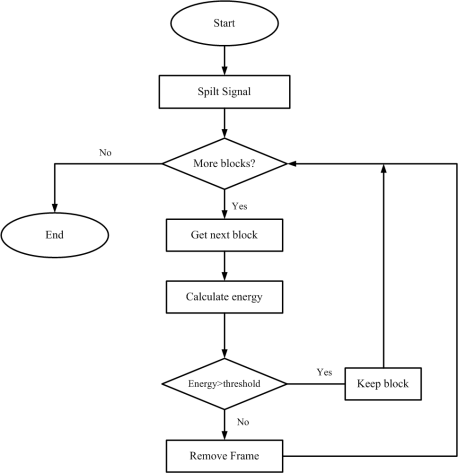
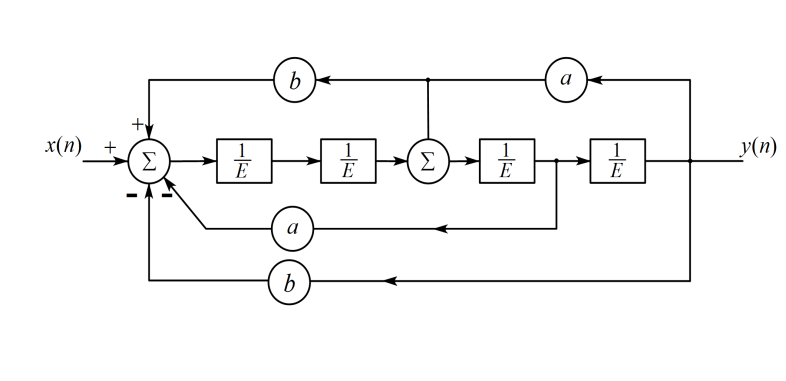
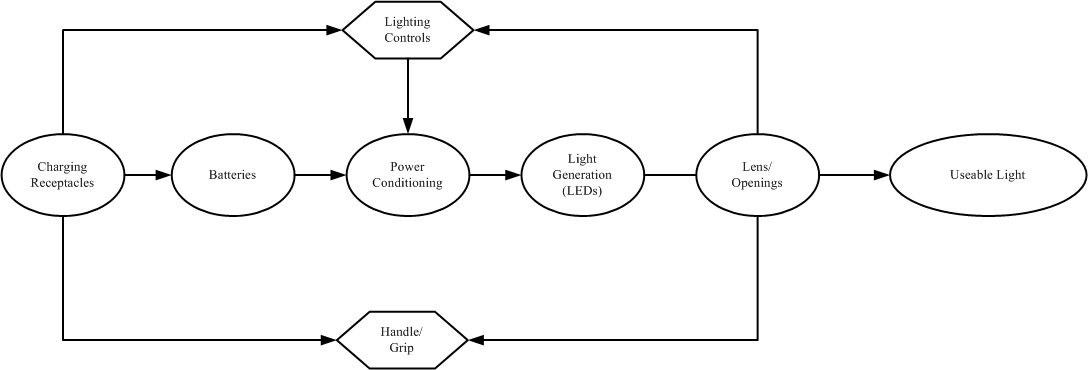
中期后的工作主要是在目前框架的基础上结合之前学习的一些理论进行了图元识别的仿真实验以及设计了箭头定位的方案。下面先进行介绍：

对于框图来说，不管是系统框图、流程框图，若只以图论的角度看，都为节点与边的组合。

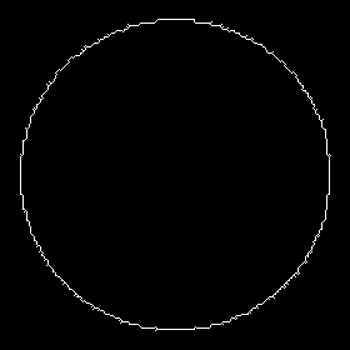
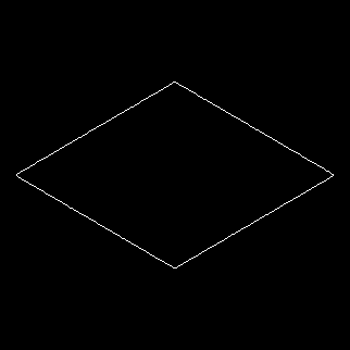


