SCC210 GROUP PROJECT DESIGN REPORT

Group 1

Lyu Yunbo; Mao Chenhui; Meng Peng;
Wang Haoze; Li Xinlong; Sang Haotian

CONTENTS

Introduction **1**

Overview 1

Initial Ideas 1

Software Design **4**

Chosen Idea and Features 4

Implementation 5

Principle 7

Software Engineering **9**

Requirements 9

Use Case 10

Accessibility Tests 14

Implementation Plans **17**

Task List 17

Active Network 18

Gantt Chart 18

Milestones 19

Deliverables 20

Appendices **21**

Reference List 21

A Visual Approach of Fall Detection Based on Convolutional Neural Networks

Introduction

Overview

For the influence of social and economic motivation, improving the health of the elderly has attracted considerable attention. And the protection of the fallen elderly has always been a common research topic in different fields. Although a variety of wearable devices have provided accurate fall detection results, the insufficient popularity of these devices and the preference of some users who are unwilling to wear them commonly limits the application scope of these devices. Fortunately, the concept for Internet of Things and the increasing number of cameras in public places provide excellent hardware support for computer vision recognition systems, which makes us able to detect fall actions through the cameras. We can build an intelligent fall detection system and analyze whether the frame sequence of video image contains human body falling action. But there are various difficulties to this method, such as the picture being blurry, or the fallen person being blocked by obstacles. In this design report, we try to overcome these obstacles through the Convolution Neural Network (CNN), and use some preprocess methods to assist. In addition, we believe that the activity of the surrounding crowd (if any) can also improve the accuracy of the judgment of fall actions.

Initial Ideas

Falls caused by unexpected events among the elderly is a major health concern, especially in the aging society where the elderly population is suffering from physical weakness. Approximately half of all injury-related hospitalizations are aged over 65. Falling causes not only fracture and other physical injuries of disability, even death, but also causes psychological trauma, which will reduce the independence and confidence of elderly patients, and then produce a series of life issues. In the last few years, much research on the topic of fall detection has been widely conducted. Wearable devices and Image vision is becoming the most favored approach for many researchers. We mainly consider the latter, which analyzes events from body shapes via video streaming using image processing methods to monitor and detect falls. Therefore, our group focus our investigation in this field.

Due to the complexity and changeableness of fall situations, normal cameras might not be suitable for fall detection. Thus, we contemplated using the deep camera. There are many kinds of it, such as Rangefinder, dual-lens cameras or Infrared camera. However, most of them are expensive or difficult to deal with the data. We also considered the Kinect, as shown in Figure 1.1, which is a cheap machine which was released in 2010 as an accessory of Xbox [1]. Using Red Green Blue (RGB) camera in the middle, we can get a clear figure in good light conditions and the infra-red

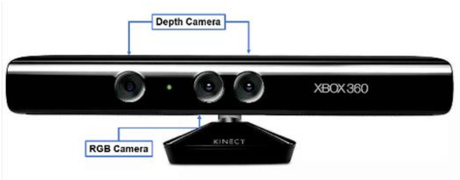


Figure 1.1 Microsoft Kinect

LED provides images in weak light condition. And the video caught by Kinect can be preliminarily consulted and output by extract the human joint which is easier to judge the motivation of human being. After constructing the module of robust fall and modify by suitable data set, we can obtain a good solution.

An idea is to use the model called the Adaptive Directional Bounding Box, which uses the skeleton 3D joint points retrieved from RGB-D information of Kinect, proposed by Apichet [2]. This type of box is adaptable to different movement configurations and flexible enough to apply with different body shapes of humans. There are four main new features used in this system to analyze the fall events: (1) height/width Ratio, (2) Center of Gravity Ratio, (3) Diagonal Ratio, and (4) BB-Height Ratio. In addition, this method also introduced the dynamic state machine to encompass both forward and backward tracking to ensure that a particular movement led to a true falling state. Experimental results showed that it provides high performance in both accuracy and speed.

