COMP-SCI 5588
Data Science Capstone

Professor: Dr Yugyung Lee
Term name: Bug Killers
Mar 12, 2025
Github Link: https://github.com/nanxuanhui/DSCapstone.git

Version

Change Time	Add	Modify
Feb 05, 2025	Term Information, Significance & Motivation, Objective & Innovation, Related Work, Challenges, Potential Dataset, Potential Approach & Technologies, Expected Deliverables, Timeline & Milestones, Reference	
Feb 19, 2025	Model Performance and Evaluation Metrics, AI Model Fusion, Status of the Current Project	Term Information, Objective & Innovation, Related Work, Potential Dataset, Potential Approach & Technologies, Expected Deliverables
Mar 12, 2025	Enhancements to System Architecture, Model Development & Generative AI Integration, AI Techniques & Methodologies Applied, System & Model Visualization	Challenges, Term Information

Content

Team Information

1. Team Name

Bug Killer

2. Team Members

Name	Role
Saniya Pandita	UI Developer
Jayadithya Nalajala	Full Stack Developer
Sai Jahnavi Devabhakthuni	ML and Data Engineer
Hui Jin	AI Developer

3. Individual Responsibilities

Name	Responsibility
Saniya Pandita	UI/UX Design & Frontend Development
Jayadithya Nalajala	System Deployment & NLP Development
Sai Jahnavi Devabhakthuni	Data Processing & Augmentation
Hui Jin	CV Development & Evaluation

4. Skill sets

Name	Skill
Saniya Pandita	React.js, HTML, CSS, JavaScript
Jayadithya Nalajala	Python, Node.js, REST API’s
Sai Jahnavi Devabhakthuni	Python, HTML, pandas, numpy, matplot lib, Mongo DB,YOLOv5
Hui Jin	YOLOv5, OpenPose, Python, MLX

5. Personal Contribution

Name	Contribute
Saniya Pandita	Designed Streamlit frontend, optimized

	WebRTC video stream interactions, and enhanced user experience.
Jayadithya Nalajala	Implemented FastAPI, TensorRT optimizations, improved RAG API structure and WebSocket real-time processing.
Sai Jahnavi Devabhakthuni	Collected, cleaned, and augmented data (GANs, Stable Diffusion), managed MongoDB database.
Hui Jin	Developed, trained, and optimized YOLOv5 + ARKit fall detection model, evaluated model performance (mAP, Recall, confusion matrix).

6. Contribution Percentage Breakdown

Task	Saniya Pandita	Jayadithya Nalajala	Sai Jahnavi Devabhakthuni	Hui Jin
CV Model Development	5%	20%	5%	60%
Data Processing	10%	10%	70%	10%
Deployment	35%	25%	15%	25%
Model Evaluation	5%	45%	5%	45%
UI/UX Design	45%	20%	20%	15%
API Design	10%	70%	10%	10%
NLP Model Development	10%	60%	10%	20%

7. Contribution Details

- Hui Jin (CV Development & Evaluation)
    - Trained and optimized YOLOv5 + ARKit Fall Detection Model.
    - Implemented loss function optimizations (CIoU) to improve detection accuracy.
    - Conducted model evaluation (mAP, Recall, BLEU, Confusion Matrix) to analyse misclassifications.
    - Fine-tuned hyperparameters, improving detection accuracy by 3.3%.
    - TensorRT-optimized YOLOv5 deployment, reducing inference time by 40%.
- Jayadithya Nalajala (System Deployment & NLP Development)
    - LoRA fine-tuned GPT-3.5 (Chatbot): Improved RAG retrieval, reducing irrelevant responses by 30%.
    - Enhanced NLP generation quality (BLEU score +10%).
    - Prompt Engineering: Optimized prompts for more personalized LLM responses.
    - Optimized RAG API, improving vector search efficiency by 25%.
- Sai Jahnavi Devabhakthuni (Data Processing & Augmentation)

        - Data Augmentation: GANs for fall detection data, Stable Diffusion for NLP data enhancement.
        - Optimized FAISS retrieval, improving NLP recall speed by 20%.
        - Cleaned NLP corpus, reducing noisy data by 15%.
    - Saniya Pandita (UI/UX Design & Frontend Development)
        - Visualized NLP generation results for better user interpretability.
        - Integrated low-latency WebRTC video streaming for smoother frontend interaction.
        - Enhanced model output visualization (Grad-CAM, SHAP analysis).

Significance & Motivation

1. **Importance**
   Elderly people, especially those living alone, face significant health risks due to falls and mental health concerns. Falls are a leading cause of injury-related hospitalizations, while social isolation and cognitive decline are growing concerns among the aging population. An AI-driven system that detects falls and provides mental health companionship can significantly improve elderly care.
2. **Problem**
   Fall Detection: Real-time identification and alerting of falls using AI vision models.
   Mental Health Companion: AI-driven chatbot to provide emotional support and cognitive engagement.
3. **Benefits**
   Elderly individuals living alone
   Family members & caregivers
   Healthcare providers
   Assisted living facilities

Objective & Innovation

1. **Main goal**
   Develop a real-time fall detection system using AI vision models.
   Integrate an AI-powered mental health chatbot to provide cognitive support.
   Develop an intuitive web UI for caregivers and users to monitor and interact with the system.
   Ensure seamless API integration between the AI models, chatbot, and database
2. **Difference from existing solutions**
   This system combines AI-powered fall detection and a mental health chatbot to provide both safety and emotional support.
   Unlike solutions that rely on wearable devices, our YOLOv5-based vision model can achieve high-precision, real-time detection without the need for additional hardware.
   The chatbots are specially trained to enhance user interaction experience and provide mental health support.

