

# 第十八届中国研究生电子设计竞赛

## 技术论文

中文题目：基于手势自然交互技术的无人机远程控制

英文题目：Gesture-Based Natural Interaction Technology  
for Remote Control of Unmanned Aerial Vehicles (UAV)

## 摘要

无人机如今已广泛应用在摄影、农业、物流、安全等多个领域。其性能随着通信技术与控制技术的优化也逐渐得到提升,越来越多的新兴技术开始出现在无人机上。但是目前大多数控制无人机的方式依旧停留在遥控器、手机等设备上,其中遥控器是目前无人机最为常用的控制方式。基于遥控设计的无人机操控方式存在着交互复杂、学习难度高等缺点,而无人机操控方式的便捷化、智能化是无人机领域发展的重要趋势。通过将手势自然交互技术应用于无人机的远程控制,本设计使无人机操作更加便捷。通过在 EAIDK-610 开发板中部署手部骨骼点检测模型,得到视频流中的手部骨骼点坐标信息,以此计算手指曲率,实现对飞行指令手势的识别,以符合自然交互的手势操纵无人机飞行。为实现上述功能,本文的设计主要如下:

### (1) EAIDK610 与多传感器的软硬件适配

本文采用 APM2.8 作为四旋翼无人机飞行控制器。通过飞行控制器内部的陀螺仪、气压计、GPS 等多个传感器,实时获取和处理包括无人机的姿态、角速度、加速度等信息。陀螺仪提供了姿态变化的测量,气压计则用于测量高度变化。同时,以 EAIDK610 搭载摄像头捕捉和传输实时图像,利用这些图像进行用户的手部检测和手势识别,实现更准确的手势分类和识别效果。这些设计可以对无人机飞行过程中的手势进行实时识别,并根据识别结果实现无人机的位置和姿态改变,从而实现对无人机的精准操控。

### (2) 基于深度学习的手骨骼点检测算法

本文采用了基于深度学习的手骨骼点检测算法,结合了谷歌开源代码和 mediapipe 框架,实现了高效、精确的手部姿态估计。首先,通过谷歌开源的训练代码,练了自己的手部骨骼点检测模型。这个模型经过了大规模的训练和优化,在多样化的手部图像数据集上表现出色。它融合了卷积神经网络和关键点回归技术,能够准确地预测手部各个关键点的位置。最后,将训练好的模型与 Mediapipe 框架相结合,使得手势检测算法具备了实时性和高效性。

### (3) 基于手骨骼点的手势识别算法

本算法首先利用传感器设备或摄像头捕获手部骨骼点信息,并使用图像处理和骨骼追踪技术将其转换为数字化的骨骼点坐标数据。由于 EAIDK 计算能力有限,便放弃了基于深度学习的识别算法网络,而是通过观察骨骼点与控制手势之间的特征联系,采用数学几何学进行特征提取,并计算手指骨骼的弯曲度来区分不同手势姿态的控制含义。虽然这种方法可能限制了对复杂手势的分类能力,但对于简单到中等复杂度的手势识别任务是简单而高效的选择。通过计算手骨骼的几何特征,可以实现实时性,并且满足多应用场景中手势分类的需求。

### (4) APM2.8 飞控与 ESP8266 无线模块结合设计

通过在无人机上添加 ESP8266 模块并改进飞控代码,本文实现了 APM2.8 飞控与 ESP8266 无线模块结合设计。该设计的关键在于将 ESP8266 模块与现有的无人机飞控系统连接起来。ESP8266 模块是一种低成本、高性能的 Wi-Fi 模块,可实现无线通信。集成 ESP8266 模块到无人机上,能够实现与地面设备(如手

机、电脑或边缘计算设备)的无线连接。同时,在飞控代码方面,对原有的飞控程序进行了改进,新增了手势识别和通信协议解析功能。

实验结果表明基于手势自然交互技术的无人机远程控制具有较好的效果,实现过程中,操纵者通过符合人体自然交互的动作语言操作无人机飞行平稳,对于突发情况,也有紧急制动的处理。因此,本设计在操纵无人机进行拍摄、农业和物流等方面都具有较大的实用价值。

**关键词:** 手势自然交互; 无人机; ESP8266 无线模块

## Abstract

Drones have been widely used in various fields such as photography, agriculture, logistics, and security. With the optimization of communication and control technologies, their performance has been gradually improved, and more emerging technologies are being applied to drones. However, the majority of drone control methods still rely on devices like remote controllers and smartphones, with remote controllers being the most commonly used control method. The conventional remote control design for drones has disadvantages such as complex interaction and high learning difficulty. The convenience and intelligence of drone control are important trends in the field of unmanned aerial vehicles (UAVs). By applying gesture-based natural interaction technology to remote control drones, this design aims to make drone operations more convenient. By deploying a hand skeletal point detection model in the EAIDK-610 development board, the coordinates of the hand skeletal points can be obtained from the video stream. This information is then used to calculate finger curvature and recognize flight command gestures, enabling natural interaction-based control of drone flight. To achieve these functions, this design includes the following main aspects:

(1) Hardware and software adaptation between EAIDK610 and multiple sensors

This study adopts APM2.8 as the flight controller of the quadcopter. It obtains and processes real-time information about the drone's attitude, angular velocity, acceleration, etc., using various sensors inside the flight controller such as gyroscope, barometer, and GPS. The gyroscope provides measurements of attitude changes, while the barometer measures altitude changes. Simultaneously, the EAIDK610 is equipped with a camera to capture and transmit real-time images. These images are utilized for hand detection and gesture recognition, enabling more accurate gesture classification and recognition results. These designs allow real-time recognition of gestures during drone flight and enable precise control based on the recognition results.

(2) Hand skeletal point detection algorithm based on deep learning

This study adopts a deep learning-based hand skeletal point detection algorithm that combines Google's open-source code and the Mediapipe framework to achieve efficient and accurate hand pose estimation. Firstly, a hand skeletal point detection model is trained using Google's open-source training code. This model has undergone extensive training and optimization, demonstrating excellent performance on diverse hand image datasets. It combines convolutional neural networks and keypoint regression technology to accurately predict the positions of each hand keypoint. Finally, the trained model is integrated with the Mediapipe framework, enabling real-time and efficient gesture detection algorithms.

(3) Gesture recognition algorithm based on hand skeletal points

This algorithm first captures hand skeletal point information using sensor devices

or cameras. It then uses image processing and skeleton tracking techniques to convert the information into digitized coordinate data of skeletal points. Due to the limited computational capacity of EAIDK, deep learning-based recognition algorithm networks are abandoned. Instead, mathematical geometry is employed for feature extraction by observing the correlation between skeletal points and control gestures. The algorithm calculates the curvature of finger bones to differentiate the control meanings of different gesture poses. Although this method may limit the classification ability for complex gestures, it is a simple and efficient choice for recognizing simple to moderately complex gestures. Real-time calculation of hand skeletal geometry features meets the requirements of gesture classification in various application scenarios.

#### (4) Design combining APM2.8 flight controller with ESP8266 wireless module

By adding an ESP8266 module to the drone and improving the flight controller code, this study combines the APM2.8 flight controller with the ESP8266 wireless module. The key to this design is connecting the ESP8266 module to the existing drone flight control system. The ESP8266 module is a low-cost, high-performance Wi-Fi module that enables wireless communication. Integrating the ESP8266 module into the drone allows wireless connectivity with ground devices such as smartphones, computers, or edge computing devices. In terms of flight control code, the original flight control program is improved to add gesture recognition and communication protocol parsing functions.

Experimental results demonstrate that remote control of drones based on gesture-based natural interaction technology has excellent performance. During the implementation, operators can smoothly control drone flights through natural gestures, and emergency braking is also available for unexpected situations. Therefore, this design has practical value in controlling drones for photography, agriculture, logistics, and other applications.

**Keywords:** gesture-based natural interaction; drone; ESP8266 wireless modul

# 目录

<b>第一章 作品难点与创新</b>	<b>1</b>
1.1 作品难点	1
1.2 作品创新点	2
1.2.1 硬件电路设计创新	2
1.2.2 算法设计与应用创新	2
<b>第二章 方案论证与设计</b>	<b>4</b>
2.1 无人机手势控制系统的总体设计	4
2.2.1 设计思路	4
2.2.2 设计框图	5
2.2.3 系统可行性分析	6
2.2 系统开发环境搭建	7
<b>第三章 原理分析与硬件电路图</b>	<b>9</b>
3.1 算法原理	9
3.1.1 基于传统算法的图像优化	9
3.1.2 实时手势识别算法	10
3.1.3 基于坐标计算手指形态自定义飞行手势指令	13
3.2 系统实现的硬件构架	16
<b>第四章 软件设计与流程</b>	<b>20</b>
4.1 系统驱动程序设计	20
4.1.1 Arduino 驱动程序设计	20
4.1.2 ESP8266 的远程控制指令接收与解译程序设计	21
4.2 系统主程序设计	23
<b>第五章 系统测试与分析</b>	<b>25</b>
5.1 EAIDK610 手势控制终端与无人机通信链路测试	25
5.2 基于手势识别的无人机远程交互功能测试	26
<b>第六章 总结</b>	<b>33</b>
6.1 总结	33
6.1 展望	33
<b>参考文献</b>	<b>35</b>

