# 卷积神经网络识别手势动作（AlexNet结构）

# 文件解释：

1.data文件里面存放测试数据集，训练数据集，testImage实时保存的图像（用于在线检测）。

2.model文件里面存放训练好的CNN网络的模型参数。

3.GetNewImage.py是利用OpenCV获取实时图片，保存到/data/testImage中，调用TestGesture.py判断其手势动作。

4.GetTestImage.py是利用OpenCV获取图片并标记，用来制作测试集。

5.GetTrainImage.py是利用OpenCV获取图片并标记，用来制作训练集。

6.TestGesture.py是将实时获取的图片送入已训练好的模型中判断其手势动作。

7.TestInTest.py是将测试集送入已训练好的CNN中判断该网络模型的准确率。

8.Train.py是训练CNN模型函数，并将训练好的模型参数保存在本地。

9.Train\_inputdata.py是用来读取数据集的图像和标签，并打包成batch形式。

10.Train\_model.py是模型结构，这里用的是AlexNet结构。

## 实验过程：

### 1.创建数据集

相关文件：GetTestImage.py，GetTrainImage.py。

一共识别5种手势动作 ：1. 剪刀动作，2.石头动作，3.布动作，4.OK动作，5.good动作，通过OpenCV获取图片并标记。

训练集： 1~5号动作各有1300张照片，测试集： 1~5号动作各有200张照片。

### 2.建立神经网络

相关文件：Train\_model.py

使用AlexNet结构,使用ReLU作为激活函数。

输入层：图像大小为 227×227×3，其中 3 表示输入图像的 channel 数（R，G，B）为 3。

卷积层：filter 大小 11×11，filter 个数 96，卷积步长 s=4，padding 使用 vaild convolution,即pad=0。

池化层：max pooling，filter 大小 3×3，步长 s=2。

卷积层：filter 大小 5×5，filter 个数 256，步长 s=1s=1，padding 使用 same convolution，即使得卷积层输出图像和输入图像在宽和高上保持不变。

池化层：max pooling，filter 大小 3×3，步长 s=2。

卷积层：filter 大小 3×3，filter 个数 384，步长 s=1，padding 使用 same convolution。

卷积层：filter 大小 3×3，filter 个数 384，步长 s=1，padding 使用 same convolution。

卷积层：filter 大小 3×3，filter 个数 256，步长 s=1，padding 使用 same convolution。

池化层：max pooling，filter 大小 3×3，步长 s=2；池化操作结束后，将大小为 6×6×256 的输出矩阵 flatten 成一个 9216 维的向量。

全连接层：neuron 数量为 1024。

全连接层：neuron 数量为 1024。

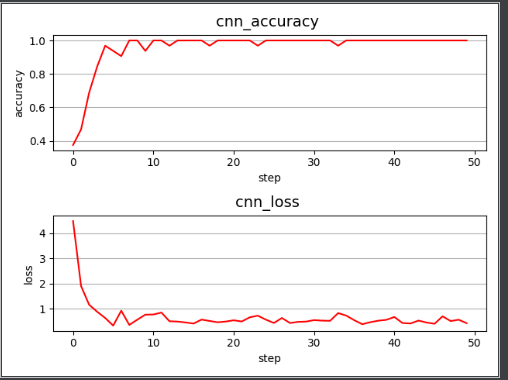
全连接层，输出层：softmax 激活函数，neuron 数量为 5，代表 5 个类别。

### 3.训练并调参

相关文件：Train.py,Train\_inputdate.py,TestGesture.py.

通过Train\_inputdata.py来读取数据集的图像和标签，并打包成batch形式，因为数据集较小，batch\_size大小设定为32。

使用AlexNet网络进行训练，迭代1000次，学习率0.0001,在AlexNet网络的全连接层进行dropout正则化，keep-prob设为0.5,即每个节点的保留和消除概率都为0.5。讲训练好的模型保存在model文件下。因为数据集较小，存在过拟合现象。



通过TestInTest.py将测试集送入已训练好的CNN中判断该网络模型的准确率。

### 4.保存并调用

相关文件：TestGesture.py,GetNewImage.py。

GetNewImage.py调用OpenCV捕获手势照片，保存在/data/testImage/文件中。TestGesture.py文件调用训练好的模型判断手势动作。