# 《人工智能导论》学习报告

曹琛 2016202171

## 一、前言

随着硬件、算法的发展，人工智能技术在不断完善，并且被广泛应用到各个领域，人工智能已经成为新一轮科技革命和产业变革的重要驱动力量。这一学期，我选修了胡鹤老师的《人工智能导论》，收获颇丰。与传统教学不同，胡鹤老师选择让我们在一个学期内完成一个大型作业，并且以此为考核方式，让我们能够及时将实际应用与所学知识相结合。实际应用的需求促进算法理论的探索，而理论本身又为应用的完善提供保障，两者相辅相成，我们很是受益。

这一学期我们大作业的主题是“人工智能小车”。在本项目中，我们通过自己搭建小车，熟悉了基本硬件构造以及Arduino编程，并且基于蓝牙实现了与手机、电脑的通信，在后续实际功能实现中，我们了解到许多实用的库与API，并且实际运行了人工智能领域的许多算法。

我们组的最终项目目标是实现初步自动驾驶。项目初期，我们基于小车的超声波、红外传感器模块实现了避障与循迹功能；项目中后期，我们通过图像处理、识别技术实现交通标识识别功能，小车可以根据交通标识做出相应反应，然后我们还考虑加入语音识别的功能，我们基于双向RNN模型实现了中文语音分词，然后小车可以根据语音指示做出相应反应。

## 二、项目陈述（个人部分）

我负责的部分主要是小车的基础功能实现、小车与电脑的通信、以及图像识别功能的完成。

### （一）小车的基础功能实现

小车功能的基础实现主要是依赖小车自带的超声波传感器模块红外传感器模块。小车的运动主要是通过调整信号来实现。以下以“前进”运动函数为例。

void run(){ // 前进

digitalWrite(Right\_motor\_go, HIGH); // 右电机前进

digitalWrite(Right\_motor\_back, LOW);

analogWrite(Right\_motor\_go, 150); //0~255调速，左右轮差异略增减

analogWrite(Right\_motor\_back, 0);

digitalWrite(Left\_motor\_go, LOW); // 左电机前进

digitalWrite(Left\_motor\_back, HIGH);

analogWrite(Left\_motor\_go, 0); //PWM比例0~255调速，左右轮差异略增减

analogWrite(Left\_motor\_back, 150);

}

（Code1：运动函数示例）

避障功能与循迹功能的具体实现过程是以传感器返回的信号为参数，执行各种小车运动的函数，代码逻辑如下。

void loop(){

keysacn(); //调用按键扫描函数，当按键没被按下时处于忙等待状态；当按键被按下时，跳出死循环开始超声波避障。

while (1){

Distance = Distance\_test(); //调用测距函数测量前方最近障碍物的距离

if (Distance < 10) //当前方的障碍物的距离小于某个阈值时，开始后退并向右转一个角度{

back(2);

right(1); //右转

digitalWrite(beep, HIGH); //蜂鸣器响起

brake(1); //停车

}

else{

run(); //无障碍物，直行

digitalWrite(beep, LOW);

}

}

}

（Code2：避障功能代码逻辑）

void loop(){

keysacn(); //调用按键扫描函数，当按键没被按下时处于忙等待状态；当按键被按下时，跳出死循环开始红外循迹。

while (1){ // 检测到黑线输出高，信号为HIGH，小车底部同侧LED灯亮；检测到白色区域输出低，信号为LOW，同侧灯灭

SR = digitalRead(SensorRight);

SL = digitalRead(SensorLeft);

if (SL == LOW && SR == LOW) run();

else if (SL == HIGH & SR == LOW)

// 左循迹红外传感器输出高，检测到黑色区域，车子向右偏离轨道，此时应该向左转

left();

else if (SR == HIGH & SL == LOW)

// 右循迹红外传感器输出高，检测到黑色区域，车子向左偏离轨道，此时应该向右转

right();

else// 左右循迹红外传感器都输出高，车子处在黑色区域内，小车停止

brake();

