



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY



自动化学院
SCHOOL OF AUTOMATION

自动化学院 2021-2022 第二学期

《自动控制理论课程设计》

课程设计报告

姓 名： 于世然

学 号： 1120200810

班 级： 06012002

时间： 2022 年 6 月 1 日

项目	实验一	实验二	实验三
得分			

自动控制理论课程设计——垃圾分拣系统

一、项目任务书要求

设计垃圾分拣系统，以两台 qube 电机 (QLAB 数字孪生平台或数学模型代替) 和 1 台摄像头作为检测装置，在 simulink 环境下实现系统逻辑功能，并设计子系统控制器，并满足以下条件：

- (1) 能够区分最少 3 种颜色的垃圾。
- (2) 每分钟完成分拣不少于 10 件，效率越高系统性能越优秀。
- (3) 系统传输模块 (转盘) 转停稳态误差不大于 2% (具体请写明每次转动角度)，超调量不大于 5%。
- (4) 系统分拣模块逻辑设计合理。

二、项目分工和计划

小组成员与分工：于世然 (图像识别)、郑明松 (转停)、刘雅萱 (分拣)。

计划：

- 1、图像识别实现 alexnet 迁移学习网络进行颜色识别
- 2、转停模块使用串级 pid 实现稳定的转停
- 3、分拣模块实现稳定精准的动作。
- 4、完成 PPT 与答辩文案

Deadline:

- 5.25 所有小组成员完成所有实验任务
- 5.29 汇总总结完成答辩 ppt



图表 1 项目分工和进度 (板栗看板)

三、项目完成情况

1、项目总体设计要求分析

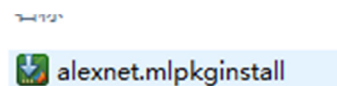
通过 matlab 中 alexnet 神经网络进行迁移学习，同时利用 matlab 相机函数实现实时对于白绿红黄四种颜色的辨认，并且通过 s-function

将 .mat 文件整合进 simulink，实现实时对颜色的判断为剩余部分输出判断信号，白绿为 0，黄 1，红 2。

2、项目设计思路和实施过程

1) matlab alexnet 神经网络进行迁移学习

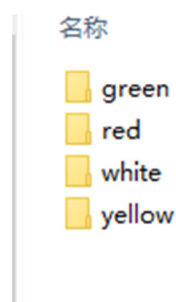
<1>安装 alexnet 工具包和 camera 工具包



图表 2 alexnet 工具包

<2>训练集图片预处理

收集训练集图片，分为四个文件夹：white、green、yellow、red，因为是采用已有的图像识别网络 alexnet 进行迁移学习，我们采集少量图片即可。



图表 3 数据集



图表 4 数据集内容

然后开始编写训练集图片预处理程序，我们

需要适配 alexnet 的要求使用 227*227*3 的 rgb 图片。

```
function [ ] = photoresize( readdir )
% 图像预处理函数
% 输入参数: readdir 需要处理图像所在目录(要求只有图像文件, 格式无所谓)

% 写入图像的格式
writetype = 'jpg';
% 写入图像的目录
writedir = ["F:\test\yellow"];
% 大小改变因子
resizefactor = [227 227];

% 创建改大小之后图像目录, 如果目录已经存在会报警告, 但是不影响使用
mkdir(writedir);
% 读取目录内所有所有图像目录信息
imnames = dir(readdir);
% 去掉目录信息中的无用项( . 和 .. )
imnames(1:2)=[];
% 统计图像个数
imcnt=length(imnames);

% 针对每一个图像
for imidx = 1:1:imcnt
    % 读入图像
    imtemp = imread(fullfile(readdir,imnames(imidx).name));
    %调整图像RGB通道数量, 若不为3, 则改为3
    if numel(size(imtemp)) ~= 3
        imtemp = cat(3,imtemp,imtemp,imtemp);% 用于将图片改为3通道
    end
    % 改变图像大小
    imtemp = imresize(imtemp,resizefactor);
    % 按照需要格式写入图像
    imwrite(imtemp,fullfile(writedir,[imnames(imidx).name(1:end-3),writetype])
end
end
```