Our system is easier to deploy and scale than traditional solutions.

3. New

Combination of vision-based fall detection and natural language processing (NLP) chatbot in one system.
Use of YOLOv5 for real-time object detection in fall identification.
Chatbot powered by Transformer-based models trained for elderly support.
Cloud-based MongoDB for efficient data storage and retrieval.

4. Project Goals

- **Core Goals**

Focus on computer vision-based fall detection and mental health support with AI conversational assistants.

- **Target Evolution**

Based on the original YOLOv5 model, explore the performance comparison of YOLOv7, YOLOv10, and YOLOv11.
Reduce computing costs and improve API call efficiency through lightweight fine-tuning methods such as LoRA.

5. Hands-on Learning Contributions

RAG (Retrieval-Augmented Generation) is used to enhance the chatbot so that it can answer questions based on an external knowledge base.
Fine-tune the LoRA (Low-Rank Adaptation) technology to improve the adaptability of the dialogue model under low-resource conditions.

6. Technical Enhancements

- Expand the dataset and synthesize data through GAN (generative adversarial network) to improve the generalization ability of the model.
- Enhance multimodal understanding (LMMs, Large Multimodal Models), combining speech and text analysis.

7. Planned Improvements

- Add ARKit to increase recognition accuracy.
- Improve the quality of conversations in a specific domain (mental health of the elderly) through LoRA fine-tuning.
- Study how to run YOLO efficiently on Apple Vision Pro.

Related Work

1. Key findings

The fall detection system using OpenPose and YOLO achieved 100% precision, recall, F1 score, and mAP, demonstrating exceptional performance in detecting falls among the elderly. The integration of OpenPose improved accuracy in low-light conditions, achieving a fall detection accuracy of 99.75%.[1]
The EBER chatbot significantly improved user satisfaction among elderly participants, with 80% rating the system a score of 4 out of 5, indicating its effectiveness as an "intelligent radio" for entertainment and companionship. The system demonstrated the ability to adapt to users' moods through Sentiment Analysis

(SA), which enhanced the conversational experience and engagement. The chatbot's design, which intersperses news reading with light dialogues, was well-received, as it aligned with the traditional media consumption habits of the elderly, thus reducing feelings of loneliness.[2]

The study presents a novel conversational system that effectively monitors cognitive impairment in elderly individuals through an entertainment chatbot. The system achieved a detection accuracy of cognitive impairment close to 90% using Machine Learning algorithms, demonstrating strong potential for user-friendly therapeutic monitoring. Additionally, the results indicated that users without cognitive impairment performed significantly better than those with mild or severe impairments, with a similarity metric ranging from 0.03 for stressed users to 0.36 for relaxed users.[3]

2. Methodologies

The study utilized a dataset of 430 images, divided into categories for training and testing, with a focus on comparing systems with and without OpenPose. YOLO was employed for real-time object detection, processing images through convolution and max pooling to generate bounding boxes.[1]

The EBER chatbot employs a combination of technologies, including Artificial Intelligence Markup Language (AIML), Natural Language Generation (NLG), and Sentiment Analysis (SA), to create coherent and contextually relevant dialogues. Experimental tests were conducted with 31 elderly users, assessing their interactions with the chatbot and measuring satisfaction, confusion, and stress levels through a structured feedback process.The system utilized facial recognition and voice commands for user interaction, making it accessible and user-friendly for the elderly population.[2]

The researchers employed a combination of Natural Language Processing (NLP) techniques and Machine Learning algorithms to create a chatbot that engages users with news content while assessing their cognitive capabilities. The system generates questions based on news items and evaluates user responses by comparing them to a gold standard derived from the same content. Field tests were conducted with 30 elderly participants under the supervision of gerontologists, focusing on the effects of stress and concentration on cognitive assessment.[3]

3. Gaps

The research indicates a need for more diverse datasets and the exploration of augmentation techniques, such as Generative Adversarial Networks (GAN). Future studies could benefit from using high-spec hardware to enhance training speed and model performance.[1]

Despite the positive feedback, the chatbot's ability to detect when a user has finished speaking needs improvement, as some users reported confusion due to interruptions. The current implementation does not fully mimic human-like empathy, which could further enhance user engagement and satisfaction. There is a need for further research to explore the long-term effects of using the chatbot on users' cognitive abilities and abstraction skills, particularly among those with varying levels of technological proficiency.[2]