# 第一章 作品难点与创新

## 1.1 作品难点

本团队通过分析目前的人机交互的前沿技术，提出四旋翼无人机与手势自然交互技术融合的控制方案，但结合目前计算机视觉的模型部署算力需求与 EAIDK-610 等嵌入式移动设备的有限计算能力，产生有如下问题：

### (1) EAIDK610 与多传感器的软硬件适配

将 PC 端与其他开发板的程序设计移植并融合到 EAIDK610 平台是一项复杂的任务。同时，需要通过合适的通信协议和接口实现 PC 端与 EAIDK610 平台及其他开发板之间的数据交换。通过充分利用平台功能和整合软硬件，可以实现一个规范且高效的软硬件结合系统。

### (2) 轻量级手势检测与分类算法的设计

本文的通过设计合理的手势实现对无人机的远程控制。在该场景下要求算法对无人机操作者的手势具有高准确与低失误率，从而保证无人机的安全驾驶。相较传统的目标检测与分类算法，基于深度学习的目标检测与分类算法能够更好的实现对手部区域以及手部区域的手骨骼点实现好的检测。然而，EAIDK610 等嵌入式边缘设备其计算能力有限，因此如何对深度学习算法进行优化和选择适合的部署方式，成为能否完成此作品的关键。

### (3) 无人机控制手势的用户态设计

当涉及到无人机控制手势时，最关键的是如何设计出令人容易理解且易于执行的手势。在设计无人机控制手势时，需要选择合适的手势类型，并确保每种手势均与特定的控制指令相关联。例如，可以考虑使用基于方向的手势，以区分方向和被控制对象的运动方式。这样就可以让用户轻松地掌控无人机飞行。为了让用户能够直观地掌控无人机，必须设计出符合实际操作的手势。该手势需要在不同的环境中可靠地工作，并考虑到不同的用户在使用无人机时可能会遇到的情况。

### (4) 无人机远程交互平台的安全设计

本文主要将 F450 四旋翼无人机作为智能手势控制的实验平台，如何无人机功能与手势进行合理匹配。实现包括远程指令控制飞行、定点悬停和紧急刹车等功能。在实现这些功能的同时，需要思考如何进行合理设计，使得无人机实验平台在远程控制信号丢失时，确保无人机自主进行有效的安全措施。

## 1.2 作品创新点

### 1.2.1 硬件电路设计创新

#### (1) APM2.8 飞控与 ESP8266 无线模块结合设计

通过将 ESP8266 嵌入无人机 APM2.8 飞行控制器中，这一设计的优势在于其创新性和简便性。通过将 ESP8266 模块与飞控代码改进相结合，本文能够实现无需复杂遥控器和繁琐设置的手势远程控制无人机。用户可以使用手势远程控制无人机的行为。只需简单的手势动作即可实现无人机的起飞、降落、悬停、调整飞行方向和速度等操作。通过无线通信，实现了更加直观且灵活的无人机控制。这种设计不仅提升了用户体验，还拓宽了无人机使用的场景，并为未来的无人机技术发展带来了更大的潜力。

#### (2) 多传感器环境感知设计

本文采用 APM2.8 作为四旋翼无人机飞行控制器。通过飞行控制器内部的陀螺仪、气压计、GPS 等多个传感器，实时获取和处理包括无人机的姿态、角速度、加速度等信息。陀螺仪提供了姿态变化的测量，气压计则用于测量高度变化。同时，以 EAIDK610 搭载摄像头捕捉和传输实时图像，利用这些图像进行用户的手部检测和手势识别，实现更准确的手势分类和识别效果。这些设计可以对无人机飞行过程中的手势进行实时识别，并根据识别结果实现无人机的位置和姿态改变，从而实现对无人机的精准操控。

### 1.2.2 算法设计与应用创新

算法设计与应用创新主要分为四个方面，分别是基于深度学习的手骨骼点检测算法、基于手骨骼点的手势实时识别算法、基于计算机视觉的图像白平衡智能矫正和用户态手势交互逻辑设计。下文对此做具体阐述。

#### (1) 基于深度学习的手骨骼点检测算法

本文采用了一种基于深度学习的手骨骼点检测算法，该算法结合了谷歌开源代码和 Mediapipe 框架，实现了高效、精确的手部姿态估计。通过谷歌开源的训练代码，本文训练出自己的手部骨骼点检测模型，这个模型经过大规模的训练和优化，在多样化的手部图像数据集上取得了出色的性能。通过融合卷积神经网络和关键点回归技术，该模型能够准确地预测手部各个关键点的位置。最后将本文的训练模型与 Mediapipe 框架相结合，从而使得手势检测算法具备实时性和高效性。

#### (2) 基于手骨骼点的手势实时识别算法



本算法首先借助传感器设备或摄像头捕获手部的骨骼点信息，并采用图像处理和骨骼追踪的技术将这些信息转化为数字化的骨骼点坐标数据表示。由于 EAI DK610 开发板的计算能力有限。为了实现实时性，本文摒弃基于深度学习的识别算法网络，通过观察骨骼点与本文控制手势的手势之间的特征联系，采用数学几何学进行特征提取，通过坐标点计算手指骨骼的弯曲度，根据弯曲度区分不同手部姿态的手势控制含义。尽管不使用深度学习技术可能会限制本文对复杂手势的分类能力，但对于简单到中等复杂度的手势识别任务，本文的设计方案是一种简单而高效的选择。通过计算手骨骼的几何特征，本文能够实现实时性，并且在许多应用场景下都能满足手势分类的需求。

### (3) 基于计算机视觉的图像白平衡智能矫正

基于应用场景，考虑输入视频流图像信息会因角度因素和真实世界中亮光源等原因导致输入数据存在差异化，致使计算机视觉算法检测效率不佳等问题。在不同的场景或光线条件下，图像可能会受到白平衡的干扰，从而使图像的颜色失真。本文通过图像识别技术自动检测图像的环境光源的颜色温度，并根据特定的算法智能地调整色彩平衡，以达到更加真实的色彩还原效果。本文实现的基于计算机视觉的智能矫正可大大提高白平衡矫正的精确性和速度，同时自动适应各种光线条件下的图像，并且开放 API 也让其他软件可以调用接口来使用该智能矫正工具。

### (4) 用户态手势交互逻辑设计

自然交互设计要从无人机操控方式的用户需求与生活习惯出发。无人机主要的飞行姿态由的偏航角、俯仰角、滚动角、油门来控制。因此，无人机常见的飞行控制需求包括：航向的左转和右转、前进和后退、横滚的左平移和右平移。无人机还需要解锁、下降定高、上锁等控制命令。针对这些需求，本团队结合人们日常的手势操作习惯。设计出了令人容易理解且易于执行的手势。在设计无人机控制手势时、确保了每种手势均与特定的控制指令相关联。

## 第二章 方案论证与设计

### 2.1 无人机手势控制系统的总体设计

本文的作品旨在实现基于手势识别的无人机控制系统，通过使用 EAIDK-610 边缘计算设备上部署的手部骨骼点检测模型来获取双手的骨骼坐标，并根据坐标进行手势定义与识别。通过整体设计实现无人机控制系统的常规操作与控制简化。使用手势识别替代传统遥控器，使用户只需通过手势即可轻松操纵无人机，提供更直观、自然且互动性强的控制方式。这使得无人机技术的应用更加普及，并拓宽了人机交互的应用范围。

#### 2.2.1 设计思路

随着无人机技术的快速发展，无人机已广泛应用在生活中的各个领域。如今无人机的远程操控大都依赖于传统的遥控器和硬件设备，具有很大的应用局限性，这对无人机的普及和推广带来了巨大阻碍。为了提高无人机远程操控的智能化以及用户的沉浸感。越来越多的研究人员开始探索自然交互技术在无人机远程交互中的应用。

针对上述的发展趋势，本团队设计了一套基于手势操作无人机的方案与设计。手势识别一直是自然交互技术领域的一个重要研究方向，传统的手势识别算法鲁棒性差、精度低，成熟的手势识别又依赖于一些昂贵的设备，比如数据手套。随着目标检测算法取得了巨大成功，手势识别领域开启了新的研究方向。基于深度学习的手势识别算法应用越来越受到青睐，然而在 EAIDK610 等嵌入式移动设备上，需要针对其有限的计算能力设计满足实时性的目标检测模型。为达到预期设计目标，本团队采用如下设计思路：

##### (1) 选取合适的手部骨骼点检测算法进行部署

由于设备的计算能力有限，本文选取谷歌开源的手势骨骼点检测算法作为本项目的骨骼点检测模块。该算法与 mediapipe 框架相结合可以实现较低的计算复杂度和内存占用，同时保证算法的准确性和稳定性。该算法能够准确地检测手部骨骼点，而且对于不同的手势和手型都能够有良好的适应性。在部署过程中，本文会对所选算法进行模型压缩和优化，以适应设备的资源限制。这可能包括移除冗余参数、量化模型权重以减少内存占用，并采用轻量级的神经网络架构来提高运行速度。

##### (2) 设计满足实时行的手势识别算法

本文设计了一种基于计算手骨骼二维坐标点的弯曲度等几何特征来进行手

势分类的方法。通过捕获手部动作，本文提取手掌和手指的关键节点，在每个时间步上记录它们的坐标点。接下来，本文计算手指关节点之间的距离、角度以及手指之间连接线的弧长等几何特征，并结合这些特征来描述手势形状。为了实现实时性，本文选择不使用深度学习技术。相较于深度学习需要大量的数据和复杂的计算过程，本文的方法更加轻量级和高效。通过仅依赖于手骨骼坐标点的几何特征，本文能够快速地进行实时的手势分类。