An alternative idea is by recording the acceleration and angular velocity of human motion (like walking, falling, jogging, standing up, going up-stairs, sitting down) which are very different from one to another, and using it as a training set [3]. By setting this training set as an input of Convolutional Neural Networks (CNN), we can obtain the category of the human motion state. However, how to extract the characteristic value of human motion state from a video? One way is by mapping the values of human motion (acceleration and angular velocity) in 3-axes (X, Y, Z) into values of 3 channels of RGB image. Namely, each 3-axial data can be converted into RGB pixel through normalization processing. And through recording the changing trend of red, green, and blue value in a video, we can extract the characteristic value of the human motion. Figure 1.2 shows an

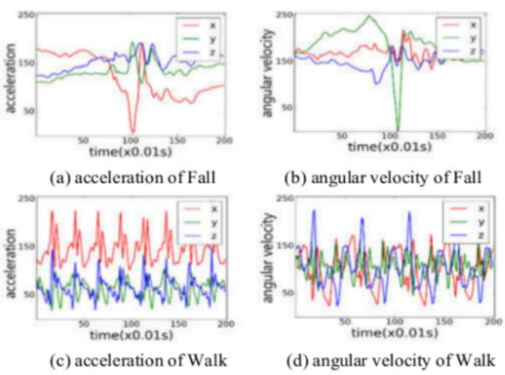


Figure 1.2 Acceleration and Angular Velocity of Falling and Walking

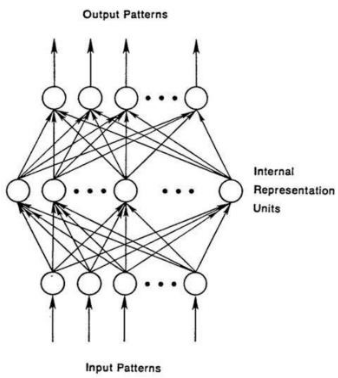


Figure 1.3 Recurrent Neural Networks

example of angular velocity and acceleration value that mapping into RGB pixel between fall action and walking. As a consequence, with existing categories which can be obtained from the CNN's output layers, we can judge whether the object in a video is falling or not.

Another idea that followed is to create a fall detection system which is based on recurrent neural network (RNN), which is shown in Figure 1.3. The recurrent neural network mainly solves the problem of how to deal with changes in time series [4]. Firstly, a serialization representation method on image sensor data, training and test data is designed as the basis for intrinsic relationship exploration. According to the characteristics of RNN, the data obtained by video capture devices are reconstructed and transformed into a list of entries which are suitable for RNN. Then, the training algorithm for RNN based on fall detection is proposed, where the fall detection is transformed into a classification problem of the input sequence. By comparing the probability that user's behavior is non-falling, falling or near-falling, the type of user's behavior is judged. Finally, using the large-scale RNN system based on distributed neurons, the fall detection system is implemented on the Spark platform. Evaluations are carried out on a specific dataset. In addition to distinguishing between normal and falling behaviors, as already existed fall detection systems can do, it can also accurately identify near-falling behaviors that are more dangerous.

Besides, we come up with an idea that we can train a principal components analysis network (PCANet) model[5], which applies feature learning methods to detect a fall action. Two models are obtained after the training stage: one is a single frame detection model and another is an action model. The first step of training stage is to extract the frame, which includes humans from video sequences of different views, and they form the training set. The training samples for the single frame detection model are labeled to three classes, namely Standing, Falling and Fallen. Then we can use all the samples to train a PCA Net model. Since a fall event is contained in many continuous frames, the reliable fall detection should analyze a video sequence instead of single frame. The next step is based on the prediction result of the trained PCANet model for each frame, which can be used to obtain an action model in sub video clips.

Software Design

In Chosen Idea and Features, we proposed seven features that should be included in our system. Then we discuss the process details in implementation.