}

}

（Code3：循迹功能代码逻辑）

### （二）小车与电脑的通信

在实现交通标识识别功能时，需要实现小车与电脑的通信。小车与电脑的通信主要是两个方面，其一摄像头视频传输，其二PC指令返回。我们遇到的第一个问题是由于网络问题，手机摄像头视频传输给PC端十分缓慢，第二个问题是动态视频处理较为困难。通过查阅相关资料，我们解决了这些问题。第一个问题的解决方法是在手机与电脑上均安装DroidCamX，PC端再借用opencv库提供的接口即可实现摄像头视频流共用；第二个问题的解决方法是通过python截屏工具，实现单帧图像的处理。在电脑给小车发送指令的功能实现中，我们考虑在小车上安装蓝牙，然后基于Pyserial监听蓝牙。

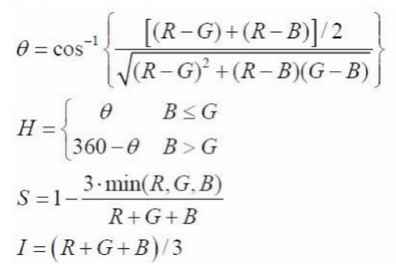
### 图像识别功能

#### 1、交通标识的检测

在PC端处理图像时，主要分为两个模块。其一是交通标识的检测，其二是交通标识的识别，我主要参与了交通标识的检测工作。

交通标识检测方面，我们采用了基于颜色分割的图像二值化处理。具体流程主要由四步构成：①加载图片，转换为HSV通道；②颜色过滤；③优化；④裁剪区域。

最开始我提出直接利用RGB颜色空间来描述图像的色彩情况，但是RGB色彩空间极易受到光线情况的影响。我曾用一些测试集进行测试，发现RGB色彩分割效果在图像成像质量较为理想的时候效果非常好，但也很容易受到干扰，我们查阅相关资料后，决定使用HSI色彩空间模型进行色彩分割。其中从RGB色彩空间转换到HSI空间的转换方式如下：



①H：色调，与光波的波长有关，它表示人的感官对不同颜色的感受。

②S：饱和度，表示颜色的纯度。

③I：亮度：对应成像亮度和图像灰度。

我们可以很清楚看到HSI色彩空间将饱和度和亮度信息独立出来，可以一定程度上降低亮度和色彩的关系，因此理论上效果会有所提升，实际证明我们是正确的。

Python中有直接将图像转换为HSV通道的函数，代码如下：

import cv2

import numpy as np

#加载原图

img=cv2.imread('walks.jpg')

print('img:',type(img),img.shape,img.dtype)

cv2.imshow('img',img)

#转换为HSV通道

hsv=cv2.cvtColor(img,cv2.COLOR\_BGR2HSV)

cv2.imshow('hsv',hsv)

（Code4：加载图像并转换为HSV通道）

我们不能确定交通标志的底色，因此为了简化处理，我们要考虑指定标志的底色，然后将图像中相同底色的区域提取出来，这一步就是进行颜色过滤，即可通过inRange函数确定提取的颜色范围。

#提取蓝色区域

blue\_lower=np.array([100,50,50])

blue\_upper=np.array([124,255,255])

mask=cv2.inRange(hsv,blue\_lower,blue\_upper)

print('mask',type(mask),mask.shape)

cv2.imshow('mask',mask)

