

基于轻量化网络的课堂学生学习状态判别

刘秋会, 王 林, 梁明秀

贵州民族大学数据科学与信息工程学院, 贵州 贵阳
Email: 1753879994@qq.com

收稿日期: 2021年3月29日; 录用日期: 2021年4月22日; 发布日期: 2021年4月29日

摘 要

课堂学习状态判别是了解学生课堂学习情况及教师授课情况的关键步骤, 轻量化网络能够提高学习状态的判别精度, 轻量化网络是在神经网络算法的基础上进行优化的网络, 在本文中, 我们采用改进的轻量化网络Mobilnetv2对教室课堂学生学习状态进行判别, 实验结果表明, 通过提出的方法得到的课堂学生学习状态判别最高达到了99.00%的精度。

关键词

深度学习, 课堂学生学习状态判别, Mobilnetv2, Swish

Learning State Discrimination of Classroom Students Based on Lightweight Network

Qiuhui Liu, Lin Wang, Mingxiu Liang

School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang Guizhou
Email: 1753879994@qq.com

Received: Mar. 29th, 2021; accepted: Apr. 22nd, 2021; published: Apr. 29th, 2021

Abstract

Classroom learning status discrimination is a key step to understand students' classroom learning status and teachers' teaching status. Lightweight networks can improve the accuracy of learning status. Lightweight networks are optimized networks based on neural network algorithms. In this article, the improved lightweight network Mobilnetv2 is used to discriminate the learning status of classroom students. The experimental results show that the accuracy of class student learning status discrimination obtained by the proposed method is up to 99.00%.

Keywords

Deep Learning, Classroom Student Learning State Detection, Mobilenetv2, Swish

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

学生课堂学习状态的研究与学生课堂学习效率及教师授课情况反馈息息相关,可作为学生课堂学习效率及教师授课情况的评价指标之一。目前课堂场景中对于学生课堂学习状态的研究主要从课堂学习行为[1]、课堂疲劳状态[2]、课堂人脸检测及关注度研究方面[3]展开,本文主要的工作是在已经检测出学生头部的基础上,对课堂中学生头部状态进行分类判别。

在计算机视觉和模式识别中,图像分类常用基于卷积神经网络的分类模型进行研究,1998年,LeNet模型[4]诞生,是最早的卷积神经网络之一,最早应用与MNIST [5]手写识别数字的识别;该模型参数少且结构单一,并不适用于复杂的图像分类任务。2012年,AlexNet模型[6]作为ILSVR的冠军网络被人们认识,该模型使用ReLU [7]激活函数,大大提高了训练速度,同时使用了随机失活操作,避免了一定程度上的过拟合,但对图像特征的描述及提取仍然有限。GoogLeNet模型[8]是ILSVR2014的冠军网络,最大特点在于引入Inception模块,该模块有四个分支且可以跨通道组织信息,大大提高了参数的利用效率,但同时也增加了计算量和出现过拟合。VGGNet模型[9]是ILSVR2014的亚军网络,由AlexNet模型发展而来,在卷积层使用较小的卷积核并采用多尺度训练策略,但是模型深度增加导致训练速度缓慢。ILSVR2015的冠军网络是ResNet模型[10],该模型旨在解决网络模型加深而错误率提高的问题,当模型复杂以后,随机梯度下降(SGD) [11]的优化变得更加困难,而模型达不到好的学习效果。Residual结构的提出,对卷积神经网络的发展产生了重要的意义。2016年,ResNeXt [12]通过重复一个构建块来构建,聚合了一组具有相同拓扑结构的转换,在保持复杂的限制条件下,增加基数也能很好地提高分类精度。SENet模型[13]是ILSVR2017的冠军网络,该网络通过额外的分支来得到每个通道的权重,自适应地校正原各通道激活值响应。2017年Mobilenet模型[14]提出,作为Mobilenet系列的第一个版本,该模型主要采取可分离卷积的方式,降低参数量却不降低精度。2018年Mobilenetv2模型[15]在Mobilenetv1的基础上加入倒残差结构,从原来的残差结构进行降维到升维变成从升维到降维。保留了低维的信息特征。

图像分类虽然层出不穷,但是对于不同的研究目标有不同的适用网络,本文针对教室场景,以学生头部为目标,改进Mobilenetv2模型对学生头部图像进行分类,最后进行学习状态的判别。

2. 基本原理

2.1. Mobilenetv2

MobileNetv2 [15]网络由google团队在2018年提出,在MobileNetv2网络中的重点是倒残差结构,如图1所示,(a)是Resnet网络中的残差结构,(b)是MobileNetv2中的倒残差结构。在残差结构中是 1×1 卷积降维到 3×3 卷积再到 1×1 卷积升维,在倒残差结构中正好相反,是 1×1 卷积升维到 3×3 卷积再到 1×1 卷积降维。这样做的原因在于高维信息通过ReLU激活函数后丢失的信息更少,而低维的运用ReLU激活函数信息丢失比较多,所以在最后一个 1×1 的卷积层使用线性激活函数,来保留低维的信息特征。