### (3) 对四旋翼无人机进行硬件适配

通过在无人机上加装 ESP8266 模块，并改进飞控代码以适配手势远程控制。该方案的关键是将 ESP8266 模块与现有的无人机飞控系统进行连接。ESP8266 模块是一种低成本、高性能的 Wi-Fi 模块，可以实现无线通信。通过在无人机上集成 ESP8266 模块，本文可以实现与地面设备（如手机、电脑或边缘计算设备）的无线连接。同时在飞控代码方面，本文对原有的飞控程序进行了改进，添加了手势识别和通信协议解析的功能。

通过上面的设计思路可以实现将 EAIDK610 边缘计算设备连接至无人机 wifi 热点，利用 socket 和 UDP 协议进行通信，一旦识别到特定手势，将对应的指令发送给无人机。这些指令可以是飞行方向、高度、速度以及其他飞行参数的控制命令。

## 2.2.2 设计框图

本文的作品设计框架基于 EAIDK610 设备，结合了开源的骨骼点检测算法和手势识别算法。通过这些算法，本文可以实时检测人体的手部动作，并准确地识别特定的手势。在实际应用中，本文将该框架部署在 EAIDK610 设备上，然后，本文通过连接 esp8266 控制 arduino 开发板，利用其 PWM 输出功能来控制无人机的输入通道。

当 EAIDK610 检测到特定的手势时，它会发送控制指令给 esp8266 模块，这些指令包含了特定手势所对应的操作。接着 esp8266 将解析这些指令并将其传递给连接的 arduino 开发板。Arduino 开发板根据接收到的指令进行解析，并通过适当的 PWM 波信号控制无人机的输入通道，从而实现对无人机的控制。整个过程实现了人体手势与无人机控制之间的有效交互。系统总体设计框架如图 2-1 所示：

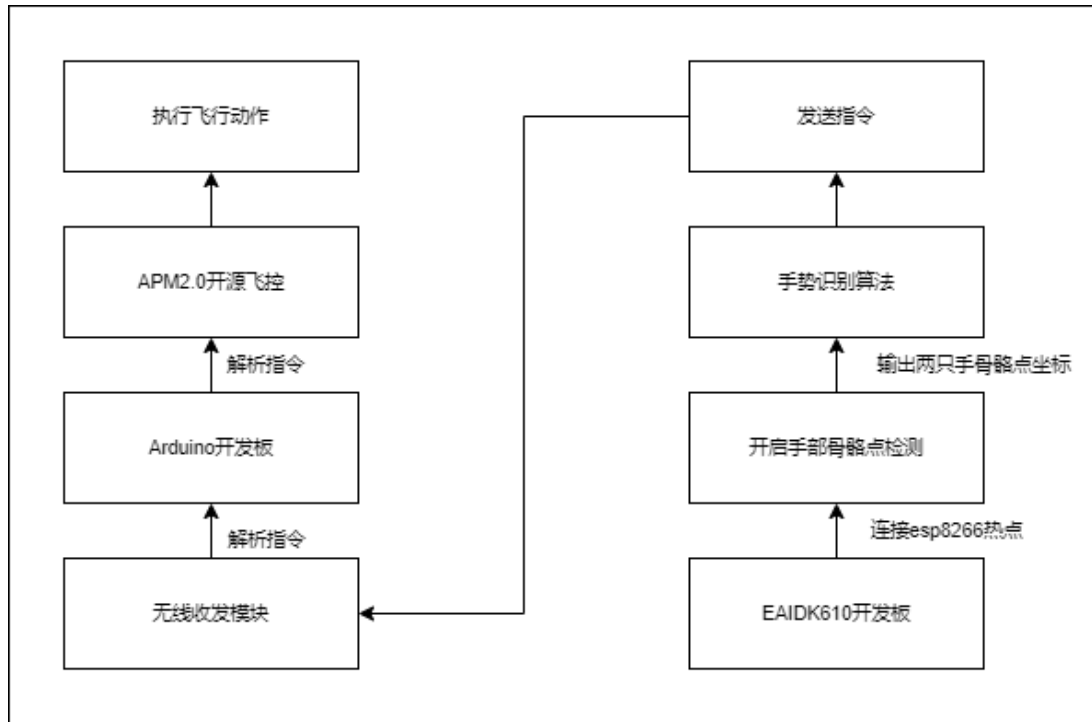


图 2-1 系统总体设计框架图

### 2.2.3 系统可行性分析

本作品设计基于深度学习、无线通信、无人机等多领域。本文对思路进行了系统可行性分析，该思路在以下几个方面被认为是可行的：

**硬件资源：**EAIK610 设备具有强大的计算能力和丰富的接口，可以满足骨骼点检测算法和手势识别算法的运行需要。同时，ESP8266 控制 arduino 开发板提供了 PWM 输出功能，适用于控制无人机输入通道。

**软件支持：**开源的骨骼点检测算法和手势识别算法已经在许多平台上得到验证，并且被广泛应用。这些算法可以有效地从图像或视频中提取手部动作信息并进行识别，符合本文的目标。

**通信协议：**使用 ESP8266 与 arduino 进行通信是合理的选择，因为它们之间的通信速度快且可靠。这样，当板子检测到特定手势时，可以通过发送指令给 esp8266 来触发相应的动作。

**数据传输延迟：**由于使用本地端设备进行处理和控制，系统的响应速度较快。这使得本文能够实现实时的手势识别和无人机控制，为用户提供即时的反馈。

总而言之，本文的思路在技术和实际应用层面上是可行的。通过在

EAIDK610 设备上部署骨骼点检测算法和手势识别算法，并结合 esp8266 控制 arduino 开发板输出 PWM 波来控制无人机输入通道，本文能够将手势识别与无人机控制相结合，实现基于手势的无人机控制系统。

## 2.2 系统开发环境搭建

本文的作品涉及到多个环境的搭建，以实现在 EAIDK610 设备上部署骨骼点检测算法和手势识别算法，并利用 ESP8266 控制 arduino 开发板输出 PWM 波来控制无人机输入通道。以下是系统开发环境搭建的步骤：

### (1) EAIDK610 开发板

确保 EAIDK610 设备具有 python 代码编译器，并根据项目的需求安装对应的库。本作品的环境配置如下表 2-1 所示。

表 2-1 EAIDK610 开发环境

模块名	版本
pysrial	3.5
numpy	1.21.5
matplotlib	3.5.1
opencv-python	4.5
tqdm	4.64.0
Pillow	8.4.0
mediapipe	2.1.2
cvzone	1.2.1

### (2) 四旋翼无人机实验平台

在四旋翼无人机上搭载 APM2.8 开源飞控，并了解其通信协议和控制方式。将 Arduino 的 pwm 输出与飞控的输入连接。以获得通过 arduino 解析远程指令。

控制无刷电机输出姿态。

### (3) Arduino 远程指令解析模块

设计 Arduino 与 ESP8266 的通信链路，确保它与 ESP8266 模块兼容，并编写其与 ESP8266 的交互逻辑代码，并实现控制信号的解析逻辑。连接 ESP8266 模块和 Arduino 开发板，确保它们之间的通信正常。根据无人机的硬件接口和控制需求，将 ESP8266 与 Arduino 开发板相连，以实现 PWM 波的输出控制无人机输入通道。

### (4) 手部骨骼点检测算法

本作品在谷歌开源代码的基础上，对骨骼点检测的模型进行了重新训练，对模型进行训练的配置要求如下表 2-2。

表 2-2 模型训练配置

Python	3.7
Pytorch (gpu)	1.3.1
torchvision	0.14.1
torchaudio	0.13.1
CUDA	11.6
操作系统	Ubuntu

## 第三章 原理分析与硬件电路图

### 3.1 算法原理

#### 3.1.1 基于传统算法的图像优化

本作品深度学习算法都基于摄像头来提取实时信息，当摄像头观察现实世界中的 3D 物体并将其转化维 2D 图像时，就会发生一系列的畸变。畸变实际上改变了现实世界中 3D 物体的形状和大小。故本文在对实时图像信息进行预处理的第一步，即为消除这种失真缺陷，这样摄像头便可从真实世界图像中得到更优质有用的信息。

