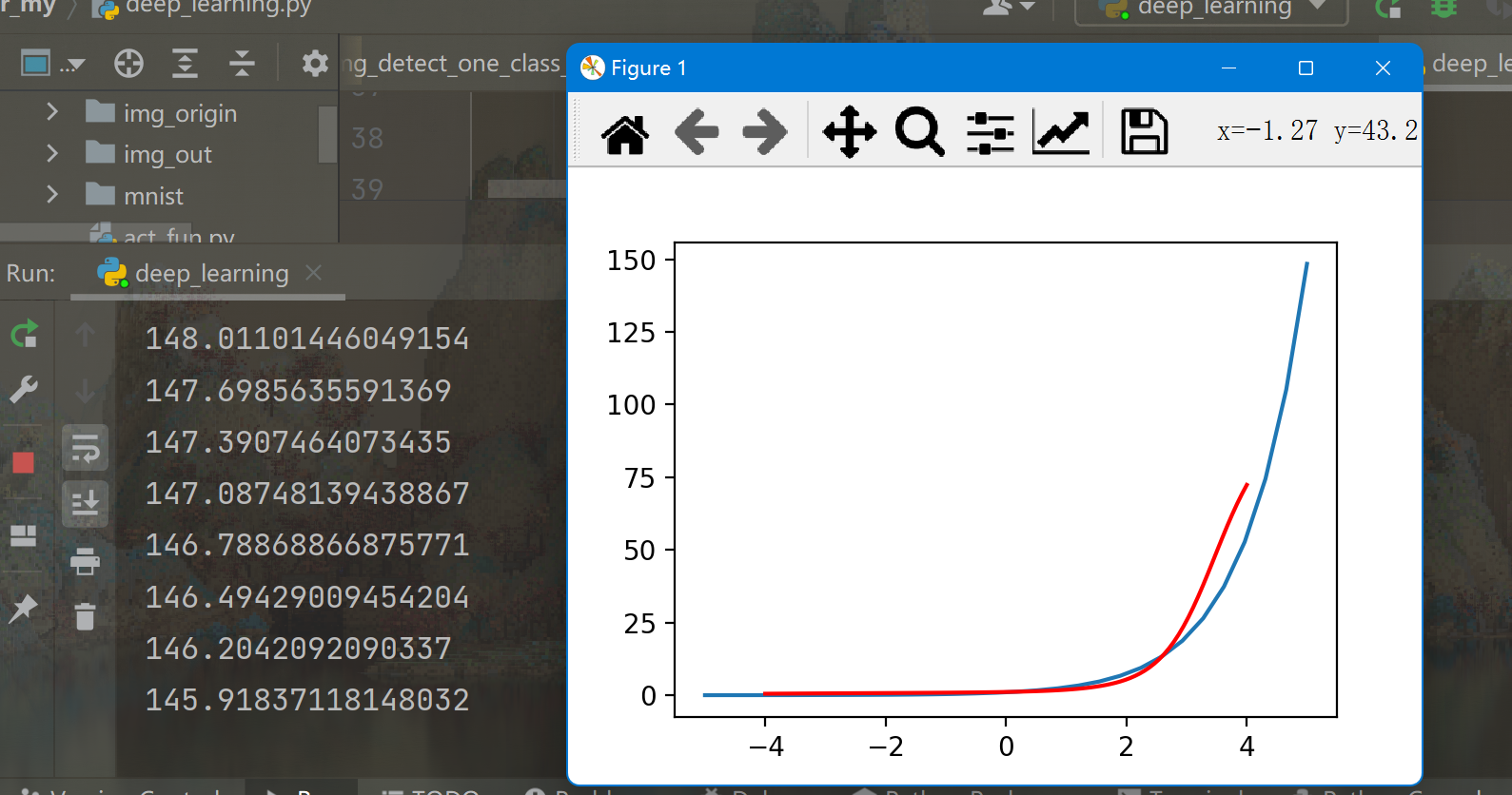
1. **deep\_learning.py**

主要的作用就是拟合曲线，输入的曲线为numpy linspace转化为exp，最后使用的是根据深度学习前向传递和反向传递使用numpy函数库，使用矩阵作为输入，并计算过程也是矩阵的运算过程，得到的结果如下：

