

Empowering Senior Healthcare with VLMs and LLMs: Enhanced Fall Detection and Proactive Care Planning

Timothy Wei*
Saratoga H.S.
Saratoga, California, USA
timswei@gmail.com

Max Peng*
Ridge H.S.
Basking Ridge, New Jersey, USA
maxpeng123678@gmail.com

Ryan Tellado*
Mountain View H.S.
Mountain View, California, USA
ryantellado@gmail.com

Bryan Zhao*
Saratoga H.S.
Saratoga, California, USA
bryanzhao08@gmail.com

Evan Li
Stratford H.S.
San Jose, California, USA
520evanli@gmail.com

Orlando Ding
University of California, Santa Cruz
Santa Cruz, California, USA
lding25@ucsc.edu

Abstract

In an age where technology is increasingly intertwined with our lives, AI-backed healthcare systems significantly enhance seniors' quality of life and provide peace of mind to their families. This paper presents a novel AI-driven aid system that integrates Vision-Language Models (VLMs) and Large Language Models (LLMs) to deliver comprehensive care to the elderly, Home Harmony. The system employs a multi-layered approach to improve both fall detection and action planning. VLMs monitor real-time video feeds, while audio transcription helps the LLM assess the severity of a fall and confirm its occurrence. Safety scores evaluate potential injuries and provide additional information for emergency services. Upon detecting a fall, LLMs determine the next best course of action, such as calling emergency services or notifying emergency contacts, which is then executed by an Android Agent, making the system an end-to-end solution.

ACM Reference Format:

Timothy Wei, Max Peng, Ryan Tellado, Bryan Zhao, Evan Li, and Orlando Ding. 2024. Empowering Senior Healthcare with VLMs and LLMs: Enhanced Fall Detection and Proactive Care Planning. In *Proceedings of August 05–09, 2024 (SIP '24)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/XXXXXXX.XXXXXXX>

*Authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. SIP '24, Santa Cruz, CA,

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
<https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Falls among older adults are a significant public health issue, resulting in serious consequences for both individuals and the healthcare system. According to the Centers for Disease Control and Prevention (CDC), approximately one in four adults aged 65 and older experiences a fall each year in the United States, amounting to roughly 36 million falls annually[4, 5]. These falls frequently lead to significant injuries; over 3 million older adults are treated in emergency departments for fall-related injuries annually, with more than 800,000 hospitalizations, primarily due to head injuries or hip fractures[5]. Falls are also the leading cause of injury-related deaths in this age group, with a 41% increase in the death rate from falls between 2012 and 2021, reaching 78.0 per 100,000 older adults[4, 5].

The rising number of fall-related injuries and fatalities highlights the urgent need for effective detection and prevention strategies. The financial burden is also significant, with direct medical costs for fall-related injuries estimated at \$50 billion annually[4]. Our research addresses this critical issue by developing an AI-driven end-to-end system that leverages Vision-Language Models (VLMs) and Large Language Models (LLMs) to enhance fall detection and proactive care planning for seniors.

Existing fall detection systems often rely on single-modal data and simple threshold-based alerts, which can result in high false positive rates and delayed responses. To address these limitations, we propose a novel AI-driven end-to-end system that leverages the combined strengths of Vision-Language Models (VLMs) and Large Language Models (LLMs) to enhance both the detection accuracy and the proactive care planning for seniors. Unlike traditional models, our system employs a multi-layered approach that integrates real-time video monitoring with audio transcription and contextual analysis, providing a more comprehensive understanding of the environment and the events leading up to a fall.

Our innovative approach uses VLMs to monitor real-time video feeds and detect falls with high precision, while LLMs

analyze audio transcriptions to assess the severity of the fall and confirm its occurrence. The system also incorporates segmentation to define "safe zones" within the environment, such as beds, reducing false positives and ensuring a more accurate assessment. Furthermore, safety scores evaluate potential injuries and provide additional information for emergency services, enabling timely and appropriate interventions. Upon detecting a fall, LLMs determine the best course of action, such as calling emergency services or notifying emergency contacts, which is then executed by an Android Agent, making the system an end-to-end solution.

This research aims to offer a robust solution that not only detects falls but also plans and executes necessary actions, ensuring timely assistance and improving the overall safety and quality of life for older adults. By integrating advanced AI technologies, our system sets a new standard in fall detection and proactive care planning, addressing a critical need in elderly care.