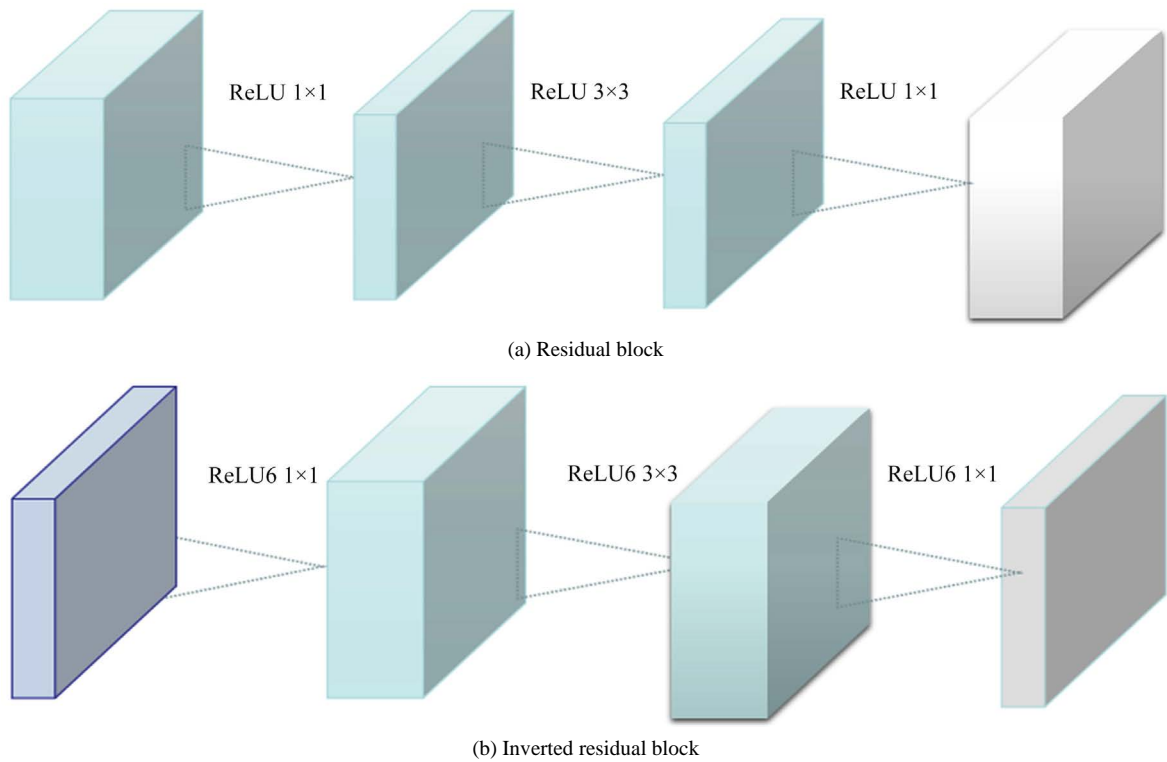


Figure 1. Residual block and inverted residual block

图 1. 残差块与倒残差块

表 1 是 MobileNetv2 网络的结构表, 其中 t 代表的是扩展因子(倒残差结构中第一个 1×1 卷积的扩展因子), c 代表输出特征矩阵的通道, n 代表倒残差结构重复的次数, s 代表步距(注意: 这里的步距只是针对重复 n 次的第一层倒残差结构, 后面的都默认为 1)。步距不同的网络结构如图 2 所示。

Table 1. MobileNet v2 network structure

表 1. MobileNet v2 网络结构

Input	Operator	t	c	n	s
$224^2 \times 3$	Conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	Con2d 1×1	-	1280	1	1
$7^2 \times 1280$	Avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	Con2d 1×1	-	k	-	-

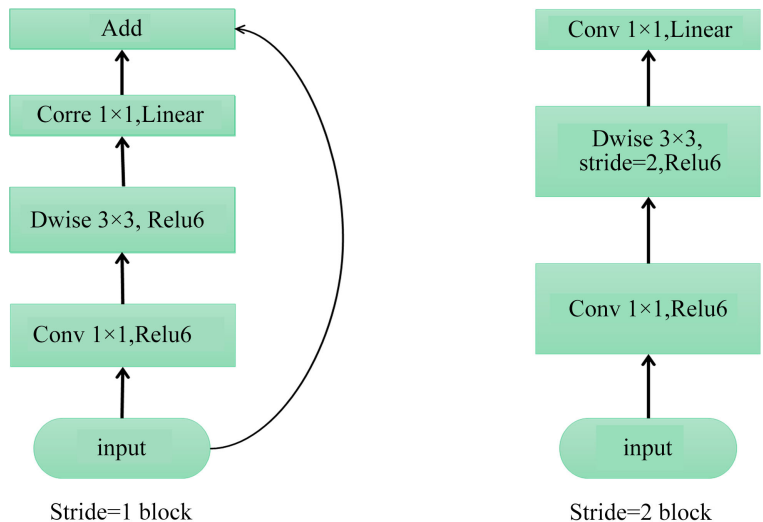


Figure 2. Different network structures of Stride
图 2. Stride 不同的网络结构

2.2. 激活函数

2.2.1. ReLU 函数

ReLU 函数是目前深度学习中用得较多且广受欢迎的非线性激活函数。在 $x \geq 0$ 时才有非零输出，在 $x < 0$ 时输出为 0，将 ReLU 函数定义为：