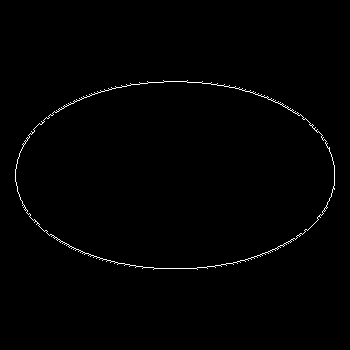
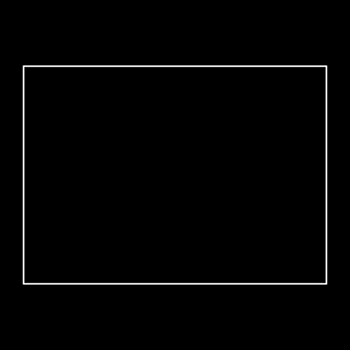


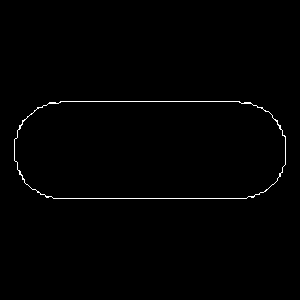
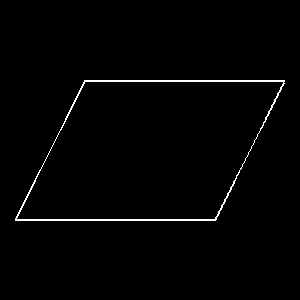
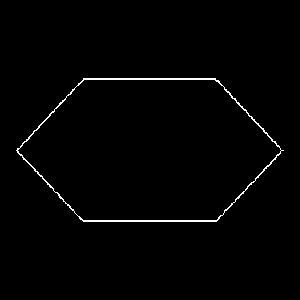
若将连接线看为边，图元看为节点，则即可构建图结构，但是由于连接线存在分支点，以及类似系统框图上存在流入点和流出点，因此需将分支点和流入流出点也作为节点，这样一副框图就完全形成图结构，每个连接线确定一对节点的连接关系。节点类型为基本图元、分支点、框图外部流入或流出点。因此需对连接线作进一步分割，使每个连接线分为二端点连接线。从图论的角度看，框图为单连通的有向图，且为稀疏图，即边数远少于完全图。

识别主要步骤如下：

1. 图像预处理：上采样、灰度化、二值化，中值滤波去噪。
2. 由于图元具有闭合性，以及节点与变之间形成闭合区域，首先进行边缘检测，提取内边缘，这里采用基于凹凸性方法筛选出图元。目前主要考虑以下7种图元。

