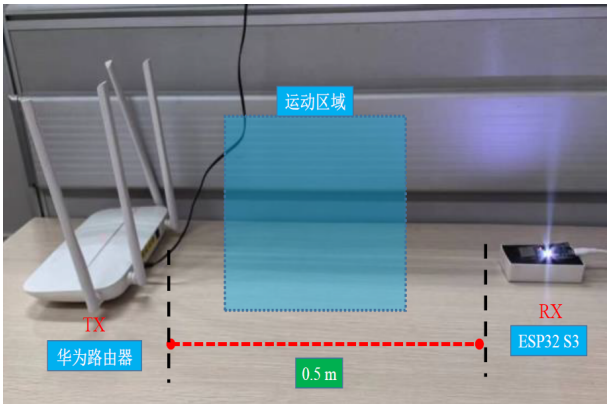
基于WI-FI信号的人体感知

以下基于本人所做工作时使用的方法，其中的参数和使用的方法后续仍需修改、优化。

本系统分为数据采集、数据预处理、特征提取和学习识别四个部分。

数据采集



其中路由器为WI-FI信号发射端，ESP-S3为接收端。

发射端路由器只要处于正常开启状态即可。

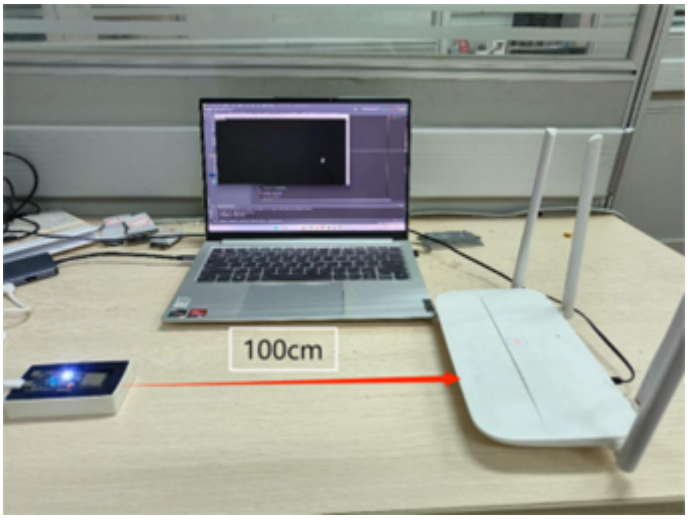
接收端ESP32-S3需要经过烧录连接电脑才能接受WI-FI信号。烧录教程请参考https://github.com/espressif/esp-csi。电脑需要安装esp-csi-master命令行，文件及说明在esp-csi-master中。

采集的代码为mulitiSerials.py和sametimeProcess.py，直接运行即可采集数据。数据为15秒一组。**在采集数据时需要创建名称为动作标签的文件夹，同时在更改采集代码中的CSV储存地址。**

采集到的数据格式如下所示：

type	seq	timestamp	taget_seq	taget	mac	rssi	rate	sig_mode	mcs	cwb	smoothing	not_sounding	aggregation	stbc	fec_coding	sgi	noise_floor	ampdu_cnt	channel_primary	channel_second	local_timestamp	ant
CSI_DATA	0	3162	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	2947769	0
CSI_DATA	1	3179	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	2961077	0
CSI_DATA	2	3244	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3032452	0
CSI_DATA	3	3256	0	unknown	d0:d7:83:e	-72	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3042207	0
CSI_DATA	4	3269	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3047836	0
CSI_DATA	5	3282	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3048646	0
CSI_DATA	6	3297	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3049825	0
CSI_DATA	7	3309	0	unknown	d0:d7:83:e	-73	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3061797	0
CSI_DATA	8	3321	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3071288	0
CSI_DATA	9	3332	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3080887	0
CSI_DATA	10	3344	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3081922	0
CSI_DATA	11	3356	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3086041	0
CSI_DATA	12	3371	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3086837	0
CSI_DATA	13	3386	0	unknown	d0:d7:83:e	-70	11	1	3	1	0	1	0	1	0	0	-97	0	6	2	3089130	0
CSI_DATA	14	3400	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3096703	0
CSI_DATA	15	3414	0	unknown	d0:d7:83:e	-72	11	1	5	1	0	1	0	1	0	0	-97	0	6	2	3114765	0
CSI_DATA	16	3434	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3131448	0
CSI_DATA	17	3446	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3133376	0
CSI_DATA	18	3458	0	unknown	d0:d7:83:e	-72	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3133765	0
CSI_DATA	19	3472	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3134217	0
CSI_DATA	20	3486	0	unknown	d0:d7:83:e	-70	11	1	2	1	0	1	0	1	0	0	-97	0	6	2	3134856	0
CSI_DATA	21	3501	0	unknown	d0:d7:83:e	-72	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3147303	0
CSI_DATA	22	3513	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3148965	0
CSI_DATA	23	3525	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3151001	0
CSI_DATA	24	3539	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3152045	0
CSI_DATA	25	3552	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3161267	0
CSI_DATA	26	3564	0	unknown	d0:d7:83:e	-71	11	1	4	1	0	1	0	1	0	0	-97	0	6	2	3174967	0