2 Related Works

2.1 Fall Detection Model

There has been extensive research on using machine learning models to detect falls. Currently, there are three main ways of detecting falls: Convolutional Neural Networks (CNNs), Graph Convolutional Networks (GCNs), and Transformer models. Researchers have focused on improving such models, many of which extract and track the skeletal structure of a human to determine if the human has fallen. For example, Kasturi et al.[6] proposed using a 3D CNN to consider the temporal relationship between frames, while others focused on improving GCNs. To illustrate, Wang et al.[17] considered creating a network of hyper-edges between non-adjacent groups of bones to model the relationships between specific body parts during certain actions. Zhou et al.[20] also introduced improvements to maintain information about the connectivity between bones through distances between joints. Others focused on synthesizing these approaches. Liu et al.[9] made a model which is comprised of a neural network, including both a GCN and a Transformer branch, to create a hybrid network that leverages both models to decide whether someone has fallen.

Large Language Models (LLMs) have also emerged in the field of fall detection. Qu et al.[15] considered a framework to convert skeletal structures into tokens that can be inputted into an LLM, which was capable of detecting the action performed by a human.

2.2 Action Planning Model

There has also been extensive research on Action Planning Models. Most existing solutions use some kind of LLM to achieve this task. Autodroid[18], for example, used UI transition graphs, HTML code and explored apps to understand how to complete certain tasks through a LLM. GPTDroid[10]

also recorded all available testing information, such as actions and widgets, before inputting them into an LLM to obtain an ensuing course of action. Others focused on improving the accuracy of the LLM inference process. For example, Shen et al.[16] used three LLMs instead of one large LLM: a Planner, Caller and Summarizer LLM. This resulted in a deeper understanding of what the user wants. Yin et al.[19] made a LLM with three modules: planning, grounding and execution. This LLM is capable of breaking down high-level instructions into smaller steps, yielding better accuracy than GPT-4o. Pan et al.[12] reviewed papers which combines Knowledge Graphs (KGs) with LLMs, allowing LLMs to access the vast information present in KGs to make a more informed decision.

Action Planning Models are not just limited to LLMs. For example, Fan et al.[3] made a Tree-of-Lens (ToL) agent, a Vision Language Model (VLM), which is capable of understanding the hierarchical structure of an app. It can also get the position of a certain UI element, and infer the content of the element and its position in the hierarchical structure of an app.

3 Methods

We utilize a Vision-Language Model (VLM) and integrate multiple layers, including video, audio, and segmentation, to enhance the accuracy and effectiveness of fall detection and action planning.

Real-time video feeds are monitored using the VLM to detect falls. In parallel, the audio from the environment is transcribed and combined with the video data to provide contextual information that aids in confirming the occurrence of a fall and assessing its severity. To further improve accuracy, we incorporate segmentation by defining "safe zones" within the environment, such as beds, to reduce the occurrence of false positives.

Upon detecting a fall, the system follows a sequence of actions. First, the VLM processes the video feeds to identify abnormalities indicative of a fall. Once a fall is detected, the information is fed into the LLM to determine the appropriate course of action. During this stage, the LLM uses safety scores to evaluate the location of the fall, helping to identify areas where falls may have subtle effects compared to more hazardous locations like the floor. This segmentation aids in assessing potential injuries and provides critical information for emergency medical services (EMS).

When combined with the audio context, the LLM action planner prioritizes actions, such as calling emergency services or notifying emergency contacts. These actions are executed by an Android Agent directly on the device. After the initial actions are carried out, the agent receives responses from the senior, if available. These responses are fed back into the LLM, which then decides whether to continue monitoring, take additional actions, or terminate the

process. This multi-layered approach enhances both fall detection accuracy and subsequent action planning, ensuring comprehensive and timely care for seniors.

3.1 Fall Detection Model

Firstly, we obtain videos from the Raspberry Pi camera[14]. Utilizing the footage, we extract frames that are one second apart and feed them into the VLM. We additionally obtain a transcription of the audio in the form of text. Our training dataset, the Le2i fall dataset[1], comprises hundreds of videos of people falling. We also prompt GPT-4o to segment the room into a list of "safe zones", where risk of serious falls are less likely. In the prompt, we tell GPT-4o[11] to consider the surfaces (floor vs bed), locations (bathroom vs bedroom), the materials the surfaces are made of (carpet vs hardwood floors), whether surfaces have sharp points (such as the corners of a granite countertop) and other possible risk factors that are present in the environment. Then, we use Video-LLaVA[8] to process our data by feeding the extracted frames into image and video encoders simultaneously. We generate a prompt that includes this list of "safe zones" and areas the model should pay extra attention to. We concatenate the audio transcription to the end of the prompt, before feeding the processed videos and prompt into an LLM for processing. This LLM outputs whether a fall is detected, and generates a description to justify why they believe so.