（Code5：颜色过滤）

处理结果如下图，可以看到实际场景中有一些颜色干扰，如白框旁边的一些白点，因此我们需要进一步优化。



（图1：颜色过滤完成图）

进行优化处理时我们分为以下几步：

①模糊处理，可以去除一些小的干扰，如下图，我们可以看到模糊后旁边的小白点消失。



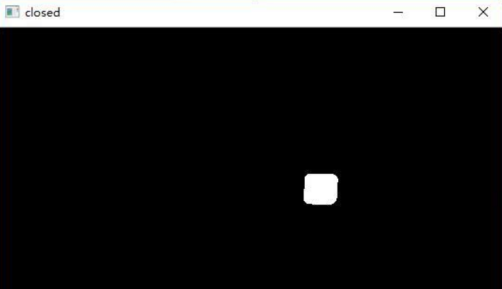
（图2：模糊处理完成图）

②二值化，将模糊后的图像转换为二值图，只有两种颜色，更便于区域划分。



（图3：二值化处理完成图）

③闭运算，即封闭区域，使其无间隙。



（图4：闭运算处理完成图）

然后我们即可进行目标区域（白）的裁剪，即可显示“检测”出交通标识。

#### 2、对于CNN的个人理解

后续交通标识识别以及语音识别的工作中，我查阅了很多相关模型训练的资料，但最后具体实现工作参与度不是很高，这里不再对实验过程进行赘述，只谈一些自己的理解。

我们交通标识的识别主要是基于简单卷积神经网络（CNN）的训练，这也是老师在课程中提到过的。我之前只听过卷积神经网络的大名，实际接触之后才深刻体会到它的巧妙。

滤波器和卷积神经网络（CNN）有什么关系？我在上这门课之前经常听到这两个名词同时出现。其实这就是一个识别问题：我们要识别图像中的某种特定曲线，也就是说，这个滤波器要对这种曲线有很高的输出，对其他形状输出很低，这也就像是神经元的激活。

我们对一张图像进行一次卷积，得到结果中，在特定曲线和周边区域就有很高的输出，而其他区域相对较低，这就是一张激活图，对应的高值区域就是我们要检测曲线的位置。在训练CNN的某一个卷积层时，我们实际上就是在训练一系列的滤波器，而训练CNN就是在训练每一个卷积层的滤波器，让这些滤波器对特定模式有较高的激活，以达到CNN分类检测等目的。

CNN的第一个卷积层的滤波器用来检测低阶特征，如边、角、线等，随着卷积层的增加，对应滤波器检测的特征就更加复杂，第二层如半圆、长方形等......最终可以达到人脸、手写字体等复杂特征的识别。即相当程度上来说，构建CNN的任务就是构建这些滤波器，而“训练”就是改变滤波器矩阵的值，使其能识别特定的特征。

我们这次实验采用的是简单卷积神经网络训练，只能实现二层卷积神经网络，即检测交通标识中的图形。

以上就是我在这个项目中主要参与的工作。总的来说，充实、有趣贯穿其中，很感谢胡鹤老师给我们这个机会能做一个真正“有自己想法”的东西。

## 三、结语

人工智能是一个非常有挑战的领域，在这个学期胡鹤老师开设的《人工智能导论》课程中，我感触颇深，收获颇丰。

**主要有以下收获：**

①对AI领域有了系统的认识。目前人工智能行业中最引人关注的九个发展热点领域分别是：芯片、自然语言处理、语音识别、机器学习应用、计算机视觉与图像、技术平台、智能无人机、智能机器人、无人驾驶。我们课程涉及了其中绝大部分的内容。

②硬件操作能力。小车底层搭建让我了解到许多硬件方面的知识，并且对Arduino编程有了一定了解。

③软件使用能力。对相关平台的使用更加熟练，对Tensorflow这个深度学习框架有了一定了解，并且发现了很多“神奇”的库。

④相关算法学习及其应用。监督学习算法：人工神经网络类，贝叶斯类，决策树类，线性分类器；无监督学习类算法：关联规则学习，聚类分析，异常检测等。其中我们主要是对人工神经网络类进行了实际应用。

**主要有以下几点感受：**

①全面性。我觉得这种布置贯穿整个学期的项目的课程方式非常好，在其他课程中我们虽然会按时布置对应阶段的实验，但是始终没有机会进行一个有充足时间的、规模大且完备的项目设计，而《人工智能导论》这门课为我们提供了一个很好的机会——我们能够从硬件、软件两个层次去进行属于自己的“工业设计”，底层搭建、功能设计、功能实现、测试与调整，这一个完整的流程的每个环节均使我们受益匪浅。同时，我们也绝非“闭门造车”，老师对人工智能诸多领域的技术都做了详尽介绍。

②趣味性与挑战性并存。功能设计中，同学们都能大大发挥了自己的想象力，而具体实现当然也绝非易事，除了对于算法的了解及具体实现，调试工作也非常繁琐，但是整个过程回味起来仍旧是美好、充实的。

③团队合作的重要性。多人头脑风暴，项目进行中的沟通都是非常重要的。

《人工智能导论》这门课不仅让我对AI有了较为系统的认识，更是给了我兴趣的钥匙，我希望以后能有更多机会参与到人工智能领域的研究中。

最后，感谢胡鹤老师这一个学期的教导，老师您辛苦了。