本文采用棋盘格来标定相机然后做畸变矫正，本文采用相机拍摄棋盘格 20 组照片，用于生成畸变模型。具体方法为：将原图像进行灰度化处理，然后使用 Opencv 中 `cv2.findChessboardCorners` 方法来确定图像中是否存在棋盘图案，并定位棋盘格上的内角点。而后使用 `cv2.calibrateCamera()` 来求解畸变系数。修正径向畸变需要三个系数： $k_1, k_2, k_3$ ；而修正切向畸变需要两个系数： $p_1, p_2$ 。计算畸变点坐标的公式如下所示：其中  $r$  为未畸变图像中某点与图像畸变中心之间的已知距离，而该中心通常就是图像的中心  $(x_c, y_c)$ 。具体计算公式如 3-1 和 3-2 所示：

$$x_{distorted} = x_{ideal}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (3-1)$$

$$y_{distorted} = y_{ideal}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (3-2)$$

$$y_{corrected} = y + [2p_2 xy + p_1(r^2 + 2y^2)]$$

采用棋盘格进行测试可以展示如图 3-1 效果：

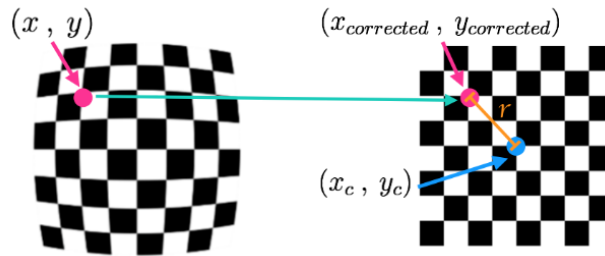


图 3-1 棋盘格畸形矫正实验对比图

对图像的畸变矫正后，本团队对人群聚集应用场景进行再分析。不同光照强度和光照类型，会导致摄像头采集图像时对每帧图像的 RGB 效果产生不同影响并导致基于深度学习的算法在数据进入网络进行实时计算造成影响。为了消除该影响本团队采用图像白平衡算法对实时视频流信息进行逐帧处理。

白平衡，是描述显示器中红、绿、蓝三基色混合生成后白色精确度的一项指标，可以解决色彩还原和色调处理的一系列问题。本方法采用灰度世界法来对输入实时视频流图像进行白平衡优化。算法可大致分为三个步骤：

- 1、计算 R、G、B 三个通道的平均灰度；令  $K = (R_{aver} + G_{aver} + B_{aver})/3$ ，其中  $R_{aver}$ 、 $G_{aver}$ 、 $B_{aver}$  分别表示红、绿、蓝三通道的平均值。
- 2、分别计算三个通道的增益系数， $K_r = K/R_{aver}$ 、 $K_g = K/G_{aver}$ 、 $K_b = K/B_{aver}$ 。
- 3、根据 Von Kries 对角模型，通过计算图像中每个像素的新 R、G、B 值，其值等于原始值乘上增益系数： $R_{new} = R * K_r$ 、 $G_{new} = G * K_g$ 、 $B_{new} = B * K_b$ 。

该算法优势在于采用灰度世界算法对图像进行白平衡，不仅具有良好的效果，同时该算法的执行速度较快，适用于边缘计算设备。

### 3.1.2 实时手势识别算法

在相关工作的基础上，为实现手掌骨骼的实时采集，本文采用了一种基于深度学习的手骨骼点检测算法，该算法结合了谷歌开源代码和 Mediapipe 框架，Mediapipe 框架可以提取每只手上的 21 个关键点，并支持多手检测，根据两者集合实现了高效、精确的手部姿态估计。

下图 3-2 是 Mediapipe 模型的架构。该模型有三个输出，共享一个特征提取器。每个头部都由标记为相同颜色的相应数据集进行训练。



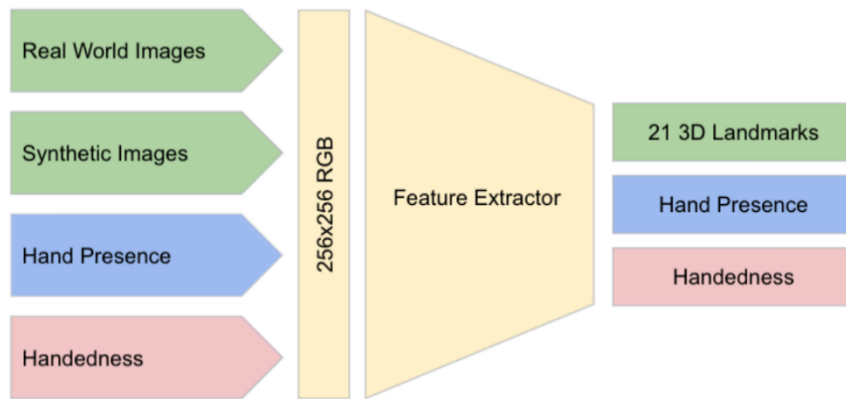


图 3-2 Mediapipe 模型架构

该模型经过大规模的训练和优化，在多样化的手部图像数据集上取得了出色的性能。通过融合卷积神经网络和关键点回归技术，该模型能够准确地预测手部各个关键点的位置。最后将本文的训练模型与 Mediapipe 框架相结合，从而使得手势检测算法具备实时性和高效性。

### (1) 基于深度学习的手骨骼点检测算法

MediaPipe 由手掌检测模型和手部关键部位模型组成。基于 BlazePalm 算法的手掌检测模型生成裁剪良好的手掌图像，然后将其传递给手部模型。其具体流程如下：

- 1) 当第一次输入视频流或手掌第一次出现在输入视频流中时，BlazePalm 检测器模型被调用以确定手掌在视频帧中的位置。
- 2) BlazePalm 检测器将切割的手掌图像传输到 HandLandmark 的手部关键部位模型，确定手掌在 3D 坐标中的 21 个关键点。通过该方法获得手掌的必要关键点信息。
- 3) 从关键点坐标中提取手势的相关特征，以实现手势的识别和分类。

训练过程中使用了预处理数据，使用镜头检测器方法检测手的运动图像，将手的形状检测为完整的手势。在手骨骼点移动中，进行评估，以显示在开始执行时设置的数据集中出现的骨骼点值的适用性。如果从评估结果来看不合适，则将执行调整。在调整过程中，参数执行迭代，直至模型中输出手上的 21 个骨骼点，并将数据点保存在一个包含 64 个组件的文件中（21 个点\*3 个参数+1 个标签）。在数据预处理阶段，图像、类名和手部 Landmark 数据将被存储为 NumPy(.npy)

文件。

通过谷歌开源的训练代码，本文训练出自己的手部骨骼点检测模型，这个模型经过大规模的训练和优化，在多样化的手部图像数据集上取得了出色的性能。通过融合卷积神经网络和关键点回归技术，该模型能够准确地预测手部各个关键点的位置。

## (2) 基于手骨骼点的手势实时识别算法

本研究使用 Mediapipe Hands 从手掌检测模型的结果中提取手部的骨骼点。根据数据集的大小，Landmark 提取过程需要 5 到 7 分钟，包括对图像进行过滤和归一化。

静态手势，依靠手掌的手指延伸和几何信息来做出判断。如下式 3-3 所示，以此判断手指伸展。使用关键点 0 作为坐标原点，计算手掌方向向量和水平方向向量之间的角度，以确定手掌的方向。

$$\begin{cases} (y_i - y_{i+1})(y_{i+1} - y_{i+2})(y_{i+2} - y_{i+3}) > 0 \\ (x_i - x_{i+1})(x_{i+1} - x_{i+2})(x_{i+2} - x_{i+3}) > 0 \end{cases} \quad (3-3)$$

如下图 3-1 所示中，从原点指向第 18 个坐标点的矢量被用作手掌的方向矢量。静态手掌的方向是基于两个向量和旋转角度来确定。

其中  $V_1$  表示手掌的方向矢量， $V_2$  表示手掌的水平矢量， $\alpha$  是手掌的旋转角度。本文更多的采用动态手势。动态手掌的方向是根据手掌的旋转角度确定的。

在图 3-4 中标有  $V_1$ 、 $V_2$  和  $\alpha$  的内容与图 3-3 中的含义相同。对于动态手势，记录的数据不仅包括手掌的伸展和几何信息，还包括动作的时序信息，这些信息形成了关键点坐标的时间序列。为提取动态手势序列的开始和结束状态，可以使用静态手势识别方法进行识别。当手势识别框架识别出手势是动态手势的开始状态时，它自动记录手势动作的序列，直到框架识别出动态手势的结束状态为止。最终，系统停止手势关键点的记录并开始计算。因此，可以通过收集动态手势的关键点序列来确定动态手势的类型，并使用相关方法计算相似度进行分类。

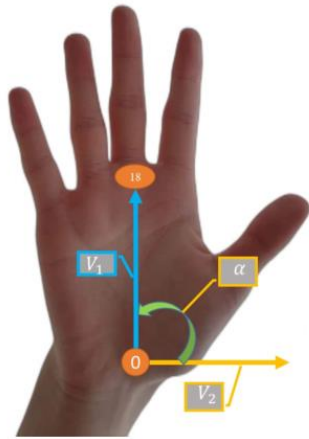


图 3-3 静态手势方向矢量

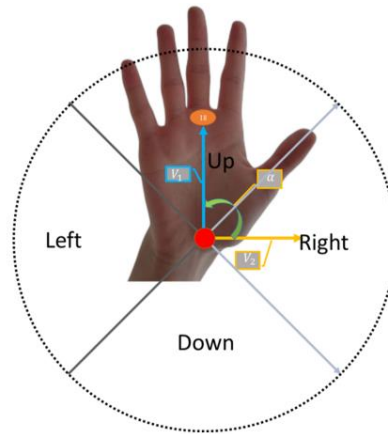


图 3-4 动态手势方向矢量角度

折叠手指的指尖和指关节在手部轮廓中形成曲率，可以通过将圆形模板与手部面具匹配来提取。应用固定大小像素的模板来检测标准化的手罩中的指尖。超过某个阈值的局部最大值被视为指尖候选者，每个指尖候选者都使用简单的启发式规则进行验证。根据检测到的 Landmark，估计手腕线，并对手部区域进行迭代侵蚀，以确定手掌中心。这些 Landmark 可以在视频序列中被跟踪。最后将本文的训练模型与 Mediapipe 框架相结合，从而使得手势检测算法具备实时性和高效性。