为尽量减轻环境因素变化带来的影响，我们在桌面采集动作时的环境设计为



为保证效果尽量在连线处做动作。我们初步设计的动作为：人不在电脑前、人在电脑前注视屏幕但无动作、人在电脑前敲击键盘、人在电脑前移动鼠标。动作幅度尽量大，初期每种数据采集150组。根据实验效果来确定后续的动作设计。

数据预处理

采集到的数据当中data列为子载波的信息，我们需要使用process_raw_data.py将数据转换为复数形式并且提取数据中的振幅与相位信息。

处理得到的数据如下所示：

data	timeSever	complex_data	amp	phase
[-10, 0, -1	02:24.7	[(-10+0j), (-10-1j), (-9-1j), (-10-2j), (-11-2j	[10.0, 10.0499, 9.0554, 10.198, 11.1803, 11.1803, 1	[3.1416, -3.0419, -3.03
[2, 8, 2, 8,	02:24.7	[(2+8j), (2+8j), (1+9j), (1+10j), (1+10j), (1+1	[8.2462, 8.2462, 9.0554, 10.0499, 10.0499, 10.049	[1.3258, 1.3258, 1.4601
[7, -6, 7, -	02:24.8	[(7-6j), (7-5j), (8-5j), (8-5j), (9-5j), (9-5j), (9-	[9.2195, 8.6023, 9.434, 9.434, 10.2956, 10.2956, 1(	[-0.7086, -0.6202, -0.5
[-9, 3, -9,	02:24.8	[(-9+3j), (-9+3j), (-9+2j), (-10+2j), (-10+2j),	[9.4868, 9.4868, 9.2195, 10.198, 10.198, 11.1803, 1	[2.8198, 2.8198, 2.9229
[1, -9, 2, -	02:24.8	[(1-9j), (2-9j), (2-9j), (3-10j), (3-10j), (3-10j)	[9.0554, 9.2195, 9.2195, 10.4403, 10.4403, 10.440	[-1.4601, -1.3521, -1.3
[3, 7, 2, 8,	02:24.8	[(3+7j), (2+8j), (2+9j), (1+9j), (1+9j), (1+10j)	[7.6158, 8.2462, 9.2195, 9.0554, 9.0554, 10.0499, 1	[1.1659, 1.3258, 1.3521
[-5, -8, -5	02:24.8	[(-5-8j), (-5-8j), (-4-9j), (-4-9j), (-4-10j), (-	[9.434, 9.434, 9.8489, 9.8489, 10.7703, 10.7703, 11	[-2.1294, -2.1294, -1.9
[-9, 1, -9,	02:24.8	[(-9+1j), (-9+1j), (-9+1j), (-10+0j), (-10+0j),	[9.0554, 9.0554, 9.0554, 10.0, 10.0, 11.0, 11.0, 11.0	[3.0309, 3.0309, 3.0309
[9, 1, 8, 2,	02:24.8	[(9+1j), (8+2j), (8+3j), (8+3j), (9+3j), (10+3j)	[9.0554, 8.2462, 8.544, 8.544, 9.4868, 10.4403, 10.	[0.1107, 0.245, 0.3588,
[-10, 0, -1	02:24.8	[(-10+0j), (-10-1j), (-10-1j), (-11-2j), (-11-	[10.0, 10.0499, 10.0499, 11.1803, 11.1803, 11.180	[3.1416, -3.0419, -3.04
[-8, 4, -9,	02:24.9	[(-8+4j), (-9+4j), (-9+4j), (-10+4j), (-10+4j)	[8.9443, 9.8489, 9.8489, 10.7703, 10.7703, 11.7047	[2.6779, 2.7234, 2.7234
[3, -9, 3, -	02:24.9	[(3-9j), (3-9j), (3-9j), (4-9j), (5-10j), (5-10j),	[9.4868, 9.4868, 9.4868, 9.8489, 11.1803, 11.1803,	[-1.249, -1.249, -1.249
[0, -10, 0,	02:24.9	[-10j, -9j, (1-10j), (1-10j), (1-10j), (1-11j), (1	[10.0, 9.0, 10.0499, 10.0499, 10.0499, 11.0454, 11.	[-1.5708, -1.5708, -1.4
[6, -7, 6, -	02:24.9	[(6-7j), (6-7j), (7-7j), (7-7j), (8-6j), (8-7j), (7-	[9.2195, 9.2195, 9.8995, 9.8995, 10.0, 10.6301, 9.8	[-0.8622, -0.8622, -0.7
[8, 0, 8, 1,	02:24.9	[(8+0j), (8+1j), (8+2j), (8+2j), (9+2j), (9+2j),	[8.0, 8.0623, 8.2462, 8.2462, 9.2195, 9.2195, 10.19	[0.0, 0.1244, 0.245, 0.24
