指导教师：宋晓威 于树强 参赛队员：朱 镇 段金宇 李姝影

魔方机器人设计报告

天津大学代表队

目录

[**一、系统概述** 3](#_Toc80654406)

[1.1 项目背景 3](#_Toc80654407)

[1.2 需求分析 3](#_Toc80654408)

[1.2.1颜色识别 3](#_Toc80654409)

[1.2.2 魔方还原算法 4](#_Toc80654410)

[1.2.3控制算法及电路系统 4](#_Toc80654411)

[1.2.4执行机构 4](#_Toc80654412)

[1.3 项目预期 4](#_Toc80654413)

[**二、作品简介** 4](#_Toc80654414)

[2.1 总体构成 4](#_Toc80654415)

[2.2 机械结构 5](#_Toc80654416)

[2.3 算法设计 6](#_Toc80654417)

[三、硬件结构 8](#_Toc80654418)

[3.1 输入元件 8](#_Toc80654419)

[3.2 控制系统 8](#_Toc80654420)

[3.3 执行系统 11](#_Toc80654421)

[3.3.1 电磁铁 11](#_Toc80654422)

[3.3.2 机械臂 11](#_Toc80654423)

[3.3.3 机械手爪 12](#_Toc80654424)

[3.3.4 步进电机 13](#_Toc80654425)

[3.3.5 电滑环 14](#_Toc80654426)

[3.4 动力系统 15](#_Toc80654427)

[四、控制算法及软件实现 16](#_Toc80654428)

[4.1 颜色识别 16](#_Toc80654429)

[4.1.1图象腐蚀 17](#_Toc80654430)

[4.2 解魔方 20](#_Toc80654431)

[4.3 魔方控制 22](#_Toc80654432)

[五、总结 24](#_Toc80654433)

[5.1 优势 24](#_Toc80654434)

[5.2 可以改进的方面 25](#_Toc80654435)

[六、实物图 25](#_Toc80654436)

# **一、系统概述**

## 1.1 项目背景

中国高校智能机器人创意大赛是一项有广泛影响的全国性的学科竞赛。2017年首届中国高校智能机器人创意大赛由中国高等教育学会、教育部工程图学课程教学指导委员会、中国高校智能机器人创意大赛组委会共同主办，浙江大学机器人研究院、中国高等教育学会工程教育专业委员会承办，决赛由浙江省余姚市人民政府承办。之后该大赛每年举办一次，至今已经连续举办3届，大赛以“更好、更快、更强”为主题，既培养学生提问题的能力，也培养学生解决问题的能力，高校参与该竞赛的积极高、参与面广，本竞赛已列入中国高等教育学会发布的《2020年全国普通高校学科竞赛排行榜》。

为进一步推进高校学生创新意识和创造能力培养，激励广大学生踊跃参加课外科技实践，有效推动新工科人才培养，经中国高校智能机器人创意大赛组委会研究，决定启动第四届中国高校智能机器人创意大赛工作。

第四届中国高校智能机器人创意大赛设三个主题和一个邀请赛，鼓励学生根据自己的专业特长和兴趣爱好选择其一参赛。其中，主题二（创意竞技）为“魔方机器人——挑战更快”：参照人类魔方竞速规则，设计制作魔方机器人，综合运用机械、电子、信息和自然科学知识，实现比人“计算”更快、“翻动”更加灵活迅速的目标。

## 1.2 需求分析

在研读历届魔方机器人决赛程序和裁判规则的基础上，我们发现，比赛对还原魔方的速度、准确性，机械结构的稳定性均提出了一定的要求，经过分析，我们认为魔方机器人至少需要具备以下功能：

## 1.2.1颜色识别

魔方机器人需要在一定的光照条件下（可加人工光源），识别魔方各色块的颜色，从而获得RGB信息。颜色识别需要对每个方块的颜色精确把握。近年来ｏｐｅｎｃｖ的发展使得此项成为现实。

1.2.2 魔方还原算法

获取魔方各面的颜色数据（包括颜色信息和位置／顺序信息）后，魔方机器人通过算法将数据转化为特定形式作为输入量求解还原步骤，还需要一个辅助程序优化理论还原步骤，进一步简化得到实际还原步骤，然后输出可执行步骤给执行系统，从而还原魔方，这要求还原魔方算法的简洁高效；

## 1.2.3控制算法及电路系统

得到还原魔方算法输出数据后，需要一套控制算法及相应配套的电路系统将还原步骤传递给执行系统，而控制系统不宜复杂，应保证稳定高效，能快速响应。

## 1.2.4执行机构

执行机构应满足精度要求，保证手爪对中性以及和魔方的匹配程度，确保在还原魔方的过程中，将误差控制在容错范围内，尽量避免误差积累带来的影响。在控制系统的控制下，执行机构需要稳定且准确地完成给定动作，为防止魔方在还原过程中受力移位、脱落，执行系统及主要机械结构对相关构型及材料具有一定的受拉、受压等强度要求。

## 1.3 项目预期

在参考往届众多优秀作品的基础上，我们将针对性地对视觉识别算法及机械结构进行大幅改进，预计可使颜色识别更加准确快速，机械爪结构更加稳定，制作及装配精度更高，识别准确率预期在95%以上，还原魔方速度预计在20s内，总时长预计在25s内，还原正确率预计在90%以上。

# **二、作品简介**

## 2.1 总体构成

本组的魔方机器人总体使用了Arduino 1块，电磁铁2个，PCB板1块，MOS管2个，220V/AC-24V/DC开关电源1个，220V/AC-12V/DC开关电源1个，720P摄像头4个，采集卡1块。电机及支架和驱动器2组，电滑环及碳刷2组，6mm光轴2根，锁紧环4个，光轴支承座4个，3D打印件手爪2个，LED小灯4条、铝型材、轴承、螺母、连接件、角码、导线等若干。