Chosen Idea and Features

We analyzed the methods we proposed before and found some insufficient areas. Firstly, the 3D structured light's machine is expensive, and devices like Kinect can only detect objects in 4 to 5 meters and cannot be used in high light situations. Then, for the system based on recurrent neural network as well as the system based on recording the acceleration and angular velocity of human motion, despite its high accuracy, these approaches have one main drawback – namely, that they must rely on the data captured by multiple sensors which are attached to the object, which makes this system not suitable for our daily use. Our group would prefer a pure vision-based method. However, for the Adaptive Directional Boundary Box method, when obstacles block the camera's ability to see all the nodes clearly, the boundary box method cannot be implemented. And if the surveillance camera is not placed sideways or at the same level as a human, the PCANet will not work effectively. After weighing up various ideas, we decided to apply an alternative way that adopts a pure vision-based approach, and takes into account some extreme situations like when part of an object is sheltered, or the object is exposed to extreme light condition. Those aforementioned ideas about our project will be illustrated more in detail in the following part.

At first, our system should include the following features:

- The system can identify and extract the dynamic human falling motion and filter out the influence of background factors.
- If the sample size is sufficient, the accuracy of posture identification can be improved through machine learning.
- The diverse postures are rotated to upright position (normalize postures), in order to identify the fall event from different arbitrary viewpoints.
- The system does not require a high-quality image source.
- We use twenty specific frames for detection to improve speed and reduce the dependence on high-performance devices.
- We preprocess images in extreme illumination situations to make them normalized, which improves the rate of target detection.
- We totally base our method on computer vision instead of wearable devices or multiple sensors.
- The system is mainly used in public places, especially where pedestrians may fall.
- The system has high robustness since it can recover from failure.

Implementation

The system uses a camera to capture the live video from different circumstances, so that it can be applied to a wide variety of scenarios which are especially fit for public use. After the target has entered into the camera's field of view, the system extracts 20 frames from the video. Then the system preprocesses the frames in order to improve the quality of the image and normalize the targets from different viewpoints. And the system applies an optical flow algorithm to separate the dynamic target from the static background. Based on the optical flow images already recognised, the system extracts 16 skeleton points. We train the CNN on FDD dataset [6] and apply transfer learning [7] by reusing the network weights. We build the classifier to test whether the target falls or not. Besides, the system analyses the movement trend of the crowd in order to help judge the falling action. The whole process is shown in Figure 2.1.

More implementation details are showed in the following paragraphs.

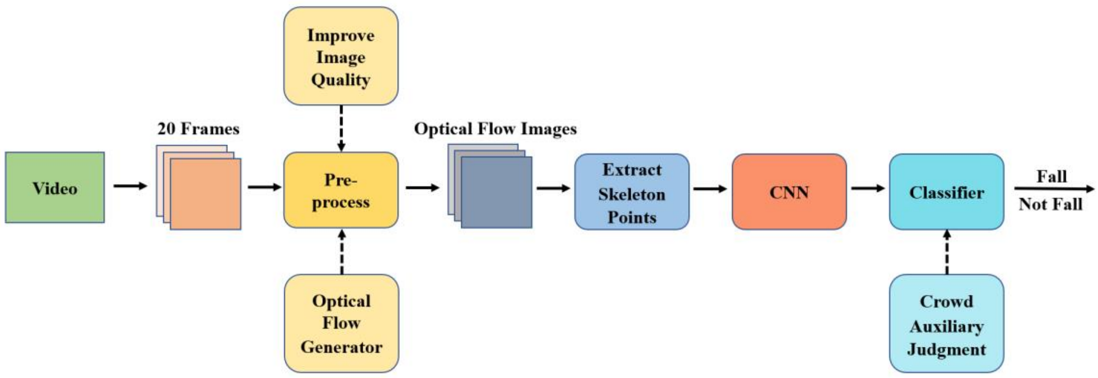


Figure 2.1 The whole process of detection

- (1) If we want to improve the practicality of fall detection system, the samples should be collected from different circumstances, such as cameras in roads, and monitors in banks or shops. That means the data we need has a wide range of resources. However, the 3D structured light camera is expensive and has many limitations. For example, the Kinect is the most common device which is used in scientific research, but it only can be used at most from 5 meters and cannot be exposed to a high light situation. Therefore, our algorithm is designed to suit different spaces resolving power and frame rate by detecting and extracting the dynamic target, and generic devices just meet our standards.