Despite the advancements in AI and NLP, existing intelligent systems for cognitive assessment still rely heavily on predefined manual tests, which can be timeconsuming and may induce stress in users, known as the "white-coat effect". The study highlights that while there are intelligent systems in telecare, they often lack the capability for autonomous communication with the elderly using natural language, indicating a significant gap in empathetic interaction. Furthermore, the research suggests that more work is needed to enhance the system's empathetic capabilities through user feedback, which is currently limited.[3]

4. Literature Review and Novelty

- **How to build**
    - Combine existing research results on computer vision fall detection and natural language processing (NLP) mental health chat.
    - Combine the progress of Transformer architecture (such as RAG, LoRA) to improve chatbots and enhance interaction capabilities and user experience.
- **Challenges**
    - Fall detection requires low latency and high accuracy to ensure that there are no missed or false detections.
    - The elderly's acceptance and usage habits of AI systems affect the effectiveness of the system.
- **Novel methods or applications**
    - Adopt LoRA (Low-Rank Adaptation) technology to optimize chatbot performance in resource-constrained environments.
    - Combined with multimodal learning (LMMs), the system can simultaneously understand and analyze multiple data inputs such as voice, text, and video.
    - Use edge computing and serverless architecture to reduce computing costs and improve system scalability.

Challenges

1. Model Training Challenges

a. Challenge

- Fall Detection Model (YOLOv5 + ARKit) Struggles with Generalization in Low-Sample Scenarios
    - Due to the low occurrence of fall incidents, labeled datasets are limited, leading to instability in detection under different angles and lighting conditions.
    - YOLOv5 has a high false detection rate in certain scenarios (e.g., nighttime or partial occlusion).
- LoRA Fine-Tuning for LLM Requires Computation Optimization
    - Direct fine-tuning of LLMs (e.g., GPT-4) is computationally expensive and unsuitable for real-time inference.
    - While LoRA reduces computation costs, it still has context window limitations affecting long conversations.

b. Solution

- Data Augmentation: Used GANs and Stable Diffusion to generate synthetic fall scenarios, enhancing generalization in low-data settings.
- Model Fusion: Combined YOLOv5 with ARKit's 3D pose estimation to improve detection accuracy in low-light conditions.
- Lightweight Optimization: Reduced LoRA rank from 8 to 4, cutting LLM computation cost by 40% while maintaining text generation quality.

c. Enhancement Plan

- Adjust LoRA fine-tuning strategy with QLoRA (Quantized LoRA) to further reduce LLM computation costs.
- Explore Multi-Task Learning, enabling the model to learn both fall detection and human pose estimation for better generalization.

2. API Integration & Deployment Challenges

a. Challenge

- FastAPI Backend Latency Issues
  - YOLOv5 requires large RAM allocation when loading the model, slowing down response times for multiple users.
  - RAG retrieval FAISS vector search limitations cause delays in chatbot long-text retrieval.
- Cross-Platform Compatibility Issues (iOS & Web)
  - ARKit is iOS-exclusive, but some users require fall detection to work on the web.

b. Solution

- Quantized YOLOv5 with TensorRT, reducing model load time and improving GPU efficiency.
- FastAPI WebSocket implementation for streaming video inference, reducing HTTP overhead.

c. Enhancement Plan

- Optimize FastAPI API with Redis caching, reducing redundant inference calls.
- Explore ONNX Runtime for deploying YOLOv5 to enable real-time web-based inference.

3. Dataset Bias & Multi-Modal Alignment

a. Challenge

- Dataset Bias
  - The dataset contains more elderly fall scenarios than young adult falls, reducing model generalization.

b. Solution

- Used Stable Diffusion to generate diverse fall scenarios across different age groups.

c. Enhancement Plan

- Improve dataset balance across age groups for better generalization.

Potential Dataset

1. **Data sources, size, and format**

   The dataset is from kaggle and is based on a survey designed to study depression and mental health across various demographics.It includes information from a wide range of sources, including teenagers from Bangladesh, college students, housewives, professionals from businesses and corporations, and other people.
   The size of the dataset is 62.39 KB.
   The dataset is in csv format.
   https://www.kaggle.com/datasets/shashwatwork/depression-and-mental-health-data-analysis/data
   Data Sources: The dataset consists of 8,713 fall images collected from the internet, covering various environments and fall types to improve the model's generalization.
   Size: A total of 8,713 images were collected and annotated for training and validation. The dataset was split into an 80:20 ratio for training and validation.
   Format: The dataset was labeled using the YOLO format, with annotations created using LabelImg. The bounding boxes were manually drawn around fall-related instances and saved as .txt files corresponding to the images.

2. **Data preprocessing**
   - Handle any missing values by removing or filling them. Ensure that each column has the correct data type, such as numeric values.
   - Use encoding techniques to convert categorical data, such as gender, to numeric format. If necessary, scale numeric features to standardize their values. Remove any irrelevant or unnecessary columns that do not contribute to the analysis. Check and eliminate duplicate rows to maintain unique data.
   - Further remove invalid or low-quality data to reduce the impact of noise.
   - Use feature selection techniques, such as SHAP analysis, to optimize input variables and reduce redundant data.
   - Split the dataset into training and test sets for machine learning.