$$f(x) = \max(0, x) \tag{1}$$

且函数曲线如图 3 所示。

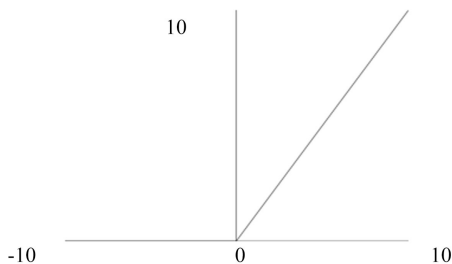


Figure 3. ReLU function curve
图 3. ReLU 函数曲线

经过求导计算，可得(1)的导函数为：

$$f'(x) = \max(0, 1) \tag{2}$$

由(2)可知，ReLU 函数其实是最大值函数，与 Sigmoid 及 Tanh 函数相比，当输入值为正数时，ReLU 函数运算非常简单且快速，同时也避免了梯度消失。拥有更加好的网络性能。当输入为负数时，ReLU 是完全不被激活的，ReLU 函数就会和 Sigmoid 及 Tanh 函数存在一样的问题。

2.2.2. Swish 函数

Swish 函数是 x 与 sigmoid 函数的结合，公式为

$$f(x) = x * sigmoid(\beta x) \tag{3}$$

且函数曲线如图 4。

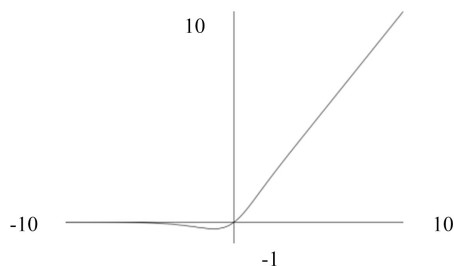


Figure 4. Swish function curve ($\beta=1$)

图 4. Swish 函数曲线($\beta=1$)

对(3)求导可得:

$$f'(x) = \begin{cases} \frac{x}{2}, & \beta = 0 \\ 2 \max(0, x), & \beta \rightarrow \infty \end{cases} \quad (4)$$

由(4)可以看出, β 是个常数且 Swish 激活函数具备有下界、平滑、非单调的特性。其也被看成是介于线性函数与 ReLU 函数之间的平滑函数。Swish 激活函数在正值区域可以达到任何高度,避免了由于封顶而导致的梯度饱和。对负值区域,与 ReLU 相比,有轻微的梯度流。平滑的激活函数允许更好的信息深入神经网络,从而得到更好的准确性和泛化。

3. 方法

3.1. 数据增强

在深度学习中,样本数量越多,训练出来的模型效果越好,模型的泛化能力越强。由于自建数据集的样本数量不是很多,为了提高模型的泛化能力,本文采用了三种方法对原有的数据进行增强,包括颜色扰动、高斯噪声滤波以及水平翻转。颜色扰动,指在某一个颜色空间通过增加或减少某些颜色分量,或者更改颜色通道的顺序。高斯噪声就是符合高斯正态分布的误差。有时我们需向标准数据中加入合适的高斯噪声来让数据变得有一定误差,从而让数据变得更具有实验价值。水平翻转是翻转中的一种,它是以坐标 y 轴为对称轴进行操作的过程,我们在这里使用水平翻转而没有使用全方位翻转的原因在于我们的数据具有一定程度的相似性,而且我们识别抬头低头的状态多数都是从正常的视觉出发,为了让数据增强具有合理性,我们使用了水平翻转。操作效果图如图 5。

其中(a)表示原图,(b)表示经过颜色扰动后的图,(c)表示添加高斯噪声后的图,(d)表示水平翻转后的图。

3.2. 更换激活函数

激活函数是在激活层起主要作用的非线性函数, Mobilenetv2 模型用到的激活函数为 ReLU6, ReLU6 其实是 ReLU 函数的变形,它的定义如下式:

$$\text{ReLU6} = \min(6, \max(0, x)) \quad (5)$$

且函数曲线如图 6。

ReLU6 函数实际上是把 ReLU 函数取了上界,但当在负值区域,容易出现函数失活的问题,所以考虑一种更好的激活函数 Swish 来替代 ReLU6。



Figure 5. Data enhancement effect diagram
图 5. 数据增强效果图

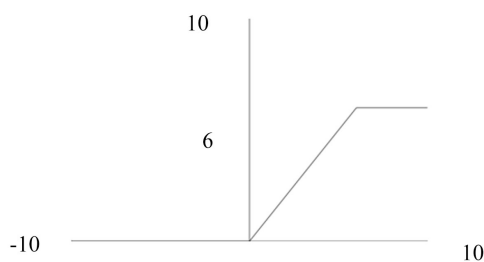


Figure 6. ReLU function curve
图 6. ReLU 函数曲线图

4. 实验结果

4.1. 实验环境

本节实验软件为 Python3.6 计算机编程语言，框架 Tensorflow2.2.0 及 Keras2.2.4 环境，硬件为 64 位 Windows 操作系统，Intel(R)Core(TM)i7-9700 CPU 和 16GB RAM，独立显卡 AMD Radeon Pro WX3100。