图表 5 图像预处理程序

通过上述程序的处理我们就可以将四个文件夹中的所有照片变为 227*227*3 的训练集图片



图表 6 预处理后的图像

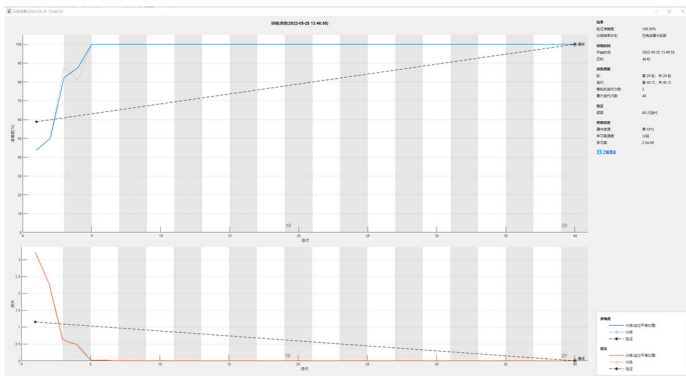
<3>编写迁移训练主函数进行训练

我们可以采用 matlab 成熟的深度学习函数进行迁移学习。程序如下

```
1 clear all;
2 clc;
3
4 net = alexnet; %加载在ImageNet上预训练的网络模型
5 imageInputSize = [227 227 3];
6
7 %加载图像, 修改预处理后的图像地址
8 allImages = imageDatastore('..\test',...
9     'IncludeSubfolders',true,...
10     'LabelsSource','folderNames');
11 %划分训练集和验证集
12 [training_set,validation_set] = splitEachLabel(allImages,0.7,'randomized');
13
14 %由于原始网络全连接层1000个输出, 显然不适用于我们的分类任务, 因此在这里替换
15 layersTransfer = net.Layers(1:end-3);
16 categories(training_set.Labels)
17 numClasses = numel(categories(training_set.Labels));
18
19 %新的网络
20 layers = [
21     layersTransfer
22     fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor',1,'BiasL
23     softmaxLayer('Name','softmax')
24     classificationLayer('Name','classOutput')];
25
26 lgraph = layerGraph(layers);
27 plot(lgraph)
28 %对数据集进行扩增
29 augmented_training_set = augmentedImageSource(imageInputSize,training_set);
30
31 %'MaxEpochs' 即训练次数, 根据需要调整
32 opts = trainingOptions('adam', ...
33     'MiniBatchSize', 32,... % mini batch size, limited by GPU RAM, default 100 or
34     'InitialLearnRate', 1e-4,... % fixed learning rate
35     'LearnRateSchedule','piecewise',...
36     'LearnRateDropFactor',0.25,...
37     'LearnRateDropPeriod',10,...
38     'L2Regularization', 1e-4,... constraint
39     'MaxEpochs',20,...
40     'ExecutionEnvironment','cpu',...
41     'ValidationData', validation_set,...
42     'ValidationFrequency',80,...
43     'ValidationPatience',8,...
44     'Plots','training-progress');
45
46 %训练
47 net = trainNetwork(augmented_training_set, lgraph, opts);
48
49 %保留训练出来的net网络, 需要随后另存
50 save Alex_Public_32.mat net
51
52 %数据的可视化显示
53 [predLabels,predScores] = classify(net, validation_set);
54 plotconfusion(validation_set.Labels, predLabels)
55 PerItemAccuracy = mean(predLabels == validation_set.Labels);
56 title(['overall per image accuracy ',num2str(round(100*PerItemAccuracy)),'%'])
57
```

图表 7 迁移训练主函数

他的主要步骤可以看作: 1、加载图像, 修改处理后的图像地址 2、划分训练集验证集 3、替换连接层 4、产生新的网络对数据集进行扩增 5、训练 6、保存新的网络。同时还可以运用 matlab 库中的一些函数进行训练数据的可视化。



图表 8 学习进度可视化



图表 9 学习准确度可视化

最终得到我们想要的网络文件

2) 编写图像识别主程序

这里用到了 webcam 的库可以在 matlab 中下载。
程序如下：

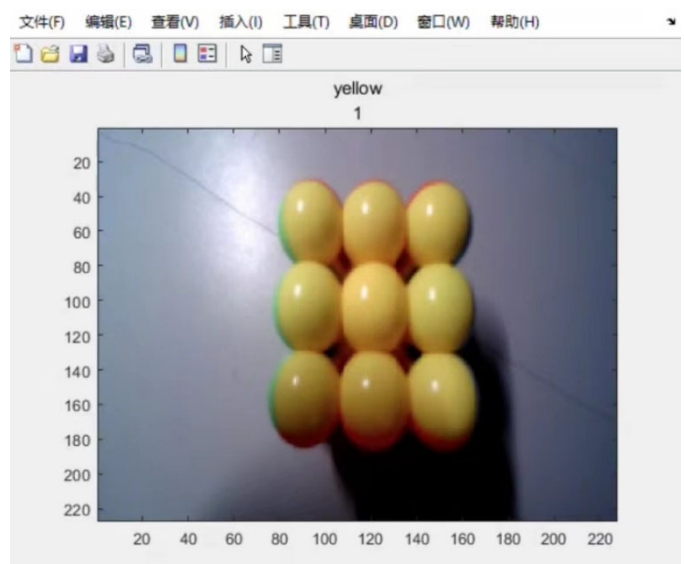
```

1 clear
2 camera = webcam;%调用摄像头
3 load('Alex_Public_32');%载入训练好的网络
4 c=0;%c是用来输出的标志位
5 while true%循环进行摄像头拍摄，然后分类
6
7     picture = camera.snapshot;%摄像头分类，设定大小
8     picture = imresize(picture,[227,227]);
9
10    label = classify(net,picture);%分类
11
12    if label=='yellow'%分别对不同颜色的标志位赋值
13        c=1;
14    elseif label=='red'
15        c=2;
16    elseif label=='white'
17        c=0;
18    elseif label=='green'
19        c=0;
20    end
21
22    image(picture);%显示图片
23    title(char(label),c);
24 end

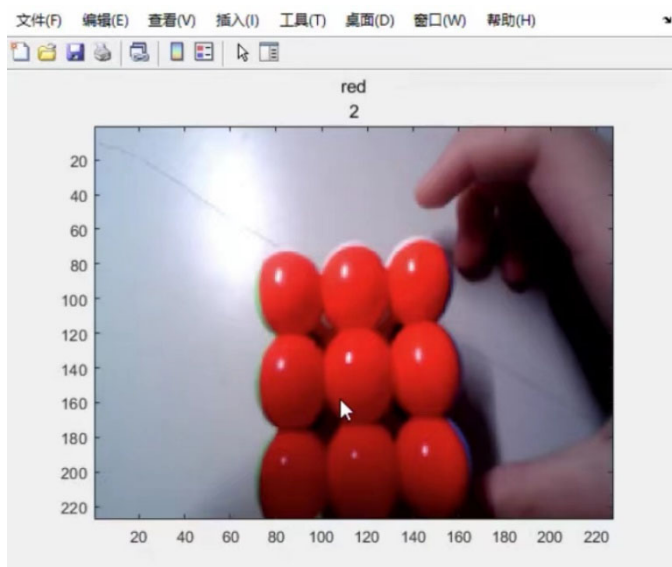
```

