项目报告

姓 名：董亚辉

学号：2019102932

**一、项目简介**

1、项目名称：图片标注小工具

2、项目功能：对图片中的图像进行分块标注，标注出图像不同部分并且给予命名。标注完成的样例展示如（图1.1）：

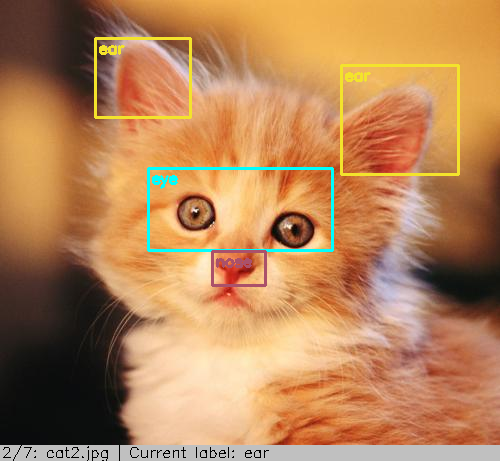
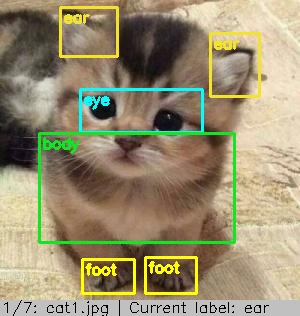


图 1.1 标注样例图

3、项目思路

对要标注的图像建立一个窗口循环，然后每次循环的时候对图像进行一次复制，鼠标在画面上画框的操作、画好的框的相关信息在全局变量中保存，并且在每个循环中根据这些信息，在复制的图像上重新画一遍，然后显示这份复制的图像。

4、操作过程

（1）首先运行 begain.py,启动项目

（2）选择是一个文件夹的路径，包含了所需标注物体框的图片。

如果图片中标注了物体，则生成一个相同名称加额外后缀.bbox的文件，来保存标注信息。选择路径图片如图1.2 ，标注完成后缀为.bbox文件图片如图1.3

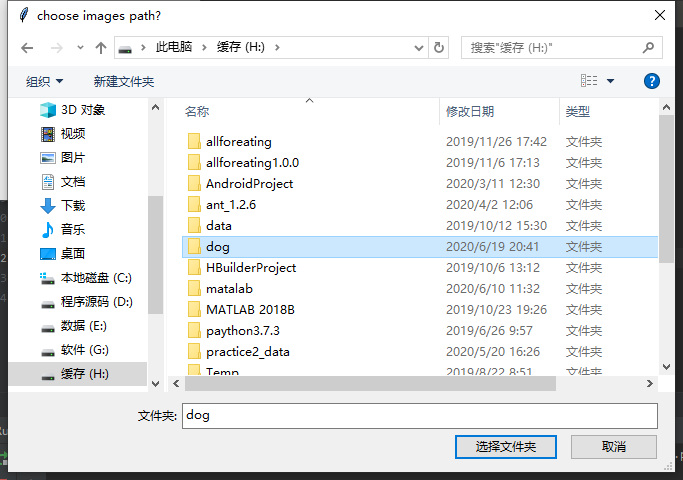
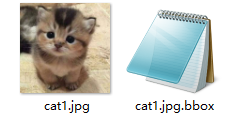
 

图 1.2 路径选择图片 图1.3 .bbox文件图片

（3）标注的方式：按下鼠标左键选择物体框的左上角，松开鼠标左键选择物体框的右下角，

①按下鼠标右键删除上一个标注好的物体框。

②所有待标注物体的类别和标注框颜色由用户自定义（.labels文件），label文件的内容如下的图1.4 所示

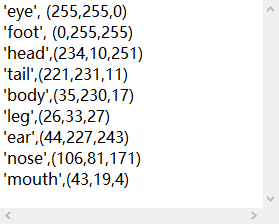


图 1.4 .labels文件内容图片

③如果没有定义则默认只标注一种物体，定义该物体名称为’eye’。

④标注完成的图片(图1.5) 以及形成的.bbox文件内容（图1.6）

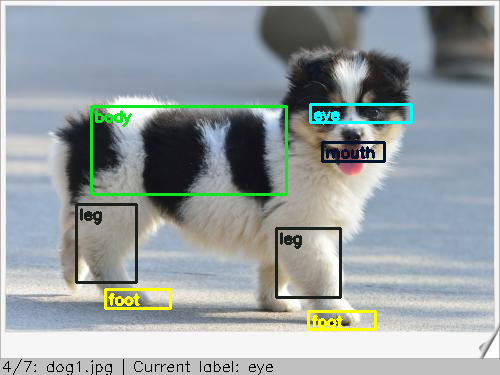
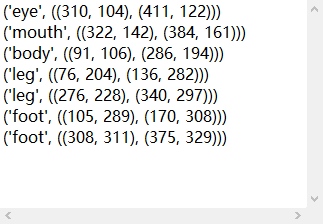
 

