**项目简介**

1. **高压输电线路中的绝缘子定位**

**1、背景：**

1）目的：实现图片中绝缘子检测。

2）**传统方法**，先经过图像分割，然后根据绝缘子边缘、形状等特征进行检测（如图1）。由于绝缘子背景复杂（如图2），图像分割效果不佳，绝缘子难检测。

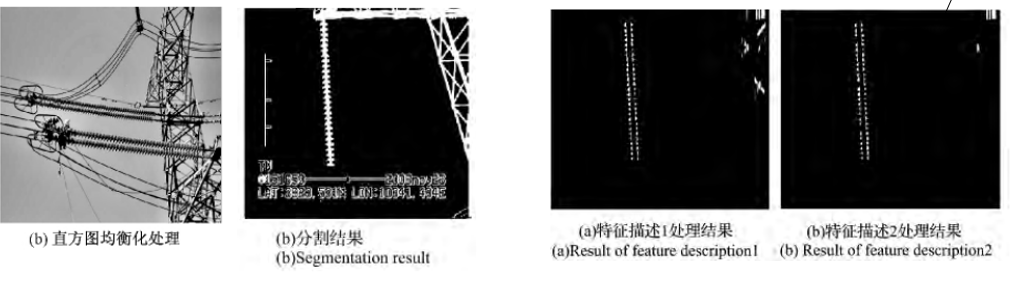


图 1

****

图 2

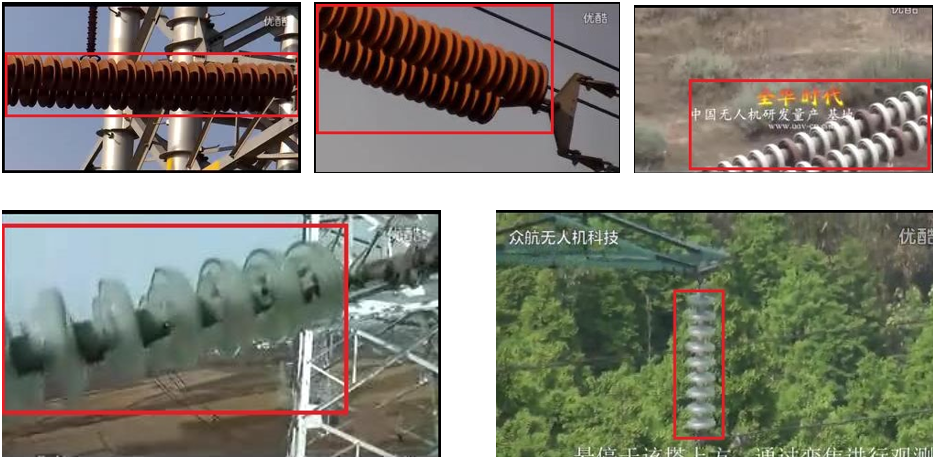
3）拍摄角度不固定导致绝缘子目标长宽比不固定，而**现有的机器学习目标检测方法**要求样本或者检测窗口大小固定。强行伸缩监测子窗口适应样本大小，易导致图像变形。

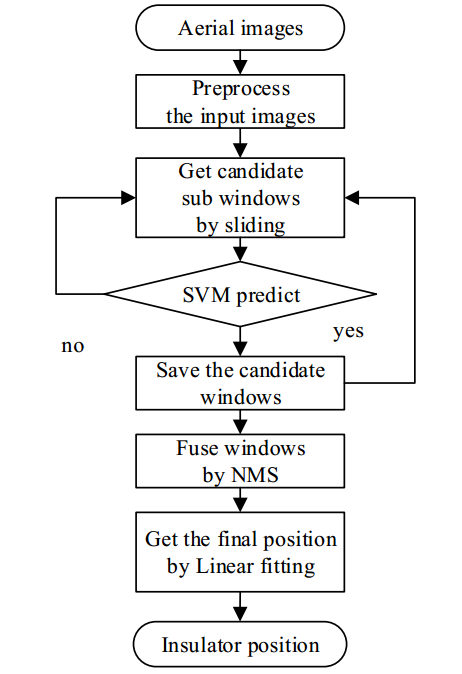
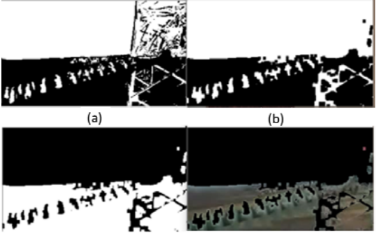
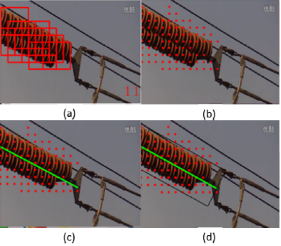
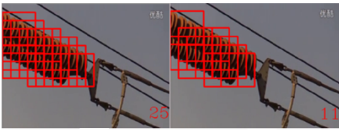
图 3

