**摘要 ：**

近年来，校园抑郁、校园暴力等青少年身心健康问题越来越影响青少年的发展，问题的原因主要是孩子早期情绪倾向没有被察觉和受到关注。面对一系列的

青少年校园成长问题，提前发现并进行心理疏导等成为保障学生身心健康的重要环节。在提前发现和治疗的过程中，情绪变化成为重要的参考标准。在发现的过程中要留意一些情绪焦躁、愤怒和情绪低落等早期信号，及时对孩子的情绪进行关注和疏导。为防止暴力行为和校园抑郁的发生，在学生身心健康遇到问题时及时给予关怀，才能从根本上解决问题。（这段话应该强调校园关怀的重要性，没有体现关怀的迫切性。要有数据和其他人的研究来支撑）心理学研究表明，在语言和表情表达的信息不一致时，表情传达的信息更准确，因此我们可以通过表情的识别和跟踪来判断孩子的身心健康状态。针对上述问题本文设计了一种校园情绪关怀系统，系统通过卷积神经网络的表情模型研究并集成摄像头实现了实时的监测系统。首先通过采集表情样本并制作训练集，然后使用TensorFlow构建训练网络并训练得到表情识别的模型。实时学生表情的识别是通过摄像头等图像捕捉工具获取视频流，然后使用OpenCV获得每一帧图片并使用模型对学生表情进行判断。本文采用的网络结构巧妙有效，样本多样具有普遍性，实时监测系统稳定可靠。通过对网络的大量训练,系统已经被证明是准确、有效的，这使得本文设计的校园情绪关怀系统具有良好的市场前景。

**关键词：**

校园情绪关怀，OpenCV，卷积神经网络，表情识别，TensorFlow

目录

[1.绪论 2](#_Toc526675447)

[1.1研究背景及意义 3](#_Toc526675448)

[1.2发展现状 4](#_Toc526675449)

[1.3 创新点 5](#_Toc526675450)

[2. 软件设计 6](#_Toc526675451)

[2.1数据集和图像处理设计 7](#_Toc526675452)

[2.11 数据集描述 8](#_Toc526675453)

[2.12摄像头的图像的处理 9](#_Toc526675454)

[2.2 卷积神经网络设计 7](#_Toc526675455)

[2.21网络结构说明 7](#_Toc526675456)

[2.22局部连接（添加一些数学公式） 7](#_Toc526675457)

[2.23卷积神经网络结构（添加一些数学公式） 7](#_Toc526675458)

[3. 实验测试和结果分析 10](#_Toc526675459)

[3.1 表情识别精度的测试 10](#_Toc526675460)

[3.2 模型应用和演示 11](#_Toc526675461)

[4.总结与展望 12](#_Toc526675462)

[4.1 总结 12](#_Toc526675463)

[4.2 展望 12](#_Toc526675464)

[参考文献 12](#_Toc526675465)

1.绪论

## 1.1研究背景及意义

校园情绪关怀系统是为中小学校园学生身心健康发展设计的产品,对校园情绪关怀系统研发的灵感来源于身边的现象。作为一名学生，生活在一个大集体中，学生的一些不良初期的情绪信号更多时候是不能被同学和老师察觉，倘若这种负面的情绪不被受到关注和关怀，很容易导致学生在学业上的力不从心，甚至造成一些生活上的障碍，这也往往是校园抑郁的导火线。校园暴力等事件大多也是孩子们早期情绪倾向没有被察觉和受到关注。从当地到全国、从国内到国外，校园暴力普遍存在，在个别国家和地区已经成为突出的社会问题。据调查，我国县城各校都存在校园暴力的现象，初高中问题尤为明显，群殴及强要钱物比较突出。就国外而言，校园暴力也令人头疼。法国巴黎大区议会曾公布的一份调查报告显示，巴黎大区中学的校园暴力问题严重，45.4%的高中生遭遇过暴力，77.39%的学生曾目睹校园暴力，20.96%的学生称自己有过暴力行为。另外，像英国、日本、韩国等国家也都存在严重的校园暴力事件。对于校园暴力的施暴者，在管教的同时也应给予更多的关注和关怀，不要让他们觉得被放弃了，看不到希望，进而自我放弃。在发现学生的一些负面情绪信号及时给予关怀，在才能从根本上解决问题。校园情绪关怀系统的设计思路可以解决上述人群的问题，经广泛调研,目前市场上还没有出现具有精准检测的校园情绪关怀系统。因此，对校园情绪关怀系统的研发应运而生。本文提出了一种基于卷积神经网络的易于实现，成本低廉的系统。实验表明，该系统可以准确的对学生的表情变化进行密切跟踪和分析，具有很高的准确率。