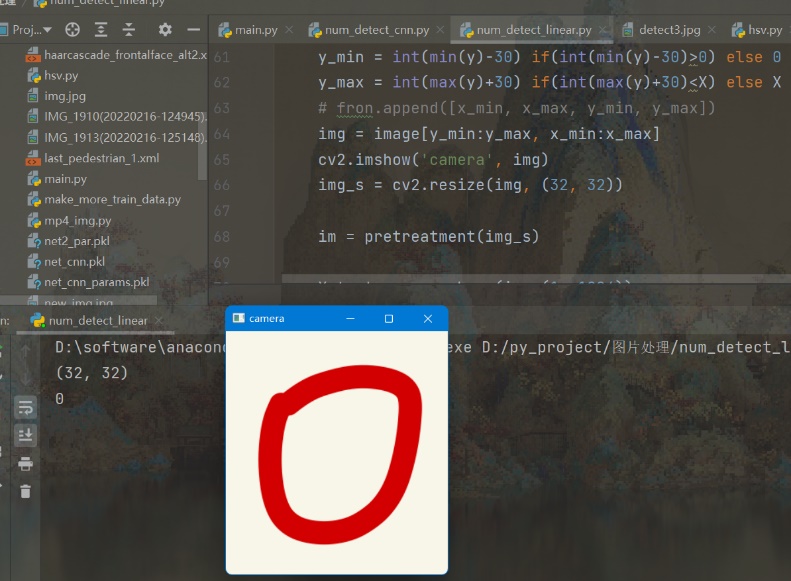


1. **handwrite.py**

主要作用将32\*32的0-1图像文件转化为[1, 1024]的数组，读取的数据信息储存在D:\py\_project\tor\_my\data1内，文件名为0\_1.txt，0为内部图片信息，1为第几张，使用神经网络，输入层为32\*32，隐藏层为10000，输出层为10，得到的结果存入net2\_par.pkl内

1. **num\_detect\_linear.py**

与上一个代码配套使用，主要作用就是将图像输入到社交网络中进行处理，得到预测的类别：



主要问题就是必须先颜色区域识别才能进行转化。

1. **handwrite\_cnn\_train32.py**

采用的是卷积神经网络处理1\*32\*32的0-1图像，从data1内读取数据输入到神经网络，最后通过全连接层得到10个输出，即表示对于这10个数的概率，选择其中最大的一个为最可信的输出，得到预测，最后保存结果到’ net\_cnn.pkl’, 'net\_cnn\_params.pkl'。

1. **cnn\_get\_img.py**

主要作用就是采集数据集，数据集的大小为128\*128，得到的结果存入img内，主要修改部分就是ball\_color，num和epo，分别为字体颜色，检测的数字，图片的数目，得到结果如下：



但是缺点就是需要修改识别的颜色，且可能会有误操作，主要作用还是收集数据，为后面的整图识别做准备。

1. **make\_more\_train\_data.py**

主要作用就是将原有数据集扩展，得到的数据集将达到更大，采用的运算主要有镜像，旋转，开运算，加噪声等等，可以将数据集扩展为原来的6倍多，最后由上一个函数的图像结到如下效果：