3.2 Action Planning Model

Once the fall is detected, another prompt is generated containing the actions that are available on an Android device. This prompt is fed into an LLM (which functions as the action planning agent), along with the aforementioned description, so that the model can determine the seriousness of the fall and the best or most efficient course of action. The model is asked to generate and order a series of actions that can be executed, including attempting to converse with the individual verbally, contacting emergency contacts, or directly messaging emergency services to declare an emergency. If the agent tries to converse with the person, is prompted with the information and formulates a conversation with the person. Unless the person confirms that they are fine in a set amount of time, the agent takes further action at its own discretion. It is also capable of terminating this process once it deems fit (such as when help has arrived or when an emergency contact confirms that the person is fine).

4 Experiments

4.1 Datasets

The Le2i Fall Detection dataset is a publicly available dataset containing 221 videos of multiple people falling or performing other activities. We used 130 of these videos, depicting activities and falls in coffee rooms and homes.

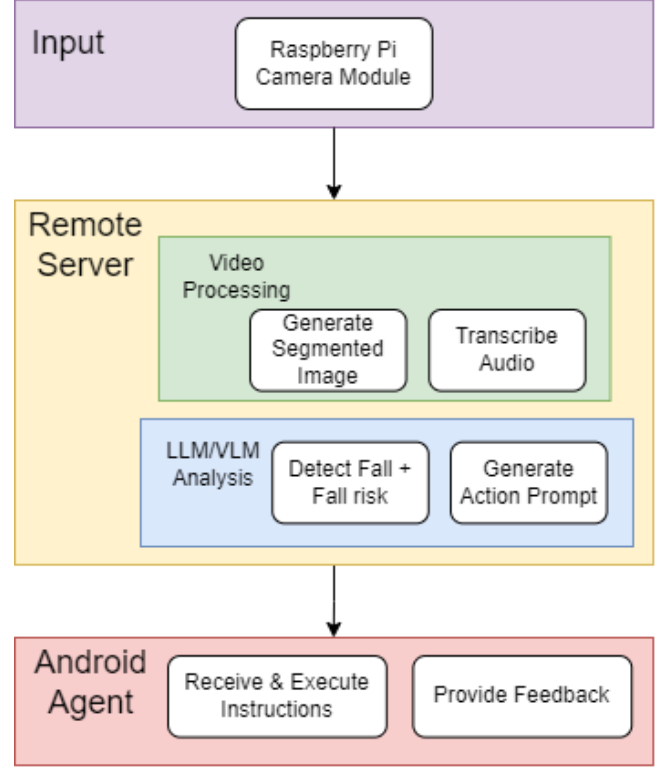


Figure 1. Here is a simplified diagram illustrating how our model works, as described in the Methods section.

However, the Le2i Fall Detection dataset is not sufficient for our purposes because all the actors fall on mattresses, which suggests to our model that the fall is less likely to be serious. Therefore, we collated 100 other videos from the UR Fall Dataset[7], which has videos of people either falling or performing Activities of Daily Living (ADLs). 60 of these videos depicted falls while 40 of the videos depicted ADLs. For each fall in the dataset, there were 2 corresponding videos inputted into the model: one from the side view and one from the top view, resulting in footage from 30 actual falls that were used.

Furthermore, we compiled another 100 videos from the CAUCAFall Dataset[2]. They used 10 subjects, who each fell 5 times in different directions and performed 5 different ADLs commonly confused with falling by AI models. In total, this dataset contributed 50 fall videos and 50 ADL videos.

We then analyzed our datasets. Firstly, we counted how many video falls and ADLs were present in each dataset.

We input these videos into our model and our baseline, which generates text indicating whether a fall has occurred and why it believes that a fall has or has not occurred.

Dataset	Falls	ADLs	Total
Le2i	98	32	130
UR	60	40	100
CAUCAFall	50	50	100
Total	208	122	330

Table 1. Breakdown of videos by Dataset

4.2 Evaluation Metrics

For Fall Detection, we check if our model correctly identified if a fall has occurred. However, we noticed that sometimes a model’s explanation for whether a fall had occurred would not align with their answer. Therefore, we came up with a new metric to measure their self-consistency: the confidence score. For this, we procured the same frames that were originally inputted into their respective model and provided GPT-4o with these frames in the same chronological order, as well as the explanation generated of what happened in the video. We then asked GPT-4o to generate a confidence score from 0 to 1, allowing up to 3 decimal places, reflecting how consistent the explanation was with what happened in the frames, and to make an assertion whether the person had actually fallen given the frames and description. We also prompted GPT-4o to generate an explanation for its confidence score for ease of evaluation of failure cases, or cases where the confidence score is less than 0.500. If the confidence score was less than 0.500, or if its assertion did not align with the model’s output, we marked the test case as a failure case. We then recorded these results for each video.

We then evaluated these results against the ground truth according to three metrics: Precision, Recall, and F1 Score.