4.2. 实验数据

本文实验用到的数据集由模型 Yolov4 与 DropBlock 结合算法检测到的学生头部结果图片制成，抬头低头图片数量各为 1065 张，总计 2130 张，命名为 ClassUD。数据增强以后的数据集命名为 ClassUD-A，每一种增强方法对每一种状态的数据都进行一次操作，抬头与低头状态的数据集扩增到了 4260 张，总计 8520 张图片。数据分布如表 2。本文分类所需的数据皆是模仿监控视角的位置与高度拍摄所得，范围为一个教室的 3 至 4 排左右，且 240 张图片由 40 分钟视频以 10 秒一张图片的截取方式获得，对每张图片出现的学生头部进行裁剪且对不同的学生头部进行编号，此编号就代表这位学生，制成的数据集命名为

ClassTest, 依此类推, 每一个编号的图片数应为 240 张, 有些编号少于 240 张是由于在某些帧中头部存在被遮挡的情况, 数据分布如表 3 所示。

Table 2. Training experiment data distribution

表 2. 训练实验数据分布

数据集	抬头训练集	低头训练集
ClassUD	1065	1065
ClassUD-A	4260	4260

Table 3. Test data distribution

表 3. 测试实验数据分布

数据集	编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	ClassTest		240	240	219	240	215	240	240	240	226	240	52	240	240

4.3. 实验对比

4.3.1. 数据增强实验

本节中的实验分别在五个深度学习分类模型上进行, 迭代次数为 80 次, 批量输入尺寸为 32。第一个实验用 Vgg16 模型进行训练及测试, 第二个实验用 Googlenet 模型进行训练及测试, 第三个实验是在 Resnet50 模型下进行训练及测试, 第四个实验在 Mobilenetv2 模型下进行训练及测试。第五个实验用 Alexnet 模型进行训练及测试, 所有实验均在数据集 ClassUD 及 ClassUD-A 上进行。学习率是深度学习中的一个重要的超参, 如何设置学习率是训练出好模型的关键要素之一。为了在五个模型上对同一数据集做实验对比分析, 需要在同一实验条件下进行实验, 所以对于模型的学习率, 根据经验分别设置了 0.001 及 0.0001 两个不同的值。

当学习率为 0.001 时, 在数据集 ClassUD 及 ClassUD-A 上的实验结果如表 4 及图 7 所示。

Table 4. Experimental results when the learning rate is 0.001

表 4. 学习率为 0.001 时实验结果

数据集	方法	Vgg16	Googlenet	Resnet50	Mobilenetv2	Alexnet
	ClassUD	训练时间/(小时)	11.50	2.4	1.23	4.33
平均训练精度(%)		49.32	91.88	81.92	98.33	77.65
平均验证精度(%)		50.03	81.28	77.96	91.33	70.83
ClassUD-A	训练时间/(小时)	49.22	10.06	5.29	18.57	1.41
	平均训练精度(%)	49.54	96.38	79.05	99.03	96.38
	平均验证精度(%)	50.02	95.85	81.82	98.77	95.72

由表 4 及图 7 可知, 当学习率为 0.001 时, 在数据集 ClassUD 上, 训练耗时最少的模型为 Alexnet, 约为 0.42 小时, 耗时最多的模型为 Vgg16, 约为 11.50 小时, 训练与验证精度最高的模型为 Mobilenetv2, 分别为 98.33% 与 91.33%, 训练与验证精度最低的模型为 Vgg16, 约为 49.32% 与 50.03%。在数据集 ClassUD-A 上, 训练耗时最少的依然是 Alexnet 模型, 约为 1.41 个小时, 耗时最多的依然是模型 Vgg16, 约为 49.22 个小时, 训练与验证精度最高的模型为 Mobilenetv2, 分别为 99.03% 及 98.77%, 训练及验证精度最低的模型为 Vgg16, 约为 49.54% 及 50.02%。