## 2.2 机械结构

二轴魔方机器人结构示意图如图2.1所示，图中，魔方机器人的外部框架均为铝型材。

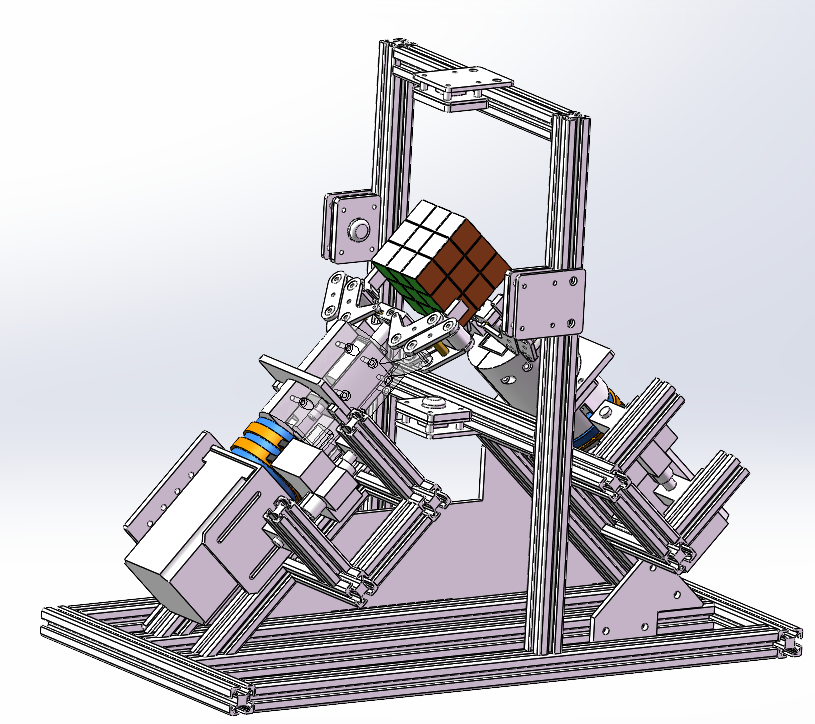


图2.1 魔方机器人solidworks模型

魔方机器人以中间门形框架所在的平面为基准面左右对称，两侧的三角支架上装有闭环步进电机，通过联轴器与机械臂相连。联轴器外固定有电滑环，配合电刷，解决电磁铁的绕线问题。手指通过手指连接件与机械臂相连，其夹放动作通过机械臂内部的推拉式电磁铁进行控制。闭环步进电机、机械臂、滑环的回转轴都位于同一轴线上，且与另一侧轴线呈90°夹角，魔方也位于该角处。机器人的图像采集通过固定在门型框架上的四个摄像头实现。

在功能分工上，机器人整体按照可分为动力系统、控制系统和执行系统。两臂的两个步进电机由驱动板供电，控制手臂的旋转；通过电磁铁控制手爪的开合。使用Arduino通过串口接收计算机发来的还原步骤，经过处理加工计算后得到魔方还原的指令，将计算机的每个指令翻译为一组信号输出。通过Arduino，实现对电机驱动板和电磁铁的控制，利用PCB简化连线，利用MOS管代替继电器。气泵控制手爪开合，电机驱动板驱动电机控制手臂旋转，两者配合完成一次完整的拧转动作。

## 2.3 算法设计

在算法设计上，本组魔方机器人的主算法由图像识别位置校准程序、图像识别程序、图像分割显示程序、图像信息转换程序、还原算法、坐标转换程序和控制程序七个子程序组成。



图2.2 程序框图

# 三、硬件结构

## 3.1 输入元件



图3.1 摄像头实物图

本魔方机器人使用4个平头广角720P摄像头作为色彩信息输入元件。

## 3.2 控制系统

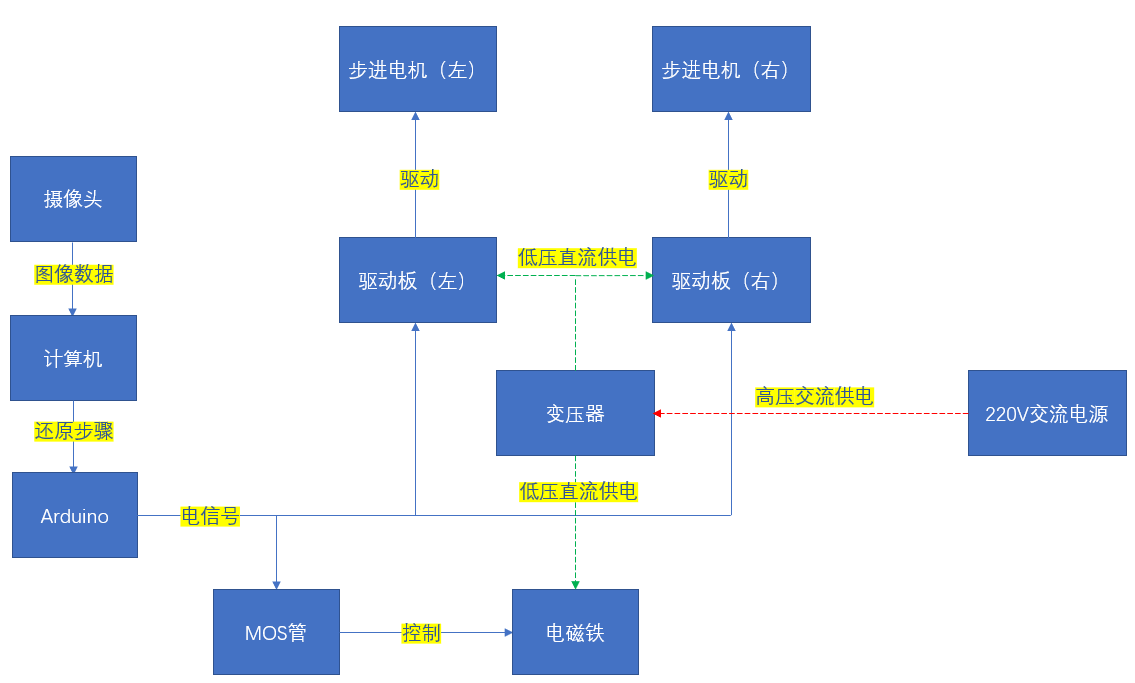


图3.2 控制系统框图

采用计算机作为本魔方机器人的“大脑”，处理摄像头输入的数据，并求解还原魔方的步骤，输出给Arduino；使用Arduino作为机器人的“脊髓”，通过串口接收计算机发来的还原步骤，经过处理加工计算后得到魔方还原的指令，并将计算机的每个指令翻译为一组信号输出，实现对电机驱动板的控制，控制手臂旋转，通过MOS管控制电磁铁，以此控制手爪开合，两者配合完成一次完整的拧转动作。