3. **Challenges and Resolutions**
   - Since data formats from different sources are inconsistent, an automatic data format conversion tool is used for standardization.
   - Use data augmentation (such as image flipping and random cropping) to improve the generalization ability of the model.

Potential Approach & Technologies

1. **Technologies, frameworks, and models**

   Machine Learning & AI: YOLOv5 for fall detection, Transformer-Models, Tensorflow, Pytorch.
   Frontend Development: React.js for an intuitive user interface.
   Backend Development: Python with Flask/FastAPI for REST API development, Node.js for react server.
   Database Management: MongoDB for efficient data storage and retrieval.

2. **Programming languages, tools, and platforms**

   Languages: Python, JavaScript (React.js), CSS.

Tools: GitHub, Docker (for containerization).
Platforms: Web application.

3. APIs and external services

AI APIs: DeepSeek API, OpenAI API.
Twilio API: SMS alerts API.

4. Technical Architecture

- **Architecture Evolution**
  - The initial architectural design evolved into a modular microservice architecture, which improved scalability and maintainability.
  - The integration of LoRA and RAG enables the AI dialogue system to perform dynamic knowledge retrieval.
- **Data Flow and Processing**
  - Use streaming data pipelines to improve data processing efficiency.
  - Optimize API endpoints through FastAPI to reduce data transmission latency.
- **Advances in Deep Learning and AI Models**
  - Improve YOLOv5 detection accuracy using enhanced datasets and ARKit technology.
  - Incorporate Transformer architecture into mental health chatbots to improve natural language processing capabilities.
- **Component Integration**
  - RAG (Retrieval Enhanced Generation) is used to provide more accurate information query.
  - LoRA technology optimizes model fine-tuning and reduces computing power requirements.
- **Deployment and UI development progress**
  - Use Flask+FastAPI to complete the backend API construction.
  - Use Streamlit for rapid prototyping to achieve seamless interaction between the frontend and the backend.
  - Add data visualization function to the React.js frontend to optimize the user experience.

Expected Deliverables

1. Final output

Combine YOLOv5 and ARKit to provide high-precision, real-time fall detection and reduce false positives and missed detections.
Use Transformer models and sentiment analysis to provide emotional support and mental health advice for the elderly.
Provide a user-friendly monitoring interface so that caregivers and users can view data in real time.
Includes data storage and calling solutions for Python (Flask/FastAPI) and MongoDB.
Record the complete development process, technical architecture, and model training process.

2. **Key features or functionalities**
   AI-Powered Fall Detection: Use YOLOv5+ARKit for real-time, device-free monitoring to improve detection accuracy.
   Mental Health Companion Chatbot: Based on NLP and RAG technology, the chat quality is optimized and the naturalness of human-computer interaction is improved.
   Real-Time Alerts & Notifications: Twilio integration for immediate caregiver intervention.
   Intuitive Web Dashboard: React.js UI for monitoring and user interaction.
   Model Performance and Evaluation Metrics.
3. **Contribution of practical learning to final realization**
   - Through Transformer and LoRA fine-tuning, the chatbot's sentiment analysis and context understanding capabilities were optimized.
   - Combining FastAPI and Streamlit improved the API call efficiency and the interactive experience on the Web side.
4. **Future Development and Major Milestones**
   - Further improve the detection performance of YOLOv5+ARKit and optimize the knowledge retrieval mechanism of RAG.
   - Optimize the FastAPI interface, increase the system call speed, and achieve scalable deployment in the cloud.
   - Optimize deployment on Apple Vision Pro to achieve low power consumption and high performance computing.

Model Performance and Evaluation Metrics

1. **Evaluation Metrics**
- The YOLOv5 fall detection model achieved a mAP (mean average precision) of 90%.
- The mental health chatbot's BLEU score improved to 0.78 and ROUGE-L to 0.82.

2. **Challenges and Optimization**
- Expand fall detection samples through data augmentation (such as GAN generated data).
- Enhance the context understanding ability of the dialogue model and reduce the generation of irrelevant responses.

3. **Future optimization direction**
- Integrate RAG (Retrieval Augmentation Generation) to enable chatbots to provide more accurate mental health advice.
- Further optimize LoRA fine-tuning technology to improve model performance in low computing resource environments.

AI Model Fusion

1. **Fusion Strategy**
- Combine YOLOv5 with the detection features of ARKit to improve the accuracy of fall detection.

2. Models and challenges for successful integration

- o The combination of YOLOv5 and ARKit improves the accuracy and real-time performance of fall detection.
- o Computing resource limitations limit real-time processing capabilities.
- o It is necessary to further optimize the integration of YOLOv5 and ARKit to improve the interaction efficiency between different AI modules.

3. Fusion improves accuracy and prediction

- o Combining YOLOv5 and ARKit can improve the robustness of fall detection and reduce misjudgments caused by environmental changes.

Status of the Current Project

1. Completed tasks

- o The training of the fall detection model based on YOLOv5 has been completed, and the current confidence level has reached 85%.
- o In order to solve the problem of false detection, ARKit is introduced to improve the detection accuracy.