图表 10 图像识别主程序

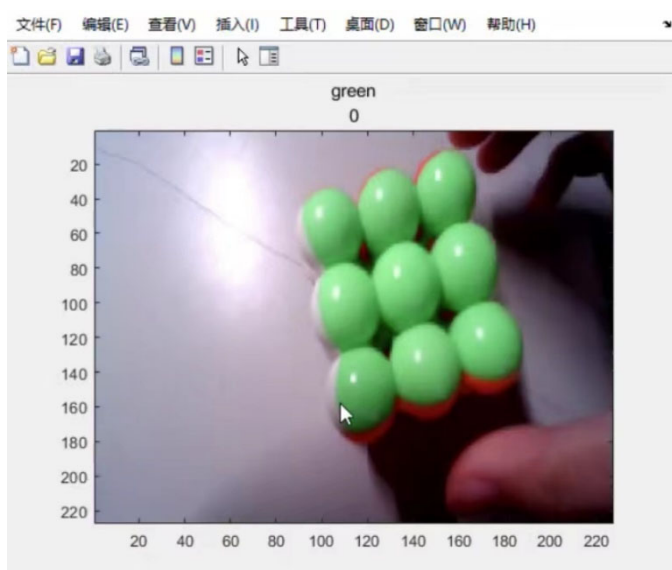
可以看到程序主要分为以下几步：1、调用摄像头和载入训练好的网络 2、循环进行摄像头拍照和 对拍照后的图片进行分类 3、得到分好结果的类别，给标志位赋值输出 4、实时显示得到结果的图像。
效果如下：