MOS管采用A0D4184型号，击穿电压电流分别为40V、50A，自带光耦隔离，控制信号和被控设备电源隔离，大大提高刚干扰，可兼容Arduino控制板，高电平启动，低电平停止，PWM调速。



图3.3 MOS管实物图

本组自行绘制一版PCB，提前将铜线布好，将MOS管，Arduino，24V，地线以及电机使能、步进、旋转方向集成在板子上，方便接线，机器的封装性更好

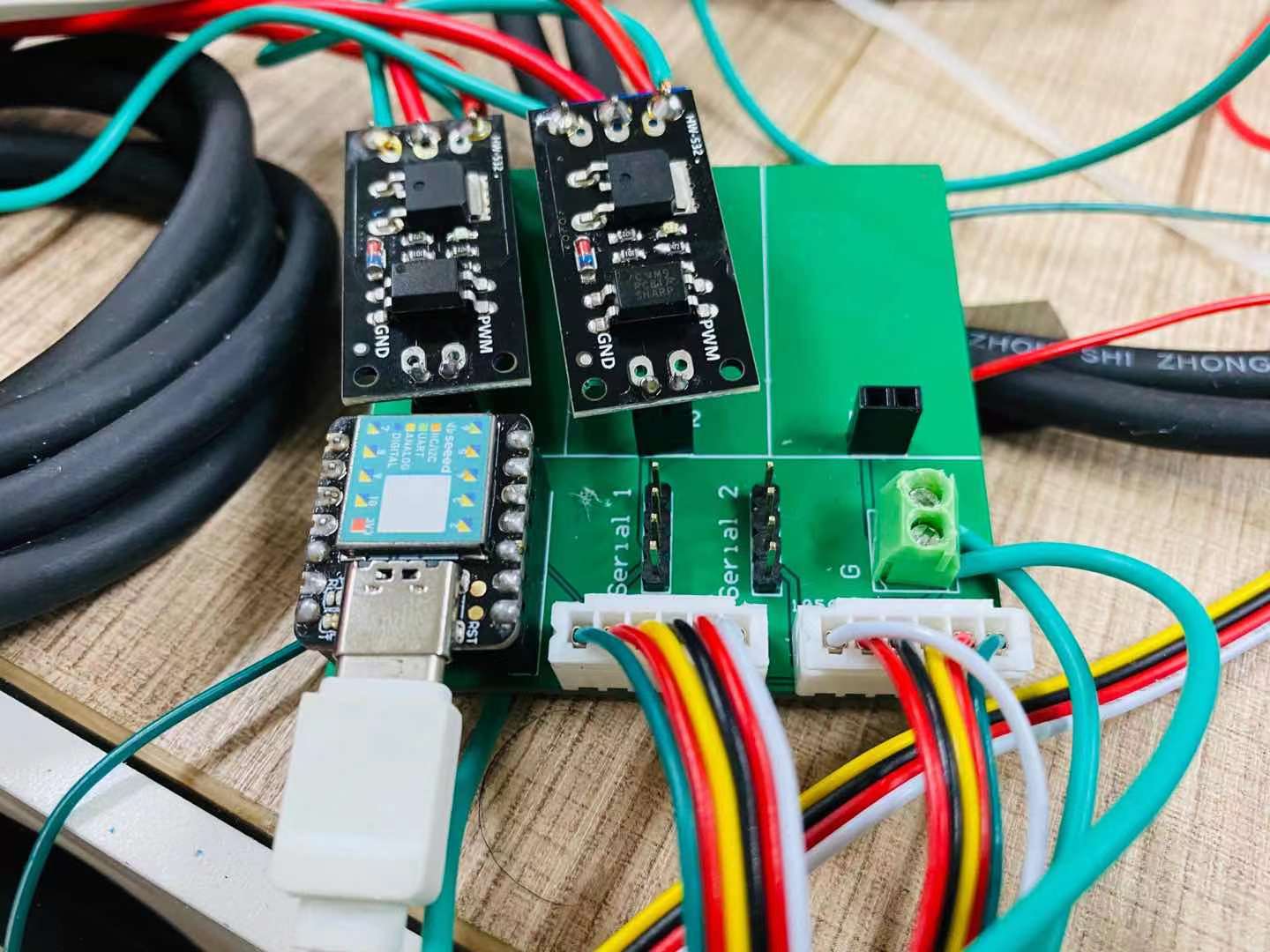


图3.4 MOS管和Arduino集成在PCB上的实物图

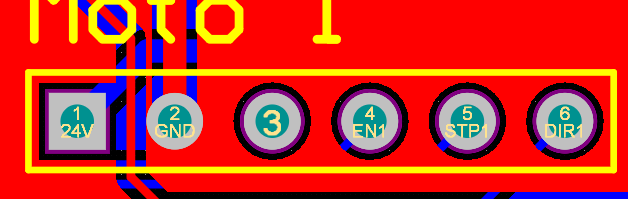


图3.5 PCB端子绘制图

## 3.3 执行系统

## 3.3.1 电磁铁



图3.6 电磁铁实物图

本组采用推拉式电磁铁，型号为伊莱科的EML-1564B，工作电压24V，工作电流2A，初始吸力3.5N，保持吸力65N，重量352g，最大可调行程20mm。通过电磁铁推动手爪结构，带动手指张开闭合，从而实现抓紧和松开两种状态。并且电磁铁响应迅速，与气缸相比执行速度更快，还原魔方的速度更快。

