

# Fall Detection and Speech Emotion Detection for elderly people living alone

Bug Killers

UI Developer: Saniya Pandita

Full Stack Developer: Jayadithya Nalajala

ML and Data Engineer: Sai Jahnavi Devabhakthuni

AI Developer: Hui Jin

## Content

<i>Overall System Architecture Diagram .....</i>	<i>2</i>
<i>AI/ML Model Diagram.....</i>	<i>3</i>
<i>Application Interface Wireframe.....</i>	<i>4</i>
<i>GitHub Repository Link.....</i>	<i>6</i>

# Overall System Architecture Diagram

The system integrates multiple AI components, including YOLOv5 for fall detection, LLMs for psychological assessment, and Whisper for speech-to-text conversion. It processes data from various sources, such as video feeds, voice inputs, and user interactions. Backend services handle data storage, API communication, and vector databases, while frontend interfaces provide a user-friendly dashboard and chatbot for elderly users.

## System Architecture for Elderly Fall Detection and Psychological Assessment

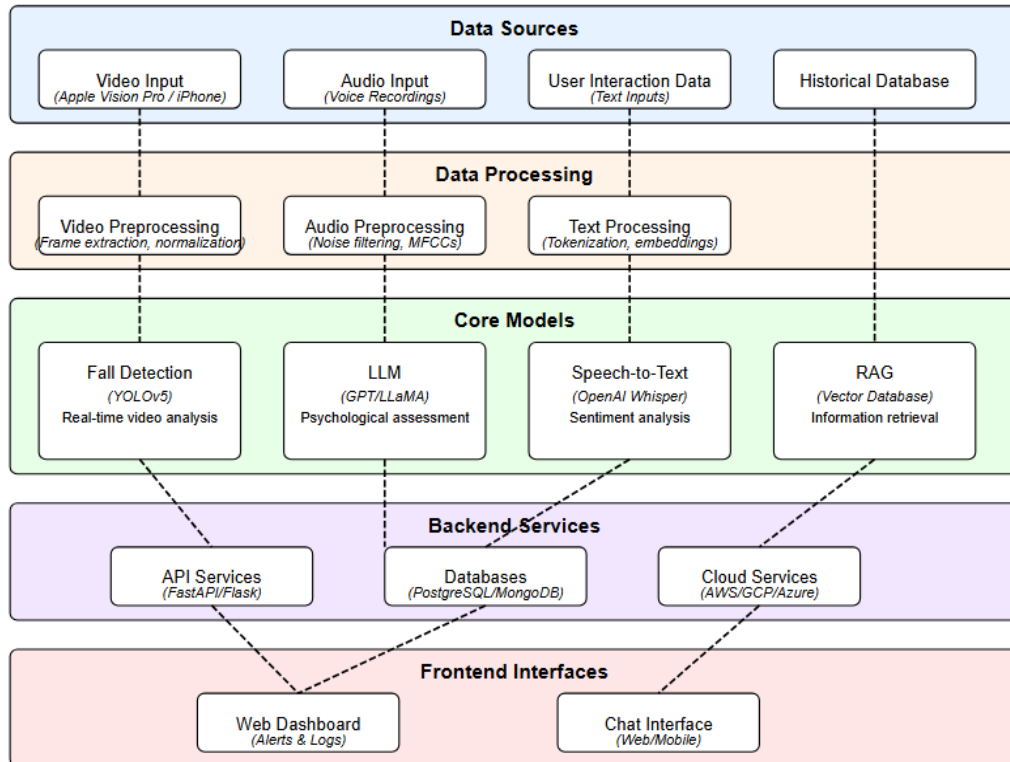


Figure 1: System Architecture

As shown in Figure 1:

### 1. Data Sources

- User Input: Audio (voice recordings), Video (Apple Vision Pro/iPhone), and Text (user interaction data).
- Document Retrieval: Historical database.
- Databases: Vector Database (used for RAG-based information retrieval).

### 2. Data Processing

- Video Preprocessing: Frame extraction, normalization.
- Audio Preprocessing: Noise filtering, MFCCs.
- Text Processing: Tokenization, embedding generation.

