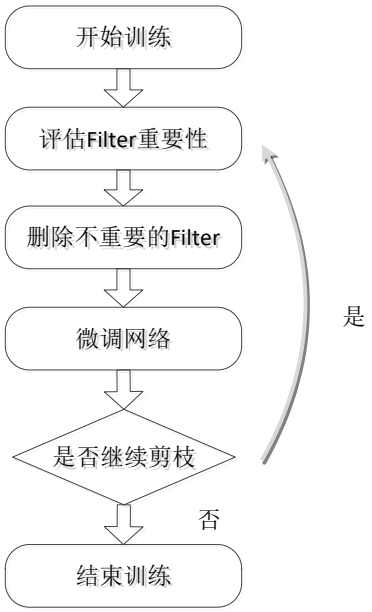

图 25 粗粒度剪枝流程图

第二，我们对剪枝后的学生模型进行量化[17]操作, 量化的核心操作是选择合适的尺度函数(scaling function),尺度函数有多种，这里我们选择线性 scaling 函数，$sc(v)=\frac{v-\beta}{\alpha}$，其中$\alpha=max_i v_i-min_i v_i$，$\beta=min_i v_i$ 将结果值映射在 [0,1]，量化函数为：

$$Q(v)=\alpha\hat{Q}\left(\frac{v-\beta}{\alpha}\right)+\beta \tag{1}$$

其中 $\hat{Q}$ 为真实的 scaling 函数，接受值的范围为 [0,1]，v 是一个向量（vector）。经过量化，得到一个新的学生模型。

最后，利用知识蒸馏[18]训练学生模型，利用教师提供的信息指导学生训练。联合训练的关键是不断优化蒸馏损失函数，定义为

$$L=\lambda L_h(y_{true},P_S)+(1-\lambda)L_s(P_T^t,P_s^t) \tag{2}$$

其中 L_h 代表 hard target 的交叉熵，L_s 代表 soft target 的交叉熵，P_s代表学生模型的预测值，P_s^t 和P_T^t 分别代表学生网络和教师网络带温度参数的预测值，λ 是用来平衡交叉熵的调节参数。经过联合训练，得到最终的学生模型。图 26 为知识蒸馏的流程框架图。

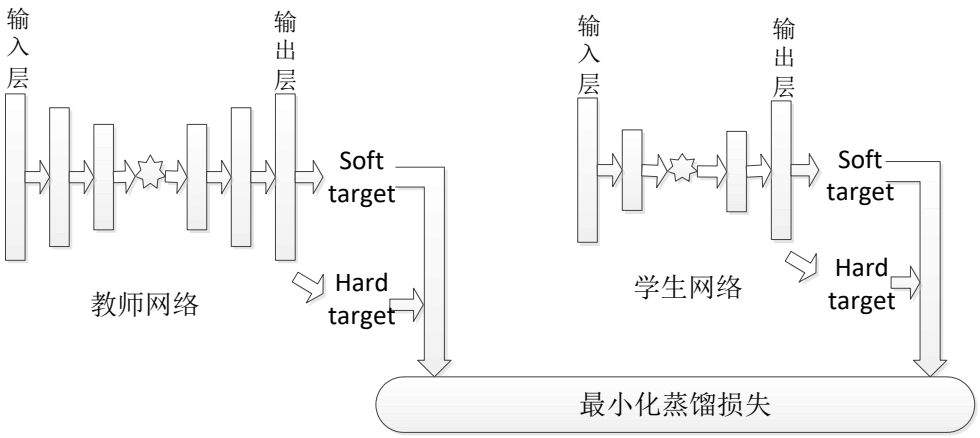

图 26 知识蒸馏框架图

我们对上述方案用一个自定义的小型网络在 CIFAR-10 数据集上进行验证，将我们的想法与每种算法进行比较，初步证明其有效性。结果表明，教师网络的准确率可以达到 top 1 的 85.84%，仅使用剪枝算法就可以减少模型 90%的计算复杂度，准确率损失为 2%。仅使用量化算法，精度损失为 12%。当剪枝量化和蒸馏算法一起使用时，精度损失仅为 1%，图 27-28 为具体结果。