## 3.3.2 机械臂

机械臂内部结构如图3.7所示：

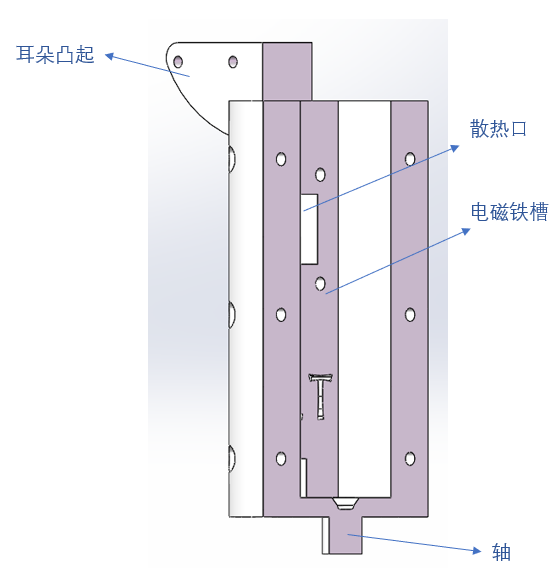


图3.7 机械臂内部结构

机械臂采用3D打印技术制作，内部有电磁铁槽，用来安装推拉式电磁铁。下端的圆柱状凸起为机械臂的轴，利用联轴器与电机轴固定。手指连接件与机械臂上端的耳状凸起相连，牵引手指的夹放动作。在一边的机械臂的轴上还设计有一个平键，用以保证机械臂和联轴器的连接，如图3.8所示。

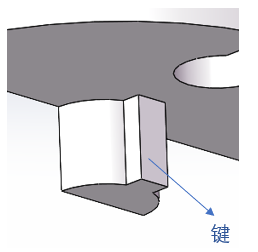


图3.8 轴上的键

## 3.3.3 机械手爪

本魔方机器人每一手臂末端有两个手指，均由3D打印材料制作，通过I型手指连接件和L型手指连接件分别与机械臂的耳状凸起及手掌连接。手掌与电磁铁的铁芯连接。机械手爪的结构图如图3.9所示：

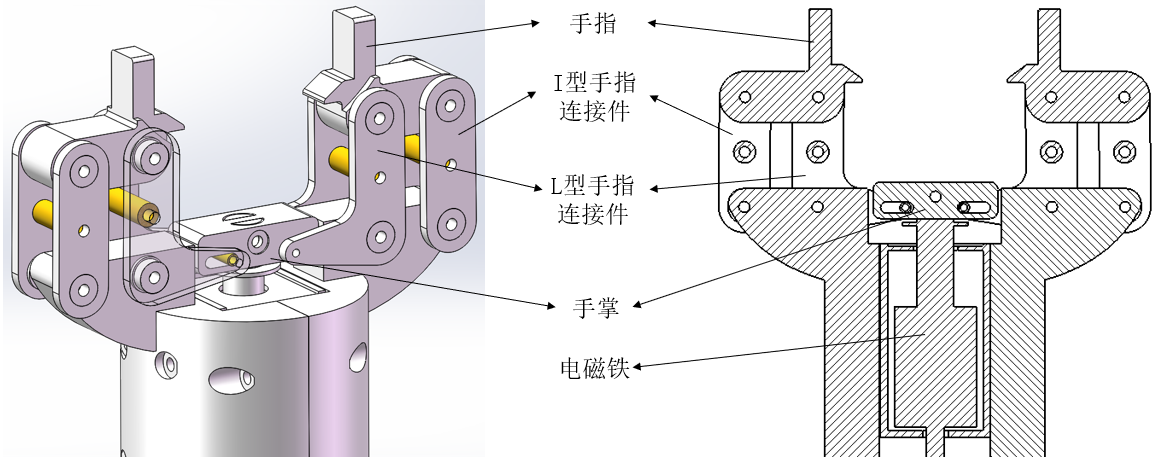


图3.9机械手爪结构示意图

当电磁铁通过电流时铁芯收缩，电流断开时通过弹簧弹出铁芯，从而带动手掌上下运动，推动L型手指连接件、I型手指连接件运动，实现手指夹放动作。

## 3.3.4 步进电机



图3.10 步进电机实物图

电机采用57闭环步进电机，型号为57HSZ2N，输出力矩2.2N·m，机身长度99mm，轴径8mm，出轴方式为单出轴，电流为4A，转速1000rpm以上扭矩下降，最高2000rpm。

电机轴与联轴器相连，联轴器又与机械臂想连，从而带动机械臂转动。

## 3.3.5 电滑环



图3.11 电滑环和电刷实物图

电磁铁的导线通过电滑环和电刷来连接电源，使得电磁铁旋转时，导线不会缠绕起来。电滑环选用人字形刷架总成二组SR20\*50\*34-2。

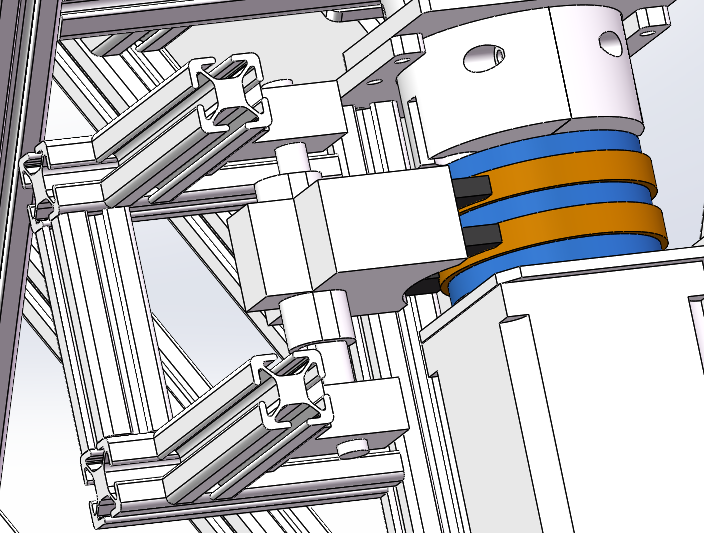


图3.12 电滑环和电刷的安装位置

3.4 动力系统



图3.11 变压器实物图

电源方面选择220V家用交流电转24V直流电，所用变压器为S-500W-24V变压器，长215mm，宽115mm，高50mm，输入电压范围为110V~220V/AC，输出电压为24V/DC，最大输出电流为20A，最大功率500W。

# 四、控制算法及软件实现

软件算法是魔方机器人核心，其大致可分为三个部分，颜色识别、解魔方、控制机构运动，其中颜色识别和解魔方我们使用Python语言使其运行在个人电脑上，控制机构部分我们使用Arduino来控制电机的转动，继电器的开闭等动作，以下将对各部分进行详述。