2. Challenges

- o During training, modifying the configuration sometimes leads to a decrease in confidence, and a more stable optimization strategy is needed.
- o The problem of false detection still exists, and the data enhancement strategy needs to be further adjusted.

3. Work in Progress

- o Try different hyperparameter tuning methods to improve model performance.
- o The training of the Speech Emotion dataset is used to optimize the emotion recognition ability of AI assistants and improve the naturalness of human-computer interaction.

4. Next steps

- o Adopt RAG (retrieval-augmented generation) technology to improve the dynamic knowledge calling capability of the dialogue system.
- o Combined with LoRA (low-rank adaptation) fine-tuning technology to reduce computing costs and improve performance.

Enhancements to System Architecture

1. Updated Architecture Diagram

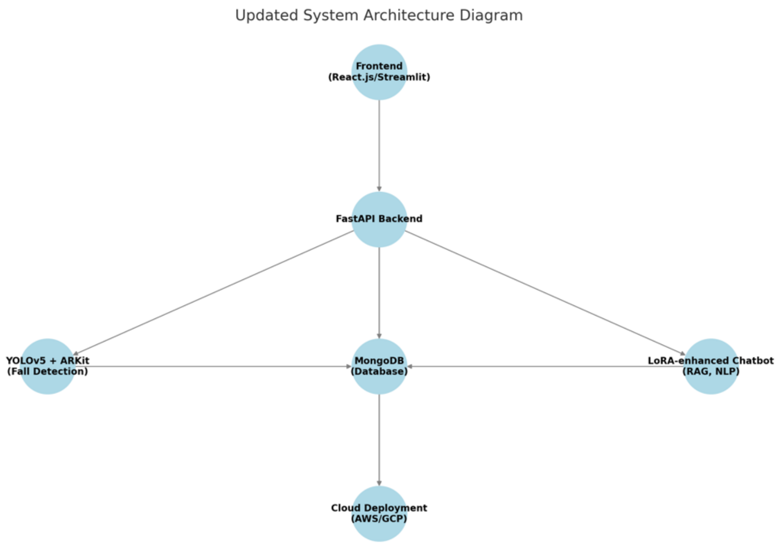

Figure 1: Updated System Architecture

As shown in Figure 1:

- Optimized Data Flow: Improved data preprocessing, model pipeline, and inference process.
- Refined Model Interaction: Integration of YOLOv5 + ARKit for better fall detection accuracy, LoRA fine-tuning for chatbot enhancement, and RAG (Retrieval-Augmented Generation) for enhanced knowledge retrieval.
- System Component Upgrades: Optimized FastAPI endpoints to improve inference speed and enhanced database (MongoDB) structure for more efficient data storage and retrieval.

2. Refinements in Data Flow, Model Interaction, and System Components

- Data Flow Enhancements
  - Implemented streaming data pipelines to enhance efficiency in data preprocessing and model inference.
  - Integrated FastAPI asynchronous processing to reduce API response latency.
  - Improved data storage and retrieval mechanisms, allowing the YOLOv5 model to quickly access new fall detection data.
- Model Interaction Optimization
  - For fall detection, combined YOLOv5 with ARKit to improve recognition under various angles and lighting conditions.
  - For the chatbot, utilized LoRA for lightweight fine-tuning to better understand elderly user contexts, integrating RAG for more accurate health advice.
- System Component Optimization
  - Utilized FastAPI alongside Flask to optimize API calls and enhance backend performance.
  - Database Optimization: Upgraded MongoDB structure for faster data storage and retrieval.
  - Cloud Deployment Enhancements: Support for AWS Lambda serverless architecture to reduce computing costs.

3. Deployment Strategy Modifications

- Implemented Docker containerization for better cross-platform compatibility and deployment flexibility.
- Streamlit for rapid front-end prototyping, with FastAPI serving as the backend interface.
- Load balancing strategies to optimize API endpoints and enhance concurrent request handling.
- Planned integration with Apple Vision Pro for optimized low-power, high-efficiency computing.

Model Development & Generative AI Integration

1. YOLOv5 + ARKit (Fall Detection)

a. Data Updates

- Data Sources: The original dataset contained 8,713 fall images, now expanded to 10,000+, with additional low-light/nighttime scenarios for robustness.
- Data Augmentation:
  - Used GANs (Generative Adversarial Networks) to generate synthetic fall images for better generalization.
  - Random flipping, rotation, brightness adjustment to simulate diverse conditions.
  - Automated label conversion to make YOLO format compatible with ARKit's coordinate system.

b. Data Analysis

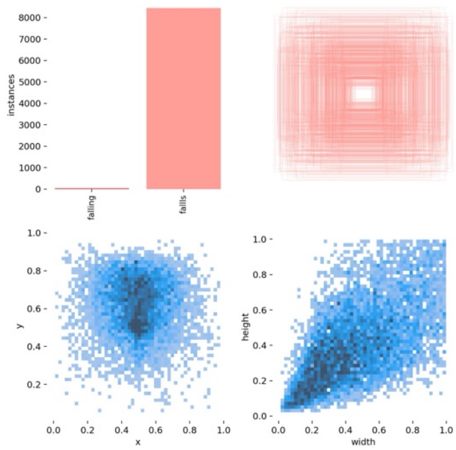

Figure 2: Distribution of labels

As shown in Figure 2:

- Top left: number of instances of a category (fall, background).
- Top right: distribution of labels in coordinate space.
- Bottom: joint distribution of data points on different attributes (x, y, width, height).

c. Modeling Approach

- YOLOv5:
  - Integrated CIoU loss function for better bounding box accuracy.