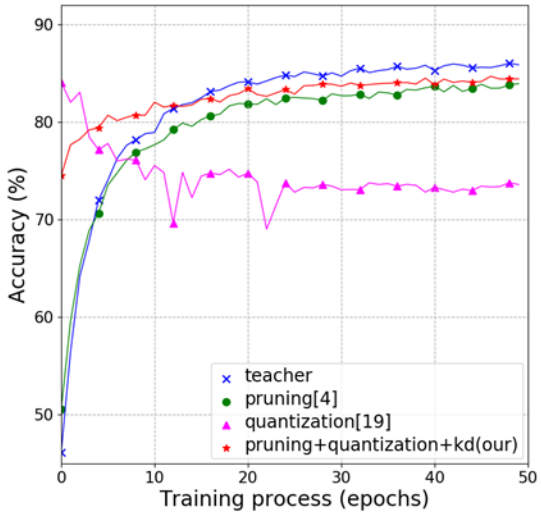

图 27 三种算法训练过程

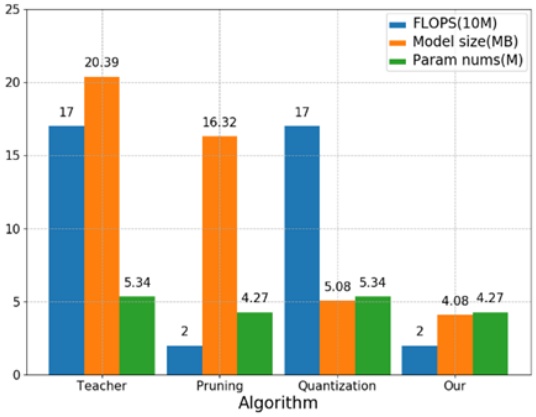

图 28 三种算法指标比较

(5) 训练分类器：

考虑到数据集较少，不适合使用神经网络来训练分类器，所以使用经典分类算法，随机森林、svm 分类器、lightgbm、XGBoost，随机森林主要是应用于回归和分类这两种场景，又侧重于分类。研究表明，组合分类器比单一分类器的分类效果好，随机森林是指利用多棵决策树对样本数据进行训练、分类并预测的一种方法，它在对数据进行分类的同时，还可以给出各个变量（基因）的重要性评分，

评估各个变量在分类中所起的作用；svm 又称支持向量机，是将向量映射到一个更高维的空间里，在这个空间里建立有一个最大间隔超平面。在分开数据的超平面的两边建有两个互相平行的超平面。分隔超平面使两个平行超平面的距离最大化。假定平行超平面间的距离或差距越大，分类器的总误差越小。lightgbm 是个快速的，分布式的，高性能的基于决策树算法的梯度提升框架。可用于排序，分类，回归以及很多其他的机器学习任务中。XGBoost 是 boosting 算法的其中一种。Boosting 算法的思想是将许多弱分类器集成在一起形成一个强分类器。因为 XGBoost 是一种提升树模型，所以它是将许多树模型集成在一起，形成一个很强的分类器。而所用到的树模型则是 CART 回归树模型。

(6) 模型融合：

要想使得使得机器学习模型进一步提升，我们必须使用到模型融合的技巧。模型融合中比较常见的一种方法——stacking，又称模型堆叠，是指将多个模型融合到一起。一般流程如下：

- 首先将数据分为 5 份。
- 在 stacking 的第一层定义 5 个基模型 [model_1 ,model_2,model_3,model_4,model_5]，其中每个模型选择做一下 5 折的交叉验证的预测，这样就相当于每个模型将所有数据预测了一遍，举个例子，最终每一个训练数据会被转换为[1,1,1,1,0]形状，维度为 5 的向量。
- 将第一层 5 个基模型的输出预测向量[1,1,1,1,0]，作为第二层模型 model_6 的特征做训练。
- 做 test 时，直接将 test 的数据输入给之前第一层训练好的 5 个基模型，5 个模型预测出的至平均后作为第二层模型的输入。

图 29 为 stacking 流程：

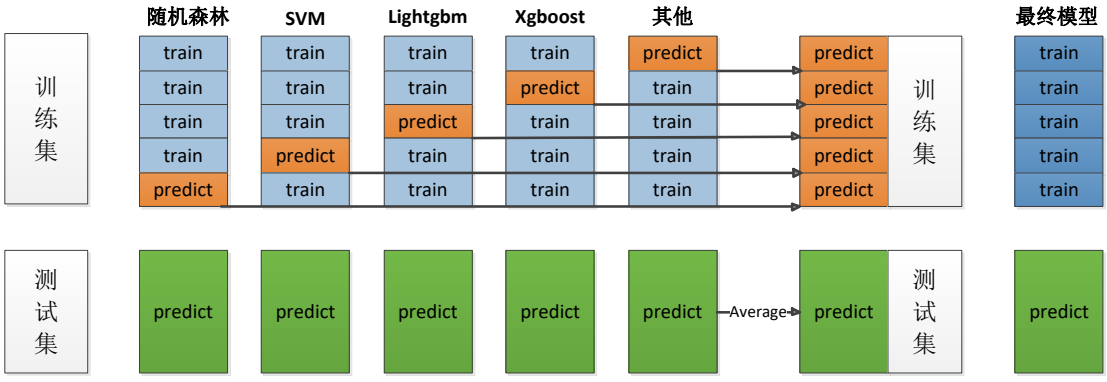

图 29 模型融合流程图

(6) 获得最终模型

经过以上步骤，最终的模型就训练好了，接下来将模型投入使用。

3.2.2 测试过程

(1)环境部署：

在测试环境教室安装 jetson tx2，及相机装置。放在教室后方中央顶部作为教师机位。

(2)采集数据：

在 jetson tx2 上配置 openpose 使用环境，并且加入训练好的分类模型，将相机采集的数据输入给 jetson tx2 设备，并启动 openpose 程序，根据视频帧率调节间隔采集画面。