Precision, which measures what percentage of the model’s fall detections are actually falls, is evaluated as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall, which measures what percentage of the actual falls are correctly identified, is evaluated as follows:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

Finally, the F1 score combines both metrics and is evaluated as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

We recorded all the Precision, Recall and F1 Scores for each batch of 30 videos. We then used this data to calculate the mean and standard deviation of these scores.

For the self-consistency analysis, we recorded the confidence scores generated by GPT-4o for all the evaluated videos. We then took these scores and calculated their mean and standard deviation.

4.3 Methods compared

Baselines. In our research on enhancing fall detection and safety considerations for seniors, we evaluated several state-of-the-art models, including GPT-4o, GPT4o-mini, Video-LLaVA, and SAM (Segment Anything Model). The evaluation focused on two primary aspects: fall detection and safety considerations.

GPT4o-mini is OpenAI’s miniature lightweight version of the state-of-the-art GPT-4o model. Both leverage pre-trained language models to analyze sequences of images and contextual data from video feeds. This model excels in processing and interpreting human poses, actions, and environmental contexts to identify falls accurately. The strength of GPT4o-mini lies in its ability to understand and contextualize sequences of frames, providing a robust analysis of human activities.

Video-LLaVA is a state-of-the-art Vision Language Model that processes visual and textual data for analysis. For fall detection, Video-LLaVA processes real-time video feeds using video and language encoders to interpret visual cues and process information.

As for safety considerations, we used Segment Anything Model (SAM). It is capable of identifying all distinct elements present in a given image. SAM applies a color mask to the image of the environment given, accurately tracing each object’s outline, and then outputs an image with each element either outlined or colored.

Performance. Table 2 outlines our results when we implemented SAM with GPT-4o and Video-LLaVA respectively.

Model	Precision	Recall	F1 Score
SAM + GPT-4o	91% $\pm$ 0.05	92% $\pm$ 0.05	92% $\pm$ 0.05
SAM + Video-LLaVA	57% $\pm$ 0.11	41% $\pm$ 0.10	47% $\pm$ 0.09

Table 2. Results of SAM with Respective Backbone Models

For ease of analysis, Table 3 provides the confusion matrices for the implementation of SAM with Video-LLaVA and GPT-4o respectively.

SAM + Video-LLaVA				SAM + GPT-4o			
actual value	p		total	actual value	p		total
	p'	n			p'	n	
	n'				n'		
	88	128	217		161	21	182
	65	19	83		12	106	118
	total	153	147		total	173	127

Table 3. Confusion Matrix for SAM + Video-LLaVA and SAM + GPT-4o respectively

Model	Incoherent	Inaccurate	Total
GPT-4o	0%	7%	7%
Video-LLaVA	0%	38%	38%
SAM + GPT-4o	0%	11%	11%
SAM + Video-LLaVA	18%	47%	65%

Table 4. Results of SAM with Respective Backbone Models as a percentage of all videos

Table 4 provides further analysis of the failure cases of each model and the reason why they failed. We noticed that some of Video-LLaVA's output simply repeated the prompt or consisted of repeated phrases or words, resulting in an output that made no sense. We defined this output to be "incoherent". More specifically, this is when the response is just a repeat of the prompt or when any word, character, or phrase is repeated more than 15 times consecutively in the output. Consequently, "inaccurate" outputs are defined as outputs that are not "incoherent" but give wrong, conflicting or low-confidence predictions.

When SAM was incorporated, the number of "incoherent" outputs alone surpassed the total number of failure cases from GPT-4o. We noted that due to the large amount of information provided in the prompt, Vicuna, the LLM used by Video-LLaVA, was overwhelmed by the prompt and could not provide a coherent response to whether the person had fallen. GPT-4o, on the other hand, was trained to handle large prompts and could better decipher what the prompt asked it to do, therefore allowing it to output a relevant and coherent output every time. However, Video-LLaVA still made more than 4 times as many "inaccurate" predictions as GPT-4o made. This suggests that Vicuna was not able to make good use of the safety score data. Upon review of the descriptions Video-LLaVA provided, the outputs suggested that Video-LLaVA had failed to process key frames in the video, failed to recognize key indicators of falls, or failed to understand the temporal relationship between the frames, inaccuracies that GPT-4o was capable of picking up and flagging.

4.4 Ablation study

Backbone analysis without Safety consideration. We conducted an ablation study to investigate the effects of each part of our model on the final output. Firstly, for the fall detection part, we removed the safety score model, instead directly feeding frames of each video, spaced exactly 1 second apart, into the VLM or GPT-4o. Our results for Video-LLaVA against our baseline, GPT-4o, are recorded in Table 6.

