FALL DETECTION SYSTEM

Justin Haryanto

Nhan Nguyen

SYSTEM REPORT

REVISION – 1.0

30 April 2023

System Reports

FOR

Fall Detection System

Prepared by:

_____

Author Date

Approved by:

_____

Project Leader Date

_____

John Lusher II, P.E. Date

_____

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1.0	30 Apr 23	Nhan Nguyen		Draft Release

Table of Contents

List of Tables

List of Figures

1. Overview

The sponsor requested a video based fall detection system which utilized pose estimation. This meant a system where a user could provide video input and receive results on whether the people inside the video have fallen or not with good accuracy. The final result of this is seen below in figure 1. Here the user could upload a video or series of images into the website and receive results in the form of a playable video where the uploaded file is edited to include the word “Fall” to indicate when a fall occurred. The results page also allows users to upload more files for further tests and the website in general could adjust its layout to meet the size of the screen it is displayed on.

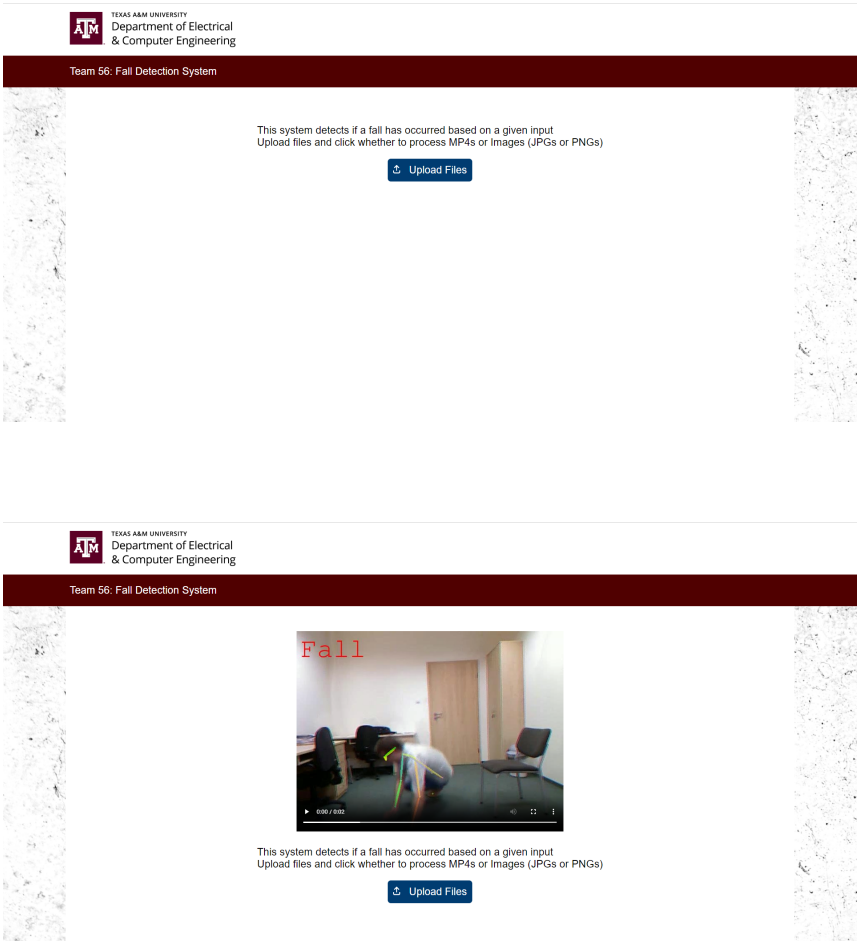

Figure 1: Website Home Page and Results Page

The system was created as two main subsystems, the Fall Detection subsystem and the Externals subsystem. A website GUI from the external subsystem allows users to upload videos and images to the Fall Detection system. The Fall Detection system processes an inputted video/image through a pose estimation system called AlphaPose to extract results on keypoints and bounding boxes for each frame. This data is sent to two machine learning (ML) models, random forest and CNN, to be analyzed for any human performing falling motions. These models were trained from collected data sets, a part of the externals. The results from this analysis are sent back to the website GUI as a video for the user to review. This system report is to allow for the review of decisions and benchmarks of the development of the fall detection system, particularly the integration of the subsystems.

2. Development Plan and Execution

2.1. Design Plan

The initial design was just the video processing and pose estimation system connected to a fall detection as seen below in figure 2. This did not originally include any form of user interface. We initially plan on refining the pose estimation system, AlphaPose, to be better able to handle unusual poses made by humans, and having the video processing as a more separate component. There are other features such as live video input.