**2、方法：**

1. **基于特征融合的绝缘子检测**

**设计技术**：采用OpenCV开源库和C++编程语言，对图像预处理、提取HOG/LBP融合特征和SVM算法分类训练，完成绝缘子检测，分析检测精度和速度。

**检测流程：**



对提取的子窗口图像，提取HOG、LBP特征，分别PCA降维，串行形成融合特征；

采用SVM训练好的模型进行分类，保存预测为绝缘子的子窗口形成候选子窗口。

图像金字塔，固定滑动子窗口大小，在整幅图像中滑动提取子窗口。

采用阈值分割得到图像掩码，与原图相乘得到待检测图像。

好处：减少需要预测的子窗口数，提高检测速度；可去除天空、较细的铁塔等背景元素，减少负样本，提高检测精度。

NMS-非极大值抑制(左，红框为SVM预测为绝缘子的子窗口)：减少最终需要合并的候选窗口数。

窗口合并，采用线性拟合方法，得到最终的绝缘子位置（右，黑色框为绝缘子位置）

检测过程主要优化手段：

1）优化1：并行预测—电脑为两核，大约提快1.5倍

2）优化2：图像预处理--对天空等元素有效，可以加快检测速度

3）优化3：交叉验证--优化svm参数，有改善

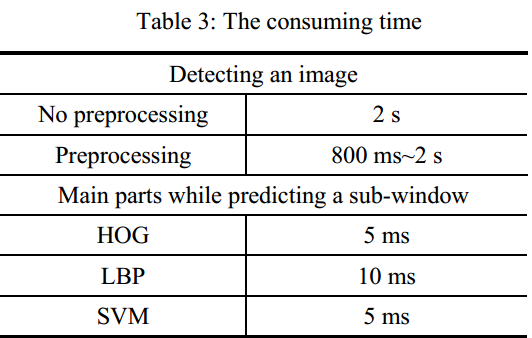
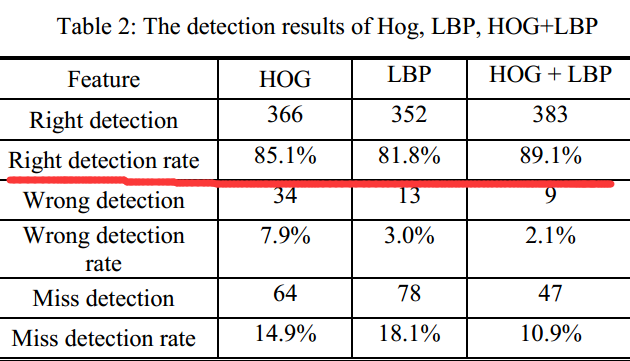
4）HOG、LBP特征融合，但LBP速度慢