## 1.2发展现状

此前的研究（列举其他的研究）中有过许多对于增强校园心理健康的尝试，但都受到了一些条件的限制。1，进行周期性课堂教学是在校学生心理健康教育的主要渠道，是帮助学生形成良好心理品质的途径。它宣传普及大学生心理健康知识，解析心理现象，传授心理调适方法，介绍增进心理健康的方法和途径，让学生学会调动自身的心理机能进行自我疏导、自我教育。然而这需要学生自己疏解问题，但当遇到一些更达到问题时，并不是靠自己就能解决，因此这个时候，理论就显得纸上谈兵。2，很多学校都有校园心理辅导员来化解在校学生的心理情绪危机。但往往到了学生需要找校园情绪辅导员时，情形定是到了恶化，后期的地步，它并不能从根本的解决问题，校园心理辅导员的存在是被动的，他们只能在面对事态发生和恶劣时去解决和化解问题不能在初期时就察觉端倪因为老师的人力资源与学生的配比是缺少的，他们并不能关注到每个学生。

## 1.3 创新点

1）利用摄像头进行图像的采集，并应用于校园情绪关怀系统；

2）本方法设计的全新校园情绪关怀系统基于CNN神经网络, 成本低，精度高。

3）摄像头的图像经过OpenCV人脸定位以后，完成一个实时的人脸表情识别模块，通过识别视频帧中的人脸并判断表情。

4）当处理一副图像时，其输入往往是高维的。传统的神经网络将下一层神经元连接到上一层所有神经元。这种方式随着网络层数的增加，参数数量会爆炸式增加，在实际运用中，会无法训练网络。而我们利用的卷积神经网络具有局部连接和参数共享的特点，可以有效的减少网络的相关参数数量，优化网络的训练速度。

在经过大量的实验验证后，已经证明它是一个有效的系统，并且具有成本效益。该系统的优势在于利用fer2013数据集和tensorflow学习框架，保证了高精度。同时对用户来说，使用起来简单明了，所以这对使用该系统的用户来说，将是一个愉快的使用过程。

2. 软件设计

## 2.1数据集和图像处理设计

### 2.11 数据集描述

目前网上公开的人脸表情数据库，数据量最多的是2013年kaggle上的一次表情识别比赛提供的数据集fer2013。该数据集提供了35887张48\*48像素值的灰度图以及表情标签，表情标签有7种，分别是：0=angry, 1=disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral。

数据分布（训练集）：

　　Angry：3995

　　Disgust：436

　　Fear：4097

　　Happy：7215

　　Sad：4830

　 Surprise：3171

Neutral：4965

fer2013中的图片以“标签 图像向量”的形式保存在.csv文件中，即每一个列向量是一张图片，第一个数字是标签，剩下的数字是图片数据（将图片矩阵按列展开存储的），可以利用matlab的reshape函数将图片还原出来，图像尺寸是48x48大小。图一，对于数据集的处理的代码