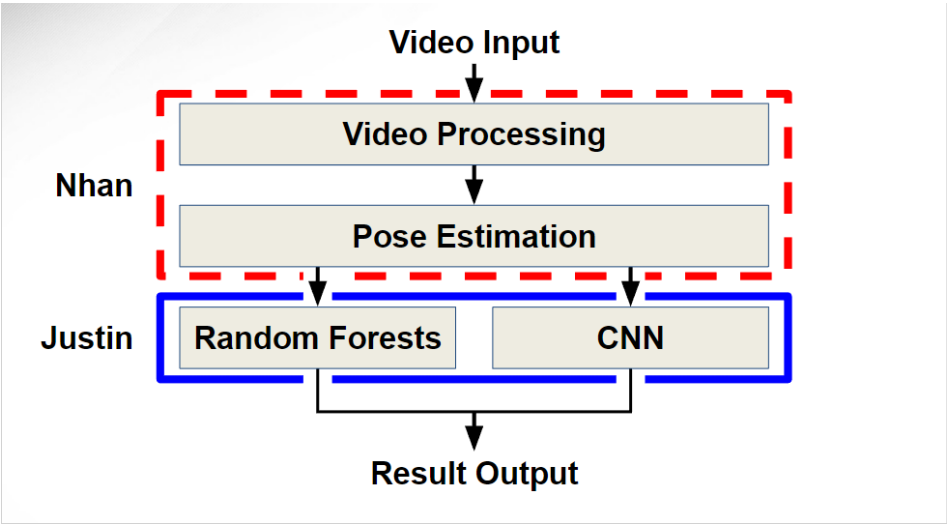

Figure 2: Original Design

Improving AlphaPose proved beyond our ability and the need for a GUI became more clear once we started integrating the systems and encountering issues with interacting with the fall detection code implemented on Google Colab. This led to the restructuring of the system design seen below in figure 3, with a Website UI as the GUI. Features like live video input were determined to be optional.

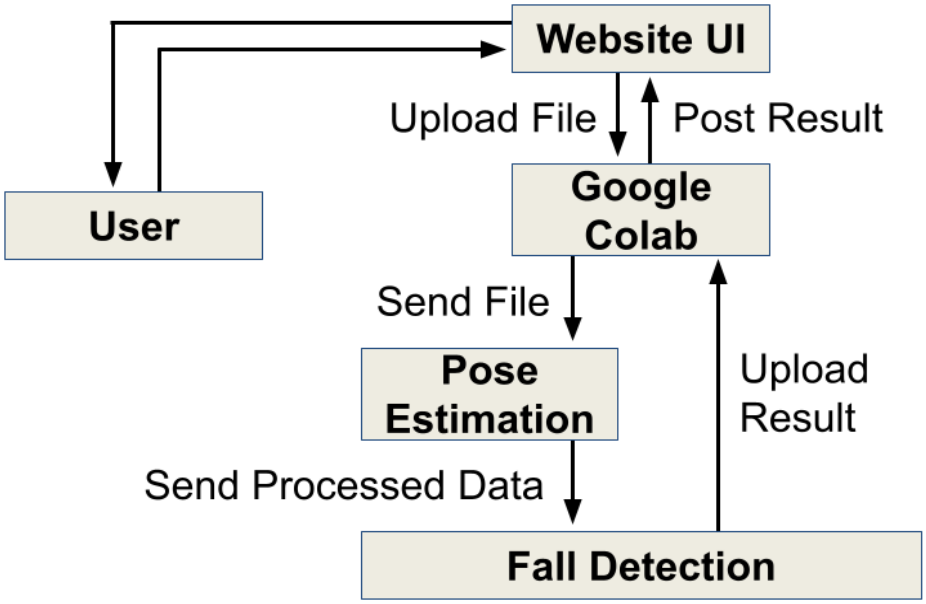

Figure 3: New Design

2.2. Execution

Before integrating, we started by finalizing the subsystems. This involves planning out, implementing, and troubleshooting the new website GUI. The fall detection subsystem was further refined with the pose estimation and machine learning models better integrated than last year, and refining the models to be more accurate by altering their parameters and retraining.

Integration begins once the subsystems are usable. The Pose Estimation component is properly integrated with the Fall Detection system, making the full Fall Detection System. Datasets were further collected and used to train and refine the ML models of this system. Video testing sets were collected to evaluate the whole system. The website GUI is integrated with the full Fall Detection system to make up the whole system. Testing is done, with further refinements, troubleshooting, and validation.