Upon evaluation, GPT-4o has higher precision, recall, and F1 scores than Video-LLaVA. GPT-4o also has a smaller standard deviation, indicating that it is more reliable and consistent in its output compared to Video-LLaVA. While GPT-4o was able to achieve similar results with or without the safety

Model	Precision	Recall	F1 Score
GPT-4o	93% $\pm$ 0.04	93% $\pm$ 0.04	93% $\pm$ 0.04
Video-LLaVA	71% $\pm$ 0.07	80% $\pm$ 0.08	75% $\pm$ 0.07

Table 5. Results of Backbone Analysis

scores, Video-LLaVA's precision score notably fell 32% and its recall score dropped 13%.

Video-LLaVA				GPT-4o			
actual value			total			total	
	p	n		p	n		
	p'	n'		p'	n'		
	178	43	221	182	10	192	
	73	6	79	11	97	108	
	total 251	49		total 193	107		

Table 6. Confusion Matrix for Backbone Model Video-LLaVA and Backbone Model GPT-4o respectively

However, backbone Video-LLaVA still performed better than when we incorporated SAM with it. We wanted to understand why Video-LLaVA performed so poorly when SAM was included. Upon reviewing the confusion matrix, Video-LLaVA was much more likely to flag an activity as a fall when the safety scores were not included. This suggests that Video-LLaVA is less capable of identifying the important instructions contained in the prompt than GPT-4o is. Our theory is further supported by our analysis of the failure cases of Video-LLaVA when combined with SAM. Some of these cases stated safety scores and used them as an indication of safety, even if the person had fallen. This suggests that Video-LLaVA could not understand its main task of determining if the person had fallen from the frames of the video, instead focusing too much on the safety scores and how they suggest this perceived "safety" of objects in the room.

Consistency check w/o Safety consideration. We also checked for consistency between the model's description and explanation of what happened, and what actually happened in the video. We did this by inputting the same frames of the video into GPT-4o and asking it to provide a confidence score between 0 and 1. We allowed for confidence scores up to three decimal places. We evaluated 10 batches of 30 videos each, calculated the Precision, Recall, and F1 Scores, and calculated their means and standard deviations. Here is a table containing the results of this evaluation:

Model	Score (no safety score)	Score (w safety score)
GPT-4o	99% $\pm$ 0.02	99% $\pm$ 0.04
Video-LLaVA	81% $\pm$ 0.23	56% $\pm$ 0.33

From the data, GPT-4o demonstrated much more accurate and reliable performance with regard to its self-consistency. Video-LLaVA, on the other hand, is less likely to be self-consistent and its performance varies wildly as indicated by the large standard deviations of its confidence score. This suggests that Video-LLaVA's output tends to lie at one of two extremes: completely right or completely wrong. This suggests that Video-LLaVA's descriptions are not as thorough as those given by GPT, hence making it easier to label its descriptions as completely right or wrong, a finding that confirmed after we looked through the failure cases for Video-LLaVA. Furthermore, when we analyzed the average length of the output description, Video-LLaVA's output length (excluding incoherent responses) was half of that of GPT-4o when SAM was incorporated into both models' prompts. This supports our theory that Video-LLaVA's outputs were less detailed and shorter.

Model	Length of output (# of characters)
GPT-4o	180 $\pm$ 43
Video-LLaVA	334 $\pm$ 91
SAM + GPT-4o	567 $\pm$ 152
SAM + Video-LLaVA	260 $\pm$ 116

Table 7. Results of SAM with Respective Backbone Models as a percentage of all videos

However, Panickssery et al. [13] noted that LLMs can recognize their own output, allowing them to better understand descriptions generated by the same LLM. Since we used GPT-4o-mini to evaluate self-consistency, cases where GPT-4 models were used have an edge over the Vicuna model (based on Meta's Llama) that is used in Video-LLaVA. In the future, to eliminate this bias, we plan to use a different state-of-the-art LLM, such as Mistral, to evaluate model outputs.

4.5 Evaluation on zero-shot setting

We also set up a Raspberry Pi camera and enabled it to live stream a video recording into our model.

However, many activities were underrepresented in our dataset. For example, common failure cases included ones where the individuals in the video were working out or performing activities in dim environments. To further refine our models, we used our Raspberry Pi camera to capture more videos of actors in dim environments performing actions often falsely identified as falls.

Ultimately, the goal of testing our model's zero-shot inference capabilities is to allow for the detection of other more dangerous falls. Other more dangerous activities that also constitute falls, such as tripping over something or slipping

on a wet floor, are not feasible or unsafe to replicate, but cannot be overlooked due to their commonness and the danger they pose to the elderly.

5 Discussion

5.1 Limitations and Assumptions

Our model only evaluates the objects in the room for their risk level at the start of each day and assumes that the makeup of the room does not change much throughout the day.