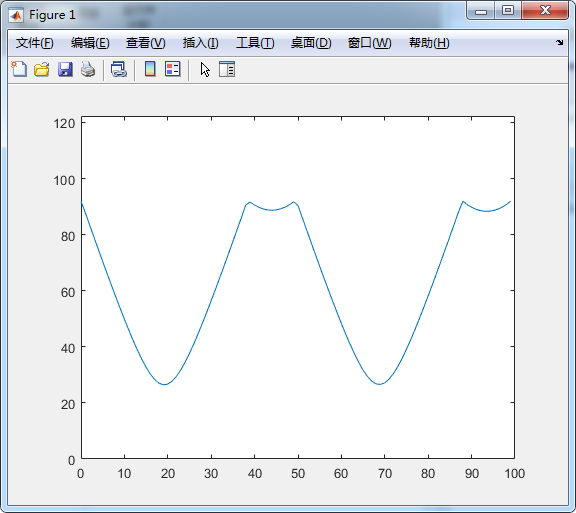
 

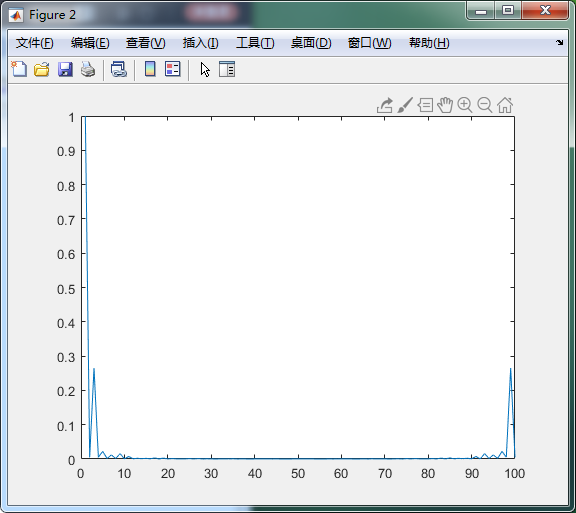
主要图元

1. 关于形状描述符，之前学习的理论主要有链码、傅里叶描述子、不变矩等，目前我主要试验了傅里叶描述子、HOG(方向梯度)特征、zernike不变矩这三种形状描述符。下面主要是仿真结果。
2. 傅里叶描述子的优点是具有旋转与尺寸不变性，我尝试了用质心轮廓距离函数，以顺时针方向沿轮廓扫描，得到质心距离函数如下，矩形为例：



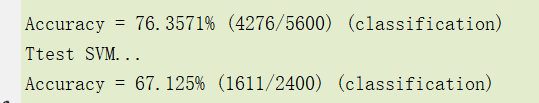
矩形的质心距离函数

对其做100点采样，然后计算其100点的傅里叶变换FFT，对傅里叶级数归一化，取前50个频谱分量做特征向量。



100点FFT

由于没有数据集，我选用了7种初始图形然后进行伸缩尺寸变换每种生成1000张图，每张图像尺寸300\*300，7种图形为圆、矩形、菱形、平行四边形、椭圆、圆角矩形、六边形。划分数据集取其特征向量，采用线性核SVM进行多分类，划分训练集与测试集为7:3，得到结果如下：

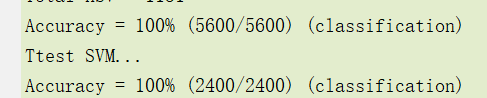


,在采用另外的实际提取图元进行测试，结果如下



可以看出识别率非常低，调试多次后也无法提升识别率，若以该识别率无法进行正常识别，所以暂不考虑采用该特征。

1. HOG特征是通过计算和统计局部区域的梯度方向直方图来构成特征，对于测试的没张300\*300的图，将其按8\*8单位进行划分，统计每个单位的梯度直方图，即形成每个单位的特征描述符，将所有特征描述符串联起来即可得到特征向量。HOG的特征的缺点主要是没有旋转不变性与尺寸不变性，所以图元集基本都为水平方向的图，大小尺寸固定。对300\*300按8\*8像素划分，可提取特征向量维数为46656，采用HOG特征+SVM线性核分类结果如下，训练集测试集7:3划分：



测试结果直接达到100%，识别率奇高，但是考虑所制作的图源数据集均为理想图形，所以选取几张实际图形进行测试：



识别率相比傅里叶描述子已经有了很大提升。

（3）zernike不变矩也是一种描述形状的描述符。其是图像函数f(x,y)在正交多项式上的投影，其中正交多项式在单位圆内正交。其优点是具有旋转不变性与尺寸不变性。1，输入预处理后的图像。2，找出目标最小外接矩形。3，计算各阶矩。4，归一化Zernike矩。关于Zernike矩的具体理论先不做过多赘述，提取特征向量维数为30，其采用SVM线性核分类后的识别率如下：



（4）尝试使用HOG+zernike不变矩组合特征向量的方式进行识别，SVM线性核：



该识别率与单独使用HOG特征持平。但由于目前没有足够的图形样本，该识别率目前只是先做参考。若样本足够的话应该能取得较好的识别率。

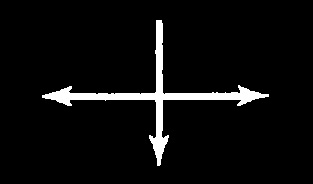
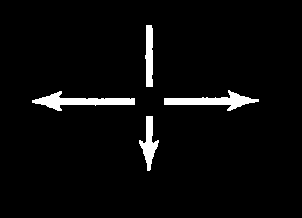
按目前得到的识别率，HOG应为最高，其次是zernike不变矩，两者组合特征识别率提升不大。但是300\*300的图像尺寸相比实际提取框图图元还是偏大，接下来考虑用100\*100,150\*150的尺寸进行分类实验，目前主要考虑以HOG特征为主进行图元识别。