### 3. Core Models

- LLMs: GPT/LLaMA for psychological assessment.
- RAG Pipelines: Uses a vector database for information retrieval.

- Speech-to-Text: OpenAI Whisper for sentiment analysis.
- Computer Vision: YOLOv5 for real-time fall detection.

#### 4. Backend Services

- APIs: FastAPI/Flask.
- Vector Databases: RAG implementation uses a vector database.
- General Databases: PostgreSQL, MongoDB for structured/unstructured data storage.
- Cloud Services: AWS, GCP, Azure.

#### 5. Frontend Interfaces

- Web Applications: Web dashboard for alerts and logs.
- Chat Interfaces: Web/Mobile-based user interaction.

#### 6. External APIs and Tools

- OpenAI: Whisper for speech-to-text processing.
- Hugging Face: Likely for LLM fine-tuning or RAG integration.
- Cloud Services: AWS, GCP, Azure.

## AI/ML Model Diagram

The AI/ML model diagram details the architecture of the core models used in the system, including Transformers, LLMs, and RAG pipelines. The Transformer-based LLMs (e.g., GPT or LLaMA) are used for psychological assessment, while the RAG pipeline enhances chatbot responses by retrieving relevant information from a vector database.

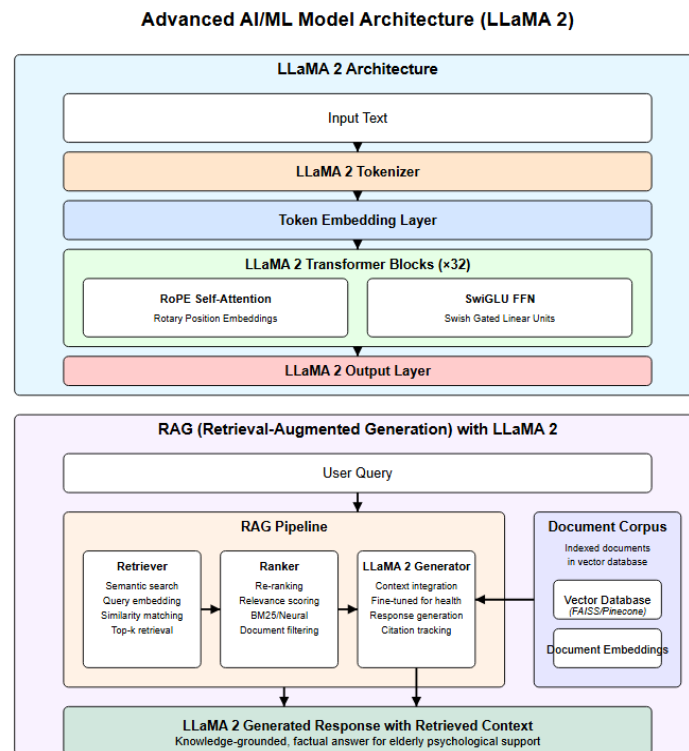


Figure 2: Advanced AI/ML Model Architecture

As shown in Figure 2:

## 1. Transformers/LLMs Architecture

- **Tokenizer:** LLaMA 2 Tokenizer processes input text.
- **Encoder/Decoder Blocks:**
  - Token Embedding Layer: Converts tokens into vector representations.
  - LLaMA 2 Transformer Blocks (×32): Contains:
    - RoPE Self-Attention: Rotary Position Embeddings for capturing positional dependencies.
    - SwiGLU FFN: Swish Gated Linear Units for non-linearity.
- **Multi-Head Attention:** RoPE self-attention mechanism is utilized for better sequence understanding.
- **Output Layers:** The LLaMA 2 Output Layer produces the final processed text.

## 2. RAG – Retrieval-Augmented Generation Architecture

- **Retriever:**
  - Semantic search
  - Query embedding
  - Similarity matching
  - Top-k retrieval
- **Ranker:**
  - Re-ranking of retrieved documents
  - Relevance scoring using BM25/Neural methods
  - Document filtering
- **Reader/Generator (LLaMA 2 Generator):**
  - Context integration
  - Fine-tuned for health applications
  - Response generation with citation tracking

## Application Interface Wireframe

The application interface wireframe illustrates the user interaction flow for elderly users and caregivers. It includes a messaging interface for text/voice input, a dashboard for psychological analysis and fall detection alerts, and app for real-time detection.