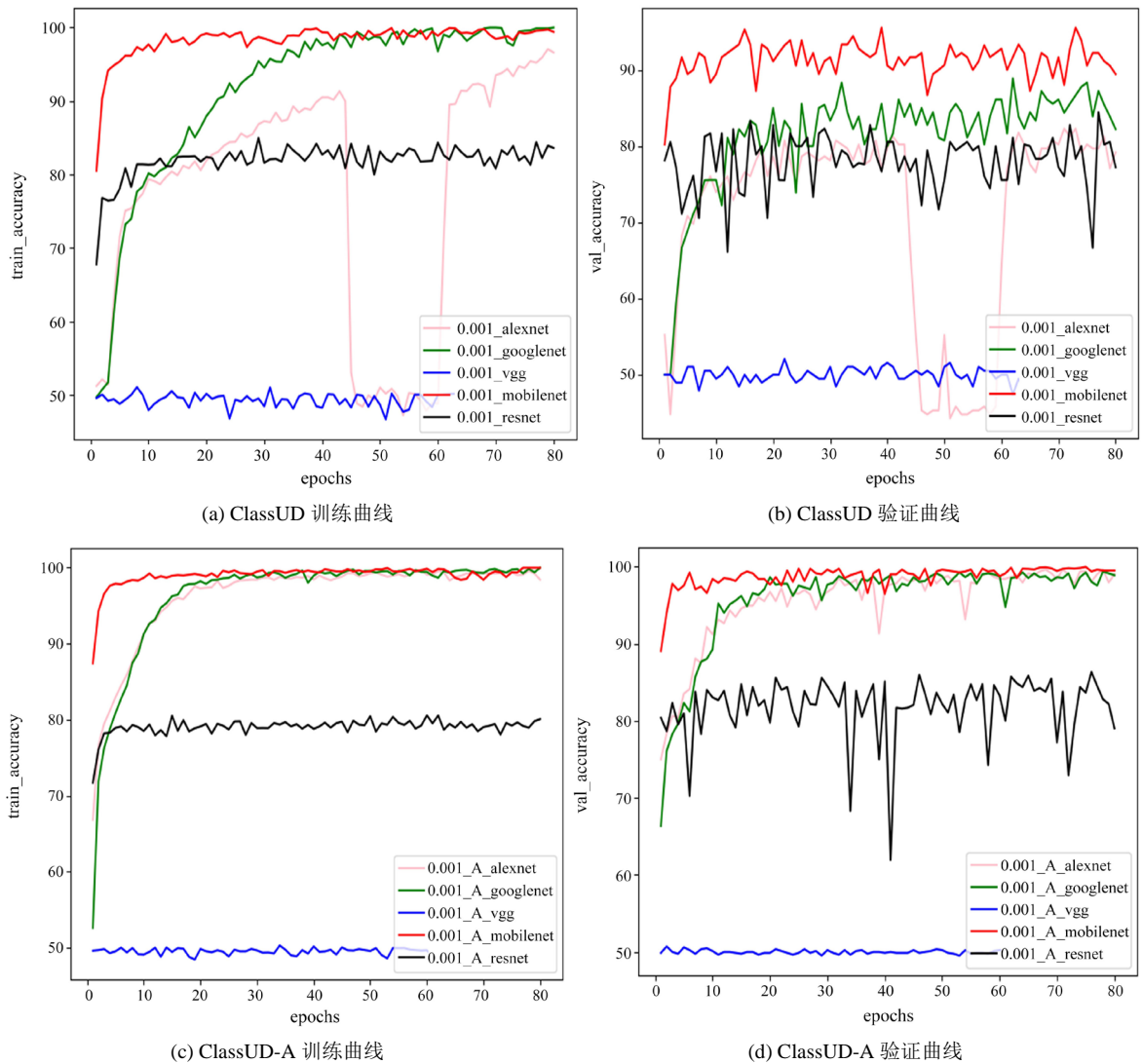


Figure 7. Training and validation diagrams of five models
图 7. 五个模型的训练及验证图

数据增强后，对于 Alexnet、Mobilenetv2 及 Googlenet 模型来说，训练及验证精度都得到了不同程度上的提升；而对于 Vgg16 模型来说，数据增强几乎没有什么效果；对于 Resnet50 模型，训练精度反而还下降了；但是训练时间也同时在成倍的增长。Vgg16 及 Resnet50 首先不是我们考虑的模型，其实按照训练时间来说，选择 Alexnet 模型最为合适，但是它较依赖于数据集的大小，而对于 Googlenet 及 Mobilenetv2 模型在 ClassUD 上表现得比较好，在 ClassUD-A 上更好，网络相对稳定且在不断的学习。

当学习率为 0.0001 时，在数据集 ClassUD 及 ClassUD-A 上的实验结果如表 5 及图 8。

由表 5 及图 8 可知，当学习率为 0.0001 时，在数据集 ClassUD 上，训练耗时最少的模型为 Alexnet，约为 0.42 小时，耗时最多的模型为 Vgg16，约为 11.49 小时，训练与验证精度最高的模型为 Mobilenetv2，分别为 98.92% 与 87.76%，训练与验证精度最低的模型为 Resnet50，约为 82.41% 与 79.13%。在数据集 ClassUD-A 上，训练耗时最少的依然是 Alexnet 模型，约为 1.47 个小时，耗时最多的是模型 Vgg16，约为 49.22 个小时，训练与验证精度最高的模型为 Mobilnetv2，分别为 99.37% 及 98.82%，训练及验证精度最低的模型为 Resnet50，约为 81.94% 及 85.61%。

Table 5. Experimental results when the learning rate is 0.0001
表 5. 学习率为 0.0001 时实验结果

