**自动控制理论课程设计——垃圾分拣系统**

1. **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。

1. **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）

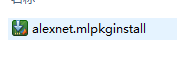
**三、项目完成情况**

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.

1. **项目设计思路和实施过程**
2. **matlab alexnet神经网络进行迁移学习**

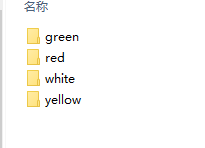
**<1>安装alexnet工具包和camera工具包**



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。



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**

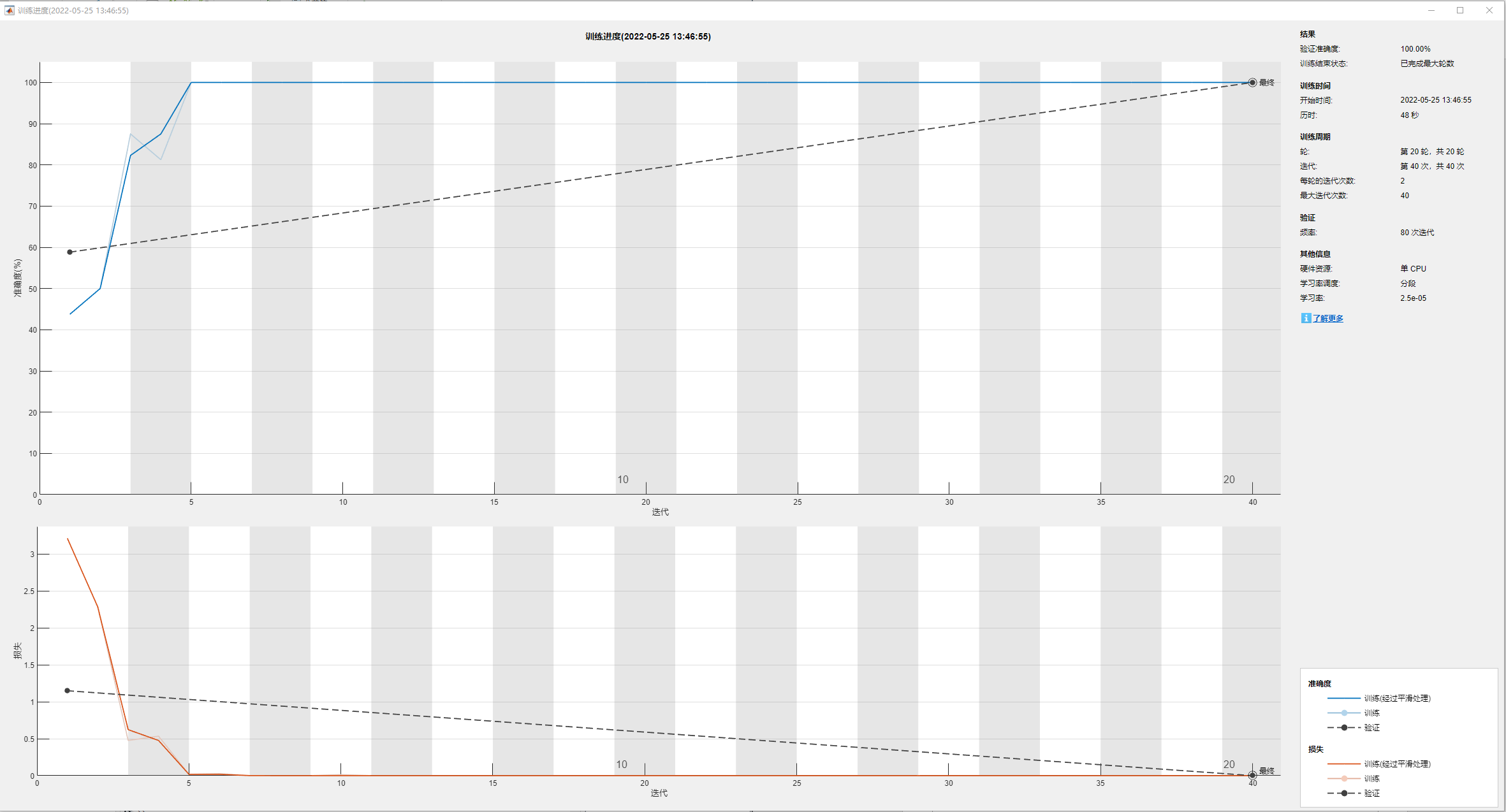
我们可以采用matlab成熟的深度学习函数进行迁移学习。程序如下



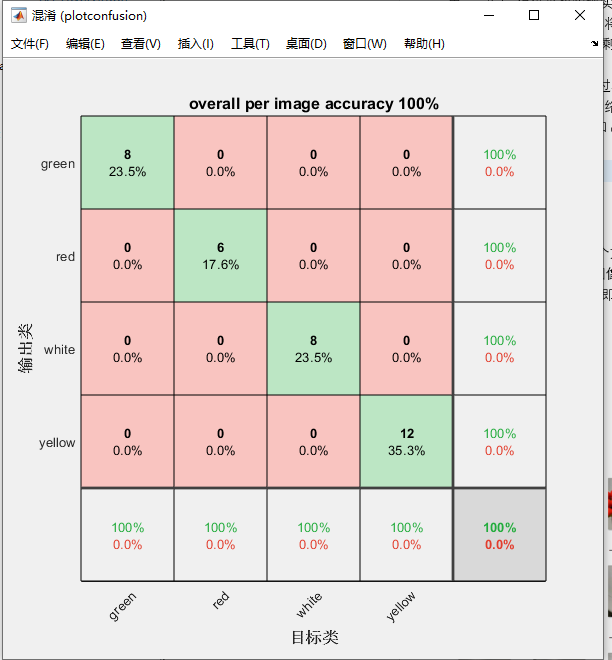


图表 7 迁移训练主函数

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



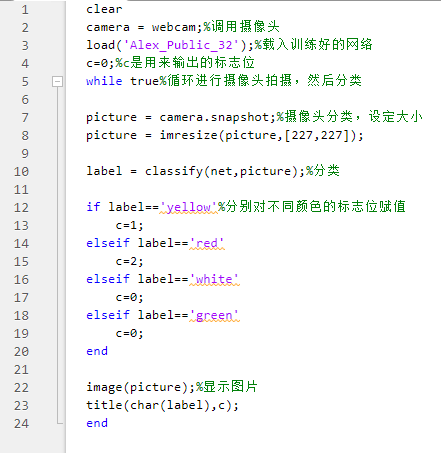
图表 9 学习准确度可视化

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。

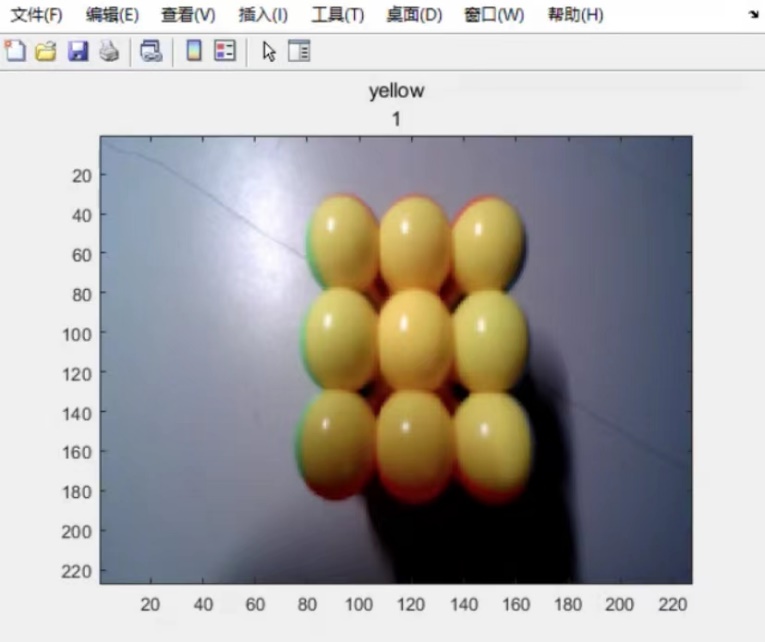