4.1 颜色识别

对于颜色识别的算法，我们首先需要进行对魔方上各个色块的颜色进行采集。我们想到的有两种方法：第一个是对魔方进行边框的识别，通过算法筛选出其9个色块，这样就可以知道我们所需要颜色的位置。这种方法虽然非常方便，但在实际的运用中效果并不是特别好。由于机械臂上会有各种的机构挡住魔方一定部分，使边缘识别的成功了大幅减小，非常不稳定，而在比赛中我们需要的是其能够稳定得运行。所以我们将这种方法运用到后面其他的地方。



图4.1 识别边框并提取、识别颜色

由于魔方在机械臂上的位置是固定的，所以便可以取第二种方法，对摄像头图像的进行定点识别。这种方法的缺点是在前期进行调试时非常麻烦，需要设各点坐标，同时需要魔方和摄像头的位置不能有一丝轻微的移动，对于魔方放的位置也提出了很大的要求，但在经过前期调试后这种方法可以非常稳定运行，所以我们使用了第二种定点识别的方法。

4.1.1图象腐蚀

但是在取得我们想要的色块之后，由于反光或者是摄像机的噪点，使得用于识别的数据十分杂乱，于是我们就使用了图形学中的“图象腐蚀”操作对数据集以及进入识别的图象进行了处理。

如下图就是图象腐蚀算法处理的结果，可以看到结果很好。



图4.2 未经腐蚀的图象

图4.3 腐蚀结束的图象

如上图所示，经过腐蚀后的图象以及完全消去了反光带来的影响。符合我们对数据的要求。

在取得我们所需要的颜色值后我们就需要通过算法来判断该颜色值是什么，首先我们尝试了对RGB数值设置阈值来判断，但这种方法需要人工来调阈值非常麻烦，同时其准确率非常低，不同环境下识别颜色的值相差非常大。所以我们又采用卷积神经网络CNN来进行颜色的识别，只需要采用第一种识别边框的方法通过摄像头采集还原好的魔方的六个面，来非常方便得获取颜色值，再来训练模型，在这种方法下训练的模型在测试训练集时可以达到基本100%的准确度。

图4.4 CNN的结构

但其在实际运用的时候却在上下摄像头的识别有非常大的失误，由于上下识别需要对两个对角面进行识别。所以我们对其训练集进行了补充，添加了许多不同光照条件下的颜色数据，这种方法后他的准确度大幅提升。但是为了以防万一，我们同时采用了集成学习的机器学习方法训练了另一套识别模式。如图就是我们采用的集成学习的结构。

图4.5　集成学习模型

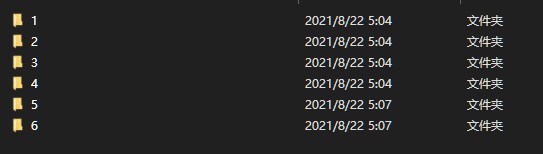


图4.6 通过机器学习所建立数据的数据库

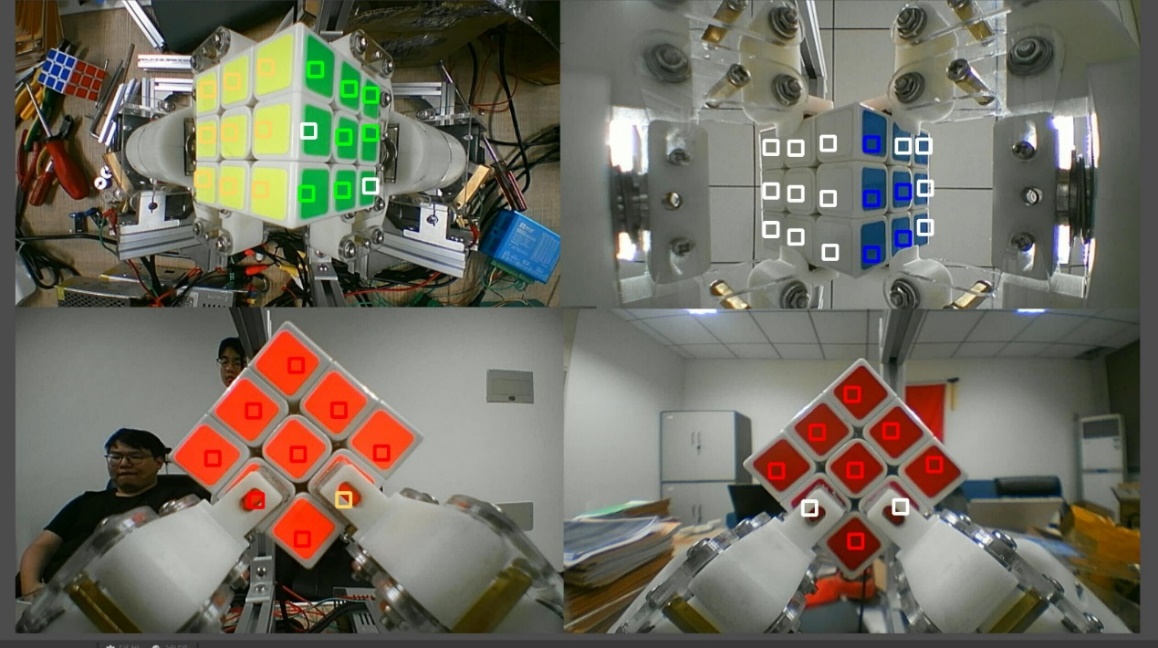
综上所述，通过首先通过识别边框的方法来获取非常方便快速得获取不同光照下的颜色的RGB值，通过这些大量的数据来进行模型的训练，再通过定点识别的方法来获取实际运行过程中魔方的颜色值，通过调用模型来进行颜色的判断。