(3)openpose 提取特征：

在 jetson tx2 上载入已训练好的分类模型，并配置模型运行所需环境，将 openpose 的预测结果（肢体坐标）直接输入给分类模型。

图 30-33 为在小样本下训练结果，由于样本采集处理较好，分类准确率较高，在实际应用中采集的画面会受各种因素干扰，会有噪声等不确定因素，使用 stacking 方法能提升模型准确率。

```
In [56]: # svm分类器
clf = svm.SVC(kernel='linear',C=100)
clf.fit(X_train,y_train)
a=clf.predict(X_test)
b=clf.get_params() #查看模型的参数
c=clf.score(X_test,y_test)#查看匹配度，正确率
print("svm = ",c)

svm =  0.972096041531
```

图 30 svm 分类器结果

```
In [57]: # xgboost

model = XGBClassifier()            # 载入模型
model.fit(X_train,y_train)         # 训练模型（训练集）
y_pred = model.predict(X_test)     # 模型预测（测试集）
accuracy = accuracy_score(y_test,y_pred)
print("xgb accuarcy: %.2f%%" % (accuracy*100.0))

xgb accuarcy: 99.16%
```

图 31 xgboost 分类结果

```
In [60]: # 随机森林
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
pre_test = rf.predict(X_test)
accuracy = accuracy_score(y_test,pre_test)
print("随机森林 accuracy:",accuracy)

随机森林 auc_score,pre_score: 0.985723556132
```

图 32 随机森林分类结果

```
In [61]: # lightgbm
train_data=lgb.Dataset(X_train,label=y_train)
validation_data=lgb.Dataset(X_test,label=y_test)
params={
    'learning_rate':0.1,
    'lambda_l1':0.1,
    'lambda_l2':0.2,
    'max_depth':6,
    'objective':'multiclass',
    'num_class':6,  }
clf=lgb.train(params,train_data,valid_sets=[validation_data])
pp = precision_score(y_test, y_pred,average='micro')
print('lightgbm precision=',pp)
# lightgbm precision= 0.991563919533
```

图 33 lightgbm 分类结果

(4) 保存结果

以 1~5s 为间隔统计课堂发生的动作，将学生与教师的预测结果保存成 csv 格式文件。后期用 pandas、matplotlib 工具包做出图形界面展示，图 34 为课堂行为分析效果。