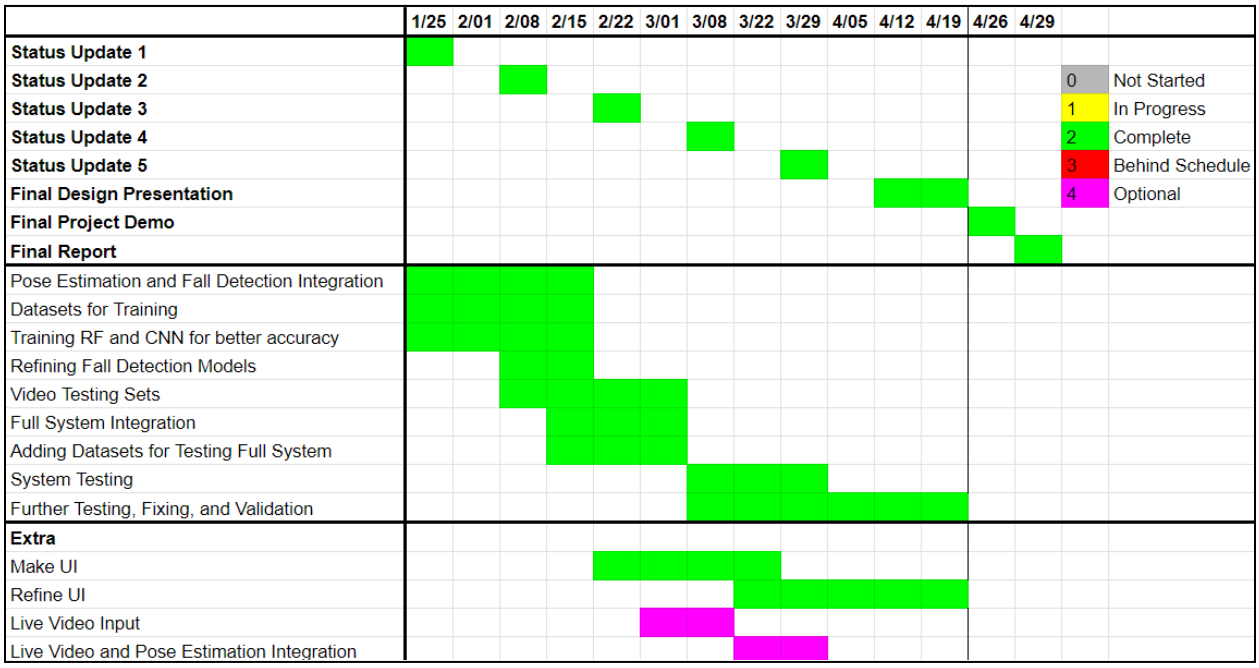

Figure 4: Execution Plan

2.3. Validation Plan

The full validation plan is shown below. The fall detection system was tested for accuracy and computation time/resources. The accuracy of the random forest, CNN, and whole fall detection system was partly validated. The computation time/resources were partly validated. The number of collected data sets and testing sets were validated. The website user interface and functions like website scaling were validated. Whole system testing and functionality was partly validated.

	Specifications	**Results**
Data Sets	Process at least 15 UR and 5 Adhikari fall detection data sets	30 UR and 10 Adhikari fall datasets
Testing Sets	Create at least 10 480p videos with a duration of 10 seconds at 15 FPS compatible with the system	78 MP4 clips with a duration of 5 to 7 seconds
User-Interface/ Website	User able to submit an MP4 or collection of images to the UI, the UI should return the processed files as a MP4 and display to the user	Designed a website, based on a typical A&M website, that allowed for mp4/images upload and result return
Website Scaling	Website display on a variety of devices with appropriate layout adjustments	Website could autoscale based on device it is displayed on
Live Video Input	The system should accept continuous series of video clips from a webcam at a minimum of 10 fps at 480p	None, specification is optional
Fall Detection Accuracy	>90% of falls successfully detected	80% accuracy on own videos
Fall Detection CNN Accuracy	>90% of fall images and bounding boxes successfully detected	67% accuracy on UR dataset,17% accuracy on Adhikari dataset, 35% accuracy on both Adhikari and UR Fall Datasets
Fall Detection Random Forest Accuracy	>90% of fall keypoints successfully detected	96% accuracy on UR dataset, 92% accuracy on Adhikari dataset, 92% accuracy on both Adhikari and UR Fall Datasets
Fall Detection Computation Time	Make Fall/Not Fall Decision in 1-30 seconds	25 seconds

Fall Detection Computation Resources	Fall detection system take up 75% or less of system GPU RAM, CPU RAM,and Memory on average	25% GPU RAM,66% CPU RAM, and 33% Memory for avg video or series of images
Fall Detection Training Computation Time	Training Times is less than 24 hours	Training Time is approx 1 hour for running over all training data sets once, Training Time approx 7.5 minute for the average dataset
Fall Detection Training Computation Resources	Training take up 75% or less of system GPU RAM, CPU RAM,and Memory on average	95% GPU RAM,51% CPU RAM, and 18% Memory for training on part of data set at any one time
Whole System Stress Test	Successfully process a stream of videos and images without system breaking down	Website have trouble displaying videos, possibly due to network issues
Whole System Functionality	Can the user submit files to the website and get back a display of the video and the fall detection results	Successfully submitted video/image files and receive results

Table 1: Validation Plan

3. Conclusion

The final design is a functional fall detector capable of determining if a fall has taken place in a video frame with reasonable accuracy. Many problems and challenges were encountered during the development of this system. The original design and plans needed to be rehauled into something more usable and within the team's capabilities. The new design allows for greater flexibility as the team could more easily use the Website GUI to work with the fall detection system situated in Google Colab. The resulting product is easier to use and meets most of the validation specifications, though there is room for improvement.

4. Future Work

Improvements to the machine learning models accuracy could be done, particularly for the CNN model. The models could also be further improved by training on a more diverse dataset which includes various motions that resemble but are not falls, or with unusual poses. This would reduce the potential false positives or negatives in the system. The system is also fairly slow, taking seconds to minutes to process videos, so research in speeding up the system could be examined. Furthermore the website GUI occasionally experiences network issues when displaying results, a problem that could be rectified by further development.