|  |
| --- |
| datasets\_path = r'./datasets'  train\_csv = os.path.join(datasets\_path, 'train.csv')  val\_csv = os.path.join(datasets\_path, 'val.csv')  test\_csv = os.path.join(datasets\_path, 'test.csv')  train\_set = os.path.join(datasets\_path, 'train')  val\_set = os.path.join(datasets\_path, 'val')  test\_set = os.path.join(datasets\_path, 'test')  for save\_path, csv\_file in [(train\_set, train\_csv), (val\_set, val\_csv), (test\_set, test\_csv)]:  if not os.path.exists(save\_path):  os.makedirs(save\_path)  num = 1  with open(csv\_file) as f:  csvr = csv.reader(f)  header = next(csvr)  for i, (label, pixel) in enumerate(csvr):  pixel = np.asarray([float(p) for p in pixel.split()]).reshape(48, 48)  subfolder = os.path.join(save\_path, label)  if not os.path.exists(subfolder):  os.makedirs(subfolder)  im = Image.fromarray(pixel).convert('L')  image\_name = os.path.join(subfolder, '{:05d}.jpg'.format(i))  print(image\_name)  im.save(image\_name) |

### 2.12摄像头的图像的处理

摄像头返回的是彩色图像，因为颜色对于表情的变化并无明显影响，首先将其转化为灰度图。考虑到图像包含不相关背景，需要进行人脸定位，将图片中的人脸识别出来，以作为检测模型的输入。通过调用OpenCV提供的haarcascade frontalface default人脸检测器进行视频帧中人脸的检测。识别成功后，由于每张人脸的尺寸不一致，将其尺寸调整到48\*48像素，这样就和fer2013数据集的格式一样了。

|  |
| --- |
| video\_capture = cv2.VideoCapture(0)  font = cv2.FONT\_HERSHEY\_SIMPLEX  while True:  ret, frame = video\_capture.read()  img\_gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY,1)  faces = faceCascade.detectMultiScale(  img\_gray,  scaleFactor=1.1,  minNeighbors=5,  minSize=(30, 30),  )  emotions = []  for (x, y, w, h) in faces:  face\_image\_gray = img\_gray[y:y+h, x:x+w]  cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)  angry, fear, happy, sad, surprise, neutral = predict\_emotion(face\_image\_gray) emotionn=emotion\_labels[np.argmax(predict\_emotion(face\_image\_gray))]  cv2.putText(frame,emotionn,(x,y),font,2,(255,255,255),7) |

## 2.2 卷积神经网络设计

搭建网络

### 2.21网络结构说明

卷积神经网络的基本思想

  在BP神经网络中，每一层都是全连接的，参数数量随着网络宽度和深度增加会指数级增长。多层网络结合BP算法对输入数据虽然有强大的表示能力，但巨大的参数一方面限制了每层能够容纳的最大神经元数量，另一方面也限制了神经网络的深度。受到动物视觉皮层中感受野的启发，效仿这种结构的卷积神经网络具有局部连接和参数共享的特点，可以有效的减少网络的相关参数数量，优化网络的训练速度。

### 2.22局部连接（添加一些数学公式）

  Hubel和Wiesel在二十世纪五十年代和六十年代的研究表明，猫和猴子的视觉皮层中的神经元只响应特定的某些区域的刺激。将这种视觉刺激影响单个神经元反应的区域称为感受野（receptive field），相邻神经元细胞具有相同或相似的感受野[5]。正是由于发现了感受野等功能在猫的视觉神经中枢中的作用，催生了日本学者福岛邦彦提出带卷积和下采样层的多层卷积神经网络[6-8]。

  当我们在处理一副图像时，其输入往往是高维的。传统的神经网络将下一层神经元连接到上一层所有神经元。这种方式随着网络层数的增加，参数数量会爆炸式增加，在实际运用中，会无法训练网络。卷积神经网络中采取的做法是将每个神经元连接到上一层的部分神经元。这种连接的空间范围是一个超参数，称为神经元的感受野，感受野实际上是神经元映射到输入图像矩阵空间的大小。