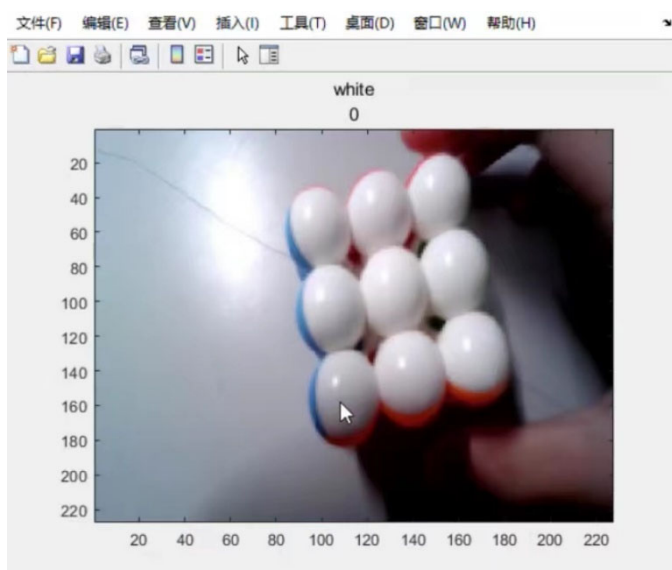
图表 11 黄色



图表 12 红色



图表 13 绿色



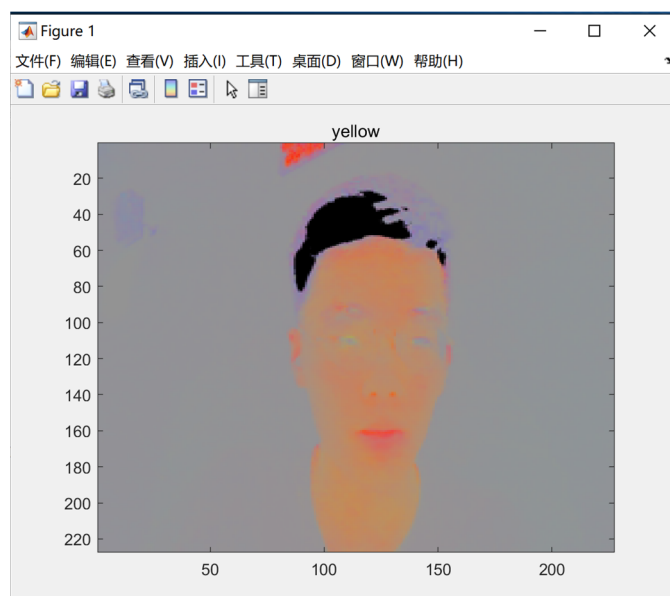
图表 14 白色

后期优化：增加了归一化色彩空间的图像预处理功能。通过对图像的 RGB 色彩空间进行归一化处理, 在某些情况下是去除光照和阴影影响的一种简单和有效的方法。

假设 RGB 代表原图像某点的像素值, rgb 表示归一化之后的值, 则

$$r = R / (R + G + B); \quad g = G / (R + G + B); \quad b = B / (R + G + B);$$

```
function ed = DoSomethingCrazy1(frame)
% 归一化彩色空间
gray = rgb2gray(frame);
mask = double(gray>0.05);
imsum = sqrt(sum(frame.^2,3));
ed = frame./imsum.*mask;
end
```



3) 通过 simulink 中的 s-function 将学习后的.mat 文件整合进入 simulink 实现信号输出。

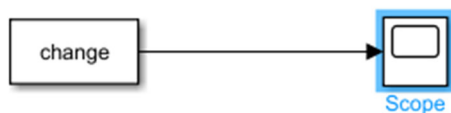
这一部分的任务主要分以下几步:

<1>Simulink 框架的搭建

因为上面所编写的程序需要调用 matlab 自带的函数, 所以我们不能使用简单的 simulink 框架, 但是我们可以使用 s 函数——s-function 模块进行搭建。经过很痛苦的 s 函数学习过程后我掌握了使用 matlab 语言编写 level-1s 函数的方法, 将他运用到了这次迁移学习辨认颜色的任务中。

Simulink 的结构极度简洁, 所有图像处理功能全部集成在编写的 s 函数中, 只输出一个颜色值给队友制作的模块。方便队友之间的合作和通信。

结构图如下:



图表 15 simulink 框架

<2>编写 s 函数

编写的 s 函数主要是移植了上一个部分写的.m 图像识别程序,将他运用在了simulink中。程序如下:

```
1 function [sys,x0,str,ts,simStateCompliance] = change(t,x,u,flag)
2 %主函数
3 %主函数包含四个输出:
4 % sys数组包含某个子函数返回的值
5 % x0为所有状态的初始化向量
6 % str是保留参数, 总是一个空矩阵
7 % ts返回系统采样时间
8 %函数的四个输入分别为采样时间t、状态x、输入u和仿真流程控制标志变量flag
9 %输入参数后面还可以接一系列附带参数simStateCompliance
10 switch flag,
11 case 0,
12     [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes;
13 case 1,
14     sys=mdlDerivatives(t,x,u);
15 case 2,
16     sys=mdlUpdate(t,x,u);
17 case 3,
18     sys=mdlOutputs(t,x,u);
19 case 4,
20     sys=mdlGetTimeOfNextVarHit(t,x,u);
21 case 9,
22     sys=mdlTerminate(t,x,u);
23 otherwise
24     DASudio.error('Simulink:blocks:unhandledFlag', num2str(flag));
25 end
26 %主函数结束
27 %下面是各个子函数, 即各个回调过程
28 function [sys,x0,str,ts,simStateCompliance]=mdlInitializeSizes
29 %初始化回调函数
30 %提供状态、输入、输出、采样时间数目和初始状态的值
31 %初始化阶段, 标志变量flag首先被置为0, S-function首次被调用时
32 %该子函数首先被调用, 且为S-function模块提供下面信息
33 %该子函数必须存在
34 sizes = simsizes;
35 %生成sizes数据结构, 信息被包含在其中
36 sizes.NumContStates = 0;
37 %连续状态数, 缺省为0
38 sizes.NumDiscStates = 0;
39 %离散状态数, 缺省为0
40 sizes.NumOutputs = 1;
41 %输出个数, 缺省为0
42
```

