gesture\_dataset/  
├── gesture\_0/ # 手势0的视频样本  
│ ├── gesture\_0\_sample\_1.avi  
│ ├── gesture\_0\_sample\_2.avi  
│ └── ...  
├── gesture\_1/ # 手势1的视频样本  
├── ...  
├── gesture\_6/ # 手势6的视频样本  
├── features.npy # 提取的特征数据  
├── labels.npy # 对应的标签数据  
├── normalization\_params.json # 归一化参数  
└── metadata.json # 数据集元数据

.2 关键点提取

使用MediaPipe提取的21个手部关键点结构：  
[https://mediapipe.dev/images/mobile/hand\_landmarks.png](https://mediapipe.dev/images/mobile/hand_landmarks.png" \o "" \t "https://chat.deepseek.com/a/chat/s/_blank)

每个关键点包含三维坐标：

x：水平位置 (0-1，0=左边缘，1=右边缘)

y：垂直位置 (0-1，0=上边缘，1=下边缘)

z：深度值 (值越小越靠近摄像头)

2.3 特征矩阵

每个视频样本的特征矩阵维度为：  
(T, 21, 3)

T：视频帧数 (约150帧)

21：手部关键点数量

3：每个关键点的坐标 (x, y, z)

3. 数据标准化/归一化

3.1 为什么需要标准化？

消除量纲差异：不同坐标轴的值范围不同

加速模型收敛：标准化数据使优化过程更高效

提高模型准确性：防止某些特征主导模型

增强泛化能力：使模型对输入变化更鲁棒

3.2 标准化方法

方法1：Min-Max归一化

公式：$X\_{\text{norm}} = \frac{X - X\_{\min}}{X\_{\max} - X\_{\min}}$

优点：

将数据缩放到[0,1]范围

保持原始数据分布形状

对异常值有一定鲁棒性

方法2：Z-Score标准化

公式：$X\_{\text{std}} = \frac{X - \mu}{\sigma}$

优点：

将数据转换为均值为0、标准差为1的分布

适用于大多数机器学习算法

对线性模型特别有效

3.3 标准化实现

本项目中使用的Min-Max归一化代码：

def min\_max\_normalize(sequence):

# 转换为(帧数, 21\*3)的形状

original\_shape = sequence.shape

sequence = sequence.reshape(original\_shape[0], -1)

# 计算每列的最小值和最大值

min\_vals = np.min(sequence, axis=0)

max\_vals = np.max(sequence, axis=0)

# 避免除以零

range\_vals = max\_vals - min\_vals

range\_vals[range\_vals == 0] = 1

# 归一化

normalized = (sequence - min\_vals) / range\_vals

# 恢复原始形状

return normalized.reshape(original\_shape), min\_vals, max\_vals

3.4 归一化效果对比

| 特征 | 原始值范围 | 归一化后范围 |
| --- | --- | --- |
| 手腕x | [0.12, 0.85] | [0.0, 1.0] |
| 食指尖y | [0.05, 0.92] | [0.0, 1.0] |
| 中指z | [-0.5, 0.3] | [0.0, 1.0] |

4. 使用说明

运行data\_collection.py采集手势视频

运行feature\_extraction.py提取特征并标准化

使用features.npy和labels.npy训练手势识别模型

text

复制

下载

### 系统使用说明

1. \*\*数据采集\*\*：

- 运行 `python data\_collection.py`

- 按提示采集0-6手势视频

- 每个手势采集50个样本，每个样本5秒

2. \*\*特征提取\*\*：

- 运行 `python feature\_extraction.py`

- 自动提取关键点并进行标准化

- 生成特征文件(features.npy)和标签文件(labels.npy)

3. \*\*模型训练\*\* (示例)：

```python

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

# 加载数据

features = np.load("gesture\_dataset/features.npy", allow\_pickle=True)

labels = np.load("gesture\_dataset/labels.npy")

# 转换为2D矩阵 (样本数, 帧数\*关键点数\*3)

X = np.array([x.flatten() for x in features])

y = np.array(labels)

# 分割数据集

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 训练SVM分类器

clf = SVC(kernel='rbf', C=10, gamma=0.001)

clf.fit(X\_train, y\_train)

# 评估

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"模型准确率: {accuracy:.2f}")

技术要点说明

数据标准化重要性：

不同用户的体型差异导致绝对坐标不同

摄像头位置变化影响坐标范围

归一化使模型关注手势形状而非位置

MediaPipe关键点选择：

使用21个手部关键点捕捉手势细节

保留z坐标捕捉深度信息

只处理单只手简化问题

时序处理：

保留完整的时序信息

后续可使用LSTM等时序模型处理

也可提取帧间差分特征增强动态信息