  局部连接的实现方式是引入卷积层，通过卷积层对应局部的图像，每一层的神经元组合在一起对应图像的全局信息。如图3所示，在网络的第m层，每个神经元感受野大小为3，能连接到上一层的3个神经元。m+1层与m层类似。随着层数增加，神经元相对于输入层的感受野会越来越大。每个神经元不会响应感受野以外神经元的变化。受启发于动物的视觉神经元只响应局部信息，这样的结构确保了卷积神经网络只响应上一层局部神经元的变化，起到过滤作用的同时，减少了网络参数。而且随着层数的增加，这种过滤作用会越来越全局。

### 2.23卷积神经网络结构（添加一些数学公式）

  卷积神经网络是一种层次模型（Hierarchical Model），其输入是RGB图像，视频，音频等数据。卷积神经网络通过一系列卷积（Convolution）操作，非线性激活函数（Non-linear Activation Function），池化（Pooling）操作层层堆叠，逐层从原始数据获取高层语义信息[10]。

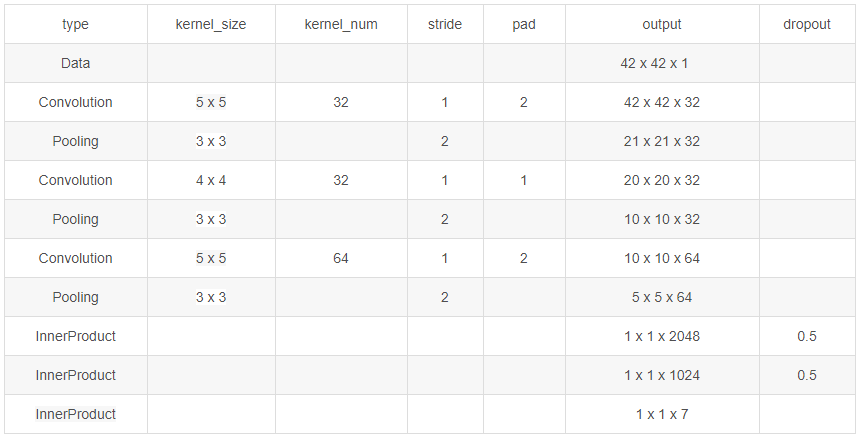
如图5所示，在结构上，卷积神经网络分类器有四种类型的网络层：卷积层、池化层、全连接层和分类器。各层次之间的有如下约束：

（1）多个卷积（C）和池化（S）层，将上一层的输出图像与本层权重W做卷积得到各个C层，然后经过下采样得到S层。

（2）全连接层：全连接层的输入是最后一个卷积池化层的输出，其输出是一个N维的列向量，维度对应类别的个数。

（3）分类器：p\_1，p\_2，p\_n的具体数值代表输入图像属于各类别的概率，分类器根据提取到的特征向量将检测目标划分到合适的类中。

本文所用的网络是在AlexNet上修改得到，减少了3个卷积层，现在的网络包含3个卷积层、3个池化层和3个全连接层。模型图如下



1）在Data层，原始图像数据是48\*48\*1的，其中1代表只有一个通道，因为原始图像是灰度图像。经过一个裁剪操作后，将48\*48的图像裁剪成42\*42的大小（裁剪操作在上述表格中没有体现，后面在介绍Data层代码时会介绍到），所以Data层的output为42\*42\*1。

2）在第1个Convolution层，pad为2，即图像的宽高各增加2个像素，特征图像尺寸变为46\*46，经过卷积之后，特征图像尺寸变为(46-5)/1+1=42，即42\*42，发现经过卷积之后图像尺寸还是42\*42没有改变，这是因为pad=2。由于卷积核的个数是32，因此在第1个Convolution层，output为42\*42\*32。