图1.5 标注图 图1.6 标注信息图

（4）方向键 ← 和 → 键用来遍历图片， ↑ 和 ↓ 键用来选择当前要标注的物体（注意：选择标注的需要提前设定然后标注）

**二、项目代码**

项目主要有annotation.py 和begain.py两个文件组成，其中begain.py是可执行文件以及项目启动文件。项目的主要功能都在annotation.py文件当中，其中主类为SimpleBBoxLabeling，调用主类当中的start函数即可开始执行。在主类当中主要的函数有鼠标回调函数（\_mouse\_ops）用来控制画框的开始和结束，以及鼠标右击删除最近画好的框，代码如下所示：

def \_mouse\_ops(self, event, x, y, flags, param):  
 if event == cv2.EVENT\_LBUTTONDOWN:  
 self.\_drawing = True  
 self.\_pt0 = (x, y)  
 elif event == cv2.EVENT\_LBUTTONUP:  
 self.\_drawing = False  
 self.\_pt1 = (x, y)  
 self.\_bboxes.append((self.\_cur\_label, (self.\_pt0, self.\_pt1)))  
 elif event == cv2.EVENT\_MOUSEMOVE:  
 self.\_pt1 = (x, y)  
 # 按下鼠标右键删除最近画好的框  
 elif event == cv2.EVENT\_RBUTTONUP:  
 if self.\_bboxes:  
 self.\_bboxes.pop()

画框标注和当前信息的函数（\_draw\_bbox）用来读取文件当中的图片、存储正在标注的物体信息、显示当前文件名文件个数、画出已经标注好的框、存储标注的框以及对应的名字，函数代码如下所示：

def \_draw\_bbox(self, img):  
 h, w = img.shape[:2]  
 canvas = cv2.copyMakeBorder(img, 0, BAR\_HEIGHT, 0, 0, cv2.BORDER\_CONSTANT, value=COLOR\_GRAY)  
 label\_msg = '{}: {}, {}'.format(self.\_cur\_label, self.\_pt0, self.\_pt1) \  
 if self.\_drawing \  
 else 'Current label: {}'.format(self.\_cur\_label)  
 msg = '{}/{}: {} | {}'.format(self.\_index + 1, len(self.\_filelist), self.\_filelist[self.\_index], label\_msg)  
 cv2.putText(canvas, msg, (1, h+12), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 0), 1)  
 for label, (bpt0, bpt1) in self.\_bboxes:  
 label\_color = self.label\_colors[label] if label in self.label\_colors else COLOR\_GRAY  
 cv2.rectangle(canvas, bpt0, bpt1, label\_color, thickness=2)  
 cv2.putText(canvas, label, (bpt0[0]+3, bpt0[1]+15), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, label\_color, 2)  
 if self.\_drawing:  
 label\_color = self.label\_colors[self.\_cur\_label] if self.\_cur\_label in self.label\_colors else COLOR\_GRAY  
 if (self.\_pt1[0] >= self.\_pt0[0]) and (self.\_pt1[1] >= self.\_pt1[0]):  
 cv2.rectangle(canvas, self.\_pt0, self.\_pt1, label\_color, thickness=2)  
 cv2.putText(canvas, self.\_cur\_label, (self.\_pt0[0] + 3, self.\_pt0[1] + 15),  
 cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, label\_color, 2)  
 return canvas

导出当前标注框信息函数（\_export\_n\_clean\_bbox），删除当前样本和对应的标注框信息函数（\_delete\_current\_sample）等等。其他函数的详细信息请参见annotation.py文件。

**三、项目总结**

标注工具能完成基本的标注功能，能对图片中的不同部分进行标注，能够采集到标注信息，并且存储下来。但是工具还有很多不足之处，获取的标注信息比较单一，图片的标框只能直线而不能沿着图像的边缘进行标注，对已经标注了的不能自动添加颜色背景以增加区分度。种种不足难以满足现在图片标注的种种任务，有待进一步的完善。