4）OpenCL并行加速LBP特征提取过程：速度没有明显改善，舍弃。分析原因：GPU中需要计算的数据量少，时间主要用在CPU与GPU传参上（这个还说么）

**精度、速度分析：**

可以实现复杂背景下多个角度的绝缘子检测，检测精度还不错。但是每个子窗口都需要特征提取和分类，速度差。

测试:500 images, 400 contain insulators, 100没有绝缘子。总共430 insulators。



**需要复习的知识：**

**图像处理（滤波、分割常用方法）、HOG/LBP、SVM、OpenCL看下。**

1. **Faster-RCNN目标检测实验**

**设计技术：**

研究Caffe框架，构建AlexNet网络进行特征提取，将CNN网络提取的特征替换方法1）中的HOG/LBP特征，速度太慢；

研究Faster-RCNN、SSD目标检测方法，采用Faster-RCNN进行试验并分析。

**分析：labelImage图像标注。**精度91%左右，但是检测的边框包含过多背景，不适合后续的故障检测。如下图，检测和没检测没什么区别，依然包含大量背景。

****

**需要复习的知识：**

**Caffe、CNN网络、Faster-RCNN、SSD**

1. **基于FCN网络的绝缘子目标分割**

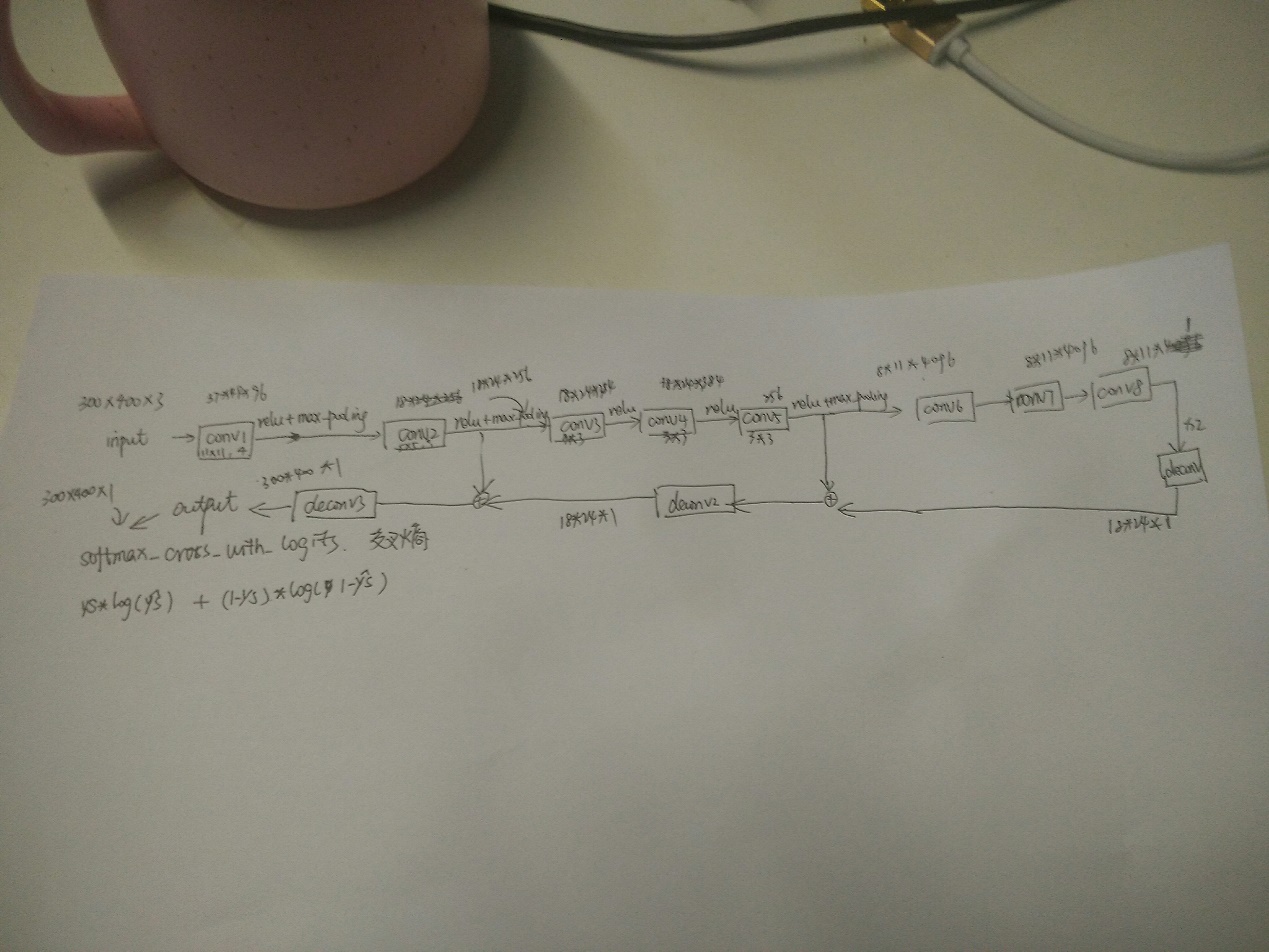
**方法：**

1. **上采样采用resize\_image:可行**
2. **上采样采用conv2d\_transpos:可行**

**要注意conv2d\_transpos的stride合kernal\_size。设计输出尺寸和视野**

**设计技术：**采用Tensorflow框架和Python编程语言，构建FCN网络对绝缘子目标进行分割。

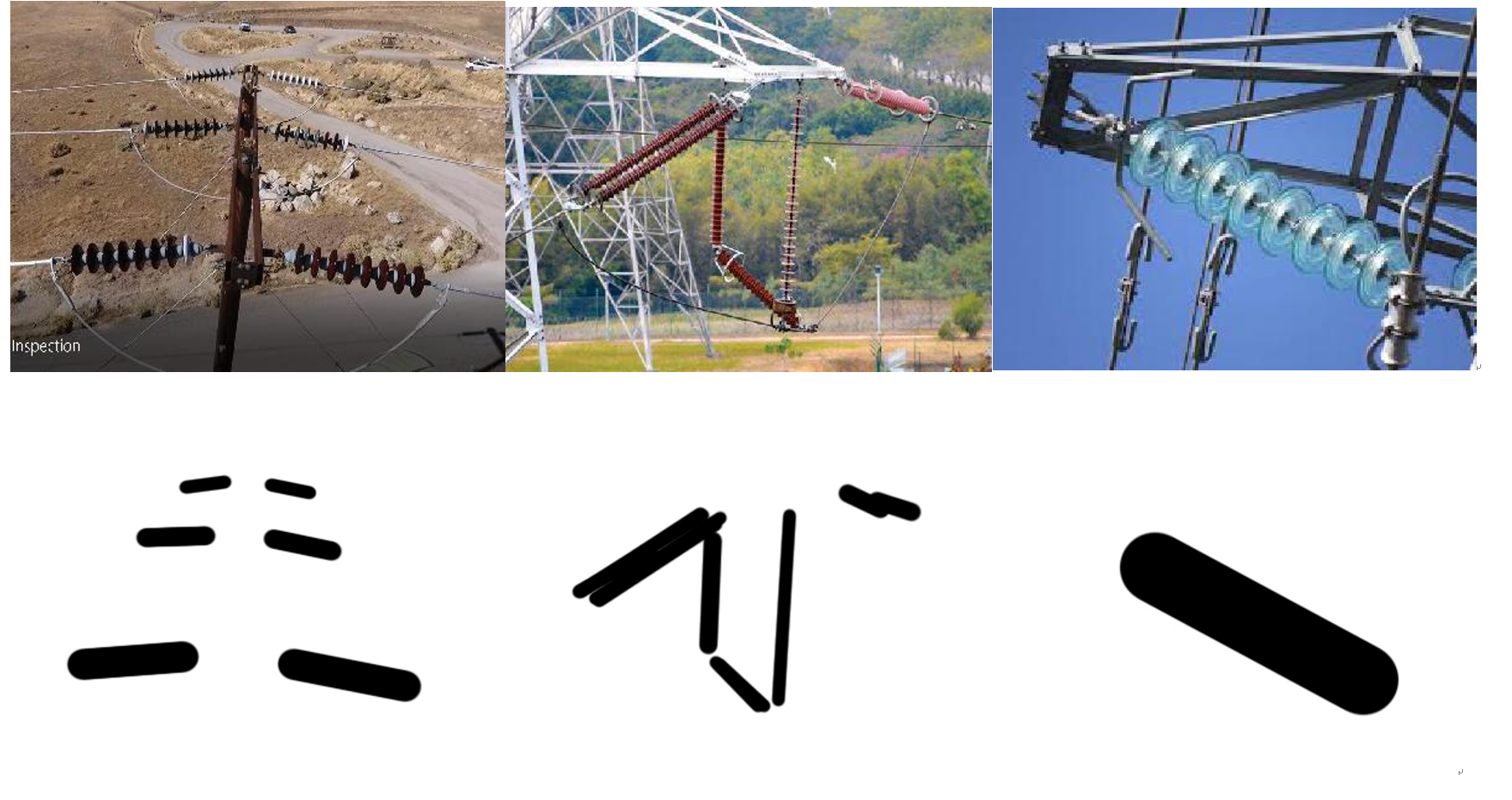