特征图像的计算公式：特征图像尺寸=[(原特征图像尺寸-卷积核尺寸)/步长]+1

3）在第1个Pooling层，使用上面公式计算得到特征图像尺寸=(42-3)/2+1=20.5，进1得到21，经过池化层的特征图像大小为21\*21。该层的output为21\*21\*32。

4）后面的第2个第3个Convolution层和Pooling层的计算方法跟上面相同。 网络结构层定义代码如下：

model = Sequential()

#第一个卷积层

model.add(Conv2D(64,(5,5),activation='relu',input\_shape=(48,48,1))

model.add(MaxPooling2D(pool\_size=(5,5), strides=(2, 2)))

#第二个卷积层

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(Conv2D(64, (3, 3), activation='relu'))

model.add(AveragePooling2D(pool\_size=(3,3), strides=(2, 2)))

#第三个卷积层

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(Conv2D(128, (3, 3), activation='relu'))

model.add(AveragePooling2D(pool\_size=(3,3), strides=(2, 2))

model.add(Flatten())

#全连接神经网络

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(1024, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num\_classes, activation='softmax'))

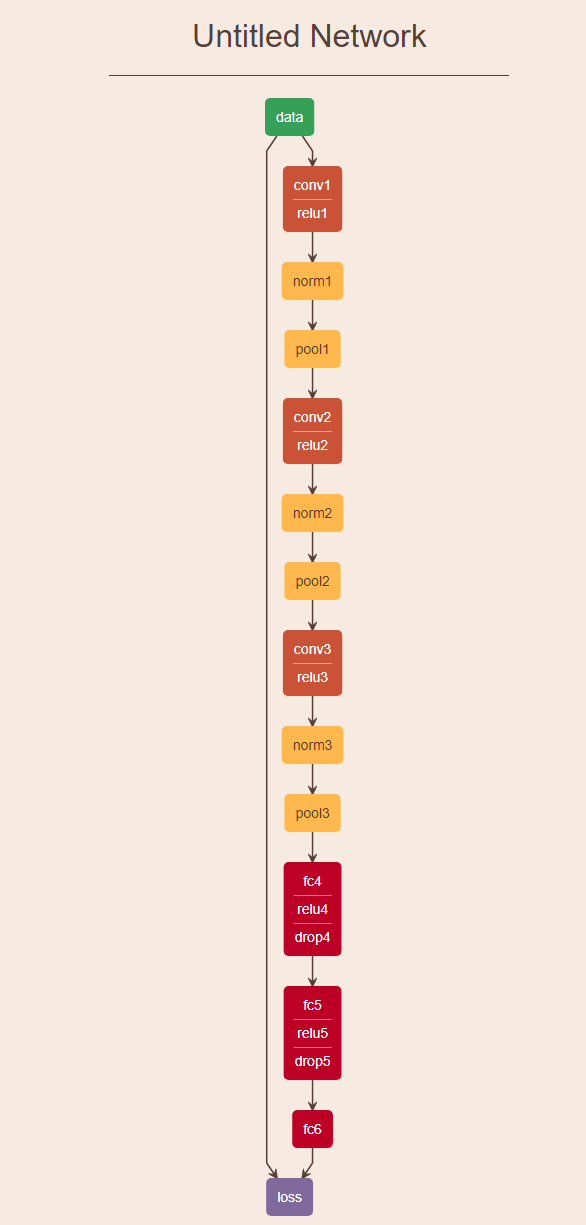


图2.8 网络结构图

3. 实验测试和结果分析

## 3.1 表情识别精度的测试

模型训练

首先，定义好训练过程中需要的占位符，分别是训练数据，标签以及不同阶段会变化的dropout保留率，因为每一批数据大小不定，shape的第一位为None，name参数是我们将来读取训练好的模型的入口：

|  |
| --- |
| **x\_data = tf.placeholder(tf.float32, shape = (None, CLIPED\_SIZE, CLIPED\_SIZE, NUM\_CHANNEL), name = 'INPUT')**  **y\_data = tf.placeholder(tf.int16, shape = (None, EMO\_NUM), name = 'LABEL')**  **keep\_prob = tf.placeholder(tf.float32, name = 'KEEP')** |