- (2) The system we designed can be applied to a wide amount of scenarios, and especially for public use. The system based on wearable devices or multiple sensors can only be used for some particular people who wear them day and night. However, our system, based on computer vision, uses a digital camera that is placed in public places to detect people falling in public. As long as people enter the area that the digital camera monitors, whoever falls down can be immediately detected by our system.

- (3) We find that if too many frames are extracted, the requirement of CPU performance and algorithm efficiency are extremely high which would not contribute much to the detection. On

(a)

(b)

Figure 2.2 (a) Images taken under normal illumination condition (b) images taken under weak illumination condition.

the other hand, fewer frames will lead to inaccurate results. Thus, in order to ensure the integrity of motion and sufficient accuracy, we can only extract twenty frames in the fall action video sequence [8].

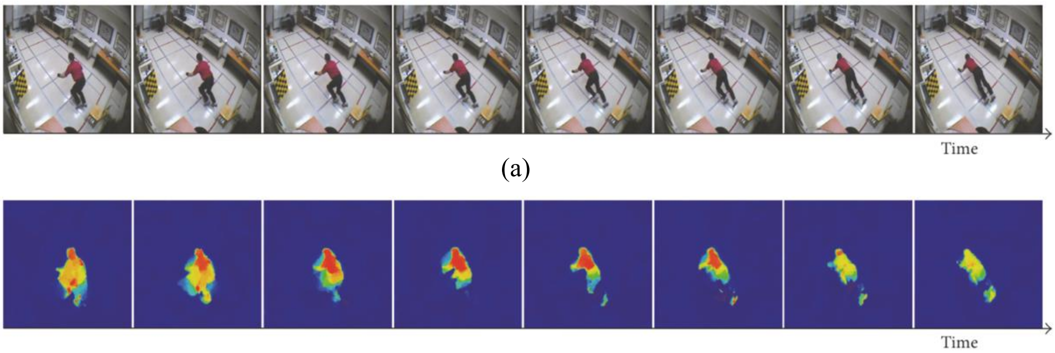
- (4) RGB values are mainly based on camera gain, illumination, object pose and the particular situation of the object. The frames which are ready to be analyzed should retain as few unnecessary changes as possible and retain the images that are crucial to the eventual decision. For example, it is hard for computers or algorithms to recognize a human from an image taken under extreme illumination condition, as two sets of pictures in Figure 2.2 shows that it is difficult to recognize target in dark condition even for a human; not to mention the computers.

Figure 2.3 Rotate diverse postures to upright postures

Hence, we need to process poor images before using them to train the model. The preprocessed and normalized image can greatly improve the rate of target detection.

- (5) In practice, the pictures taken may be people from different angles. In order to detect the same action, the cost of attitude recognition based on sample learning is really high. In order to reduce the number of learning samples, we advocate that targets from different viewpoints should be normalized to that from the same viewpoint first. As shown in Figure 2.3, which puts forward an easy approach through rotating the picture to an upright posture to achieve normalized images. In such a way, not only can the cost of learning can be reduced, but the speed of detection can also be increased.

- (6) Because of the large amount of data in video processing, it is necessary to extract the key target quickly. And since the shooting position is fixed, the physical environment captured by the camera seldom changes. We can quickly extract the target by taking advantage of the feature that the human is moving in different frames but the environment is fixed. These images can be used to generate a sequence of optical flow images [9], as shown in Figure 2.4.



(a)

(b)

Figure 2.4 (a) Sample of sequential frames of a fall (b) and their corresponding optical flow horizontal displacement images.