**网络结构：**

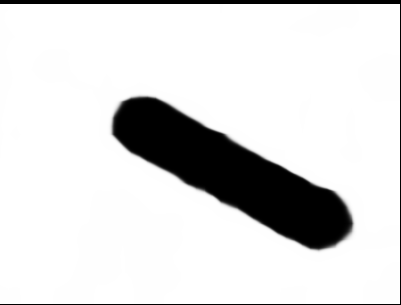
****

**结果：**

样本：840\*\*8=6720（翻转，旋转）

5720训练，1000测试，





按像素点：82%左右（正确的像素点/所有像素点）

按覆盖率，预测的图像覆盖标注的图像85%以上算检测正确：92%左右（分割正确的图像/所有图像）

**需要复习的知识：**

**Tensorflow、python**

**2、基于CNN回归的消失点检测**

**遇到的困难：**

输入不同，输出相同，神经元死掉。

解决方法：

1. 所有relu->tanh，从头开始训练，可行（人脸特征点检测，采用tanh）

Loss很快到0.1，而且经过几万次迭代后就从0.1继续收敛，见论文。

1. 所有激活使用relu, 加载AlexNet预训练网络，不可以？

不可以。loss:0.04左右，但输出相同，relu输出几乎为0。

1. Xavier初始化方式，全为relu, bach\_normalize，不可以？

不可以。

1. Xavier, leaky\_relu, bach\_noemalization？

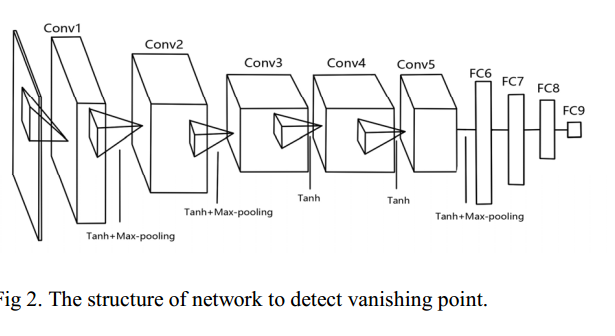
不可。

3）和4）的 loss:0.3左右不下降；3）relu输出几乎为0，输出相同；4）输出不同，但是loss就是不下降。2）-4）训练一天一夜左右。