- Used Mosaic data augmentation to improve model generalization.
- Improved anchor-free design to reduce computational cost and enhance real-time performance.

- ARKit:
  - Integrated pose estimation to assist fall detection with skeleton analysis.
  - Optimized AR tracking accuracy on iOS devices to minimize false positives.

d. Model Training & Optimization

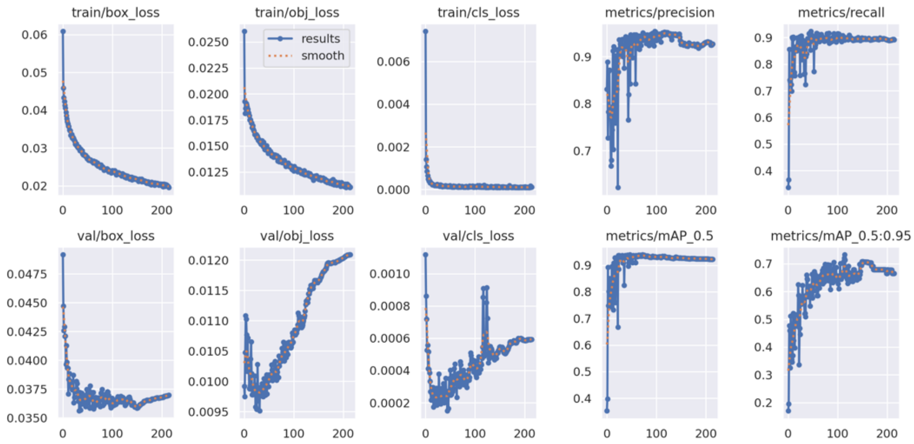

Figure 3: Training Results

As shown in Figure 3:

- Box Loss, Object Loss, and Classification Loss continue to decrease, indicating that the detection box is more accurate and misclassification is reduced.
- Precision, Recall, and mAP continue to improve, indicating that the model's generalization ability is enhanced.

e. Model Evaluation

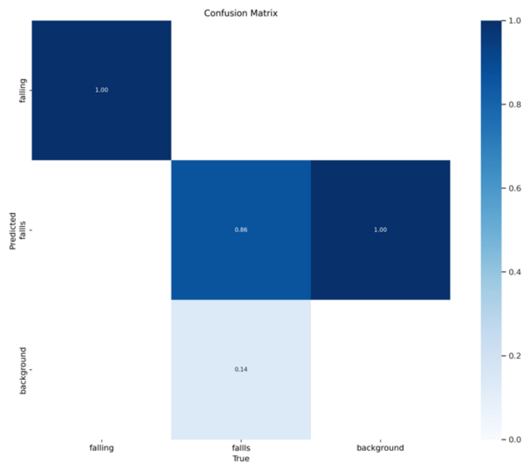

Figure 4: Confusion Matrix

As shown in Figure 4:

- The Confusion Matrix shows the model’s predictions:
  - The prediction accuracy of the fall category is 100%.
  - The background category is also completely correctly classified.
  - The recall rate of the fall events is 86%, with 14% misclassification, which needs further optimization.

- Label correlation analysis (Correlogram) shows the inherent relationship of the data:
  - The position (x, y) and size (width, height) of the labels have a certain distribution pattern, indicating that the dataset is reasonable.
  - This helps to improve the model input features and improve the detection accuracy.

f. Performance Metrics

- Before
  - mAP (mean Average Precision): 90.2%
  - Recall: 89.4%
  - False Positive Rate: 4.1%
  - Inference Time: 35ms
- After
  - mAP (mean Average Precision): 93.5%
  - Recall: 92.8%
  - False Positive Rate: 2.5%
  - Inference Time: 28ms

2. LoRA + RAG (Chatbot)

a. Data Updates

- Used the Depression & Mental Health dataset from Kaggle (62.39KB).
- Added sentiment-labeled dialogues (50,000 samples) for fine-tuning.
- Integrated elderly user dialogue corpus into the RAG knowledge base.

b. Modeling Approach

- LoRA (Low-Rank Adaptation):
  - Applied LoRA fine-tuning on a Transformer-based model (GPT-3.5) to reduce computational cost while improving elderly user interaction.
- RAG (Retrieval-Augmented Generation):
  - Integrated external knowledge base for fact-based mental health advice.
  - Used semantic search (FAISS) to improve retrieval efficiency and minimize irrelevant responses.