5.2 Dataset Limitations

There are a few limitations of significance. One limitation is that the dataset fails to account for non-serious falls like on mattresses or other soft surfaces. Furthermore, the majority of dataset videos either lacked audio or did not match up with the falling. Some of the videos also do not show the actions performed after the fall, making it hard for the model to gauge if the person was capable of getting up after the fall and by extension, the severity of the fall. Lastly, the data we currently have is skewed, with there being many more videos of falls than activities of daily living (ADLs).

5.3 Model Limitations

Firstly, the cost of calling the GPT-4o API may not be very feasible, especially if we plan on maintaining a constant stream. Furthermore, as the model currently relies on a VLM running on a server, the system will work without internet connectivity.

5.4 Future Work

With the limitations of the dataset as mentioned previously, we plan to refine our dataset by expanding the categories available from simply "fall" and "no fall" to include options such as "soft fall" when a person falls on a mattress, "hard fall" when a person falls on the ground, and "soft but dangerous fall" when the person collapses on a mattress but is evidently in distress. In terms of our fall detection component, we plan to fine-tune an LLM such as GPT-4o-mini to produce more accurate results. In addition, we would like to integrate a model such as SAM directly into an open-source VLM such as Video-LLaVA for better performance. Additionally, we would like to look into the possibilities of curbing the costs of using constant VLM monitoring for fall detection. In terms of our Android App, we plan to utilize edge computing to outsource some of the computation from the server to the app. Finally, we would like to assess the real time interaction between a person's body parts and parts of an object. This will allow us to generate better assessments of possible injuries sustained and determine how each part of the body is affected.

6 Conclusion

In an age where technology is increasingly intertwined with our lives, AI-backed healthcare systems significantly enhance seniors' quality of life and provide peace of mind to their families. This paper presents a novel AI-driven aid system that integrates Vision-Language Models (VLMs) and Large Language Models (LLMs) to deliver comprehensive care to the elderly. The system employs a multi-layered approach to improve both fall detection and action planning. VLMs monitor real-time video feeds, while audio transcription helps the LLM assess the severity of a fall and confirm its occurrence. Safety scores evaluate potential injuries and provide additional information for emergency services. Upon detecting a fall, LLMs determine the next best course of action, such as calling emergency services or notifying emergency contacts, which is then executed by an Android Agent, making the system an end-to-end solution.

The experimental results demonstrate the effectiveness of the proposed system in accurately detecting falls and planning appropriate actions, thereby addressing the critical issue of fall-related injuries and fatalities among older adults. By leveraging advanced AI technologies, our system sets a new standard in fall detection and proactive care planning. Future work will focus on further refining the system's algorithms, incorporating additional contextual data, and expanding the system's capabilities to cover a broader range of emergency scenarios, ultimately ensuring a safer and more secure environment for seniors.