打开一个session，进行全局变量的初始化，通过一个计数器step判断经过了几批次的训练，按照batch大小每次从训练集中取出数据喂给模型，记录准确率，每100次打印出当前的训练集准确率，loss，以及将验证集放入当前模型计算验证集准确率。从而通过观察训练集和验证集的acc变化判断是否过拟合。

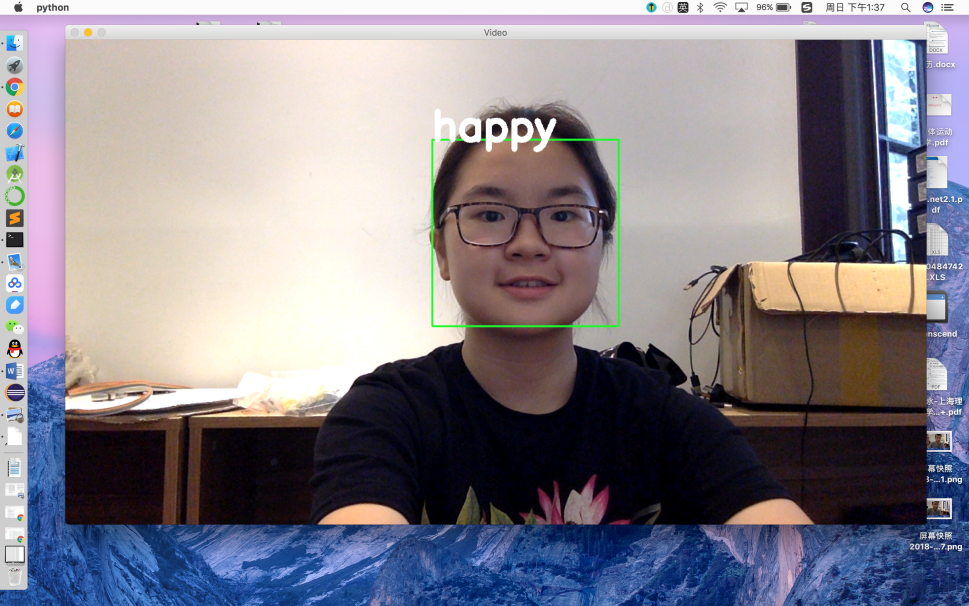
完成所有迭代次数以后，调用tensorflow的saver模块对模型进行保存，以便用于测试和实际应用。最后将上面的训练部分代码包装成一个函数train即可。