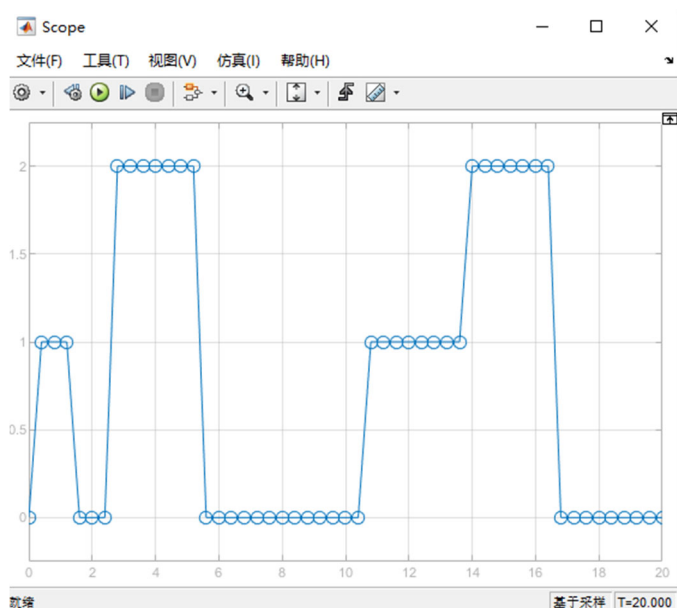
```
40 %离散状态数, 缺省为0
41 sizes.NumOutputs = 1;
42 %输出个数, 缺省为0
43 sizes.NumInputs = 0;
44 %输入个数, 缺省为0
45 sizes.DirFeedthrough = 0;
46 %是否存在直接通过, 1存在, 0不存在
47 sizes.NumSampleTimes = 1;
48 %采样时间个数, 至少是一个
49 sys = simsizes(sizes);
50 %返回size数据结构所包含的信息
51 x0 = [];
52 %设置初始状态
53 str = [];
54 %保留变量置空
55 ts = [0 0];
56 %设置采样时间
57 simStateCompliance = 'UnknownSimState';
58
59 function sys=mdlDerivatives(t,x,u)
60 %计算导数回调函数
61 %给定t,x,u计算连续状态的导数, 可以在此给出系统的连续状态方程
62 %该子函数可以不存在
63 sys = [];
64 %sys表示状态导数, 即dx
65 function sys=mdlUpdate(t,x,u)
66 %状态更新回调函数
67 %给定t,x,u计算离散状态的更新
68 %每个仿真步内必然调用该子函数, 不论是否有意义
69 %除了在此描述系统的离散状态方程外, 还可以在此添加其他每个仿真步内都必须执行的代码
70 sys = [];
71 %sys表示下一个离散状态, 即x(k+1)
72 function sys=mdlOutputs(t,x,u)
73 %计算输出回调函数
74 %给定t,x,u计算输出, 可以在此描述系统的输出方程
75 %该子函数必须存在
76 camera = webcam;
77 load('Alex_Public_32');
78 %flag1=0;
79 %while flag1<80
80
81 picture = camera.snapshot;
82 picture = imresize(picture,[227,227]);
83
84 label = classify(net,picture);
85
86 if label=='yellow'
87     sys=1;
88 elseif label=='red'
89     sys=2;
90 elseif label=='white'
91     sys=0;
92 elseif label=='green'
93     sys=0;
94 end
95
96 image(picture);
97 title(char(label),sys);
98 %flag1=flag1+1;
99 %end
100
101 %sys表示输出, 即y
102 function sys=mdlGetTimeOfNextVarHit(t,x,u)
103 %计算下一个采样时间
104 %仅在系统是变采样时间系统时调用
105 sampleTime = 1;
106 %设置下一次采样时间是在1s以后
107 sys = t + sampleTime;
108 %sys表示下一个采样时间点
109 function sys=mdlTerminate(t,x,u)
110 %仿真结束时要调用的回调函数
111 %在仿真结束时, 可以在此完成仿真结束所需的必要工作
112 sys = [];
```

图表 16 s 函数

程序主要包含以下几个部分: 1、s 函数的基本框架包括初始化、回调函数、更新回调函数、输出回调函数、下一次采样时间 2、因为我们只是通过 s 函数输出图像辨认结果, 所以我们只对输出回调函数部分进行编辑, 像上文提到的图像识别程序一样编写: 调用摄像头、训练好的网络,

采集图像，判断，输出结果。

效果如下：



图表 17 simulink 仿真效果

我们可以看到，当摄像头看到绿色、白色时输出为 0，当看到黄色时输出为 1，看到红色时输出为 2。这里得到的数据可以直接输出给队友负责的打击部分，进行判断打击。

3、项目遇到的问题及解决方案

1) matlab 上的迁移学习

matlab 上的迁移学习我参考了 matlab 官方网站上的视频，其中详尽的介绍了如何利用 matlab 和 alexnet 网络进行迁移学习。

<https://ww2.mathworks.cn/videos/deep-learning-with-matlab-transfer-learning-in-10-lines-of-matlab-code-1487714838381.html>