程序如下：



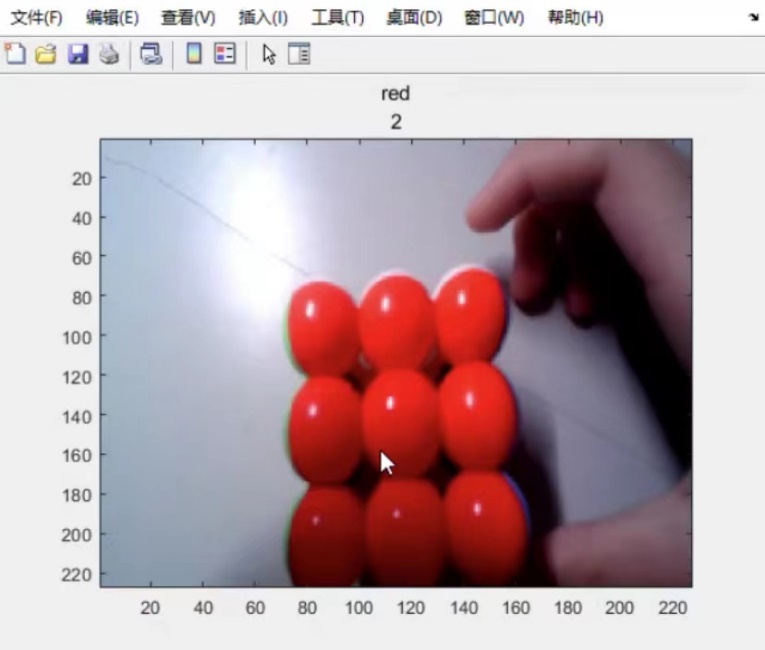
图表 10 图像识别主程序

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。

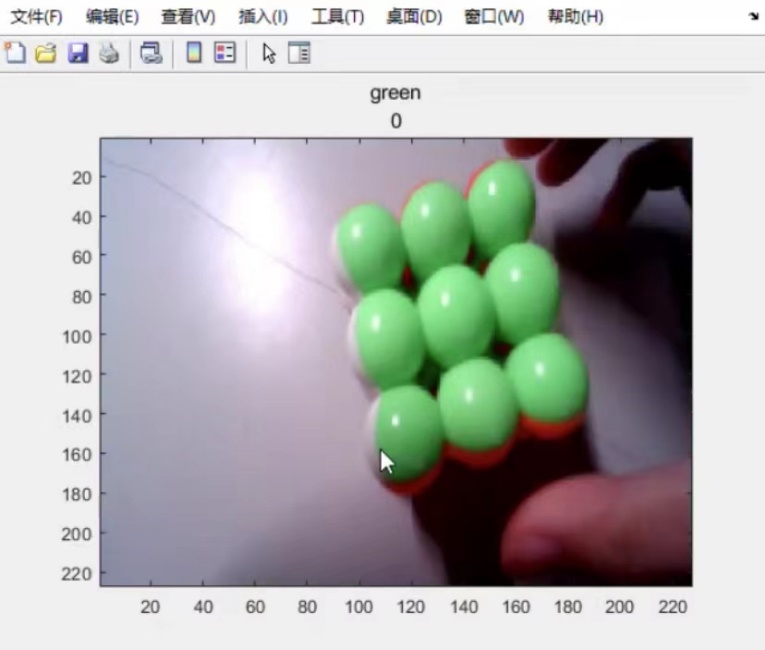
效果如下：



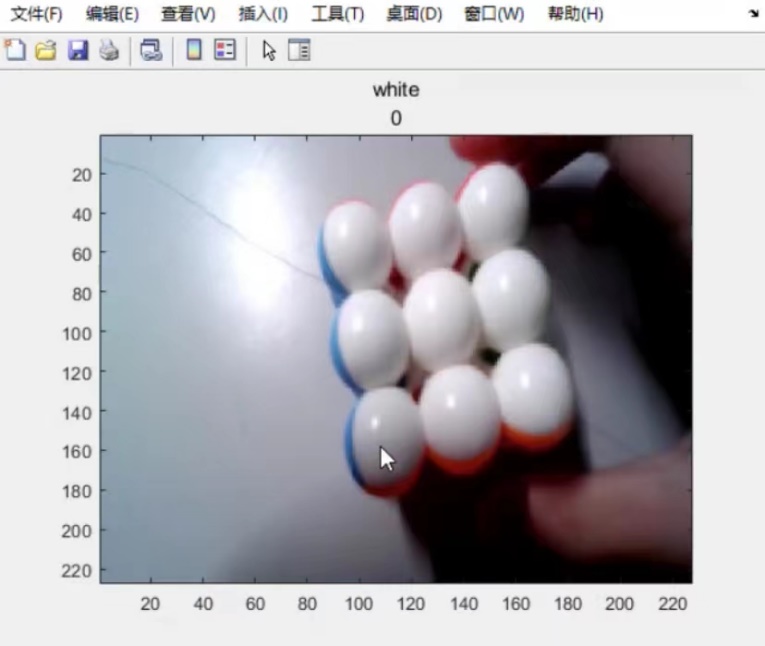
图表 11 黄色



图表 12 红色



图表 13 绿色



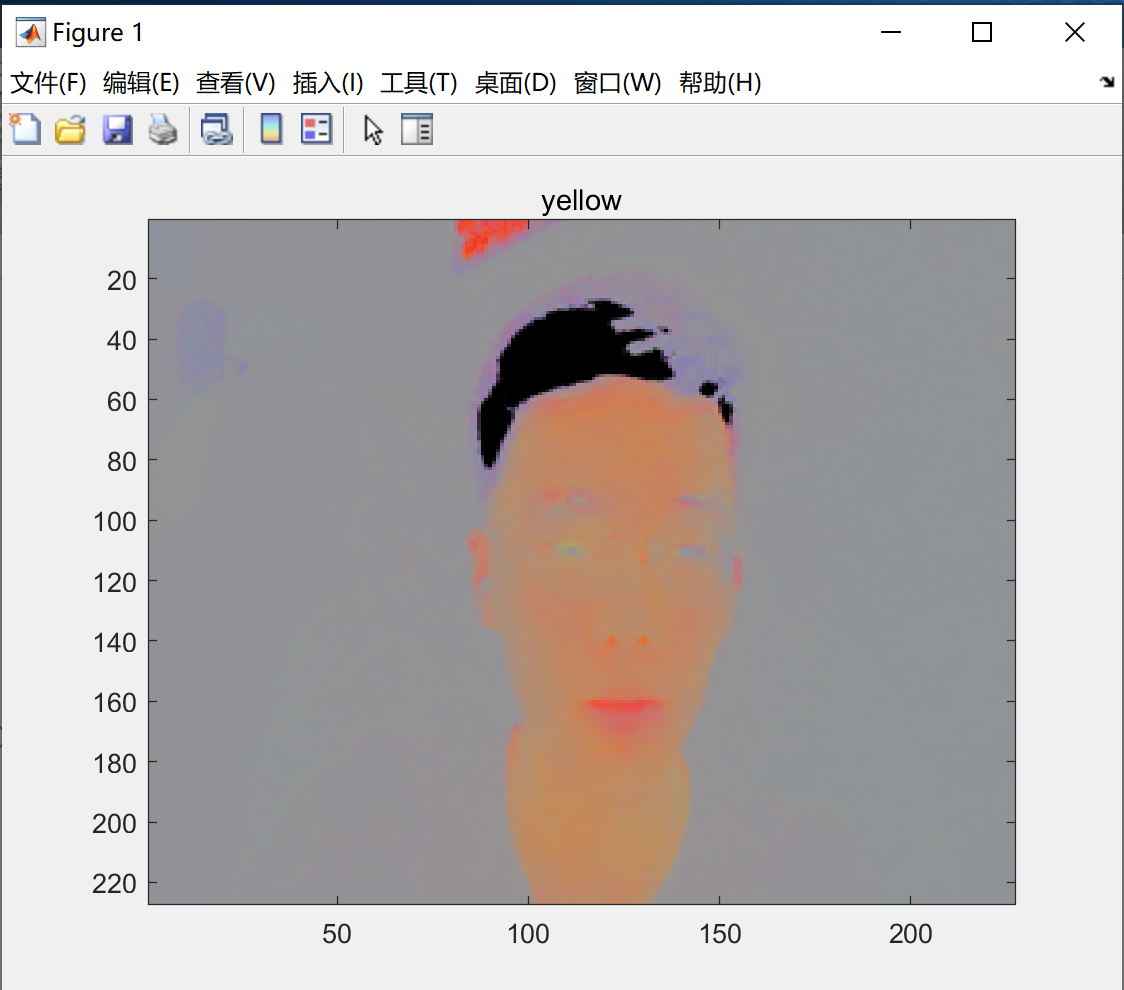
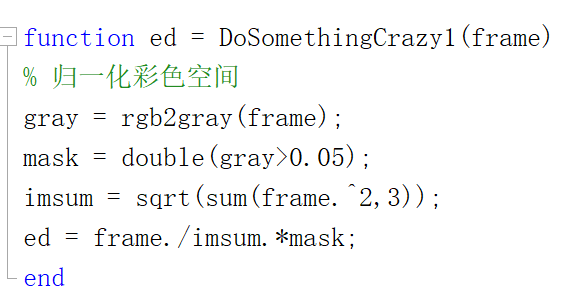
图表 14 白色

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)；g = G / (R+G+B)；b = B / (R+G+B)；