### 3.1.3 基于坐标计算手指形态自定义飞行手势指令

#### (1) 基于坐标的手指开合状态算法

通过基于深度学习的手骨骼点检测算法进行手部检测后，手部地标模型通过回归确定了检测到的手部区域中 21 个骨骼点的坐标，如图 3-5 所示。

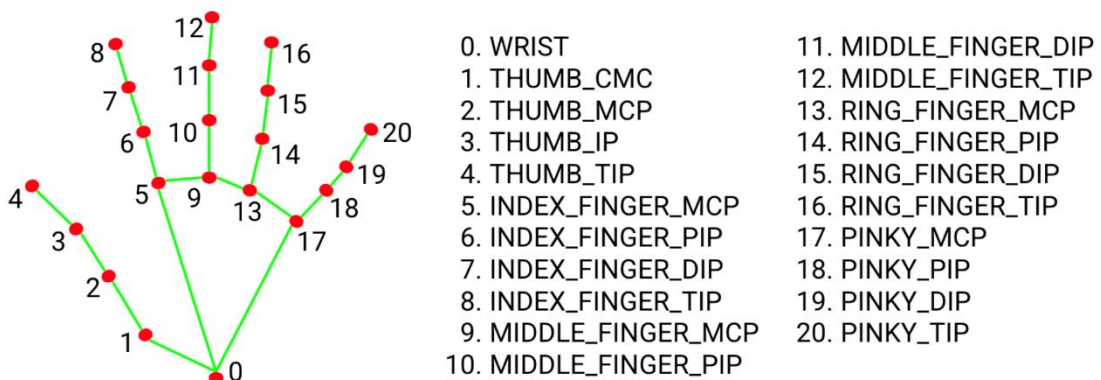


图 3-5 手区域内的 21 个骨骼点定位

根据手部骨骼点的坐标计算每根手指的角度，自定义手势，加快了计算和识别速度。通过骨骼点的坐标信息，可得到不同骨骼点形成的向量，以大拇指为例，选取图 3-5 中 0 号骨骼点与 2 号骨骼点形成的向量，以及 3 号骨骼点与 4 号骨骼点形成的向量，计算两向量间形成的角度，其他手指计算方式均相同。其公式如 3-4 所示：

$$\theta = \arccos \frac{(x_1 * x_2 + y_1 * y_2)}{(\sqrt{x_1^2 + y_1^2} * \sqrt{x_2^2 + y_2^2})} \quad (3-4)$$

再将每个手指骨骼点所形成的向量的角度，用数组保存，与所设定的各手指阈值进行比较。大拇指闭合设定的阈值为  $53^\circ$ ，大于该阈值，大拇指则视为闭合，否则视为张开。其余四指张开的阈值设为  $49^\circ$ ，小于该阈值，视为手指张开；闭合的阈值设为  $65^\circ$ ，大于该阈值，视为手指闭合。

## (2) 基于手指开合状态自定义无人机飞行指令

本文将左右手手指的开合状态组合成为一系列符合人机自然交互标准的无人机飞行指令，该指令的设计基于自然交互标准的动作语言，能极大程度地减少使用遥控器操控无人机的时间成本，快速掌握无人机飞行。

### 1) 基本手势

飞行指令依赖于手势的识别，所以，将详细描述基础手势的手指开合状态，飞行指令中的基础手势如表 3-1 所示。

表 3-1 手势的手指开合状态

手势	手掌各手指开合状态				
	大拇指	食指	中指	无名指	小拇指
one	大于 $53^\circ$	小于 $49^\circ$	大于 $65^\circ$	大于 $65^\circ$	大于 $65^\circ$
ok	大于 $53^\circ$	大于 $65^\circ$	小于 $49^\circ$	小于 $49^\circ$	小于 $49^\circ$
love	小于 $53^\circ$	小于 $49^\circ$	大于 $65^\circ$	大于 $65^\circ$	小于 $49^\circ$

three	大于 53°	小于 49°	小于 49°	小于 49°	大于 65°
five	小于 53°	小于 49°	小于 49°	小于 49°	小于 49°
fist	大于 53°	大于 65°	大于 65°	大于 65°	大于 65°
thumbUp	小于 53°	大于 65°	大于 65°	大于 65°	大于 65°
gun	小于 53°	小于 49°	大于 65°	大于 65°	大于 65°

## 2) 飞行指令

通过左右手基础手势的组合，本文将得到一组符合自然交互的动作语言，以操控无人机飞行，其手势组合飞行指令如表 3-2 所示。

表 3-2 飞行指令

飞行指令	左手手势	右手手势
解锁	ok	ok
上锁	love	ok
上升	one	five
下降	one	fist
向前	gun	gun
向后	fist	fist
向左	thumbUp	five
向右	five	thumbUp
偏航角左转	three	five

偏航角右转	three	fist
紧急制动	love	love

### 3.2 系统实现的硬件构架

#### (1) 嵌入式 AI 开发平台

本作品选用 EAIDK610 作为 AI 算法的部署平台。EAI610 主芯片采用具备高性能 Arm SoC 的 RK3399，同时搭载 OPEN AILAB 嵌入式 AI 开发平台 AID（包含支持异构计算库 HCL、嵌入式深度学习框架 Tengine、以及轻量级嵌入式计算机视觉加速库 BladeCV）。为 AI 应用提供简洁、高效、统一的 API 接口，加速终端 AI 产品的场景化应用落地。其外形及接口设计如图 3-6 所示：

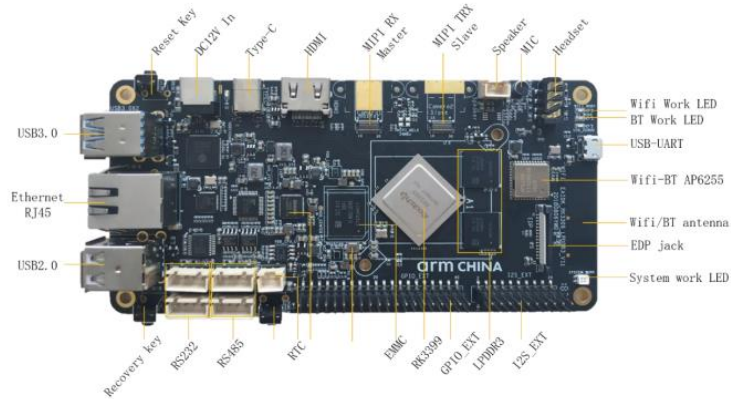


图 3-6 EAIDK-610 平台

#### (2) APM2.8 无人机开源飞行控制器

APM2.8（ArduPilot Mega 2.8）是一款基于开源飞行控制系统的无人机飞控板。它是由 ArduPilot 社区开发的，作为 ArduPilot 项目的一个主要组件之一，旨在为多种类型的无人机提供可靠的飞行控制。APM2.8 内部集成了强大的处理器、传感器和通信模块，以实现稳定的飞行和高级的导航功能。它基于 Atmel AVR 系列的 8 位微控制器，并通过添加外围设备扩展板来支持 GPS 导航、遥控器接收器、电池监测等功能。APM2.8 飞控的接线图如图 3-7 所示。

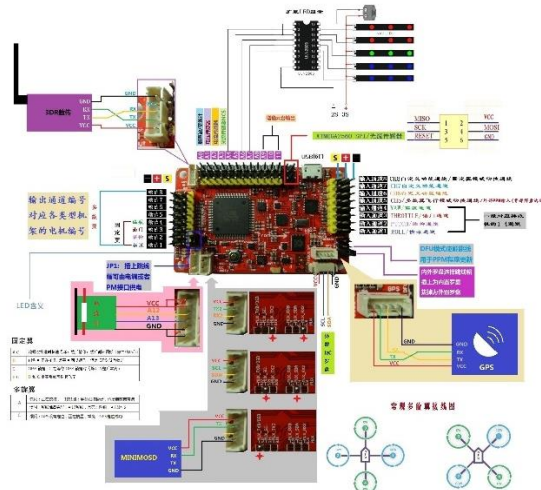


图 3-7 APM2.8 飞控的接线图

### (3) 微控制器模块

对于微控制器的选择，首先处理器性能功能需要满足设计预期，还能配备足够完善的接口设备外设等，方便本文初期的探索性学习。同时能输出多路 PWM 信号连接到无人机飞控，控制无人机的飞行姿态。因此本系统选用功能强大，简单易用的 Arduino 结合 ESP8266 模块作为无人机的远程指令解析模块，其提供的各类接口如下图 3-8 所示。

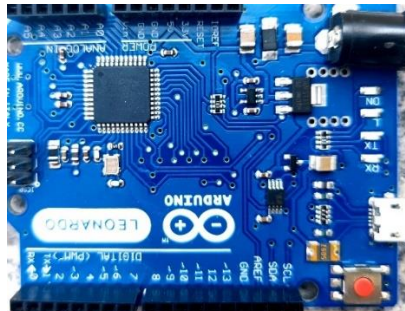


图 3-8 Arduino 微型控制器

Arduino 的 3、5、6、9、10 分别作为 pwm 波形的输出，连接到 apm2.8 的 1~5 输入通道。从而实现对无人机的俯仰角、油门、翻滚角、航向角的控制。