[3, -8, 3, -	02:24.9	[(3-8j), (3-8j), (4-8j), (4-9j), (5-9j), (5-9j), (5-	[8.544, 8.544, 8.9443, 9.8489, 10.2956, 10.2956, 10	[-1.212, -1.212, -1.107
[0, -9, 1, -	02:24.9	[-9j, (1-9j), (1-10j), (2-10j), (2-11j), (2-10j),	[9.0, 9.0554, 10.0499, 10.198, 11.1803, 10.198, 11.	[-1.5708, -1.4601, -1.4
[8, -2, 8, -	02:25.0	[(8-2j), (8-1j), (9-1j), (9+0j), (10+0j), (10+0j)	[8.2462, 8.0623, 9.0554, 9.0, 10.0, 10.0, 10.0, 11.04	[-0.245, -0.1244, -0.11
[-2, 9, -2,	02:25.0	[(-2+9j), (-2+8j), (-3+9j), (-3+9j), (-4+9j), (-	[9.2195, 8.2462, 9.4868, 9.4868, 9.8489, 10.7703, 1	[1.7895, 1.8158, 1.8925
[1, 8, 1, 8,	02:25.0	[(1+8j), (1+8j), 9j, 10j, (-1+10j), 10j, (1+10j),	[8.0623, 8.0623, 9.0, 10.0, 10.0499, 10.0, 10.0499,	[1.4464, 1.4464, 1.5708
[7, 4, 7, 5,	02:25.0	[(7+4j), (7+5j), (7+5j), (7+6j), (7+7j), (7+7j),	[8.0623, 8.6023, 8.6023, 9.2195, 9.8995, 9.8995, 10	[0.5191, 0.6202, 0.6202
[6, 6, 6, 6,	02:25.0	[(6+6j), (6+6j), (6+7j), (6+8j), (6+8j), (6+8j),	[8.4853, 8.4853, 9.2195, 10.0, 10.0, 10.0, 9.8995, 1	[0.7854, 0.7854, 0.8622
[-7, 5, -7,	02:25.0	[(-7+5j), (-7+5j), (-8+5j), (-8+6j), (-9+6j), (-	[8.6023, 8.6023, 9.434, 10.0, 10.8167, 10.2956, 10.	[2.5213, 2.5213, 2.583,
[-9, 1, -9,	02:25.0	[(-9+1j), (-9+1j), (-9+0j), (-10+0j), (-10+0j),	[9.0554, 9.0554, 9.0, 10.0, 10.0, 11.0, 11.0, 11.0454	[3.0309, 3.0309, 3.1416
[8, 2, 8, 3,	02:25.1	[(8+2j), (8+3j), (7+4j), (7+5j), (8+5j), (8+5j),	[8.2462, 8.544, 8.0623, 8.6023, 9.434, 9.434, 9.848	[0.245, 0.3588, 0.5191,
[6, -7, 6, -	02:25.1	[(6-7j), (6-7j), (6-7j), (7-7j), (8-7j), (8-7j), (7-	[9.2195, 9.2195, 9.2195, 9.8995, 10.6301, 10.6301,	[-0.8622, -0.8622, -0.8
[-3, -9, -2	02:25.1	[(-3-9j), (-2-9j), (-2-10j), (-2-10j), (-1-11j),	[9.4868, 9.2195, 10.198, 10.198, 11.0454, 11.0454,	[-1.8925, -1.7895, -1.7

理论来讲我们提取到的data中有104个数据，组成52个子载波的信息，但一部分实际测量到的数据多于104，我们在此只是简单的提取前104个。

其中complex_data列是处理后的复数形式，amp列是计算得到的幅度信息，phase是计算得到的相位信息。

处理后我们使用Hampel滤波去除离群点。

由于人体动作频率较低，再去除噪声时我们要去除环境中的高频噪声。因此我们使用低通滤波器，截止频率设置为5Hz。

离群点去除和低通滤波的具体代码为process_data.py。

特征提取

计划书中特征提取使用的PCA方法，我们使用PCA.py对于个别具体的数据进行处理可以得到前3主特征就能包含几乎所有的信息。因此我们只保留前3个特征的信息。

本实验中csv文件的读取需要大量的时间，因此我们为了后续特征学习的高效性，我们将文件中的数据输出到nparray文件中。export_array.py包括了数据的预处理和数组输出。

Concatenate_nparray可以将同一名称的amp数组和phase数据结合起来。

学习识别

在后续的识别过程中我们选择机器学习和LSTM、GRU。SVM_model.py中使用了SVM来学习。LSTM中使用LSTM来学习，每个样本的数据规模为1500*3，样本标签为文件夹名称。