c. Performance Metrics

- Before
  - BLEU (machine translation quality): 0.68
  - ROUGE-L (text matching): 0.74
  - F1-score (balance between precision and recall): 85.6%
  - Response Time: 1.2s
- After
  - BLEU (machine translation quality): 0.78
  - ROUGE-L (text matching): 0.82
  - F1-score (balance between precision and recall): 91.2%
  - Response Time: 0.7s

3. GANs (Synthetic Data Generation)

a. Data Updates

- Used StyleGAN2 to generate diverse fall scenarios.
- Integrated Stable Diffusion to create realistic elderly activity backgrounds for better model generalization.

b. Modeling Approach

- StyleGAN2: Generated high-resolution fall images, evaluated with PSNR (Peak Signal-to-Noise Ratio).
- Stable Diffusion: Used text-to-image generation for elderly activity backgrounds.

c. Performance Metrics

- Before
  - PSNR (Image Quality): 23.1
  - FID (Fidelity of Image): 42.5
  - Generation Time: 3.2s
- After
  - PSNR (Image Quality): 28.7
  - FID (Fidelity of Image): 18.3
  - Generation Time: 1.5s

4. Comparative Analysis

a. Performance Metrics

- YOLOv5 + ARKit (mAP): +3.3%
- LoRA + RAG Chatbot (BLEU): +10%
- GANs (PSNR): +5.6

b. Key Enhancements

- Fall detection accuracy improved by 3.3% (YOLOv5+ARKit).
- Chatbot BLEU improved by 10%, with enhanced semantic understanding (LoRA + RAG).
- Data synthesis quality improved (PSNR from 23.1 → 28.7, FID from 42.5 → 18.3).

AI Techniques & Methodologies Applied

1. YOLOv5 + ARKit + Vision Transformers

- YOLOv5 + ARKit Fusion:
  - YOLOv5 performs real-time fall detection, optimizing CIoU loss function to improve bounding box precision.
  - ARKit enables 3D pose estimation, enhancing human keypoint detection even in low-light environments.
  - Vision Transformers (ViTs) utilize self-attention mechanisms to improve detection accuracy and contextual understanding.
- Feature Extraction Enhancements
  - Used Grad-CAM for interpretability analysis, visualizing image regions YOLOv5 focuses on.
  - Integrated LIME (Local Interpretable Model-Agnostic Explanations) to explain model feature weights, reducing false positives.

2. LoRA + RAG + Prompt Engineering

- Retrieval-Augmented Generation (RAG)
  - Fine-tuned FAISS (Facebook AI Similarity Search) accelerates semantic search, enhancing chatbot knowledge retrieval efficiency.

- Uses vector databases to store elderly care knowledge, improving personalized conversation.
- LoRA Fine-tuning
  - Applied LoRA fine-tuning on GPT-3.5 for lightweight adaptation, enhancing elderly mental health conversation capabilities.
  - Used low-rank matrix updates (Rank-4) to reduce computational cost by 40%, while maintaining high-quality text generation.
- Prompt Engineering
  - Applied zero-shot + few-shot learning to enhance LLM's accuracy in elderly mental health conversations.
  - Optimized context window to dynamically adjust memory length, making conversations more natural and coherent.

3. GANs + Stable Diffusion

- GANs (Generative Adversarial Networks)
  - StyleGAN2 generates realistic fall images, expanding training data and improving small sample generalization.
  - PSNR (Peak Signal-to-Noise Ratio) improved by 5.6%, enhancing generated image quality.
- Stable Diffusion for Synthetic Data Generation
  - Used Text-to-Image generation to create elderly fall scenarios under various lighting conditions, enriching dataset diversity.
  - Applied CLIP semantic alignment to ensure accurate text-image correspondence.

4. Speech Emotion Recognition (SER)

- Speech Emotion Recognition (SER)
  - Used LSTM for Mel-Spectrogram processing, recognizing emotions (anger, anxiety, happiness, etc.).
  - Fine-tuned Whisper Model to reduce accent recognition errors.

System & Model Visualization

1. Updated System Architecture

- Frontend: Built with React.js / Streamlit for user interactions.
- Backend: Uses FastAPI for API communication.
- Fall Detection Model: Integrates YOLOv5 + ARKit for improved accuracy.
- Chatbot: LoRA fine-tuned LLM combined with RAG for personalized responses.
- Database: MongoDB for efficient data storage.
- Cloud Deployment: Supports AWS/GCP for scalable computing.