### (4) ESP8266 无线模块

ESP8266 是一款低成本、高集成度的 Wi-Fi 模块，由乐鑫公司开发，具有强

大的网络通信能力，可用于各种应用领域。ESP8266 通过内置的 Wi-Fi 模块，可以轻松地连接到无线网络，并实现与其他设备的互联互通。它支持 2.4GHz 频段的 Wi-Fi 标准，包括 802.11 b/g/n。尽管尺寸小巧，ESP8266 内部集成了处理器、存储器、Wi-Fi 芯片和各种接口，具备强大的计算和通信能力。ESP8266 在设计中考虑了低功耗需求，适用于需要长时间运行或依赖电池供电的应用。ESP8266 提供多个 GPIO 引脚，可以用于连接传感器、执行控制操作，以及和其他外部设备进行通信。ESP8266 使用基于 C 语言的开发环境，提供了丰富的 API 和函数库，使得开发者能够轻松地进行编程和控制。它也支持通过串口与其他设备进行通信。本项目的 ESP8266 模块如下图 3-9 所示。



图 3-9 ESP8266 模块

本作品使用 ESP8266 做无人机远程控制指令解析的前端模块，将接受到的手势指令解析输出为 GPIO 逻辑输出，本作品中通过将 ESP8266 的 D0、D1、D2、D3 接到 Arduino 的 A1、A2、A3、A4 输入口。通过 GPIO 的高低电平的逻辑输出，将指令发送到 Arduino 进行二次解译。该如下表 3-3 所示。

表 3-3 GPIO 的逻辑设计

D0	D1	D2	D3	指令含义
1	1	1	1	解锁
0	0	0	0	上锁
1	0	1	0	紧急制动
1	0	0	0	前进



0	0	0	1	后退
0	1	0	0	向左飞行
0	0	1	0	向右飞行
1	1	0	0	上升
0	0	1	1	下降
1	1	1	0	向左旋转
0	1	1	1	向右旋转
0	1	1	0	悬停

#### (5) 四旋翼无人机平台

四旋翼无人机是一种以四个电动机驱动的垂直起降飞行器，也被称为四轴飞行器。它采用了旋翼作为主要升力装置，并通过电调控制电动机的转速，以实现悬停、横滚、俯仰和偏航等飞行动作。本作品采用的四旋翼无人机结构如下图 3-10 所示。



图 3-10 四旋翼无人机结构

## 第四章 软件设计与流程

### 4.1 系统驱动程序设计

#### 4.1.1 Arduino 驱动程序设计

由于 Arduino 可以输出 PWM 信号，可以通过 Arduino 控制 apm2.8 的输入通道，操控无人机的飞行姿态。不同的 PWM 引脚控制无人机的偏航角（yaw）、俯仰角（pit）、滚动角（roll）、升降油门（thro）、飞行模式（flymod）。其逻辑控制设计如下表 4-1 所示。

表 4-1 逻辑编码图

PWM 引脚	控制通道
3	滚动角
5	俯仰角
6	升降油门
9	偏航角
10	飞行模式

Arduino 对飞控进行驱动的部分程序如下：

```
void loop()
{
    mod();
    ch_width_1 = map(data.roll,      0, 255, 1000, 2000); // 将 0~255 映射到
    1000~2000，即 1ms~2ms/20ms 的 PWM 输出
    ch_width_2 = map(data.pitch,     0, 255, 1000, 2000);
    ch_width_3 = map(data.throttle, 0, 255, 1000, 2000);
    ch_width_4 = map(data.yaw,       0, 255, 1000, 2000);
```

```

ch_width_5 = map(data.flymod,      0, 255, 1000, 2000);

Serial.print("\t");Serial.print(ch_width_1);
Serial.print("\t");Serial.print(ch_width_2);
Serial.print("\t");Serial.print(ch_width_3);
Serial.print("\t");Serial.print(ch_width_4);
Serial.print("\t");Serial.print(ch_width_5);
Serial.println("");

// 将 PWM 信号输出至引脚
ch1.writeMicroseconds(ch_width_1);//写入 us 值
ch2.writeMicroseconds(ch_width_2);
ch3.writeMicroseconds(ch_width_3);
ch4.writeMicroseconds(ch_width_4);
ch5.writeMicroseconds(ch_width_5);
}

```

#### 4.1.2 ESP8266 的远程控制指令接收与解译程序设计

ESP8266 是一款基于 WiFi 的微型开发板，可以用于连接互联网并实现物联网应用。该板子具备处理能力，在许多项目中被用作远程控制设备的核心部分。下面将介绍如何设计一个 ESP8266 的程序，用于接收和解析远程控制指令。

在设计程序之前，本文需要明确需求以便合理地开发系统。根据远程控制指令接收与解译的功能，本文可以列出以下需求：

**接收远程控制指令：**ESP8266 需要能够通过网络接收来自远程设备或服务器的指令。

**解析指令内容：**ESP8266 需要能够解析接收到的指令，并提取出需要执行的操作或参数。

**执行操作：**ESP8266 根据指令内容执行相应的操作，例如控制传感器、执行动作等。

**反馈结果：**ESP8266 需要向远程设备或服务器反馈操作执行的结果。

基于上述需求，本文可以设计一个简单的程序来实现远程控制指令接收与解译功能。以下是程序设计的基本思路：

- (1) 设置 ESP8266 为 Wifi Station 模式，连接到无线网络。

- (2) 创建 UDP 服务器，监听指定的端口，等待远程设备连接。
- (3) 接收远程设备发送的控制指令数据。
- (4) 解析接收到的指令数据，提取出需要执行的操作或参数。
- (5) 根据解析结果执行相应的操作，例如控制 GPIO 引脚读取传感器数值等。
- (6) 获取操作执行结果，并将结果通过网络返回给远程设备。
- (7) 返回结果后继续等待并处理下一个指令。

ESP8266 的远程控制指令接收与解译程序部分代码如下：

```
void loop() {
    // 检测是否接收到消息
    int packageSize = udp.parsePacket();
    if(packageSize < 0){
        currentTime = millis();
        unsigned long elapsedTime = (currentTime - startTime) / 1000;
        if(elapsedTime>1)
        {
            Serial.println(elapsedTime);
            fly_mod('p');
        }
        if(elapsedTime>1)
        {
            Serial.println(elapsedTime);
            fly_mod('p');
        }
    }
    if (packageSize > 0) {
        startTime = millis();
        // 读取发送者信息与数据包内容
        String sender = udp.remoteIP().toString();
        int senderPort = udp.remotePort();
        char packetBuffer[packageSize];
        udp.read(packetBuffer, packageSize);

        // 打印消息内容
        Serial.print("Received message from ");
```

```

Serial.print(sender);
Serial.print(":");
Serial.println(senderPort);
char rx[50];
strncpy(rx, packetBuffer, packageSize);
rx[packageSize]='\0';
// 发送回复
if(strcmp(rx,hello) == 0)
{
    Serial.println("ok");
    String responseStr = "Hello From ESP8266!";
    udp.beginPacket(udp.remoteIP(), udp.remotePort());
    udp.write(responseStr.c_str(), responseStr.length());
    udp.endPacket();
}
else
{
    String responseStr = "ok";
    Serial.print("cmd=");
    char mod;
    Serial.println(rx[0]);
    fly_mod(rx[0]);
    udp.beginPacket(udp.remoteIP(), udp.remotePort());
    udp.write(responseStr.c_str(), responseStr.length());
    udp.endPacket();
}
}
}

```

## 4.2 系统主程序设计

主程序是本作品实现手势远程控制无人机的核心，其主要设计思路如下：

- (1) 设计手部骨骼点检测算法本作品选择谷歌开源的 **mediapipe** 框架结合本文训练的深度学习模型实现其手部骨骼点检测算法。这个算法可以提供准确的手部骨骼点坐标，并且相对较轻量化，适合资源受限的设备使用。
- (2) 设计实时的手势识别算法：本文设计了一个基于手骨骼坐标点的弯曲度和几何特征来进行手势分类的算法。使用手部骨骼点坐标计算手指关节之间的距离、角度等几何特征，并结合这些特征描述手势形状。确保算法具有实时性，避免过度依赖深度学习技术以提高运行速度。
- (3) 根据识别到的特定手势，编写代码将其映射为相应的指令。针对选定的