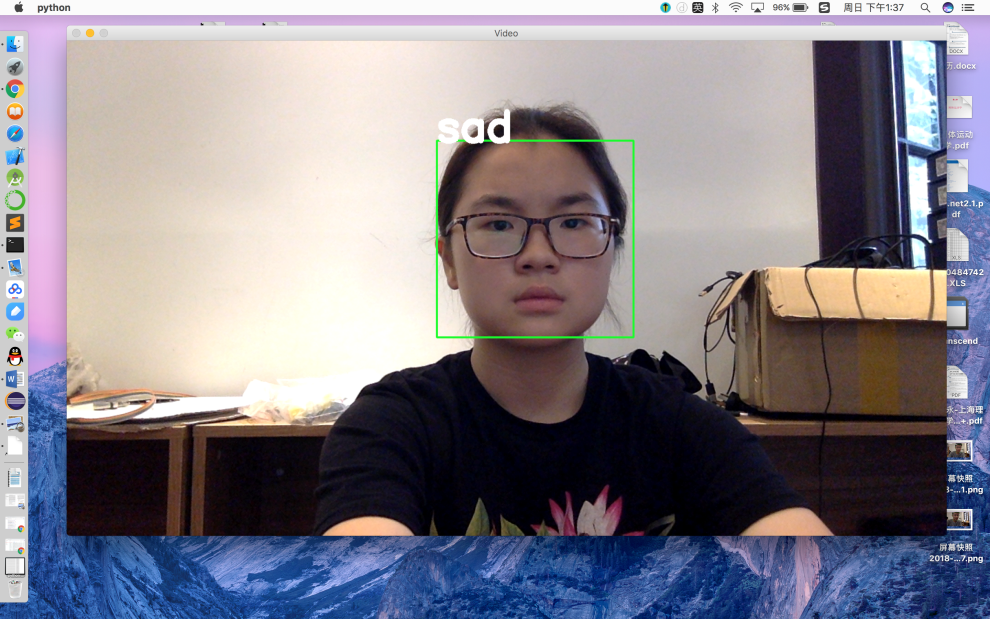
|  |
| --- |
| with tf.Session() as sess:  sess.run(tf.global\_variables\_initializer())  step = 0  for epoch in range(EPOCHS):  for batch\_i in range(TRAIN\_SIZE//BATCH\_SIZE):  step += 1  x\_feats = x\_train[batch\_i \* BATCH\_SIZE : (batch\_i + 1) \* BATCH\_SIZE]  y\_feats = y\_train[batch\_i \* BATCH\_SIZE : (batch\_i + 1) \* BATCH\_SIZE]  feed = {x\_data: x\_feats, y\_data: y\_feats, keep\_prob: 0.6}  sess.run(optimizer, feed\_dict=feed)  if step % 100 == 0:  (loss, acc) = sess.run([cost, accuracy], feed\_dict=feed)  feed\_v = {x\_data: x\_val, y\_data: y\_val, keep\_prob: 1.0}  acc\_v = sess.run(accuracy, feed\_dict=feed\_v)  print("In epoch %d, batch %d, loss: %.3f, accuracy: %.3f, validation accuracy: %.3f" % (epoch, batch\_i, loss, acc, acc\_v))  saver = tf.train.Saver()  saver\_path = saver.save(sess, SAVE\_PATH)  print('Finished!') |

## 3.2 模型应用和演示

将摄像头的图像经过opencv人脸定位以后，就可以完成一个实时的人脸表情识别模块，可以在视频界面绘制人脸定位框，并显示当前表情。

（加入一个测试的表格，表格中有一些测试数据，网上找一个类似的改一下。写一段对表格分析的话）

****

****

4.总结与展望

## 4.1 总结

本文介绍了校园情绪关怀系统的设计。为了达成这一目标，该系统通过一种简单的算法，成功地应用了卷积神经网络（CNN），并实现了对于表情动态识别的功能。通过大量实验，该系统已经被证明是准确及有效的。此外，我们的系统具有高精度，简单的算法和低成本的优点。对于使用者来说，是一款高效、便利的校园管理工具。这些优势将使这款产品在市场上处于领先地位，这将为校园老师提供了很大的便利。

## 4.2 展望

目前，此眼校园情绪关怀系统的发展现状还有很大的提升空间，做一些改进将会增加其实用性。

1）配套app开发，老师可以直接通过手机了解；

2）opencv自带的人脸检测器简陋，导致在光照条件不是很好，可以再做一个好用的人脸检测器

# 参考文献

1. 《基于深度学习的人脸表情识别》吕亚丹（天津大学硕士学位论文）
2. Jeon, Jinwoo, et al. “A Real-time Facial Expression Recognizer using Deep Neural Network.” International Conference on Ubiquitous Information Management and Communication ACM, 2016:94.
3. 关于高职院校大学生心理健康教育 李志宏（昆明冶金高等专科学校 心理健康教育中心，云南 昆明 ６５００３３）
4. 《Deep Facial Expression Recognition: A Survey》
5. （参考文献数量不够至少一页）