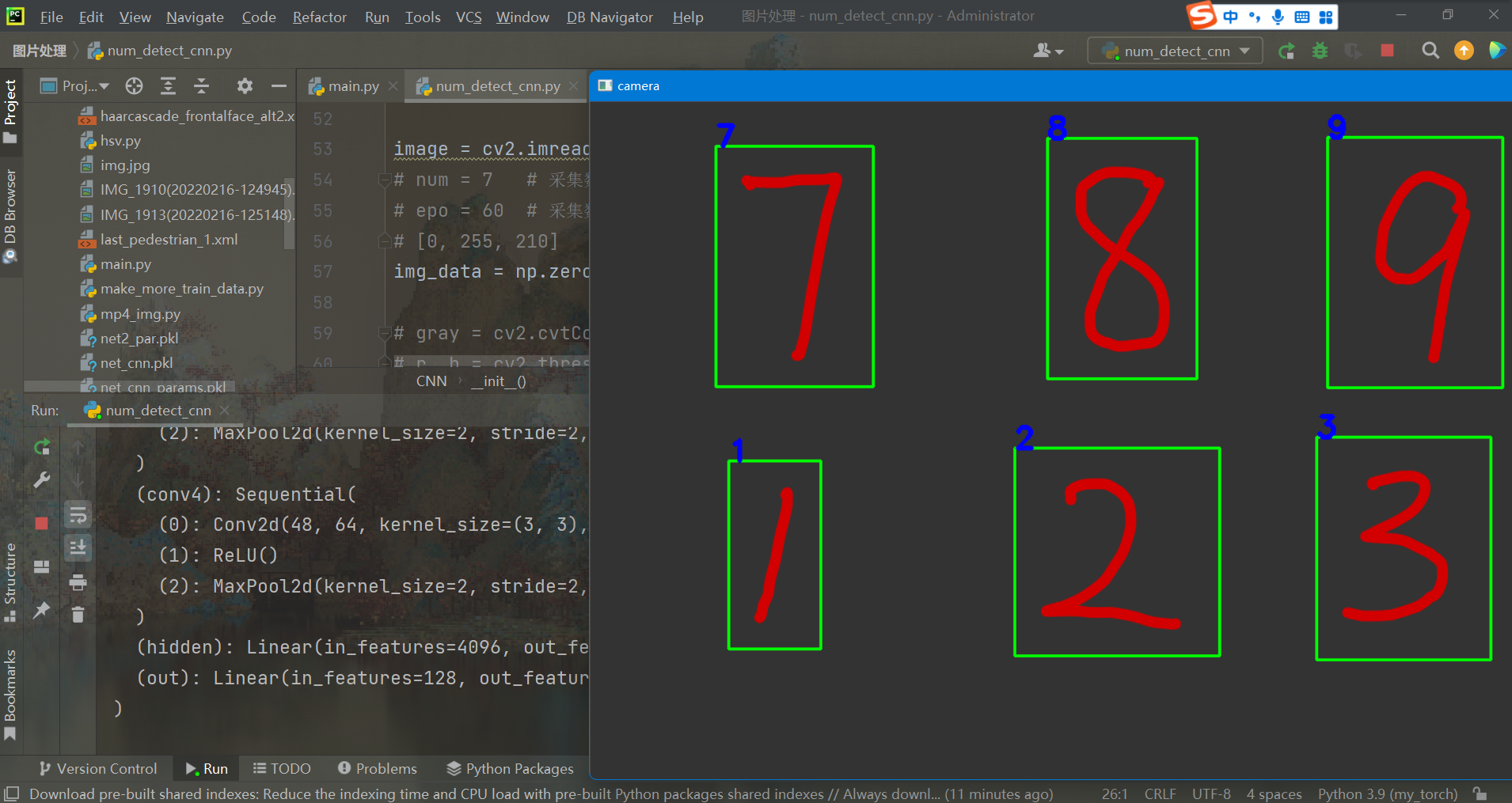
结果一共得到的图像为12290张图片。

1. **handwrite\_num\_train\_128.py**

通过上面得到的数据集，得到图像，采用的是(length, 3, 128, 128)这种输入类型，使用三维的图像输入卷积神经网络，最后通过全连接层，输出的为10层，得到的最大值就是该图像的最接近的标签，最后结果输出为'net\_cnn.pkl', 'net\_cnn\_params.pkl'

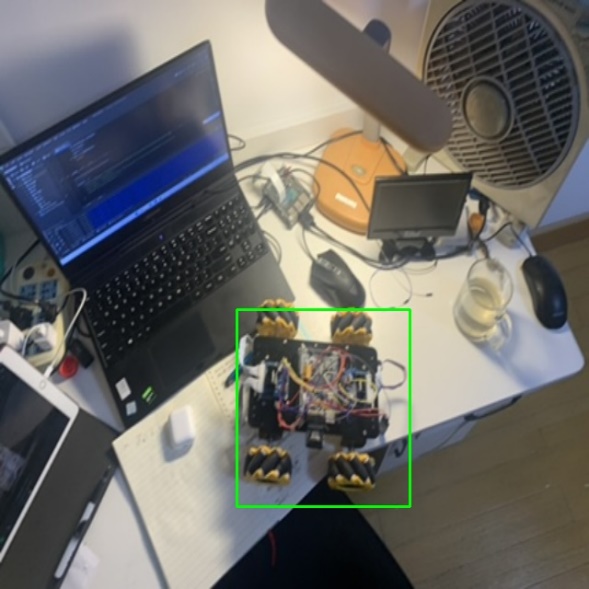
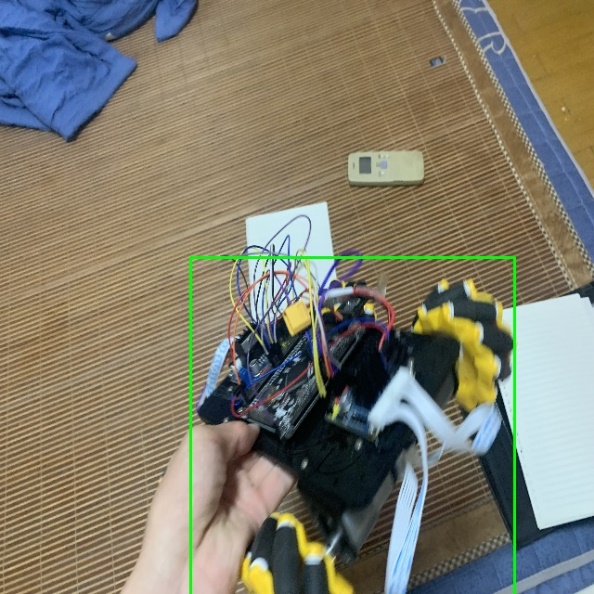
1. **num\_detect\_cnn.py**