1. 关于图元形状的识别之后，采用掩膜法在原图中去掉图元部分，只保留连接线。连接线是图元关系的识别标志。由于将分支点划分为节点类型，需对每一个连接线在分支点处进行分割。我考虑的只为实心连接线与实心箭头。具体实现为：
2. 连接线的线宽估计：线宽主要是在后面的分割检测等步骤中作为阈值标准，由于不同框图的线宽不同，所以先对连接线的线宽进行估计。对于连接线层，首先进行连通域标记。提取任意一条连接线，然后进行细化，生成连接线的骨架线，以下面的示意图为例：



对于骨架线上的每一点，计算其到连接线边缘轮廓上的每一点距离，取其最小值，记为d，求得所有骨架线到边缘距离的最小值均值d’，得到线宽的估计值linewidth=2\*d’。

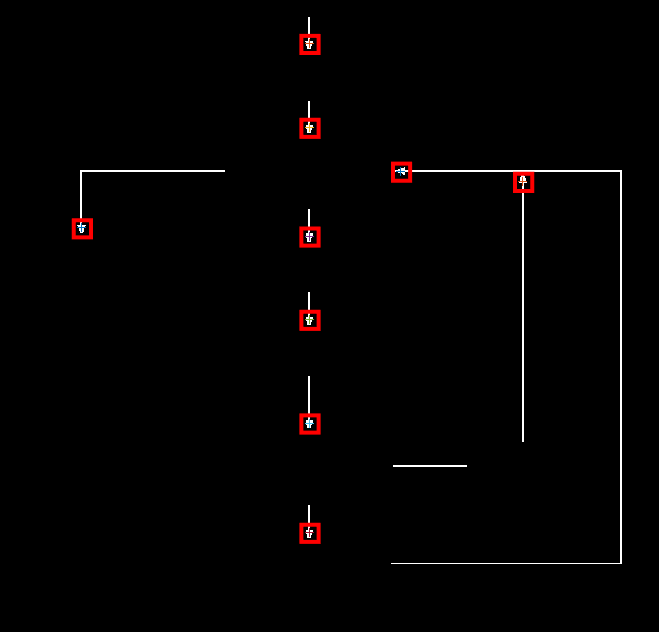
1. 对每一连接线的进行检测，检测其分支点，采用模板扫描，得到分支点后，在分支点处设置正方形的掩膜，正方形掩膜的宽设为3\*linewidth，将连接线分割如下图所示。