然后我才顺利地完成了迁移学习

2) Simulink 中 s 函数的使用

这是我第一次结束 simulink 中的 s 函数，我一开始只觉得 s 函数复杂，繁琐，上网看了很多的教程和介绍，最后在 csdn 上找到了一篇详细介绍 s 函数的文章，帮助我更好理解了 s 函数的使用，在有一定理解之后，我感觉 s 函数本身实际上是面向用户的系统搭建的，并不是特别使用于此次实验，所以过程会有些繁琐，同时我还感受到了 level-1matlab 的局限性，他的编译速度很慢，会对 simulink 仿真的速度有很大的影响，在今后希望尝试

用 c 语言编译 level-2 S 函数，提高使用 s 函数 simulink 仿真的效率。

https://blog.csdn.net/weixin_42650162/article/details/90488610

4、项目成果

- 1、图像识别模块利用 alexnet 的迁移学习实现了稳定的识别 4 种颜色颜色，并且不受光线的干扰。
- 2、转停模块利用串级 PID 实现了稳定转停，使得转停的过程平稳顺滑。系统传输模块（转盘）转停稳态误差不大于 2%，超调量不大于 5%。
- 3、分拣模块实现了精准稳定的分拣。一分钟可以至少分拣 20 个。
- 4、实现了分拣模块和转停模块的通信，只在停状态下进行分拣操作。

四、收获和建议

收获：

- 1、团队协作能力的重要性；本期实验帮助我认识到了团队协作合作能力的重要性，也让我意识到了团队的协同合作可以依托于一些云合作平台比如 github，或者通过板栗看板来监督项目的进度，实现团队合作的高效化。
- 2、matlab 深度学习库的了解；本次实验中我主要使用了 matlab 中 alexnet 这个库，我发现利用 matlab 进行深度学习虽然在现在这个时间节点已然有些过时，但是通过上面一些简单的库和函数可以帮助一个初学者快速了解一些深度学习的基本概念，是初学者入门不错的一个选择，我也打算在假期学习一些关于深度学习的基础课程，希望能加深理解。
- 3、simulink s 函数初次见面；这是我第一次使用 s 函数，之前的实验中我是用 matlab function 实现了转停条件的判断，但是今天 matlab function 已经满足不了我的需求，我开始使用了功能更加强大的 s 函数，虽然学习使用方法的过程很痛苦，在中文互联网中很难找到一个系统的学习方法，但是学习后掌握使用方法后，那种成就感令人难忘。
- 4、检索信息能力的重要性；在查阅资料的过程中我真切的认识到搜索资料能力的关键，关于如何在 simulink 中使用 s 函数这个问题，我一开始都是搜索“如何在 simulink 中使用.m 的结果”，

但是得到的都是如何保存.m中的运行结果,然后输出到simulink中,实际上不是在simulink仿真。然后我就换了一种搜索措辞”如何在simulink中调用.m文件”,然后关于simulink中用户可自定义的库的内容开始映入眼帘,点开介绍文章看下来,能满足我要求的只有s函数了。于是我才真正走上正轨,开始学习s函数的使用。

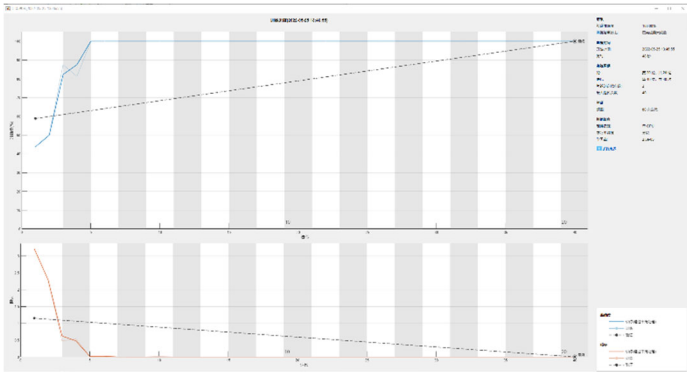

5、保持好奇,享受学习的过程;本次实验的过程中我采用了深度学习的方法完成任务,这并不是因为对于这个实验采用深度学习有绝对的优势,而是由于对于新技术和新知识想要了解的情感,能在实验过程中多掌握一些书本外的新知,使得实验本身变成了一件充满乐趣的事。

建议:

- 1、不同平台;实验不一定选择matlab,可以选择比如单片机,openmv之类的方案来实现这个实验,同时可以做到节约成本。
- 2、压缩课时,开放实验室,自由选择实验时间;自控实验给我的感觉是时间很长,我认为老师可以早早将PPT下放给同学们自学,在课堂上精讲难点。同时开放实验室,动态配置实验时间,这样可以节约老师同学们的时间,提高授课效率。

