

Weekly Report

June 27, 2017

1 2017.6.16

1.1 用LeNet训练基元草图

重新训练网络

- 1.六边形画成五边形1个，基本不像两个。
- 2.图片缩小时，采用区域差值方式。区域插值是opencv中提供的5种resize差值方式实验效果最好的差值方式。

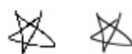


Figure 1: 左图与其他插值方式，右图为区域差值

重新训练网络，得到的识别准确率为95%。UI Demo

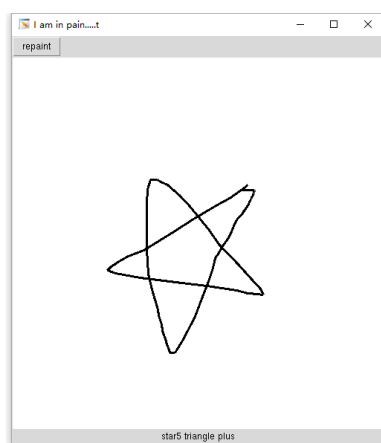


Figure 2: UI

2 2017.6.18

由于基元存在旋转的情况，不适合把基元做average。
故找出当前画的基元的fc1层feature vector 和训练集中基元的fc1 feature vector最相近的前3个作为提示给出。下图是一个demo

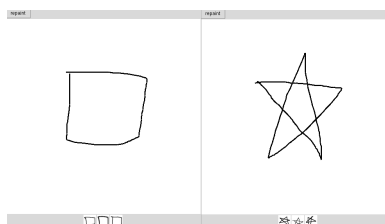


Figure 3: 主区域是当前画的基元，底下三个从训练集找出最相近的三个基元

3 2017.6.23

在网络上加dropout层，识别的准确率稳定在95.4%。
将圆形，五边形，三角形，梯形，矩形，五角星，六边形，平行四边形旋转90，180，270度，其余基元样本直接复制成原来的四倍。得到总共20000个样本，训练迭代30000次，识别准确率为94.9%。

UI demo

画一个基元，从训练样本找出和该基元在fc2 特征最像的10个样本，并且按照特征距离远近，获得10个样本的权值，再将10个图片相加，获得shadow图。

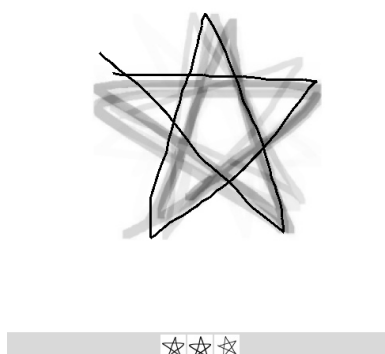


Figure 4: 五角星



Figure 5: 圓

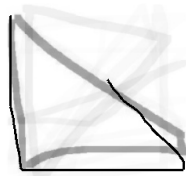


Figure 6: 三角形



Figure 7: 除

To Do:

从3340训练样本中找到距离当前画的基元10个最近的样本比较耗时，约2到3秒时间，考虑构建树结构或其他数据机构去加快搜索。

4 2017.6.27

shadow图片大部分来自识别分数最高的前3类，故只从识别分数最高的前3类选取10张最近邻。feature vector虽然20维，但是能量分布主要集中在识别分数最高的3个维度。找到10个最近邻时间为8ms-12ms,优化选取10个近邻的过程(原来是遍历整个训练集10遍，找10个最近邻，而现在实现是维护10个最近邻有序序列，只对训练集遍历一遍)。

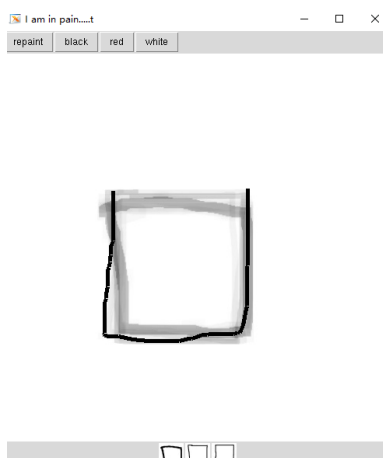


Figure 8: 矩形

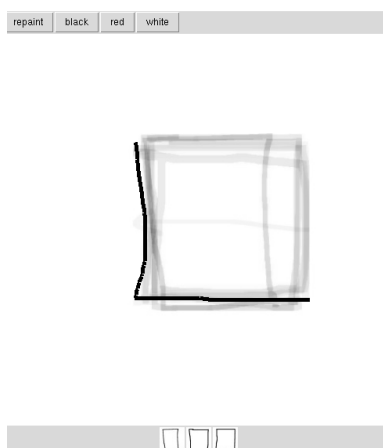


Figure 9: 矩形

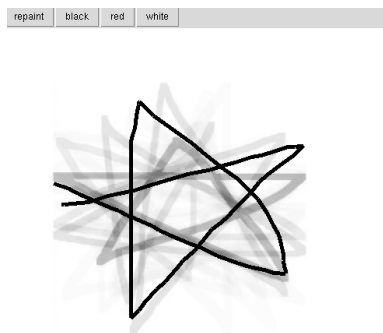


Figure 10: 五角星



Figure 11: 圆

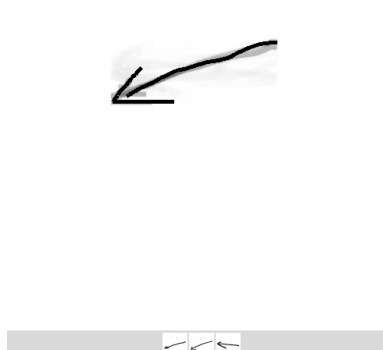


Figure 12: 左箭头

References