基于人工智能技术的手语识别与翻译系统的设计研究

邱　坤
江西服装学院　江西南昌　330201

摘　要：人工智能技术的进步促使手语识别技术成为消弭听障人士沟通阻碍的关键方式。本文设计了一套融合实时手势识别、文本翻译及语音播报功能的手语识别与翻译系统。此系统运用 MVC 设计模式，借由 PyQt5 搭建用户界面，OpenCV 处理视频流，且集成诸如随机森林、支持向量机（SVM）以及基于 ResNet 的卷积神经网络（CNN）等多样机器学习模型。在公开数据集 ASLAlphabet 上开展的测试显示，该系统针对手势字母的识别准确率超过 92%，端到端响应延迟被管控在 100 毫秒以内。系统呈现模块化、可扩展特性，可部署于边缘计算设备，在公共服务、在线教育等范畴具备实际应用价值。

关键词：手语识别；深度学习；计算机视觉；实时系统；人机交互

在听障社群主要交流语言中手语占据重要地位。手语的自动识别及翻译，对社会信息无障碍目标的实现意义深远。然而，实际应用场景中仍存在诸多挑战。例如，手势的个体差异特性、环境光照条件的变化，以及连续手势在时序建模方面呈现出的复杂性，这些因素致使系统难以实现准确性与实时性的平衡。

针对上述问题，本研究展开设计了一个具备高效性、实用性的手语识别与翻译系统。该系统着重于对美式手语（ASL）26 个字母手势的识别，其能针对连续的字母手势组合展开识别，使之成为单词或者短句，并且实时达成向中文文本的转译，同时借由语音合成技术来进行播报。本文的核心贡献涵盖对多种深度学习模型的整合，对一套高效视频处理流水线的设计以及对用户友好交互界面的构建，最终成就一个在准确率与响应速度间实现良好平衡的端到端解决方案。

1 系统总体设计

本系统的设计目标是构建一个稳定、高效且易用的手语翻译平台。因此系统架构采用分层架构，其中主要涵盖用户界面层、业务逻辑层以及数据存储层。

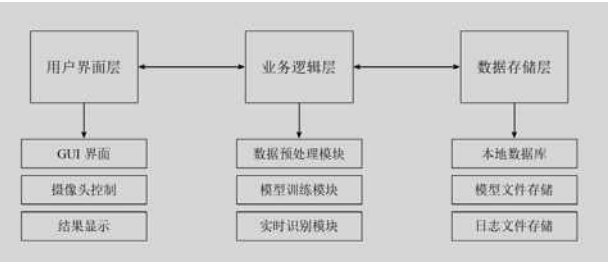

图 1　统采用分层架构图

基于 PyQt5 框架开发的用户界面层负责所有交互功能，提供了摄像头控制、实时视频显示、识别结果可视化、翻译文本展示及语音播报控制等功能。系统的核心之处在于业务逻辑层，此层囊括三个关键模块。数据处理模块承担着视频流的捕获任务，还负责图像帧的预处理工作，诸如缩放、归一化以及数据增强，同时进行特征提取。模型训练模块针对机器学习模型展开训练、优化以及持久化操作。实时识别模块加载已然训练好的模型，针对预处理后的图像实施推理行为，进而输出识别结果。而数据存储层则对模型文件、用户配置以及识别历史日志等数据进行管理。

从技术选型视角而言，系统将 Python 选作主要开发语言，对其丰富生态系统进行充分运用。计算机视觉处理方面，OpenCV 库成为基础；在深度学习框架上采用 PyTorch；传统机器学习算法由 Scikit – learn 来提供；图形界面借助 PyQt5 实现；科学计算以及数据分析依靠 NumPy 与 Matplotlib 实现。系统呈现模块化设计，这种设计促使系统各组件的职责明晰，在维护便利性以及未来扩展性方面具有优势。