- (7) Accuracy is one of the biggest potential obstacles that blocked our project. Achieving high accuracy requires both high complexity of the algorithm as well as high performance of computer itself which can be hard to achieve. To make it more specific, we prepare sufficient training sets as input of the CNNs, which have been shown to be very versatile automatic feature extractors. As long as the amount of the training set is sufficient, the CNNs will be able to learn the set of falling features, with which we can judge whether the target has fallen or not with high accuracy.

Principle

When involving AI project design, it is crucial to consider those key principles that are relevant to AI design. Thus, we combined the Google AI design principle and proposed the key design principles as follows that have influenced the design of our project.

- Our design should be beneficial to society.
- Our design should avoid unfair prejudice such as for the disabled.
- Our design should guarantee accuracy and security.
- Our design should be constructed with the goal for serving human society.
- Our design should have scalability which can be modified by real-time requests.
- Our design should respect and protect privacy so that videos and pictures will not be passed to someone else.

  Security is another issue we should consider, due to our project being applied in public places. Therefore, protecting personal privacy is particularly critical. In our project, we decided to achieve security by not passing the processed video to someone else except the system staff. Additionally, only authorized users can access the video backup, which ensure the security from the peripheral equipment.

- Our design should consider about important problems that needs to be solved.

  When designing our project, there are several problems that are crucial to our project. Among which illumination conditions, targets that are partly sheltered, or angles can especially cast a negative impact to our judgment of whether the target fell or not. Thus, solving these problems are important.

- Our design should consider circumstances which can be used in multiple situations.

  Understanding the application scenarios can contribute to the design of our system, because the function and feature of our system should focus on that specific scenario. As for our system, which is applied in public places, all the features and functions of the system should focus on the characteristic of the public places.

- Our design should learn and predict human behaviors, and repetitive learning will help promote the procedure of precise calculation.

  The program is trained to analyze characteristics through the given data set. The data set includes a large number of image sequences which includie human falling motion. The accuracy and representativeness of data sets have a vital impact on the performance as recognition rate of the program. A good data set could make the program predict similar behaviors and promote the procedure of precise calculations.

- Our design should be practical which runs fast, recognizes precisely and is cost-efficient.

  Efficiency is significant in video processing. The program must be able to process the extracted frames fast enough; otherwise, it would lead to failure of fall identification. Another assessment criterion of the recognition program is precision rate. Our design must make sure that the recognition accuracy is sufficient; otherwise, our work would make no sense.

Software Engineering

Requirements

We set up a series of requirements for our fall detection system, including both functional requirements (FR) and non-functional requirements (NFR).

Table 3.1 Requirements table

ID	Description	FR/ NFR
R.1	The system shall display the surveillance video.	FR
R.2	The system shall be able to track and detect the human joints.	FR
R.3	The system should rotate the diverse postures to upright positions (normalize postures).	FR
R.4	The system shall be able to detect the fall of humans.	FR
R.5	The system shall be able to discriminate fall actions and fall-like actions.	FR
R.6	The system should prompt manual verification when the accuracy of judgment is insufficient.	FR
R.7	The system shall alert when a fall event is detected.	FR
R.8	The system shall send a geographic location of the event when sending an alert. e.g. Ticket Gate 3, London Railway Station.	FR
R.9	The system should determine whether there is a fall event with the help of the reaction of the surrounding people.	FR
R.10	The system shall display the probability of falling.	FR
R.11	The system shall reset when a exception occur.	FR
R.12	The system should make a complete action judgment in a relatively short time, e.g. 10 seconds.	NFR
R.13	Relatively well image preprocessing should be completed in a reasonable time, which meets the need of judgment system.	NFR
R.14	The system should achieve high recognition accuracy, e.g. 70%.	NFR
R.15	The system should identify people's falls from different arbitrary viewpoints.	NFR
R.16	Preprocesses should be able to deal with extreme light conditions.	NFR

ID	Description	FR/ NFR
R.17	The preprocess should be able to deal with the occlusion of the objects.	NFR
R.18	The system shall detect the situation of multiple target falls.	NFR
R.19	The system shall keep user's privacy by not being accessible by a third party.	NFR
R.20	The system should take a hierarchical function design, which makes the system scalable, maintainable and adjustable.	NFR
R.21	The system should save the falling scene for a period of time, e.g. three months.	NFR
R.22	The system shall have friendly user interface.	NFR

Use Case

We make a use case, which is used to help us propose the requirements.