数据集 \ 方法	Vgg16	Googlenet	Resnet50	Mobilenetv2	Alexnet	
ClassUD	训练时间(/小时)	11.49	2.6	1.29	4.45	0.42
	平均训练精度(%)	94.23	93.95	82.41	98.92	95.13
	平均验证精度(%)	83.28	85.91	79.13	87.76	85.76
ClassUD-A	训练时间(/小时)	49.22	10.4	5.30	18.72	1.47
	平均训练精度(%)	98.16	97.68	81.94	99.37	98.30
	平均验证精度(%)	98.17	97.07	85.61	98.82	97.72

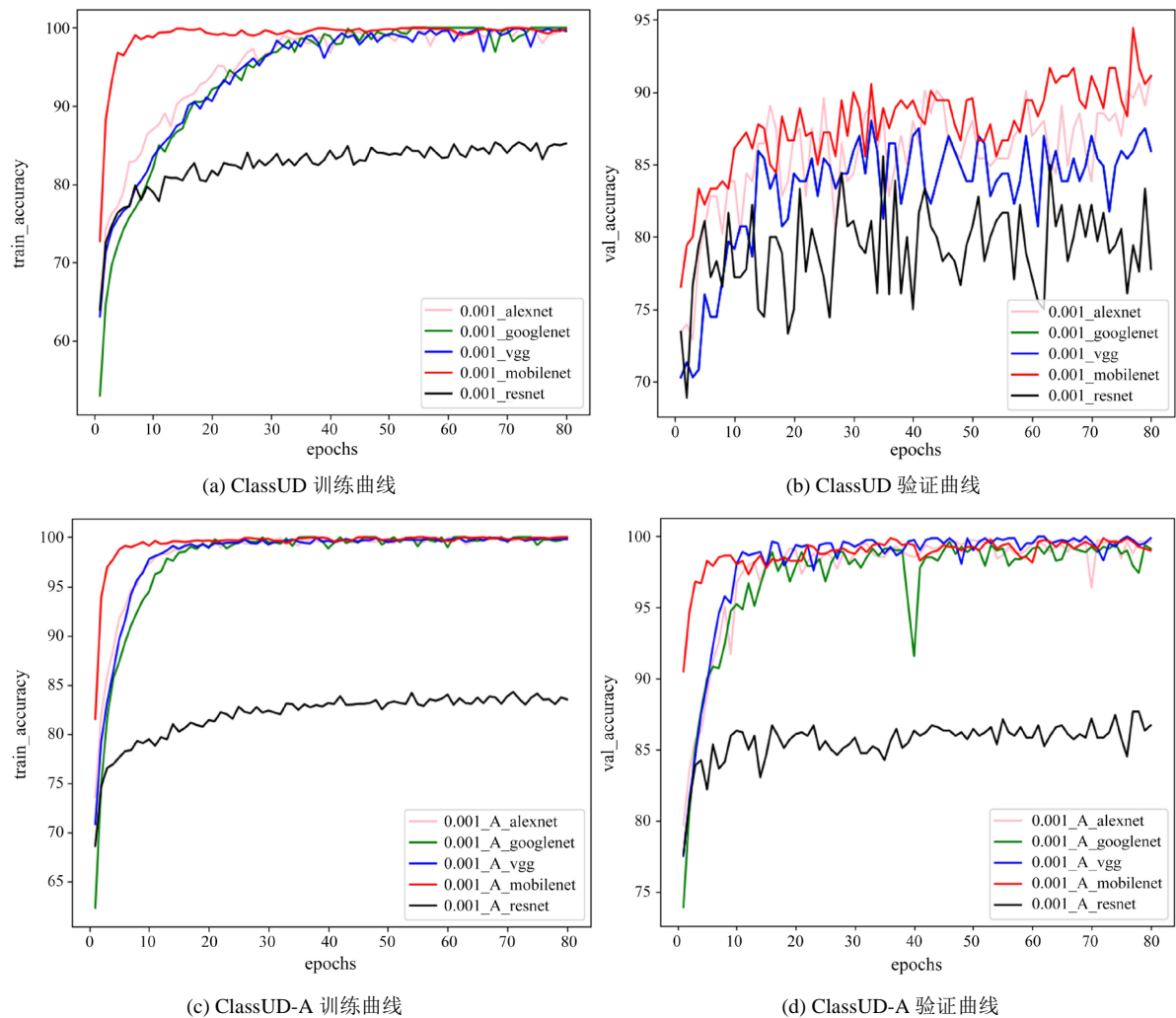


Figure 8. Training and validation diagrams of five models
图 8. 五个模型的训练及验证图

数据增强以后,对于 Alexnet、MobilenetV2、Googlenet 及 Vgg16 模型来说,训练及验证精度都得到了不同程度上的提升;对于 Resnet50 模型,训练精度反而下降了一点;训练时间也同时在成倍的增长。Resnet50 首先不是我们考虑的模型,同时 Vgg16 的训练时间太长,也不是我们考虑的模型。而 Alexnet、