图4.7 定点提取、识别颜色

4.2 解魔方

解魔方我们采用了当前最主流的计算机解魔方算法，二阶段算法（Two phase algorithm，又称Kociemba's algorithm），在当前计算机上，对于一个充分打乱的魔方，该算法平均可以在几毫秒的时间内得到一个不超过20步的解法。魔方可能的状态数实在太多了，达到4.3\*10^19，这是一个很大数量级，而二阶段算法顾名思义，就是把还原的过程分成了两个阶段。首先进行阶段一的搜索，得到了阶段一的最优解。接着通过阶段二的搜索，得到了阶段二解法，从而产生了一个步的完整解法。通过不断搜索解法再进行比较，获得其最快解。由于这个算法过于高深，所以我解魔方为调用Kociemba库来完成。虽然我们已经极力提高颜色的准确率了但是若光照改变，其所识别的颜色还是会有几率错误，某个色块的颜色会在两种颜色中跳动，而将错误的颜色信息输入Kociemba库，整个程序将会报错，从而导致运行失败，所以我们设计了检验程序，将颜色分别标为1，2，3，4，5，6，将所有面的颜色相加，若其不等于21\*9，那么便说明识别中有错误，需要重新进行识别。然后这一步遇到的最大的困难是对四个摄像头依次识别数据的处理，使其成为其库的标准输入，即每个面都有一个字符表示，U为上表面，D为下表面，L为左表面，R为右表面，F为前表面，B为后表面，解完的魔方每个面都是同一个颜色，所以UDLRFB可以代表每个面的颜色。假定我们让上表面U为白色，下表面D为黄色，左表面L为红色，右表面R为橙色，前表面F为蓝色，后B表面为绿色。当魔方打乱后，则下图表示的每个面的单元格颜色就可能是之前任何面的某一个颜色，这样便可以让魔方可任意摆放在手臂上。我们按照如下顺序将单元格颜色变成字符串：`U1`, `U2`, `U3`, `U4`, `U5`, `U6`, `U7`, `U8`, `U9`, `R1`, `R2`, `R3`, `R4`, `R5`, `R6`, `R7`, `R8`, `R9`, `F1`, `F2`, `F3`, `F4`, `F5`, `F6`, `F7`, `F8`, `F9`, `D1`, `D2`, `D3`, `D4`, `D5`, `D6`, `D7`, `D8`, `D9`, `L1`, `L2`, `L3`, `L4`, `L5`, `L6`, `L7`, `L8`, `L9`, `B1`, `B2`, `B3`, `B4`, `B5`, `B6`, `B7`, `B8`, `B9`。所以当我们输入字符串"UBL..."，表示U1为白色，U2为绿色，U3为红色，以此类推，即可定义所有面的单元格标志字符串。

以下便是各字符所在的位置：

|\*\*\*\*\*\*\*\*\*\*\*\*|

|U1\*U2\*U3|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|U4\*U5\*U6|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|U7\*U8\*U9|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|

|L1\*L2\*L3|F1\*F2\*F3|R1\*R2\*R3|B1\*B2\*B3|

|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|

|L4\*L5\*L6|F4\*F5\*F6|R4\*R5\*R6|B4\*B5\*B6|

|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|

|L7\*L8\*L9|F7\*F8\*F9|R7\*R8\*R9|B7\*B8\*B9|

|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*\*|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|D1\*D2\*D3|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|D4\*D5\*D6|

|\*\*\*\*\*\*\*\*\*\*\*\*|

|D7\*D8\*D9|

|\*\*\*\*\*\*\*\*\*\*\*\*|

再将其输出处理为我们需要数据。解魔方步骤字符串包含空格字符，含义如下：

* 单个字符表示所代表的面顺时针旋转90度。
* 单个字符带一撇，表示代表的面逆时针旋转90度。
* 单个字符带2，表示代表的面旋转180度。

比如R U R’ U R U2 R’ U, 表示右表面顺时针旋转90度，下表面顺时针旋转90度，右表面逆时针旋转90度，下表面顺时针旋转90度，右表面顺时针旋转90度，下表面旋转180度，右表面逆时针旋转90度，下表面顺时针旋转90度。Kociemba库输出的步骤只是六个面的转动，而由于我们的魔方是双臂四指来完成步骤，所以我们需要进行坐标系的转换，例如Kociemba库解出的算法为将U面进行顺时针旋转90度，而我们的机械结构需要现将魔方整个进行翻转，来让处于上面的U面转至下面，从而使手臂可以拧到这个面，而在这个过程中整个魔方的坐标系就会进行旋转，Kociemba库解出的后面的步骤就要进行改变。这样输出的步骤便是我们希望手臂所能进行的运动。我们便可将其传入Arduino。

4.3 魔方控制

最后便是电脑与Arduino之间的通信与Arduino控制机械的运动，将解出电脑来的通过调用serial库来通过串口通信来将步骤的字符串传给Arduino，Arduino每次读取两个字符那便是单个步骤，同时剩余的步骤将存在于数据缓存区，Arduino同时也会通过串口来将步骤完成的信息进行反馈，以此来达到计时的目的。在Arduino收到步骤信息后它便要输出不同PWM信号来控制电机的转速，逻辑信号控制电机的正反转，逻辑信号控制继电器的开关以此来控制电磁的开关从控制手臂的张合。对于两个逻辑型号非常简单，只需输出高低电平，需要注意的是继电器的连接端的选择使手臂在低电平的信号下为闭合状态，在输入高电平是张开状态。

而PWM信号则会复杂许多。步进电机（也称脉冲电机），将电脉冲转换为相应的角位移或线位移的电磁机械装置，具有快速启动，停能力，在电机的负荷不超过它能提供的动态转矩时，可以通过输入脉冲来控制它在一瞬间的启动或者停止。在非超载的情况下，步距角和转速只和输入的脉冲频率有关，和环境温度，气压，振动无关，也不受电网电压的波动和负载变化的影响，因此，控制较为精准，即给电机一个脉冲信号，电机则转过一个步距角。信号开始运行后，步进电机将从0度旋转目标度数，每次一个分度。而转速则为PWM信号的频率决定的，一个脉冲信号，便会转一个分度。在调试完后我们需要使电机转速提升，这就需要我们加快频率，而当频率到超过Arduino上限时我们便要调低分度值，这是便会发现电机由于转速太快会丢步。这就需要我们对电机的启动与停止来设置加速度，即使PWM信号的频率由低到目标值，再由目标值到0。这样电机的转速便会由一开始的慢变快，再由快变慢。从而使其的峰值转速达到非常高而不会丢步，而由于加速时间和减速时间其实非常短，所以总的转速时间会有非常大的提升。从而加快整个机构的速度。