* 1. **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。**

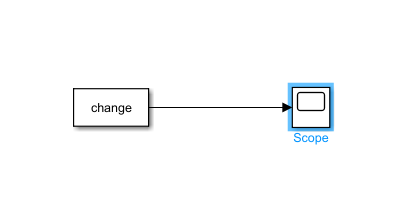
这一部分的任务主要分以下几步：

**<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：



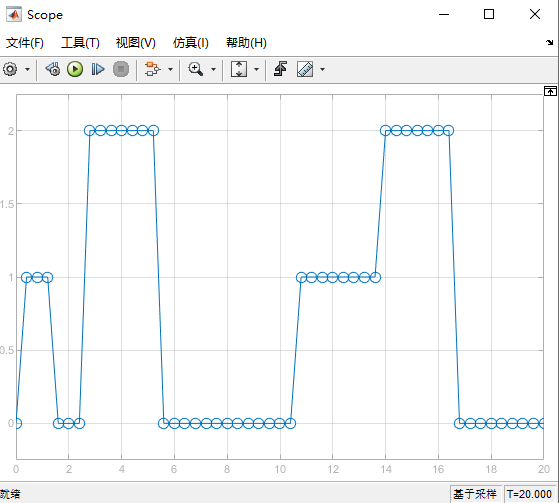




图表 16 s函数

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。

1. **项目遇到的问题及解决方案**
2. **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

1. **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>

1. **项目成果**
2. 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。
3. 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% 。
4. 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。
5. 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、团队协同能力的重要性；本期实验帮助我认识到了团队协同合作能力的重要性，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”，但是的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。