无人机平台和 API，发送适当的指令，例如起飞、降落、悬停、前进等。将指令打包成 UDP 数据包，并通过 ESP8266 模块的网络连接发送给无人机。

- (4) 无人机安全飞行的设计，本文在程序中正对无人机信号丢失、炸鸡等情况加入了安全性保障设计。使得无人机在出现上述情况时，实现自主降落与自主紧急制动。

主程序流程图如图 4-4 所示：

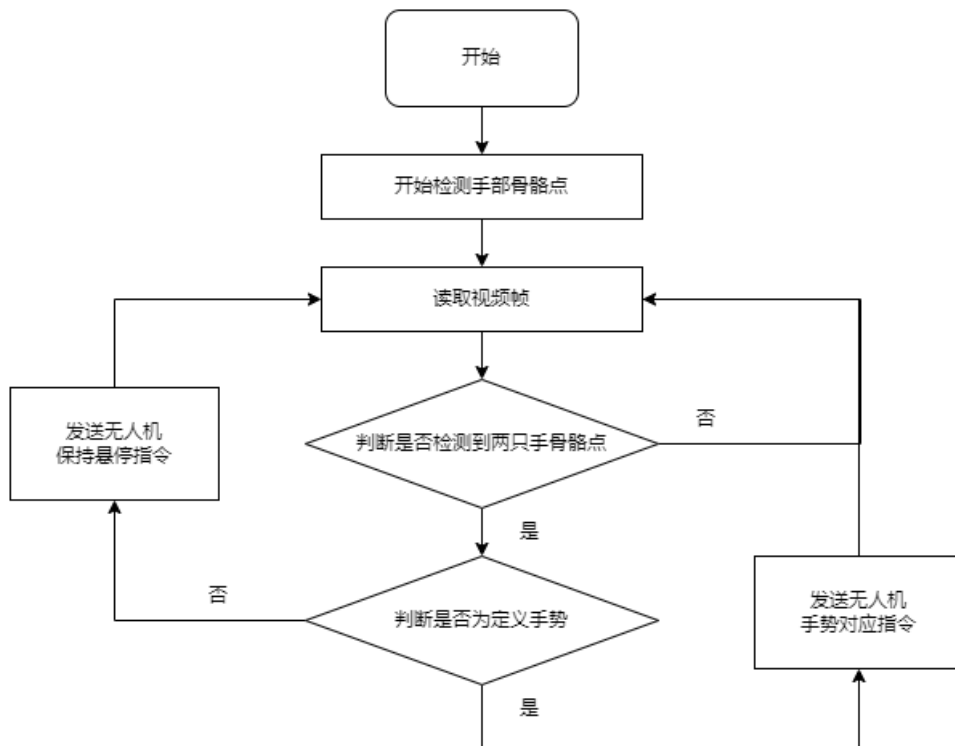


图 5-2 主程序流程图

## 第五章 系统测试与分析

嵌入式系统是软件和硬件的综合体,还包括系统机械装置,外设硬件等。因此嵌入式系统测试其实是将已经集成好的软件系统,在实际运行环境下,对计算机系统从硬件进行一系列测试活动。通过测试找出出现问题的原因或与目标不符的部分,不断对代码设计修改,从而达到最终实现目的,完善各种功能。

本文的作品是一个基于手势自然交互技术的无人机远程控制系统,通过检测手部骨骼点并计算手指的弯曲度来实现特定手势的识别。一旦识别到特定手势,EAIDK-610 可以通过 ESP8266 模块热点连接并利用 UDP 协议与无人机进行通信,向其发送相应的指令。下面是本文整个系统的完成度测试。

在测试过程中本文选择将无人机飞控连接到地面站 Mission Planner,观察无人机飞控的输出参数,作为测试评价指标。

### 5.1 EAIDK610 手势控制终端与无人机通信链路测试

#### (1) 无人机飞控连接至地面站

将无人机的飞控通过 APM2.8 连接至电脑端,选择正确的波特率,点击连接。最终连接成功的画面如下图 5-1。

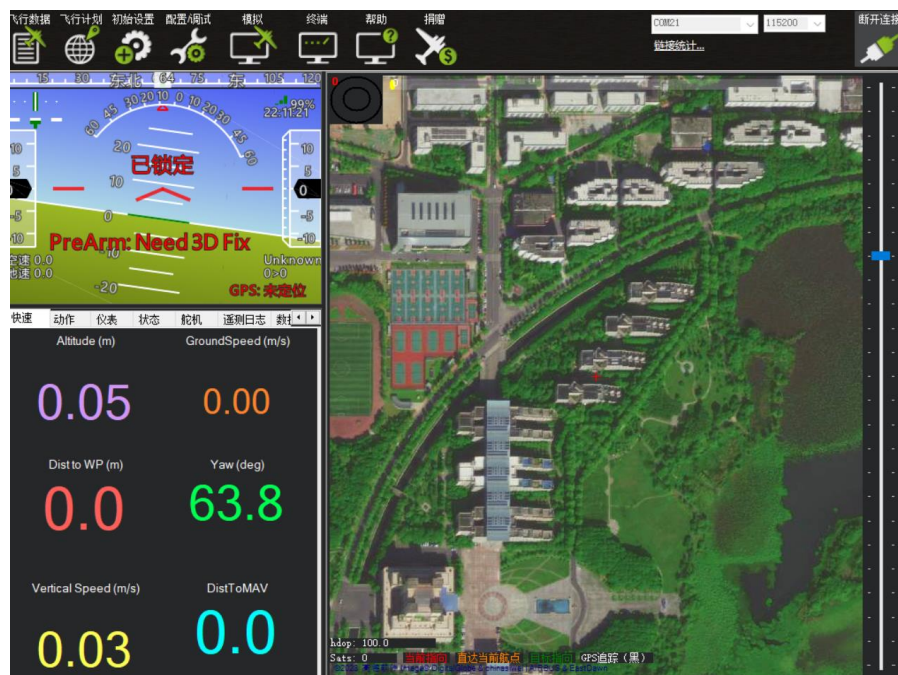


图 5-1 连接地面站

## (2) 运行手势骨骼点检测代码

在 EAIDK-610 上运行手势远程操控代码，处于安全性考虑，在 EAIDK610 没有连接无人机热点的时候，代码无法正常运行。显示效果如下图 5-2：

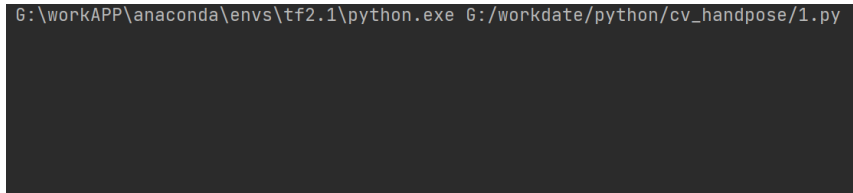


图 5-2 wifi 连接异常

在设备连接到无人机热点后，显示效果如下图 5-3：

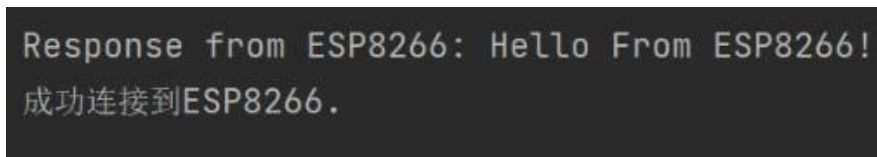


图 5-3 wifi 连接成功

在程序运行的过程中，EAIDK-610 会作为地面站定时向无人机的 ESP8266wifi 模块发送信息，并通过是否收到返回信息确认连接状态，一旦发现连接状态改变就会启动针对无人机的安全措施。wifi 返回信息的 log 打印如下图 5-4：

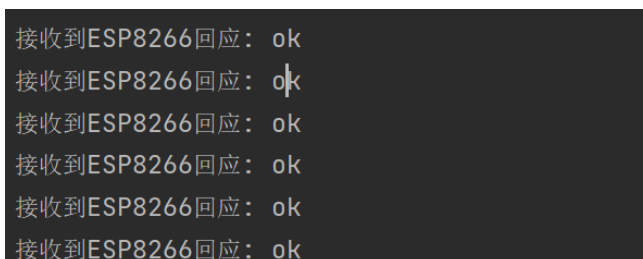


图 5-4 连接确认返回

## 5.2 基于手势识别的无人机远程交互功能测试

在软件测试与分析阶段，处于安全性考虑，本文团队采取无人机去掉桨叶，连接地面站观测飞行数据与电机转动状态的方式，验证本文的手势控制的逻辑性



与操作性。下面是本文开启 EAIDK-610 控制端的手势识别代码后的运行效果。本文设计的四旋翼无人机实验平台如下图 5-5 所示：



图 5-5 四旋翼无人机

### (1) 无人机解锁控制

在无人机开启后，无人机处于为解锁的保护状态，当无人机未解锁是的地面站显示状态如下图 5-6 所示：



图 5-6 无人机未解锁





图 5-9 无人机初始参数

由于自稳模式的飞行模式与手势控制难以匹配，在自稳模式下，手势无法控制无人机的飞行姿态，本文需要做一次无人机前进的手势，将无人机的飞行模式转换为悬停模式。无人机前进的动作设计为两只手模拟手枪向上的姿势。如下图 5-10:

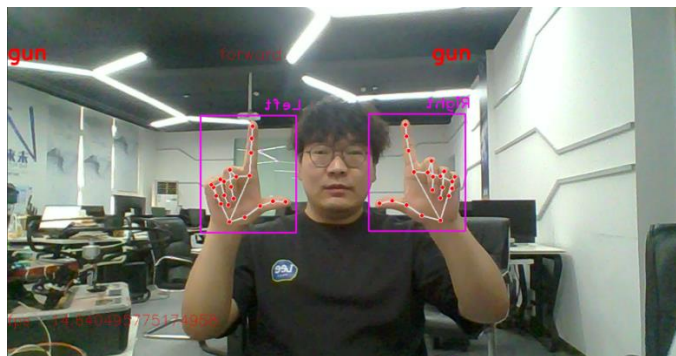


图 5-10 无人机前进手势

此时无人机切换为悬停模式，地面站的参数显示如下，第五通道的舵机量变为悬停模式对应的舵机量。如下图 5-11 所示：

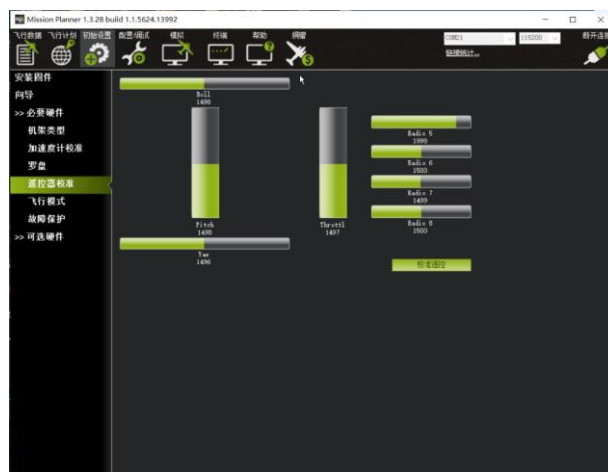


图 5-11 悬停模式

切换悬停模式后，无人机已经可以通过 EAIDK-610 终端的手势远程控制无人机进行飞行姿态的变换，针对无人机的起飞、下降、左转、右转、前进、后退、左飞、右飞设计的交互式手势如下图 5-12 所示：

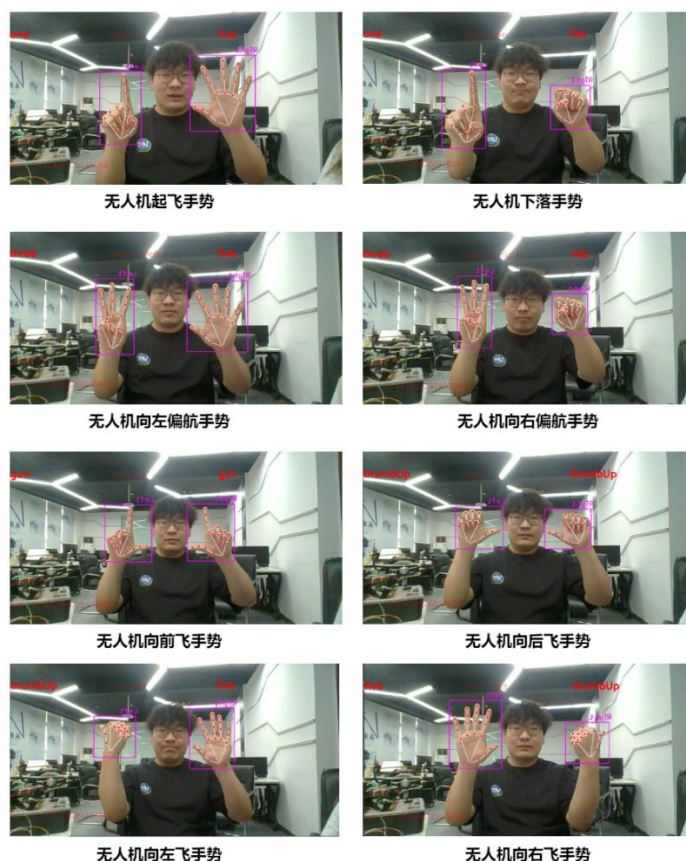
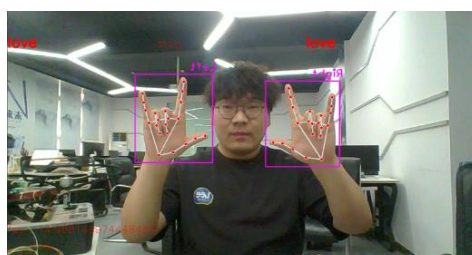


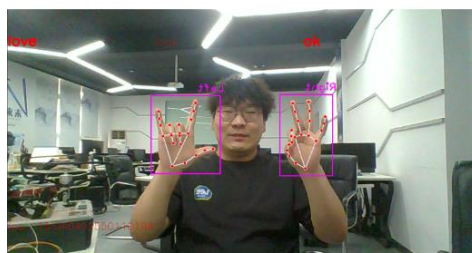
图 5-12 无人机飞行姿态控制手势。







紧急制动手势



无人机上锁手势

图 5-14 紧急制动与上锁

## 第六章 总结

### 6.1 总结

无人机如今已广泛应用在摄影、农业、物流、安全等多个领域。无人机的性能随着通信技术与控制技术的优化也逐渐得到提升,越来越多的新兴技术开始出现在无人机上。但是目前大多数控制无人机的方式依旧停留在遥控器、手机等设备上,其中遥控器是目前无人机最为常用的控制方式。基于遥控设计的无人机操控方式存在着交互复杂、学习难度高等缺点。

随着微小型无人机等产业的快速崛起与发展,越来越多的普通民众开始接触到无人机,对无人机的操控性的便捷化、智能化提出了要求。用户希望花费更少的精力在无人机操控的学习上,而更多专注于无人机的飞行体验。

本文的作品是一个基于手势识别和远程控制的系统,能够通过检测手部骨骼点并计算手指弯曲度来实现特定手势的识别,并通过与无人机的通信控制无人机的飞行。本文选择了谷歌开源的 `mediapipe` 框架和 `ESP8266` 模块进行实现,确保了算法的准确性和稳定性,并且适应了设备的资源限制。通过这个系统,用户可以轻松地使用手势来控制无人机,提高了操作的灵活性和舒适度。

### 6.1 展望

虽然本文的作品已经初步实现了手势识别和远程控制的功能,但仍有一些潜力和改进空间:

**增加更多手势支持:** 当前的系统可以识别特定手势,但可以考虑增加更多手势的支持,让用户能够实现更多种类的飞行动作或指令。

**优化实时性和准确性:** 继续优化手势识别算法,以提高实时性和准确性。可以尝试使用更先进的深度学习技术,如深度神经网络,来提高手势识别的精度和速度。

**增加交互性:** 可以在系统中添加反馈机制,如声音或振动,使用户能够更直观地了解手势识别的状态和控制结果,增加交互性体验。

**扩展到其他应用领域:** 除了无人机控制,本文的系统也可以应用于其他领域,如虚拟现实、智能家居等。进一步研究和拓展这些应用场景,将手势识别技术与更多的物体、设备进行交互。

总之，本文的作品为使用手势操控无人机提供了一个创新的解决方案，但仍有许多方面可以改进和发展。通过持续优化算法和扩展应用领域，本文可以提升系统的功能和用户体验，进一步发掘手势识别技术的潜力。



## 参考文献

- [1] 楚康波,牛芳琳. 基于注意力机制和MobileNet的手势识别算法研究[J]. 电脑知识与技术,2022,18(29):4-7+11.
- [2] 谢松. 基于机器视觉的四旋翼无人机跟踪系统设计[D]. 阜阳师范大学,2022.
- [3] 何磊. 基于深度学习的无人机手势航拍控制[D]. 西安工业大学,2022.
- [4] 廖慧玲. 基于深度学习的静态手势识别与交互[D]. 广东工业大学,2022.
- [5] 樊杰. 面向第一视角人机交互的手势识别方法研究[D]. 电子科技大学,2022.
- [6] 黄永吉,翁发禄,韦辉,侯磊磊. 基于STM32的四旋翼无人机智能控制方法设计[J]. 科技与创新,2022,(02):61-63.
- [7] 张雁鹏,陈世均,高建勇. 基于VR设备中IMU的无手交互方法研究[J]. 计算机测量与控制,2021,29(11):137-141.
- [8] 秦文强. 基于双目视觉的植物定位和测量算法研究与实现[D]. 内蒙古大学,2021.
- [9] 魏加立,曲慧东,王永宪,朱俊青,关英俊. 基于飞行时间法的3D相机研究综述[J]. 红外技术,2021,43(01):60-67.
- [10] 王博. 四旋翼无人机飞控系统硬件平台及底层软件设计[D]. 哈尔滨工程大学,2021.
- [11] 刘璇恒. 基于多模态手势交互的无人机操控方法研究[D]. 天津大学,2020.
- [12] 巩彦丽. 基于虚拟现实的远程双目视觉系统设计[D]. 浙江理工大学,2020.
- [13] 王妙云. 煤矿井下四旋翼无人机虚拟远程操控关键技术研究[D]. 西安科技大学,2020.
- [14] 王晓卫,李维宝,孟越. 基于虚拟现实的无人机智能监控系统综述[J]. 飞航导弹,2020,(04):26-29+51.
- [15] 陈华,王立军,刘刚. 立体匹配算法研究综述[J]. 高技术通讯,2020,30(02):157-165.
- [16] 蔡成林. 基于视觉的无人机人体姿态指令识别系统的设计与实现[D]. 国防科技大学,2019.
- [17] 李卓文. 无人机地面站管控系统的研究与开发[D]. 重庆邮电大学,2019.
- [18] 李鹏飞. 基于无人机三维视觉的电力巡线系统设计与实现[D]. 东北农业大学,2019.
- [19] 张开军. 基于深度学习的视觉目标检测与识别关键技术及应用[D]. 南京大学,2019.
- [20] 郭锦辉. 手势识别及其在人机交互系统中的应用[D]. 华南理工大学,2019.
- [21] 成孝刚,彭雅琴,谢世朋,闫静杰,熊健. 基于视觉传感的无人机手势交互研究[J]. 电子测试,2015,(20):19-20.