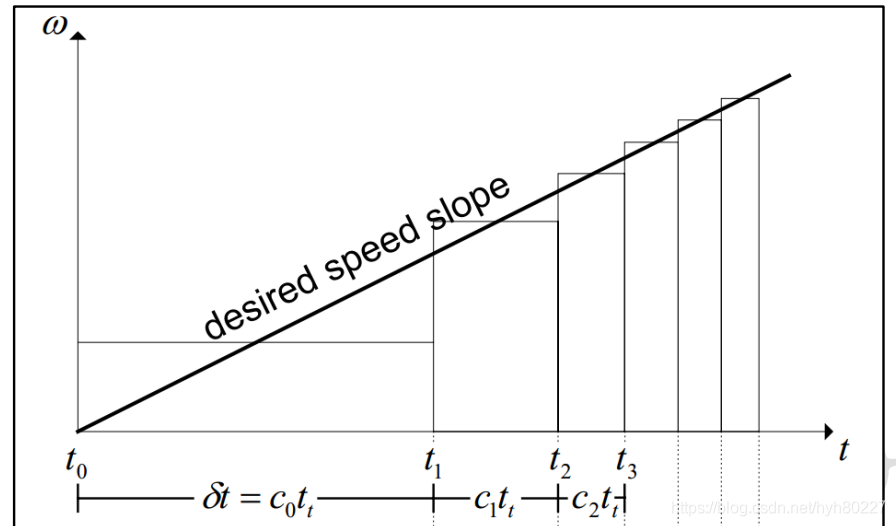


图4.8 步进电机加减速的实现

# 五、总结

## 5.1 优势

1、按系统功能分，机器人整体可分为动力系统、控制系统和执行系统。在执行逻辑上，首先4个摄像机得到指令后拍摄图片并发送至计算机。然后计算机处理图像并提取色块信息，通过算法解算出步骤后发送信号至下位机。最后下位机控制电机旋转与电磁阀的开闭，进而完成手腕拧转和手指夹取的基本动作。通过以上三个步骤还原打乱的魔方。

2、机械手爪的驱动控制。本作品的机械手爪夹、放动作的驱动装置为电磁铁，具有反应速度快、成本低、重量轻、体积小的优点。

3、机械手爪的机械结构。为配合电磁铁的驱动特性，本机械手爪结构极具创新特色。其两个手指均采用I型和L型手指连接件与机械手臂和手掌相连，通过电磁铁弹出、收进手掌来控制手指的夹、放动作。该结构性能稳定、反应迅速，在电磁铁的驱动下可高效完成手指的夹、放动作。

4、采用半遮光-多摄像头图像识别策略，提高识别精度，缩短识别时间。通过不完全透明的亚克力板提供较为柔和的光照，保证图像亮度；将包含图像信息的RGB空间数据，实现色调与饱和度、明度的分离，提升颜色识别的简便性和准确性；多摄像头的设置避免采样时手臂需多次旋转的问题，大大缩短图像识别时间识别。

5、使用Python开源算法库Kociemba，解算过程时间短效率高，解算结果步骤简单步数少。使用强大的功能函数kociemba.solve()，能够轻易在ms级的时间得到平均20步（5次测量的数据）包含还原步骤的字符串，再通过Python的字符串处理函数规范化字符串，得到能够被下位机识别的指令序列。

经过多次测试，该魔方机器人能高标准完成机器人大赛给出的任务，效果出色，性能稳定，还原速度快。

## 5.2 可以改进的方面

在调试的过程中，我们还发现了许多可以改进的方面：

1.机械结构材料及其精度。不同的零部件采用不同的材料，有各自的机械精度，在组装时又会有一定的精度差。故魔方机器人整体机械结构的精度对魔方复原的稳定性有一定影响，当转速过快时，会出现魔方滑落甚至甩出的现象。