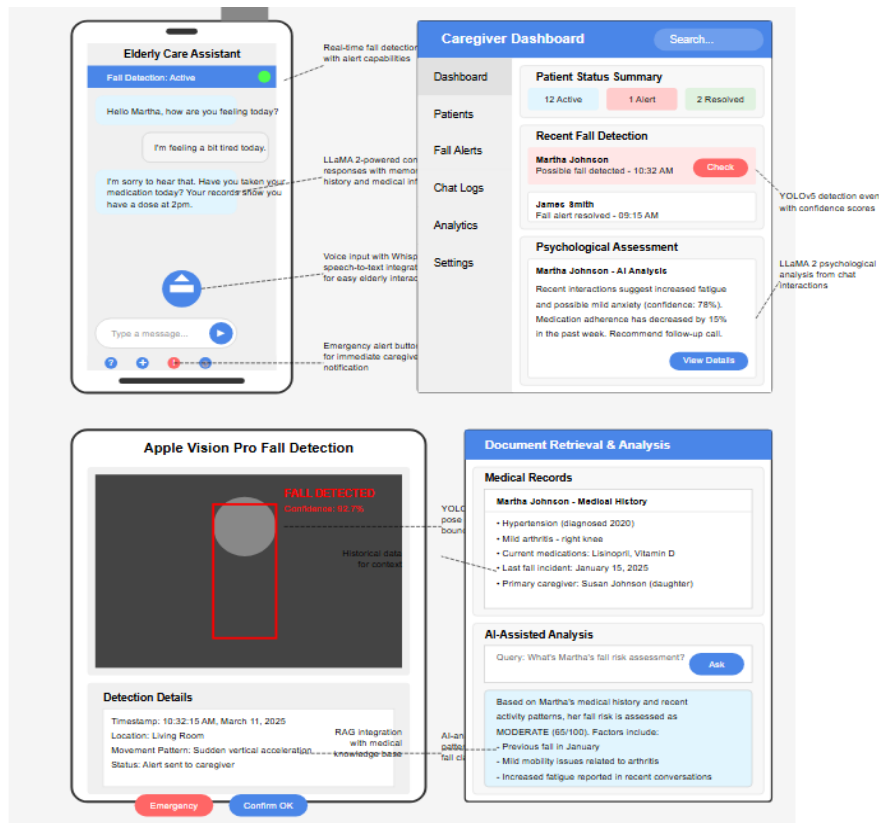


Figure 3: Application Interface Wireframe

As shown in Figure 3:

## 1. Core Screens

- Chat Interface (Elderly Care Assistant)
  - Allows voice and text-based interactions.
  - Provides AI-generated responses for psychological assessment.
  - Displays medication reminders and emergency alerts.
- Caregiver Dashboard
  - Patient Status Summary: Displays active, alert, and resolved statuses.
  - Recent Fall Detection: Logs detected falls with timestamps and confidence scores.
  - Psychological Assessment: Summarizes emotional state and medication adherence based on AI analysis.
- Apple Vision Pro Fall Detection Screen
  - Shows fall detection results with confidence scores.
  - Includes detection details such as location, timestamp, and movement analysis.
- Document Retrieval & Analysis
  - Medical Records: Retrieves patient history, medications, and caregiver details.
  - AI-Assisted Analysis: Generates risk assessment insights based on past records and patient interactions.

## 2. User Actions

- Submitting Prompts: Users interact via the chat interface to report conditions or ask about medications.
- Uploading Files: The system fetches and analyzes medical history and other stored records.
- Viewing Results:
  - Caregivers view fall alerts, psychological assessments, and AI-assisted risk analysis.
  - Users receive fall detection results and emergency alert options.

### **3. Outputs**

- AI-Generated Text Responses:
  - LLaMA 2-powered responses for psychological assessment.
  - Sentiment-based analysis from chat interactions.
- Images:
  - Fall detection visual output with YOLOv5 confidence scores.
- Retrieved Documents:
  - Patient's medical history and AI-generated risk assessments.

## **GitHub Repository Link**

<https://github.com/nanxuanhui/DSCapstone.git>