通过上一个函数得到的结果对图像进行识别并框识别，得到的结果显示在屏幕，不过需要对颜色首先进行识别，才能完成该区域的目标识别并分类，该代码不支持一张图像随机位置识别，只能先颜色识别可能的区域再检测。

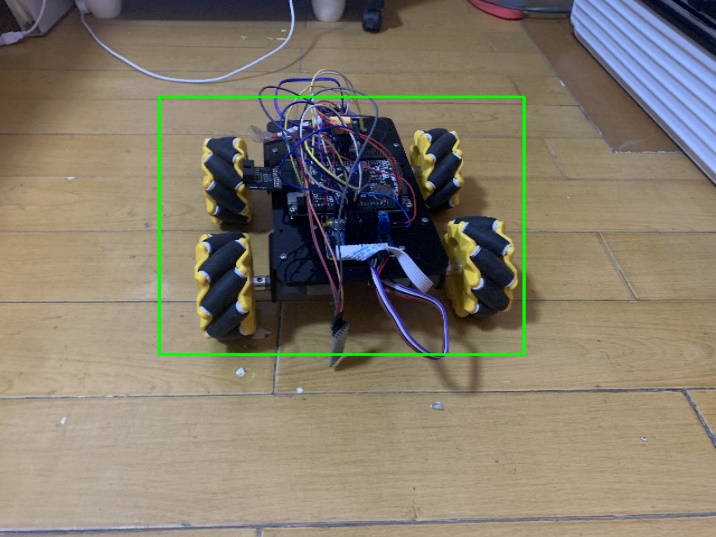
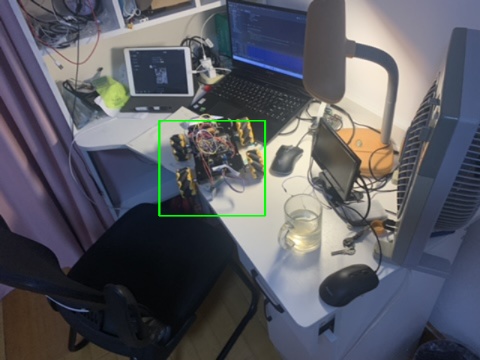
****