针对此现象，小组尝试不同的3D打印材料，以提高机械臂和机械手爪的精度；不同机械零件连接时，加强固定力度；在手指内侧粘贴橡胶膜，增大手指与魔方之间的摩擦力。

2.光线强度变化。自然条件下，外界光线会随时间而发生变化。光线强弱影响了摄像头对魔方色块颜色的识别与提取，会使上位机获得错误信息。

针对此现象，小组采用补光设备，使魔方周围光线尽量保持不变，削弱自然光变化而产生的影响。

3.电磁铁的反应周期。小组采用的电磁铁控制电路已经将电磁铁铁芯收缩的时间控制在15以内，但铁芯的释放过程仍有较长时间，约为60。

针对此现象，小组采取给电磁铁增设弹簧的方法，加大铁芯释放所受到的回程力，缩短释放时间。

# 六、实物图



图6.1 手爪部分实拍图

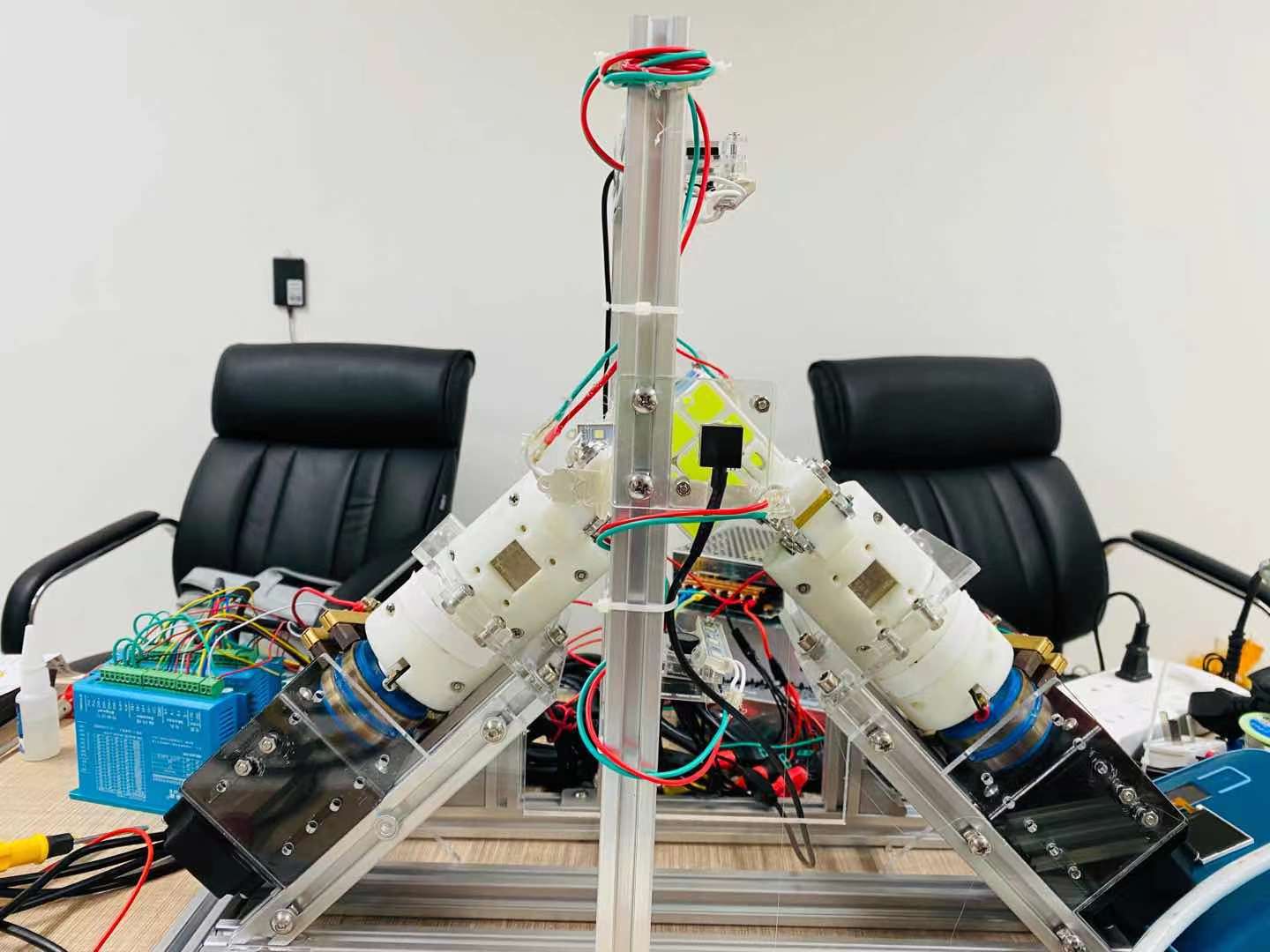


图6.2总体实拍图

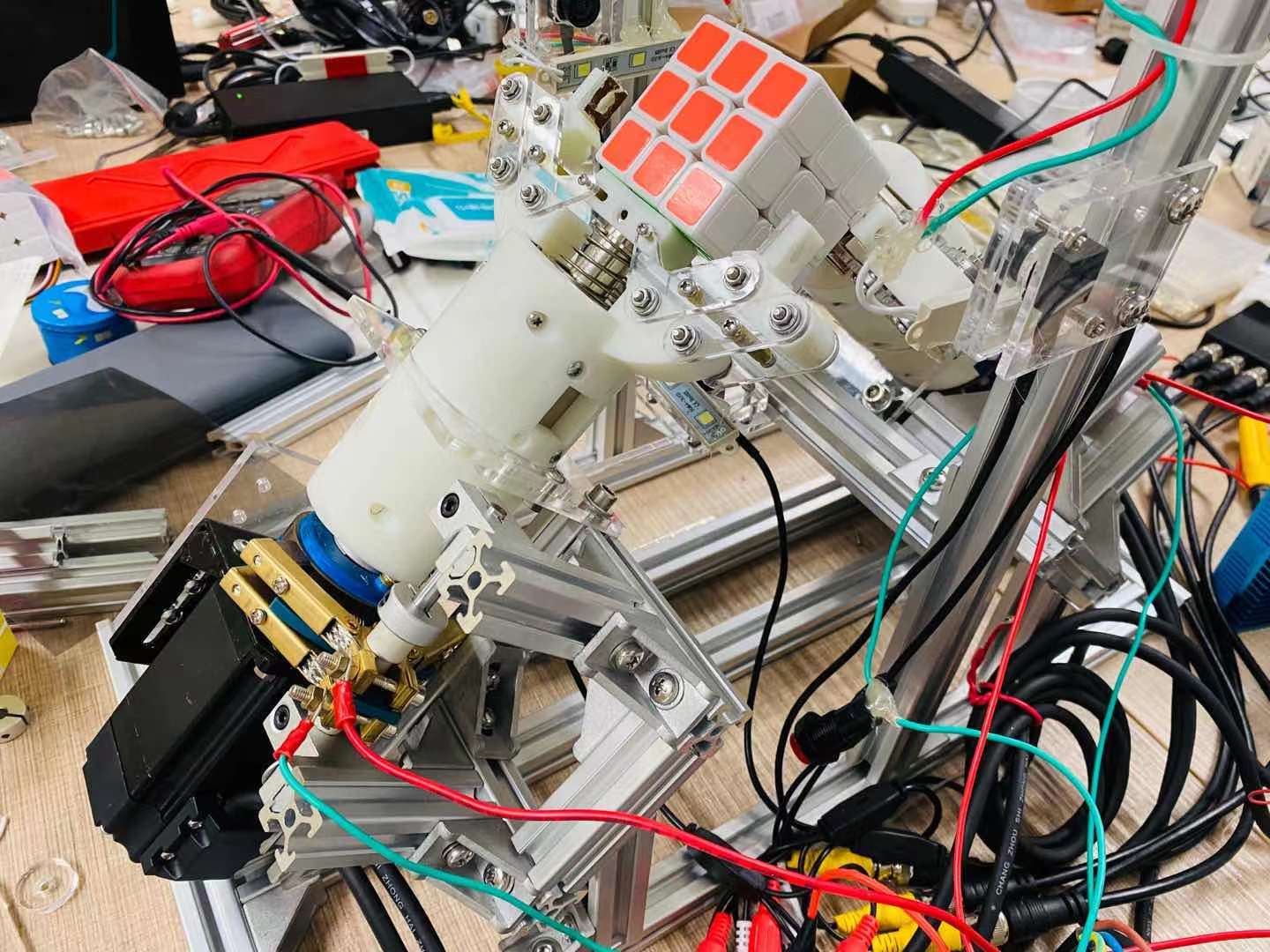


图6.3 侧视图