实验记录单

日期		5.23
组别 (班级、小组成员)		06012002-1-于世然、郑明松、刘雅萱
分工		于世然（图像处理）、郑明松（转盘）、刘雅萱（分拣）
实验记录	实验题目	自动控制理论课程设计——垃圾分拣
	时间	内 容
	5.23 13:30	<p>通过 matlab 中 alexnet 神经网络进行迁移学习，同时利用 matlab 相机函数实现实时对于白绿红黄四种颜色的辨认。</p> <p>利用 alexnet 进行这次实验的想法来自于在 bilibili 搜索如何在 matlab 中进行图像识别是跳出的一个视频：https://ww2.mathworks.cn/videos/deep-learning-in-11-lines-of-matlab-code-1481229977318.html在这个视频中 matlab 的工作人员用一个简单的程序实现了利用 alexnet 进行图像识别分类物品的功能，于是我在 matlab 网站中继续搜索关于深度学习的内容，最终决定利用 alexnet 的迁移学习来实现高效稳定的颜色识别。</p> <p>我遇到的困难有两个，第一是如何进行迁移学习和迁移学习的可视化，在这里我参考了 bilibili 一个视频中博主分享的程序：https://www.bilibili.com/video/BV1Kf4y1m7no?spm_id_from=333.337.search-card.all.click，按着他的指导，编写了迁移学</p>

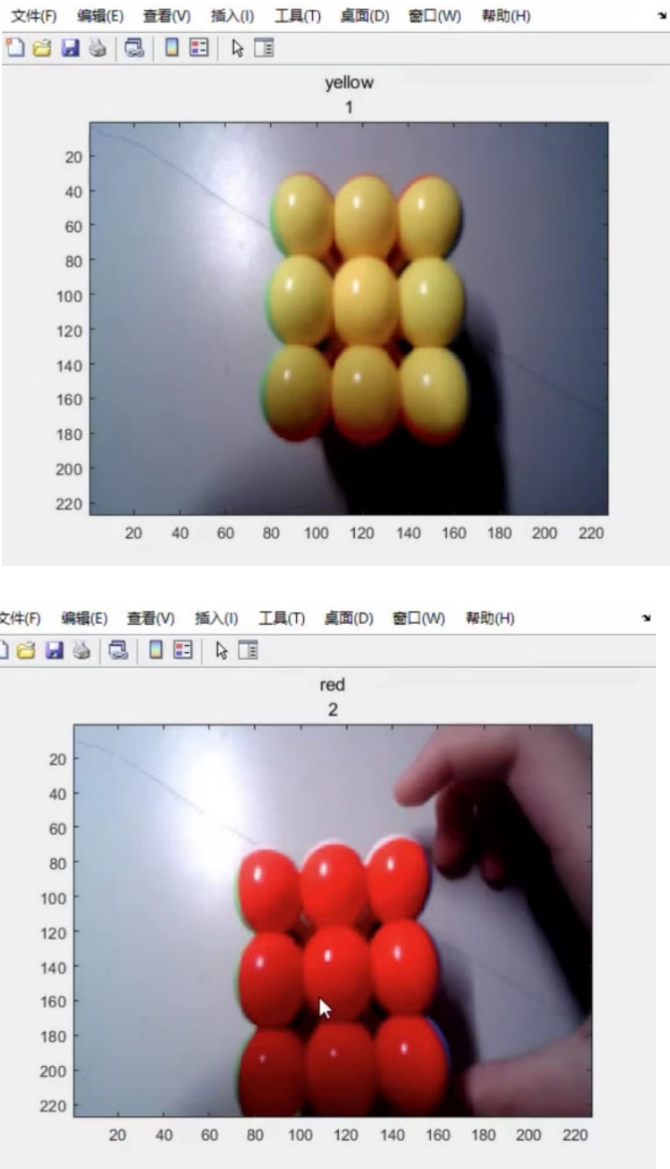
	<p>习的程序，实现了学习过程的可视化。效果如下：</p>  <p>第二个遇到的困难是数据集的收集和处理。虽然迁移学习需要的样本量并不多，但仍然需要几十张不同颜色的照片，在这里需要感谢我的舍友借给我他的彩色魔方，这实在是再好不过的样本了。接着我对着魔方各个角度各种光照拍了一些照片。接下来进行图片的处理，因为我们需要的是 227X227 的图片进行训练，一张一张用 ps 更改像素费时费力，可以利用 matlab 进行编程实现图片预处理的自动化。在这里我参考了 b 站视频：https://www.bilibili.com/video/BV1qb411W7jk?spm_id_from=333.337.search-card.all.click, 编写 matlab 程序实现了一个文件夹的照片预处理的功能。</p> 
备注	明天开始着手编写图像处理的文件。

说明：

- (1) 根据实验过程、测试数据、测试结果、出错环节及纠错过程等内容。重点记录实验数据、遇到的问题和分析、解决方法等实验实施过程性数据。
- (2) 备注填写存在问题及修改意见。