References

- [1] Lê Đình Tuyên. 2023. Le2i Fall Dataset. <https://www.kaggle.com/datasets/tuyenldvn/falldataset-imvia>
- [2] Jose Camilo Eraso, Elena Muñoz, Mariela Muñoz, and Jesus Pinto. 2022. Dataset CAUCAFall. <https://data.mendeley.com/datasets/7w7fccy7ky/4>
- [3] Yue Fan, Lei Ding, Ching-Chen Kuo, Shan Jiang, Yang Zhao, Xinze Guan, Jie Yang, Yi Zhang, and Xin Eric Wang. 2024. Read Anywhere Pointed: Layout-aware GUI Screen Reading with Tree-of-Lens Grounding. arXiv:2406.19263 [cs.CL] <https://arxiv.org/abs/2406.19263>
- [4] Centers for Disease Control and Prevention. 2021. Nonfatal and Fatal Falls Among Adults Aged ≥ 65 Years — United States, 2020–2021. *Morbidity and Mortality Weekly Report* (2021). <https://www.cdc.gov/mmwr/volumes/70/wr/mm7035a1.htm>
- [5] Centers for Disease Control and Prevention. 2024. Older Adult Falls Data. CDC (2024). <https://www.cdc.gov/falls/data-research/facts-stats/index.html>
- [6] Sowmya Kasturi, Alexander Filonenko, and Kang-Hyun Jo. 2019. Human Fall Recognition using the Spatiotemporal 3D CNN. *The International Workshop on Frontiers of Computer Vision* (2019). https://www.researchgate.net/profile/Sowmya-Kasturi/publication/333852390_Human_Fall_Recognition_using_the_Spatiotemporal_3D_CNN/links/5d08ea45299bf1f539cc6580/Human-Fall-Recognition-using-the-Spatiotemporal-3D-CNN.pdf
- [7] Michal Kepski. 2014. UR Fall Detection Dataset. <http://fenix.ur.edu.pl/~mkepski/ds/uf.html>
- [8] Peking University Yuan Lab. 2023. Video-LLaVA. <https://github.com/PKU-YuanGroup/Video-LLaVA>
- [9] Jinfu Liu, Baiqiao Yin, Jiaying Lin, Jiajun Wen, Yue Li, and Mengyuan Liu. 2024. HDBN: A NOVEL HYBRID DUAL-BRANCH NETWORK FOR ROBUST SKELETON-BASED ACTION RECOGNITION. *Multi-Modal Video Reasoning and Analyzing Competition - ICME Grand Challenge* (2024). <https://arxiv.org/pdf/2404.15719>
- [10] Zhe Liu, Chunyang Chen, Junjie Wang, Mengzhuo Chen, Boyu Wu, Xing Che, Dandan Wang, and Qing Wang. 2024. Make LLM a Testing Expert: Bringing Human-like Interaction to Mobile GUI Testing via Functionality-aware Decisions. , 13 pages. <https://doi.org/10.1145/3597503.3639180>
- [11] OpenAI. 2024. GPT-4o. <https://openai.com/index/hello-gpt-4o/>
- [12] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (July 2024), 3580–3599. <https://doi.org/10.1109/tkde.2024.3352100>
- [13] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. arXiv:2404.13076 [cs.CL] <https://arxiv.org/abs/2404.13076>
- [14] Raspberry Pi. 2024. Raspberry Pi Documentation. <https://raspberrypi.com/documentation/accessories/camera.html>
- [15] Haoxuan Qu, Yujun Cai, and Jun Liu. 2024. LLMs are Good Action Recognizers. *Conference on Computer Vision and Pattern Recognition* (2024). https://openaccess.thecvf.com/content/CVPR2024/papers/Qu_LLMs_are_Good_Action_Recognizers_CVPR_2024_paper.pdf
- [16] Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small LLMs Are Weak Tool Learners: A Multi-LLM Agent. arXiv:2401.07324 [cs.AI]
- [17] Lei Wang and Piotr Koniusz. 2023. 3Mformer: Multi-order Multi-mode Transformer for Skeletal Action Recognition. *Conference on Computer Vision and Pattern Recognition* (2023). https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_3Mformer_Multi-Order_Multi-Mode_Transformer_for_Skeletal_Action_Recognition_CVPR_2023_paper.pdf
- [18] Hao Wen, Yuanchun Li, Guohong Liu, Shanhui Zhao, Tao Yu, Toby Jia-Jun Li, Shiqi Jiang, Yunhao Liu, Yaqin Zhang, and Yunxin Liu. 2024. AutoDroid: LLM-powered Task Automation in Android. , 15 pages. <https://doi.org/10.1145/3636534.3649379>
- [19] Da Yin, Faeze Brahman, Abhilasha Ravichander, Khyathi Chandu, Kai-Wei Chang, Yejin Choi, and Bill Yuchen Lin. [n. d.]. Lumos: Learning Agents with Unified Data, Modular Design, and Open-Source LLMs. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.
- [20] Yuxuan Zhou, Xudong Yan, Zhi-Qi Cheng, Yan Yan, Qi Dai, and Xian-Sheng Hua. 2024. BlockGCN: Redefine Topology Awareness for Skeleton-Based Action Recognition. *Conference on Computer Vision and Pattern Recognition* (2024). https://openaccess.thecvf.com/content/CVPR2024/papers/Zhou_BlockGCN_Redefine_Topology_Awareness_for_Skeleton-Based_Action_Recognition_CVPR_2024_paper.pdf

7 Appendix

7.1 Engineering Design Flow

7.2 Dataset

We compiled a total of 330 videos from three datasets: the Le2i, UR and CAUCAFall Datasets. Here is an example of how we formatted our dataset in a .json file.

```
[
  {
    "id": 1,
    "filepath": "./data/Annotation_files/video_(1).txt",
    "location": "Coffee Room",
    "original_location": "Le2i Dataset",
    "groundtruth": "Fall",
    "fall_start_frame": 48,
```