Googlenet 及 Mobilenetv2 模型不管是在 ClassUD 上还是在 ClassUD-A 上都表现得比较好, 网络相对稳定且在不断的学习。

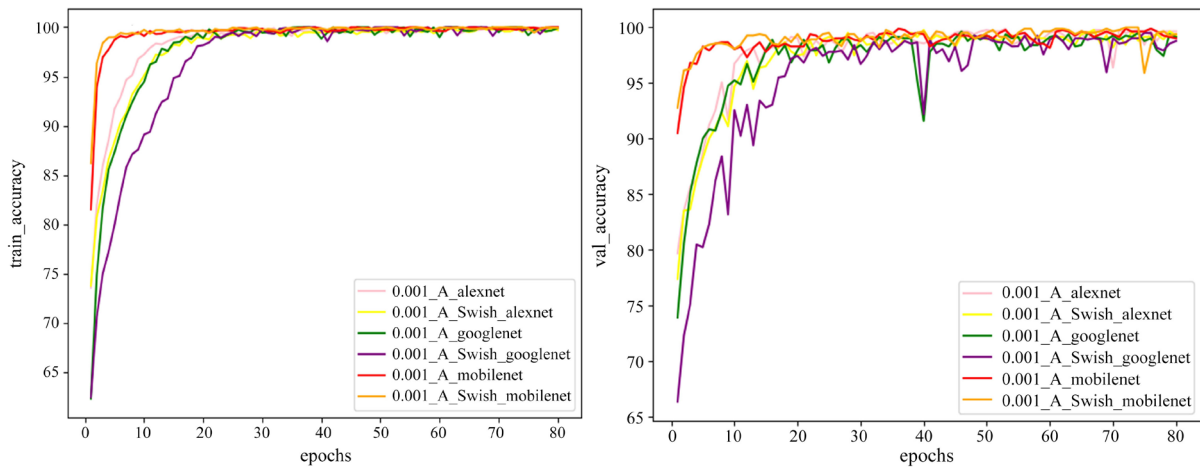
通过两个不同学习率及不同数据集下的实验, 得出在数据集 ClassUD-A 上的实验效果比数据集 ClassUD 效果更好, 在学习率为 0.0001 时比学习率为 0.001 时效果更好, 所以之后的实验将在数据集 ClassUD-A 及学习率为 0.0001 的条件下进行。

4.3.2. Swish 激活函数实验

分别进行三个实验, 把 Alexnet、Googlenet 及 Mobilenetv2 网络模型的激活函数替换为 Swish 后, 称这些网络为 Alexnet_S、Googlenet_S 及 Mobilenetv2_S, 并在数据集 ClassUD-A 上进行了 Alexnet、Alexnet_S、Googlenet、Googlenet_S、Mobilenetv2 及 Mobilenetv2_S 实验的对比。实验结果见表 6 及图 9。

Table 6. Experimental results of different models
表 6. 不同模型的实验结果

数据集 \ 方法	ClassUD-A		
Alexnet	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	1.47	98.30	97.72
Alexnet_S	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	1.65	97.86	97.32
Googlenet	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	10.4	97.68	97.07
Googlenet_S	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	11.12	96.55	95.47
Mobilenetv2	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	18.72	99.37	98.82
Mobilenetv2_S	训练时间/(小时)	平均训练精度(%)	平均验证精度(%)
	18.83	99.55	99.00



(a) ClassUD-A 训练曲线

(b) ClassUD-A 验证曲线

Figure 9. Training and verification diagrams of the three models
图 9. 三个模型的训练及验证图

从表 6 及图 9 可知, 在数据集 ClassUD-A 上, 我们更改模型的激活函数为 Swish 后, 在训练时间上, Alexnet_S 比 Alexnet 多花了 0.18 h、Googlenet_S 比 Googlenet 多花了 0.72 h、Mobilenetv2_S 比 Mobilenetv2 多花了 0.11 h; 在平均训练精度上, Alexnet_S 比 Alexnet 减少了 0.44%, Googlenet_S 比 Googlenet 减少了 1.13%, Mobilenetv2_S 比 Mobilenetv2 提高了 0.18%; 在平均验证精度上, Alexnet_S 比 Alexnet 增加了 0.01%, Googlenet_S 比 Googlenet 减少了 1.47%, Mobilenetv2_S 比 Mobilenetv2 提高了 0.18%。综合上述分析, 在 Alexnet_S、Googlenet_S 及 Mobilenetv2_S 模型中, Mobilenetv2_S 是训练时间增时最少, 也是平均精度增加最多, 更是训练精度及验证精度最高的模型。所以对于学生头部状态的判别选用 Mobilenetv2_S 模型进行实验。

4.3.3. 头部状态分类判别实验

在进行分类判别之前, 对学生头部状态数据集 ClassTest 进行了人工观察, 观察结果见表 7, 进而用 Mobilenetv2_S 模型对学生头部状态数据集 ClassTest 进行分类判别, 结果见表 8。而人工观察与算法分类判别结果的平均值误差对比见表 9。规定抬头状态比例达到 60% 及以上的学生为认真学习状态, 低头状态比例达到 60% 以上的学生为非认真学习状态, 若两种状态比例相近且不属于上面两类, 判断学生为注意力不集中状态。