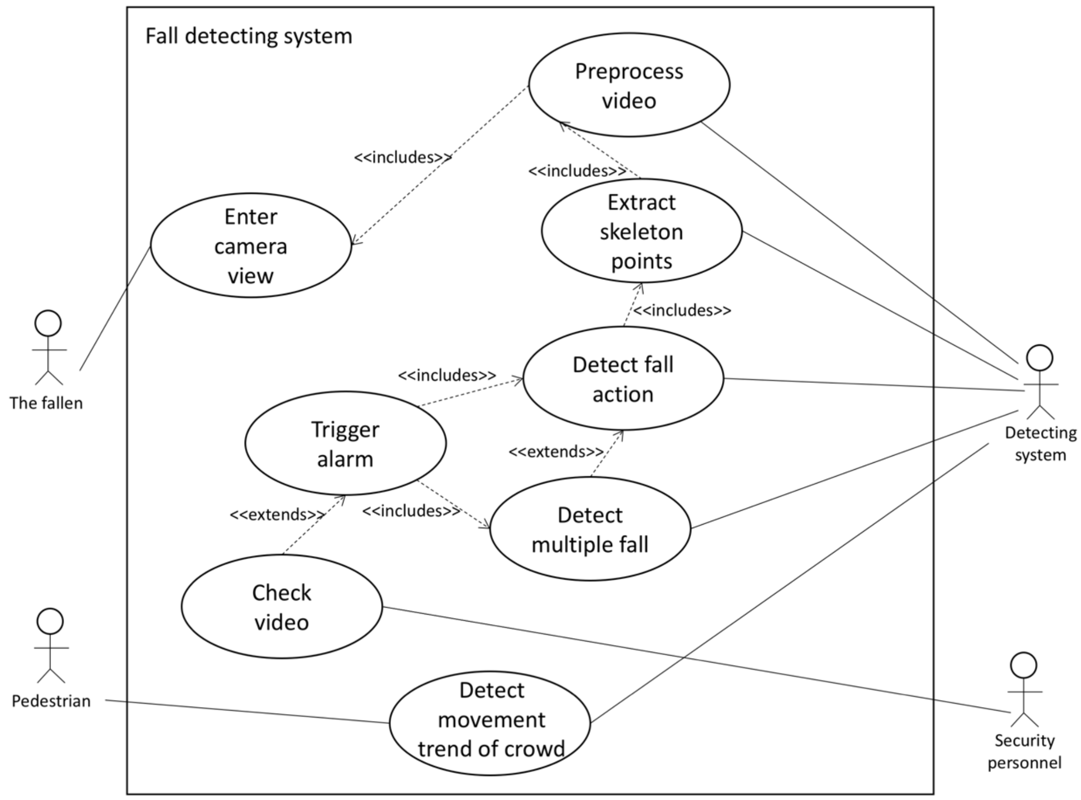


Figure 3.1 Use case

Table 3.2 Use case table for the 'Preprocess video' use case

Use case name:	***Preprocess video***
Scope:	Fall detecting system
Primary actor:	Detecting system
Secondary actors:	None
Summary:	Before detecting the falling of the target, optimized the quality of the image and separate the target from the image
Preconditions:	Targets enter into the camera's field of view
Main success scenario:	1. The target has entered into the camera's field of view 2. The system extracts 20 frames from the video 3. The system adopts smoothing method to deal with the noisy within the image and improve the quality of the image 4. The system enhances the contrast ratio of the image 5. The system applies optical flow algorithm to separate the dynamic target from the static background 6. The system rotates the separated target to upright position 7. Passing the image to the Extract skeleton point part
Alternatives:	3.a. Applying an image quality evaluating algorithm, if the image up to standard, then directly jump to step 5.
Exceptions:	5.a. The target is failed to be separated from the background 6.a. The target is failed to be rotated to upright position
Postconditions:	The target has been separated from the image