**设计技术：**采用MATLAB进行图像预处理、数据扩充和标定；采用Tensorflow框架和python编程语言，构建CNN网络，实现消失点检测。

**思路来源：**人脸特征点检测；关于深度学习的消失点检测论文（将300\*300图像分别分成10\*10,20\*20,30\*30网格，对网格进行分类，包含消失点则为正，否则为负样本；误差分别为30、15、10个像素），考虑消失点为图像的全局信息，因此，采用全图进行消失点的回归检测具有可行性。

**网络结构：**



Loss函数：(y-ys)2+(x-xs)2

遇到的问题：不同图片训练得到结果相同，relu激活函数后输出为0,relu改为tanh，虽然收敛慢，但最终收敛。（改为leaky\_relu可能更好，但当时不知道有这玩意； 在强化学习中还遇到过输入不同结果相同的问题，查到的解决方案可能更参数初始化有关，不过还没有试验）要说么

**效果分析：**

训练样本：640原图，裁剪旋转后5548张左右。

测试样本：1000张，检测的消失点与标注的消失点距离不超过15则表示检测正确。准确率为89%左右（到时候应该会问实验结果，准备一个数比较好）。



检测效果不错，在直线等特征不明显的情况下也能很好的检测消失点。

**3、基于DQN的多机器人围捕仿真研究**

**设计技术：**采用Tensorflow框架和Python语言。

目前完成情况：

构建DQN网络实现flappybird的仿真实验；

构建DQN网络，采用Pygame构建实验环境，实现一追一的仿真实验；col0

构建DQN网络，构建简单的环境，验证分体可行性；col1

构建Double-Dueling DQN实现一追一的仿真实验，遇到的问题：输入不同输出相同，可能原因权重初始化问题，还未实验；

迁移学习：col0,col1->col2（多机器人围捕时正样本难收集，未成功，目前正在做）

分析：一追一追捕路径非最优；多机器人围捕时正样本难收集，未成功，目前正在做。

**需要复习的知识：**

**强化学习，不要被问倒**