实 验 记 录 单

日 期		5. 24
组 别 (班级、小组成员)		06012002-1-于世然、郑明松、刘雅萱
分 工		于世然（图像处理）、郑明松（转盘）、刘雅萱（分拣）
实 验 记 录	实验题目	自动控制理论课程设计——垃圾分拣
	时 间	内 容
	5. 24 13: 30	<p>今天开始着手编写图像识别主程序。</p> <p>首先先要下载 webcam 的库</p> <p>然后仿照 matlab 官网中的深度学习： https://ww2.mathworks.cn/solutions/deep-learning/tutorials-examples.html 的例程编写适用于本次实验任务的程序，一定要做到：调用摄像头、载入网络进行分类，然后最后得到标志位进行输出，方便负责其他工作的队友进行判断。</p> <p>最后实现了很好的效果，帧率也很快：</p>

		
	备注	<p>下一步就是把.m 文件中实现的功能如何融合进入simulink 中，给队友的分拣模块输出正确的信号。</p>

说明：

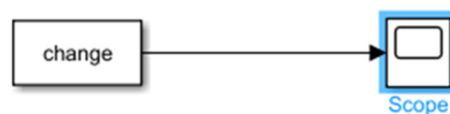
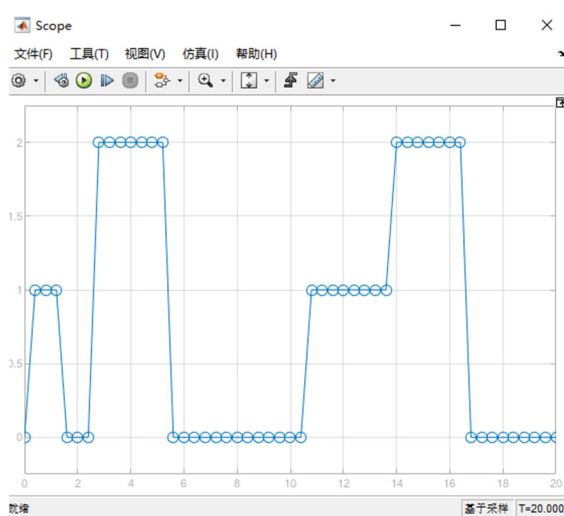
- (1) 根据实验过程、测试数据、测试结果、出错环节及纠错过程等内容。重点记录实验数据、遇到的问题和分析、解决方法等实验实施过程性数据。
- (2) 备注填写存在问题及修改意见。

实验记录单

日期		5.24
组别 (班级、小组成员)		06012002-1-于世然、郑明松、刘雅萱
分工		于世然（图像处理）、郑明松（转盘）、刘雅萱（分拣）
实验记录	实验题目	自动控制理论课程设计——垃圾分拣
	时间	内 容
	5.24 10:30	<p>下午写好了图像识别代码,网上开始学习如何讲.m文件的功能整合到 simulink 中。再查阅资料的过程中我真切的认识到搜索资料能力的关键,我一开始都是搜索“如何在 simulink 中使用.m的结果”,但是得到的都是如何保存.m中的运行结果,然后输出到 simulink 中,实际上不是在 simulink 仿真。然后我就换了一种搜索措辞“如何在 simulink 中调用.m 文件”,然后关于 simulink 中用户可自定义的库的内容开始映入眼帘,点开介绍文章看下来,能满足我要求的只有 s 函数了。于是我才真正走上正轨,开始学习 s 函数的使用。学习的过程也是十分的艰辛,因为中文互联网中鲜有使用 s 函数实现调用神经网络图像识别的,我只能全面了解 s 函数各个部分是什么意思,如何使用,然后再考虑怎么把我的.m文件整合进 s 函数。</p> <p>https://zhuanlan.zhihu.com/p/393407384</p> <p>https://zhuanlan.zhihu.com/p/23301801</p> <p>https://blog.csdn.net/weixin_42650162/article/details/9</p>

0488610

这几篇博客帮助我大致了解了 LEVEL1 的 s 函数的基本使用方法，了解了如何初始化，如何输出，怎么定义变量等基本过程。但是在有一定理解之后，我感觉 s 函数本身实际上是面向用户的控制系统搭建的需求，并不是特别适用于此次实验，所以过程会有些繁琐，同时我还感受到了 level-1matlab 的局限性，他的编译速度很慢，会对 simulink 仿真的速度有很大的影响，在今后希望尝试用 c 语言编译 level-2 S 函数，提高使用 s 函数 simulink 仿真的效率。



最终实验的效果还是很喜人的，同时这也是对我的一种锻炼，怎么样自学一个小技术，如何从中文互联网中准确找到自己需要的内容，然后再对于需要的功能和技术再官方文档或者官方论坛中找到自己需要的答案。这对于我以后成长为一个成熟的工程师或者一个科研工作者都很重要。

	备注	图像处理的部分基本上已经完成，接下来可以通过 UDP 模块尝试和对于进行通信，或者输出我的仿真数据给队友进行实验
--	----	--

说明：

- (1) 根据实验过程、测试数据、测试结果、出错环节及纠错过程等内容。重点记录实验数据、遇到的问题和分析、解决方法等实验实施过程性数据。
- (2) 备注填写存在问题及修改意见。