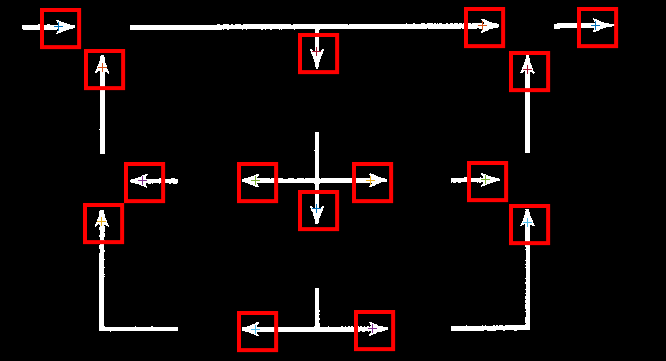
  

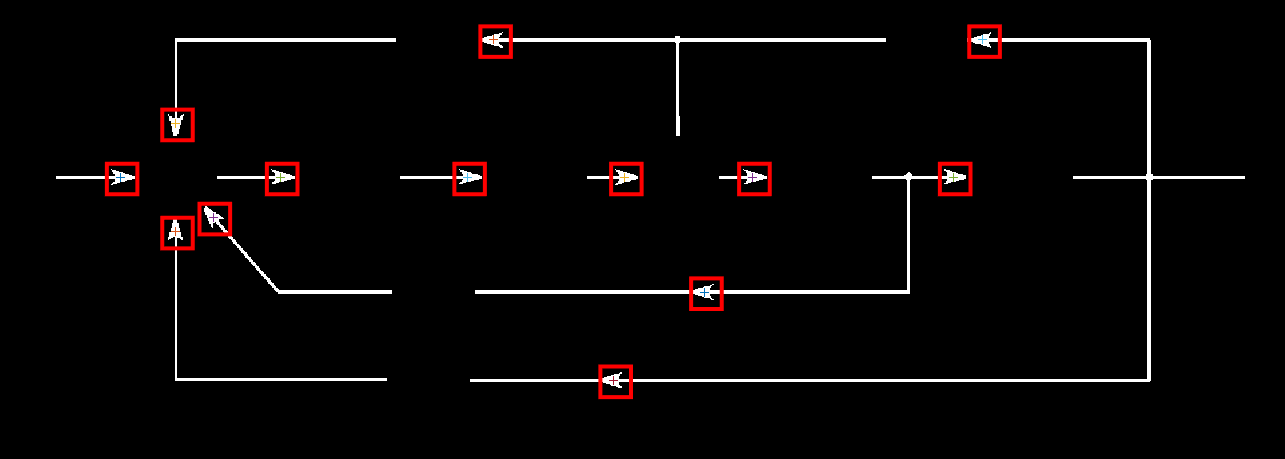
5、连接线在分支点处进行分割后，这样就保证了每一连接线均为2端点的连接线，接下来便可进行箭头位置的检测。同时，将去掉的分支点作为独立于图元与连接线的数据结构。我考虑的只为实心线段上的实心箭头。由于实心箭头在面积上相对线段本身有明显的面积差，所以考虑用面积作为阈值进行检测。具体检测方法为



对于上一步之后每条连接线都分割为只有2端点，首先进行连通域标记后，对每个骨架线端点进行检测，找到任意端点后，以该点为中心，设置正方形区域，经过多次试验后，设正方形宽设为1.4\*linewidth为合适阈值，采用链码跟踪的思想，从一端点搜索到另一端点，统计每一点内像素个数，当扫描到箭头中心位置时，即达到像素面积最大值。之所以采用这种方法，是因为可能存在箭头位于线中间位置的情况。找到箭头后，在箭头中心位置设置4倍正方形ROI，如图所示：







检索到箭头位置后，下一步打算提取足够的箭头作为箭头数据集，按方向进行分类，目前考虑8个方向，即按8邻域划分的8方向。

本文在之前等人的基础上，基于图论中最小生成树的思想构建连接关系生成算法：

1. 对所有节点进行标记，将节点依次存入集合。
2. 由于已经将连接线从分支点分割，所有连接线目前均为二端点，图为连通图，基于连通域个数判断。
3. 首先取Set中第一个结点，依次遍历栈内所有剩余结点。每次只取一个节点，保证图中只有2个节点，之后对于所有连接线，每次添加一个到图中，检测连通域个数，若连通域个数为1，则证明两节点是连接关系，若连通域个数>1则不为连接关系，由于一条连接线只连接两个节点，将此连接线删去。
4. 对于具有连接关系的两节点，保存其信息，之后继续直到遍历完栈内所有节点为止。此时栈顶结点的所有连接节点确认完毕。
5. 将栈顶元素弹出，对之后的节点重复上述步骤，由于一条连接线代表一条连接关系，，当连接线集合Set为空时，此时所有图元的连接关系已确定。

分割后，对同一连接线分割后的不同连接线，需进行融合，依据分支点和图元关系，与同一分支点具有连接关系的图元则。对于指向关系，可有箭头特征与位置关系进行判断，对于连接线无箭头的，只能依照上下文关系判断。

目前的主要问题是，现在所用框图识别思路主要还是基于2012年CLEF-IP的流程图识别的task，Rusiñol等人总结的方法上，我主要在图元识别上采用了不同的描述符，并对连接线按分支点进行分割，设计了箭头检测方案，但在整体结构上还存在诸多雷同，创新性不足。目前还没有太好的想法，希望老师给一些意见。