1. **img\_detect\_one\_class\_640input.py**

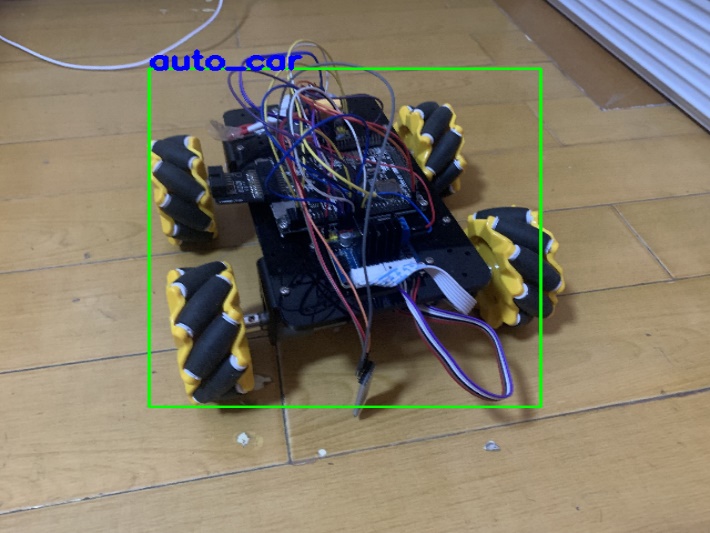
训练D:\py\_project\tor\_my\img1内的图片文件，图片原大小长宽>2000，将图片转化为640\*640输入，图片数目随便，卷积函数使用的nn.Conv2d，使用Relu激活函数，采用的类别为1类，所以暂不支持单张图片多类别识别，输出为[1, dx, dy, dw, dh, class]，但是问题是在训练过程中第一个值一直为0，暂未解决；分为两个函数，一个函数为train，主要就是训练，训练图片在img1，训练的xml文件在img1\_xml内，每张图片必须有对应的xml文件，最后得到的结果就是net\_img\_cnn.pkl，net\_img\_cnn\_params.pkl，得到的结果存在当前文件夹下，另一个函数主要是img\_test(dir, save\_dir, pkl\_name)，三个参数主要是：dir：图片文件夹, sav\_dir：保存文件夹, pkl\_name：pkl文件名，图片检测的结果存入img\_out文件夹，结果是已经变形情况，未变形结果畸变严重，得到的结果如下：



2.23日修改：9. img\_detect\_one\_class\_640input，在此基础上修改一些，使其能够实现输出原始图像及其框图，图片无畸变，webcamera还没有尝试。



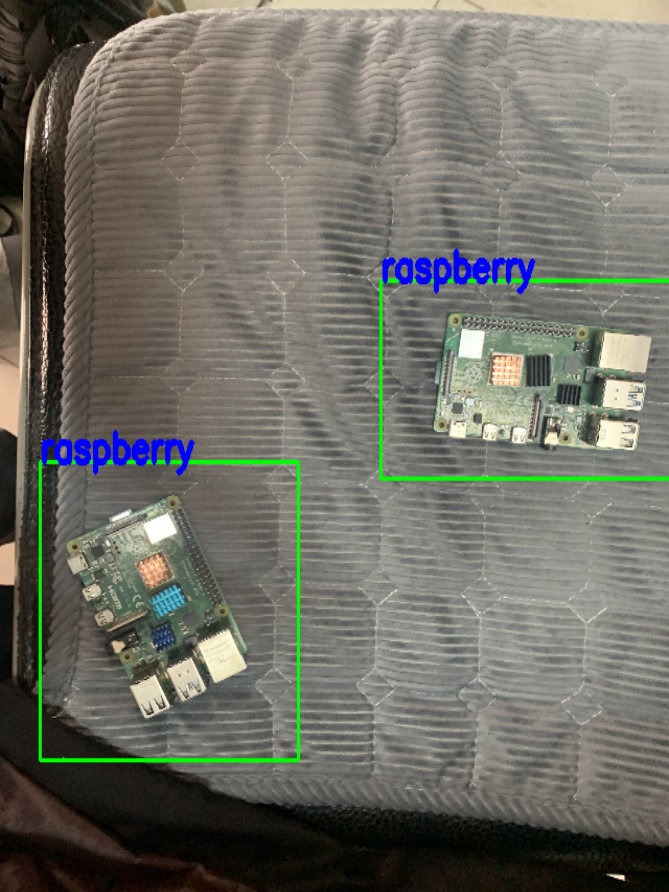
1. **img\_detect\_more\_classes\_640input.py**

这一代码主要作用是多个类别输入情况下的图像识别，图像的类别可以定义，只要增加classes的类别即可，会自动增设变量，但是如果类别变化，将train的load参数设置为False，这样就可以重新训练，神经网络的最后采取的是全连接层而不是卷积层，最后输出的格式是[1, dx, dy, dw, dh, 0, 1, 0]表示是classes的第二类别，其余与上一个相似，此代码不支持一张图片每个类别多目标识别，最后得到结果为：





**11. img\_detect\_more\_target\_640input.py**

 使用卷积神经网络，输入为3\*640\*640，等效卷积核为160\*160，stride=120，padding=0，最后得到的为5\*5\*(1+4+num\_class)的输出结果，各个列对应的为当前框的值，同上一个代码输出值，最后可以实现对多目标多类别的输出，最后再使用img\_test得到图片，结果如下：