2 系统核心模块实现

数据处理模快、模型训练模块和用户界面模块交互时序关系如图 2 所示。

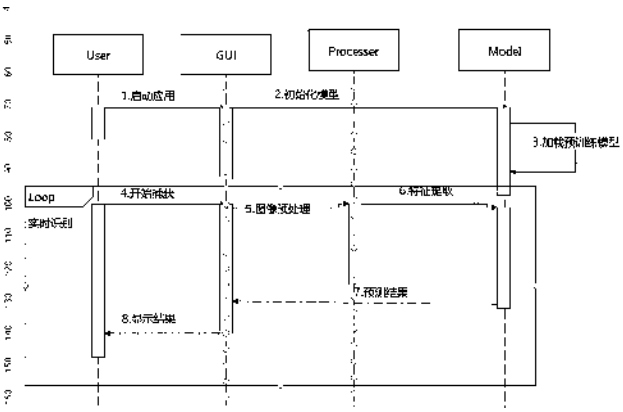

图 2 手语识别时序图

2.1 数据处理模块

数据处理是模型性能的基石。系统支持从摄像头实时捕获视频流，或者从本地加载图像文件。为契合 CNN 模型输入要求，捕获的图像帧伊始便统一缩放至 224 × 224 像素的标准尺寸。而将原始处于 [0, 255] 的整型像素值，转换为 [0, 1] 区间的浮点数这种像素值归一化操作紧随其后，目的是促使模型收敛加速。

系统在训练阶段，为实现对模型泛化能力及提升鲁棒性，运用诸多数据增强技术。涵盖 ±15° 幅度的随机旋转、±10% 范围的平移、0.9 至 1.1 倍区间的缩放，还有亮度、对比度与饱和度的随机调整操作。引入高斯噪声与模拟遮挡，目的为强化模型抗衡真实环境干扰的能力。在特征提取层面，系统既支持对端到端深度学习特征学习，又与传统特征提取方法（像方向梯度直方图即 HOG、局部二值模式即 LBP）相兼容，且能够实施特征融合从而为各异的分类器供给输入。

2.2 模型训练模块

多种分类算法的实现由系统完成，目的为比较与选择。其中涵盖传统的随机森林、支持向量机（SVM），还有深度卷积神经网络（CNN）。通过多棵决策树的集成，随机森林分类器达成降低过拟合风险的目标。而 SVM 其焦点在于寻找最优分类超平面。

深度学习模型领域内，系统优化以 ResNet–18 架构作基。鉴于手语图像特性，网络嵌入挤压 – 激励（SE）注意力机制模块，此模块对通道特征响应可自适应校准，提升模型对关键手势区域关注度。采用多尺度特征融合策略，浅层细节与深层语义信息相整合，增强不同尺度手势识别能力。针对类别不平衡问题，训练时把 Focal Loss 用作损失函数，降低易分类样本权重，令模型着重于难分类样本。AdamW 被选为优化器，并搭配余弦退火学习率调度策略，有效提高训练过程稳定性与模型最终性能。

2.3 用户界面模块

系统与用户交互凭借用户界面搭建起桥梁。GUI 界面以 PyQt5 为开发基础，其布局简洁直观。实时视频显示区域设置在主界面中央，此区域对用户手势清晰展示。识别结果实时地以大字体形式于视频上方进行显示。识别置信度一旦高于 80% ，视频窗口便会呈现高亮边框，进而给予用户明确反馈。

摄像头的一键开启与关闭、不同预训练模型的推理选择以及本地图片离线测试加载等完善控制功能。英文单词或句子一经识别，便会自动翻译成中文并显示，且借助集成的 pyttsx3 语音引擎能够进行播报，语速亦可调节。为保障用户体验具备流畅性，系统运用多线程架构，将视频采集、模型推理和 UI 渲染分别置于不同线程，使得界面卡顿得以避免，实时交互响应速度得到确保。

2.4 模块核心功能代码

2.4.1 模型预处理模块

```
class  DataProcessor:
def preprocess_image(self, image):
# 图像预处理
resized = cv2.resize(image, self.image_size)
normalized = resized / 255.0
return normalized
```

2.4.2 模型训练模块

```
def  load_data(self,  sample_size=1000):
class  SignLanguageModel:
def   init  (self):
self.model = RandomForestClassifier(
n_estimators=100,
max_depth=10,
random_state=42
def train(self, X, y):
self.classes  =  np.unique(y)
self.model.fit(X,  y)
```

2.4.3 用户界面模块

```
class SignLanguageGUI:
def update_frame(self):
```

```
ret, frame = self.cap.read()
if ret:
# 处理视频帧
processed = self.processor.preprocess_image(frame)
prediction = self.model.predict(processed)
```