Table 7. Manual observation results

表 7. 人工观察结果

学生	观察结果		ClassTest	抬头(%)	低头(%)
	数据集				
	1		240	85.83	14.17
	2		240	80.83	19.17
	3		219	33.33	66.67
	4		240	62.92	37.08
	5		215	40.93	59.07
	6		240	28.75	71.25
	7		240	67.50	32.50
	8		240	72.92	27.08
	9		226	73.45	26.55
	10		240	94.58	5.42
	11		52	80.77	19.23
	12		240	68.33	31.67
	13		240	71.67	28.33
	14		234	84.19	15.81

Table 8. Test data experimental results

表 8. 测试数据实验结果

学生	分类精度		ClassTest	精度(%)	抬头(%)	低头(%)
	数据集					
	1		240	97.50	86.67	13.33
	2		240	97.92	83.75	16.25

Continued

3	219	87.67	26.48	73.52
4	240	96.67	65.42	34.58
5	215	95.81	45.12	54.89
6	240	89.58	32.08	67.92
7	240	95.83	70.83	29.17
8	240	96.25	69.58	30.42
9	226	88.94	60.17	39.82
10	240	88.75	84.17	15.83
11	52	92.31	73.08	26.92
12	240	95.42	71.67	28.33
13	240	100	71.67	28.33
14	234	99.14	83.33	16.67

Table 9. Comparison of the results of the student's head state error of the two different methods
表 9. 两种不同方法的学生头部状态结果误差对比

状态	方法平均值	人工观察	算法分类	误差
抬头(%)		67.57	66.00	1.57
低头(%)		32.43	33.99	1.56

从表 7 来看, 学生 1、2、4、7、8、9、10、11、12、13、14 在时常 40 分钟的课堂里为认真学习状态, 而学生 3、6 为非认真学习状态, 学生 5 的抬头比例及低头比例相近, 我们认为此学生在课堂里为注意力不集中状态。

从表 8 分类精度来看, 对于 14 个学生目标, 有 10 个学生头部的分类精度均在 92.31%至 100%之间, 4 个学生头部的分类精度均在 87.67%至 89.58%之间, 基本达到检测要求, 其中 4 个低于 90%分类精度的原因是学生头部存在部分遮挡, 导致检测结果不佳, 从检测抬头低头比例来看, 学生 1、2、4、7、8、9、10、11、12、13、14 在课堂中为认真学习状态, 而学生 3、6 为非认真学习状态, 学生 5 为注意力不集中状态。

从表 9 可知, 我们的算法检测结果平均值与人工观察结果平均值相差不大, 学生学习状态也大致相同。

综合以上分析, 提出的 Mobilenetv2_S 算法适用于自建数据集并取得了良好的分类效果。

基金项目

本文研究受贵州省教育厅创新群体重大项目(KY2018018)的支持。

参考文献

- [1] 左国才, 吴小平, 苏秀芝, 等. 基于 CNN 人脸识别模型的大学生课堂行为分析研究[J]. 智能计算机与应用, 2019, 9(6): 107-110.
- [2] 李芙蓉. 基于 MTCNN 的多特征融合学生疲劳检测[J]. 信息技术, 2020, 44(6): 108-113, 120.
- [3] 唐康, 先强, 李明勇. 基于人脸检测的大学课堂关注度研究[J]. 重庆师范大学学报(自然科学版), 2019, 36(5):

123.

- [4] Lecun, Y., Bottou, L., Bengio, Y., *et al.* (1998) Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**, 2278-2324. <https://doi.org/10.1109/5.726791>
- [5] León, M.D., Moreno-Báez, A., Magallanes-Quintanar, R., *et al.* (2011) Assessment in Subsets of MNIST Handwritten Digits and Their Effect in the Recognition Rate. *Journal of Pattern Recognition Research*, **2**, 244-252. <https://doi.org/10.13176/11.348>
- [6] Krizhevsky, A., Sutskever, I. and Hinton, G. (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*, Lake Tahoe, 3-6 December 2012, 1097-1105.
- [7] 蒋昂波, 王维维. ReLU 激活函数优化研究[J]. 传感器与微系统, 2018, 37(2): 50-52.
- [8] Szegedy, C., Liu, W., Jia, Y., *et al.* (2016) Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 7-12 June 2015, 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [9] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition.
- [10] He, K., Zhang, X., Ren, S., *et al.* (2016) Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision & Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [11] Bordes, A., Bottou, L. and Gallinari, P. (2009) SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent. *Journal of Machine Learning Research*, **10**, 1737-1754.
- [12] Xie, S., Girshick, R., Dollár, P., *et al.* (2017) Aggregated Residual Transformations for Deep Neural Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 5987-5995. <https://doi.org/10.1109/CVPR.2017.634>
- [13] Hu, J., Shen, L. and Sun, G. (2018) Squeeze-and-Excitation Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-22 June 2018, 7132-7141. <https://doi.org/10.1109/CVPR.2018.00745>
- [14] Howard, A.G., Zhu, M., Chen, B., *et al.* (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.
- [15] Sandler, M., Howard, A., Zhu, M., *et al.* (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-22 June 2018, 4510-4520. <https://doi.org/10.1109/CVPR.2018.00474>