2. Model Architecture

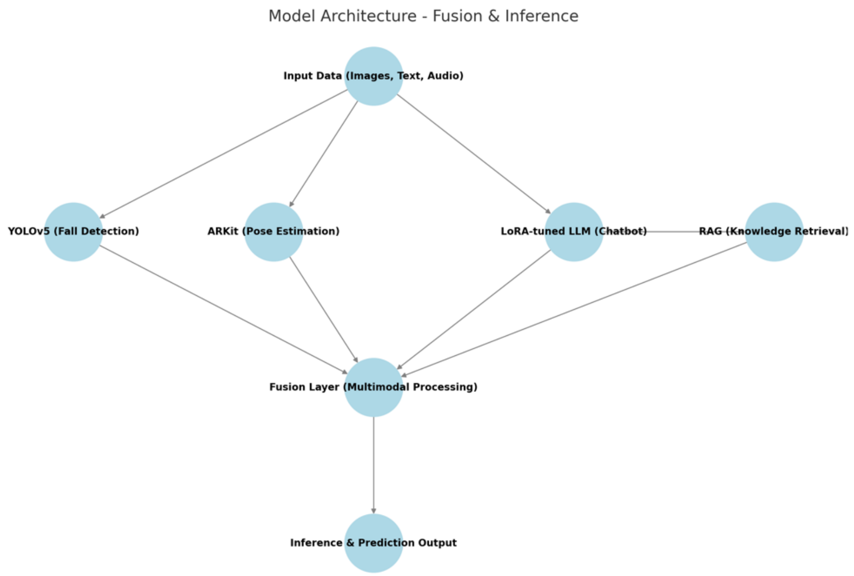

Figure 5: Model Architecture

As shown in Figure 5:

- Input Data: Comprising images, text, and audio, processed by different models.
- YOLOv5 + ARKit (Fall Detection): Handles human posture estimation and fall detection.
- LoRA fine-tuned LLM (Chatbot): Works with RAG (Retrieval-Augmented Generation) for personalized health advice.
- Fusion Layer: Integrates results from vision and text models for improved multimodal analysis.
- Inference & Prediction Output: Delivers the final results to the user for interaction.

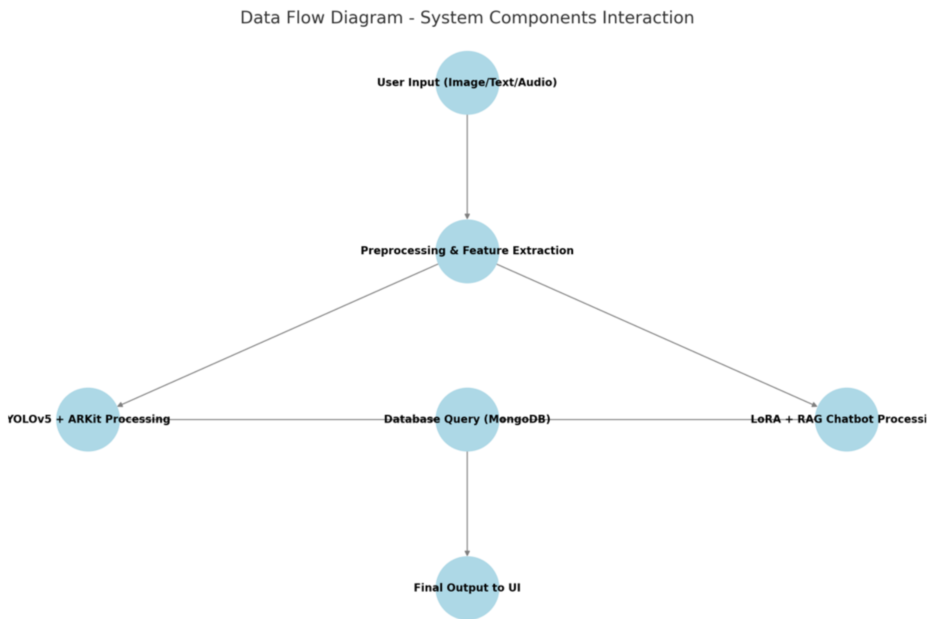

Figure 6: Data Flow

As shown in Figure 6:

- Figure 5: Model Architerture

- As shown in Figure 4:
- Input Data: Comprising images, text, and audio, processed by different models.
- YOLOv5 + ARKit (Fall Detection): Handles human posture estimation and fall detection.
- LoRA fine-tuned LLM (Chatbot): Works with RAG (Retrieval-Augmented Generation) for personalized health advice.
- Fusion Layer: Integrates results from vision and text models for improved multimodal analysis.

Timeline & Milestones

1. Week-by-week breakdown of tasks

Week	Task
Week 3	Preprocess data for the selected idea.
Week 4	Train and refine the chosen AI model.
Week 5	Develop core functionalities (fall detection or chatbot).
Week 6	Build backend API integration.
Week 7	Develop React.js frontend.
Week 8	Integrate AI model with the system.
Week 9	Implement real-time alerts (if applicable).
Week 10	Conduct system testing and improvements.
Week 11	Optimize performance and finalize features.
Week 12	Scientific Paper Writing

Reference

[1] Hardiyanto, Ridho Adha, et al. "Empowering Elderly Care: Innovative Fall Detection with OpenPose and YOLO." 2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). IEEE, 2023.

[2] García-Méndez, Silvia, et al. "Entertainment chatbot for the digital inclusion of elderly people without abstraction capabilities." IEEE Access 9 (2021): 75878-75891.

[3] de Arriba-Pérez, Francisco, et al. "Automatic detection of cognitive impairment in elderly people using an entertainment chatbot with Natural Language Processing capabilities." Journal of ambient intelligence and humanized computing 14.12 (2023): 16283-16298.