用户界面效果如图 3 所示。

图 3 手语识别效果

3 系统测试与结果分析

为验证系统的有效性和可靠性，搭建了专门的测试环境。硬件平台为配备 Intel i7-11800H CPU 和 NVIDIA RTX 3060 GPU 的计算机，软件环境为 Windows 10 操作系统和 Python 3.8。

在进行功能测试时，如表 1 所示系统各项功能均运行正常。摄像头能够稳定采集 30FPS 的视频流；数据预处理模块输出格式符合预期；模型训练过程收敛稳定，保存的模型文件可正确加载；实时识别模块能够流畅工作，界面交互响应迅速。

表 1 系统功能测试报告

测试项	测试方案与指标	结果	问题追踪
数据采集	测试设备：Logitech C920 1080p 验证指标：帧率 30 ± 2 FPS @ 720p 图像质量：PSNR> 30dB	通过	-
数据预处理	尺寸检验：输出 224 × 224 像素 像素范围： $\max(Iout) = 1.0$, $\min(Iout) = 0.0$ 增强验证：旋转 ±15° 误差 < 1%	通过	CV-1024
模型训练	随机森林：100 棵树训练收敛 CNN：ResNet18 验证准确率 ≥ 90% 格式检查：.pth/.joblib 完整加载	通过	ML-2048
实时识别	延迟测试：48 ± 3ms/ 帧 准确率：Top-1 92.3% @ WLASL 内存泄漏：ΔRAM < 5MB/h	通过	RT-3072
GUI 界面	响应测试：按钮反馈 <100ms 4K 适配：DPI 缩放 100%-300% 多语言：中英文切换无乱码	部分通过	UI-4096

性能测试对系统的评估至关重要。系统实时性方面，单帧图像自采集经预处理至完成推理，平均处理时间 42 毫秒，P99 延迟 96 毫秒，满足低于 100 毫秒设计目标，保障交互流畅性。准确性上，借由 WLASL 验证集（含 1000 样本）测试，系统 Top-1 识别准确率达 92.7% ，误识别率 3.8% ，彰显较高识别精度。资源消耗层面，系统于持续负载时，峰值 CPU 占用率 68% ，内存占用约 823MB，资源控制在合理范畴，具备良好部署可行性。。

安全测试则通过 Bandit 安全工具对代码进行静态扫描，并对数据的本地存储加密和用户权限管理功能进行了验证，结果均符合预期，确保了系统的基本安全性。

4 总结与展望

本文成功设计并实现了一个基于深度学习的手语识别与翻译系统。该系统整合了计算机视觉、深度学习以及语音合成技术，实现了端到端的自动翻译手语手势至中文文本与语音。经过测试结果，系统在识别准确率、响应速率以及资源耗费层面皆契合设计需求，为听障与非听障人群间的沟通提供一项可行的技术解决路径。

虽当前系统已然具备基础功能且展现出良好表现，然而仍存在着可供进一步优化的余地。在未来，扩展系统语言支持范畴实现更多语种间相互翻译。对于连续语句级手语，引入诸如 Transformer 这类更为强大的时序模型，实现直接识别与翻译，取代当前字母拼写方式，不失为一种可行之举。优化用户界面，增添手势引导动画以及更为详尽的错误提示信息，以此提升用户体验，亦有优化的必要。将模型轻量化，使之能在手机、嵌入式设备等资源受限的边缘计算平台更好地部署，进而拓展应用场景将会是一个重要的发展走向。

参考文献：

[1] 张磊,王振宇,连帅帅,等. 基于深度学习的手语翻译:过去、现状与未来 [J]. 计算机应用研究 ,2025,42(08):2241-2254.

[2] 崔宗兴 , 齐中正 , 盛浩男 . 智能手语翻译手套的创新设计与应用 [J]. 物联网技术 ,2025,15(06):2-3.

[3] 张恒博 , 刘大铭 . 基于时间金字塔与特征融合的连续手语识别 [J]. 计算机仿真 ,2024,41(12):297-301.

[4] 李宇楠 , 耿熙 , 苗启广 . 基于计算机视觉的手语识别与翻译研究综述 [J]. 微电子学与计算机 ,2025,42(06):15-36.

[5] 田丕承 . 基于深度学习的连续手语识别及翻译研究 [D]. 湘潭大学 ,2024.

[6] 王欣 , 陈定昌 . AI 数字人手语辅助系统的设计与应用 [J]. 广播电视网络 ,2024,31(03):91-93.