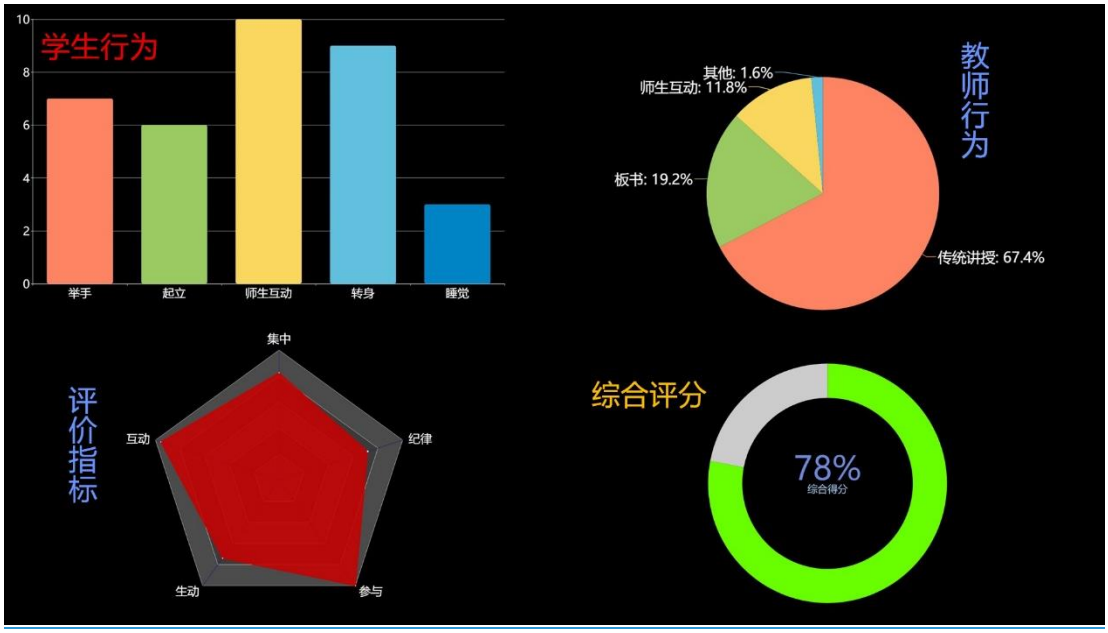

图 34 课堂行为分析

3.3 技术重点

(1) 多目标人体关键点实时检测技术

OpenPose 人体姿态识别项目是美国卡耐基梅隆大学（CMU）基于卷积神经网络和监督学习并以 caffe 为框架开发的开源库。可以实现人体动作、面部表情、手指运动等姿态估计。适用于单人和多人，具有极好的鲁棒性。是世界上首个基于深度学习的实时多人二维姿态估计应用。该框架为本实验的基础环境，为后续实验奠定基础。

(2) 基于卷积神经网络模型压缩与加速的研究

目前的压缩方法可以分为五类：剪枝（pruning）、量化（quantization）、知识蒸馏（knowledge distillation）、低秩分解（low-rank decomposition）、设计精简网络

（simplify network）。我们对剪枝方面有粗细粒+细粒度裁剪的研究成果，还有将剪枝、量化、知识蒸馏结合使用的研究成果。将模型压缩与 openpose 的结合，是本实验的重点及难点。根据 openpose 框架结构，压缩方法适用于 openpose 的第一个模块，即 vgg 前十层进行压缩。

3.4 计划和分工

计划：在七月 22 日之前完成初赛部分的要求，使用自己的硬件平台进行算法的实现以及移植的实现，并完成计划书。
在八月上旬学习 intel 至强云计算平台以及 intel Movidius 神经计算平台的使用，熟悉这两个平台的构架以及在上面开发的方法
在九月至十一月上旬这段时间完善算法，以及完成在 intel 平台上的移植，做出参赛 ppt 准备参加比赛

分工
组长：刘佳阳，负责技术的探索，主要神经网络压缩部分以及算法实现方面
组员：郭鹏，负责神经网络训练部分，训练数据标注。
组员：王悦章，负责神经网络推断部分，算法在推断平台上的迁移
组员：秘博闻，负责文档撰写，视频制作演示文档制作以及讲解

4 参考资料

[1] He A, Luo C, Tian X, et al. A twofold siamese network for real-time object tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4834-4843.
[2] Lao Y, Ait-Aider O, Bartoli A. Rolling Shutter Pose and Ego-motion Estimation using Shape-from-Template[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 466-482.
[3] Xu W, Chatterjee A, Zollhöfer M, et al. Monoperfcap: Human performance capture from monocular video[J]. ACM Transactions on Graphics (ToG), 2018, 37(2): 27.
[4] Liu Y, Chen K, Liu C, et al. Structured Knowledge Distillation for Semantic Segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 2604-2613.
[5] Qi C R, Litany O, He K, et al. Deep Hough Voting for 3D Object Detection in Point Clouds[J]. arXiv preprint arXiv:1904.09664, 2019.
[6] Zhu N, Bai X. Neural Architecture Search for Deep Face Recognition[J]. arXiv preprint arXiv:1904.09523, 2019.

[7] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks[C]. Proceedings of Advances in Neural Information Processing Systems (NIPS), 2012: 1097-1105

[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[C]. Proceedings of International Conference on Learning Representations (ICLR), 2015:1-14.

[9] Cao Z, Simon T, Wei S E, et al. Realtime multi-person 2d pose estimation using part affinity fields[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 7291-7299.

[10] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.

[11] Szegedy, C.; Zaremba, W. (2014).Intriguing properties of neural networks.arXiv:1312.6199v4.

[12] He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): 770-778.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. YOLO: Unified, Real-Time Object Detection. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[14] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[15] Osokin D. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose[J]. arXiv preprint arXiv:1811.12004, 2018.

[16] Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. 2017. Pruning convolutional neural networks for resource efficient inference. In International Conference of Learning Representation. arXiv preprint arXiv:1611.06440

[17] Polino, A., Pascanu, R., and Alistarh, D.2018. Model compression via distillation and quantization. arXiv preprint arXiv:1802.05668.

[18] Hinton, G., Vinyals, O., and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[19] 雷杰,高鑫,宋杰，王兴路，宋明黎，深度网络模型压缩综述[J]软件学报 2018,29(2):251-266

[20] 纪荣嵘,林绍辉,晁飞,吴永坚,黄飞跃.深度神经网络压缩与加速综述[J].计算机研究与发展,2018,55(09):1871-1888.