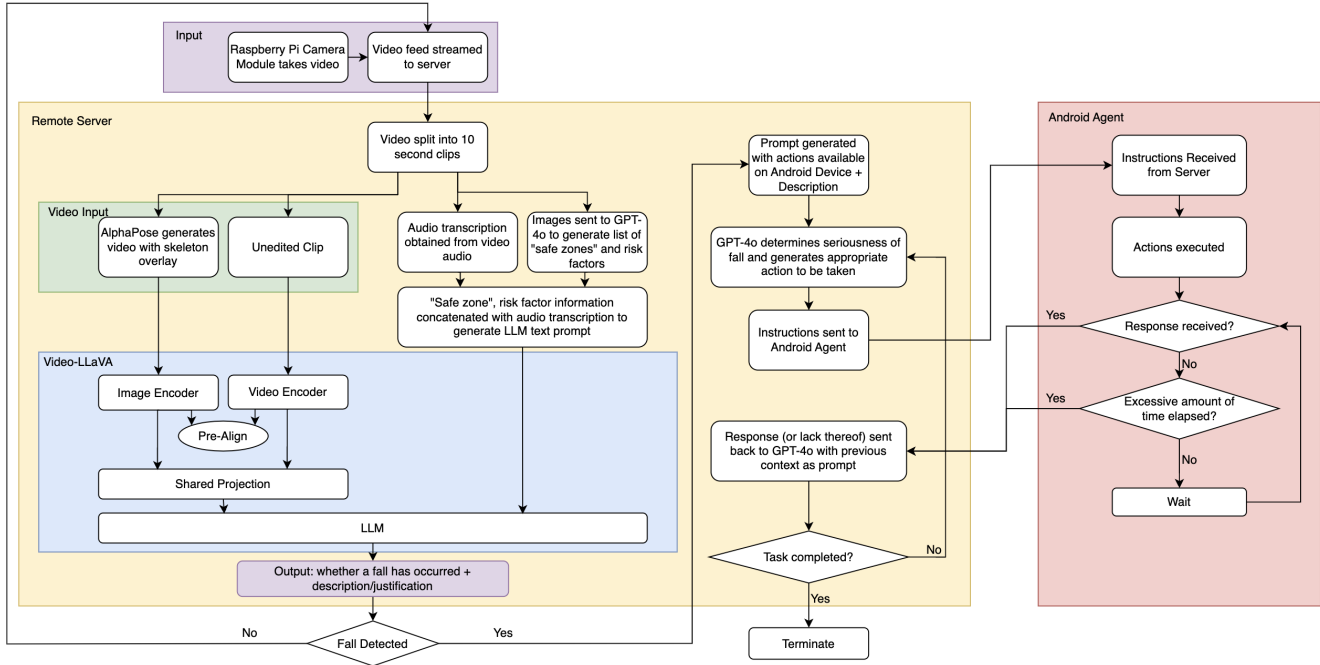


Figure 2. Here is the architect diagram of the action detection part of our model as described in the Fall Detection Model part of the Methods section.

```

    "fall_end_frame": 80,
    "point_of_view": "Wall-Mounted Camera",
    "action": "Fall"
  },
  ...
]

```

7.3 Evaluation Outputs

We also formatted our evaluation outputs as shown below to facilitate ease of failure case analysis.

```

[
  {
    "ID": 1,
    "Video IDs": [
      38,
      314,
      ...
    ],
    "Model Output": [
      "Y\n\nThe person is seen walking and then lying down...",
      "N\n\nThe person in the images is seen moving a chair...",
      ...
    ],
    "Model Predictions": [
      "Y(conflict)",
      "N",
      ...
    ],
    "Ground Truth": [
      "Y",
      "N",
      ...
    ],
    "True Positive Count": 14,
    "True Negative Count": 13,
    "False Positive Count": 2,
    "False Negative Count": 1,
    "Precision": 0.875,
    "Recall": 0.9333333333333333,

```

```

    "F1 Score": 0.9032258064516129
  },
  ...
  {
    "id": "Failure Cases",
    "failure_cases": [
      {
        "batch_id": 1,
        "ground_truth": "Y",
        "model_output": "Y\n\nThe person has fallen...",
        "model_prediction": "Y",
        "false_positive": false,
        "false_negative": false,
        "video_number": 235,
        "video_path": "../data/Videos/video_(235).avi",
        "frame_numbers": [
          0,
          20,
          ...
        ],
        "time_of_frames(seconds)": [
          0.0,
          1.0,
          ...
        ],
        "confidence_score": 1.0,
        "educated_prediction": "Y",
        "confidence_score_explanation": "Confidence score: 1.000\n\nThe description accurately matches the sequence of frames provided... The subsequent frames show the person remaining on the floor, indicating a fall.\n\nY"
      },
      ...
    ],
    "id": "Final Evaluation",
    "Total True Positive": 168,
    "Total True Negative": 101,
    "Total False Positive": 17,

```

Empowering Senior Healthcare with VLMs and LLMs: Enhanced Fall Detection and Proactive Care Planning

```
"Total False Negative": 14,
"Precision (Mean)": 0.9067...,
"Precision (Standard Deviation)": 0.0476...,
"Recall (Mean)": 0.9240...,
"Recall (Standard Deviation)": 0.0526...,
    "F1 Score (Mean)": 0.9150...,
    "F1 Score (Standard Deviation)": 0.0468...
}
```