Table 3.3 Use case table for the 'Extract skeleton points' use case

Use case name:	***Extract skeleton points***
Scope:	Fall detecting system
Primary actor:	Detecting system
Secondary actors:	None
Summary:	Based on the frame image already preprocessed, the detecting system extracts the skeleton points which are used to detect the fall action later.
Preconditions:	The detecting system successfully get the frame image which contains a human object

Main success scenario:	1. The detecting system gets the frame image already preprocessed 2. The detecting system analyses and identifies the human object from the frame image 3. The detecting system extracts skeleton points of the human 4. The detecting system matches these skeleton points to the corresponding part of the human body
Alternatives:	3.a. The current frame image does not contain a human, the detecting system processes the next frame image continuously
Exceptions:	1.a. Fail to get the frame image 2.a. Fail to identify the human object
Postconditions:	The skeleton points are successfully extracted from the frame image and ready to detect the fall action.

Table 3.4 Use case table for the ‘Detect movement trend of crowd’ use case

Use case name:	***Detect movement trend of crowd***
Scope:	Fall detecting system
Primary actor:	Pedestrian
Secondary actors:	None
Summary:	The pedestrians have been detected the movement trend
Preconditions:	The pedestrians are in a state of turmoil and rapidly gather together
Main success scenario:	1. The pedestrians are turmoil and gather together 2. Camera catch the condition and put into system 3. The system analyses the crowd behavior and preliminary judge 4. The system invokes the fall action detects’ judgmental consequence 5. The system combines the two judgements and make the final judgment 6. The system trigger alarm if judgement is fall, and keep detecting if judgment is not fall
Alternatives:	4.a. The system directly jumps to step 6 if fall action detects’ judgmental consequence reaches the line of ‘quite sure’
Exceptions:	3a. The system warns the security personal if cannot judge
Postconditions:	The system finish detecting movement trend of crowd.

Table 3.5 Use case table for the ‘Detect fall action’ use case

Use case name:	***Detect fall action***
Scope:	Fall detecting system
Primary actor:	Detecting system
Secondary actors:	None
Summary:	Determining whether the object exists falling action by calculating the parameters in the model of skeleton points
Preconditions:	The detecting system has extracted skeleton points
Main success scenario:	1. The system abstracts a three-dimensional model based on the extracted skeletal points 2. The system calculates parameters such as spatial position, motion velocity and persistent time of a special skeleton point in real time 3. The system automatically saves the current real-world image sequences, the skeleton points model and the current time when detecting falling motion 4. The system sends information to trigger alarm
Alternatives:	2.a. (Additional step) After calculation, the system checks the results according to the movement trend of the surrounding crowd 3.a. The system does not determine whether the falling detection result is accurate or not, so that it sends information to the manual check
Exceptions:	1a. The system displays warning identifier if it falls to abstract a standardized skeleton points model 3a. The system warns if it does not save the information successfully
Postconditions:	The system finish detecting fall action.

Table 3.6 Use case table for the ‘Manual’ use case

Use case name:	***Check video***
Scope:	Fall detecting system
Primary actor:	Security personnel
Secondary actors:	None
Summary:	To determine whether the target actually falls through manual checking
Preconditions:	There is a fall or fall-like event and the system is not very sure about it

Main success scenario:	1. The system detects a fall or fall-like event 2. The system is not very sure that there is a fall event, e.g. the certainty is less than 50% 3. The system transfers video recordings to security personnel for manual check 4. The security personnel confirm that there is a fall, then triggers an alarm
Alternatives:	4a. There is not actually a fall event, then the security personnel ignore it
Exceptions:	4a. Security personnel can't determine whether someone has fallen by video recordings
Postconditions:	Alarm is triggered or not.

Accessibility Tests

Table 3.7 Accessibility tests table for preprocessing

ID	Description	Input	Excepted Output	P/F
1	The system can improve the image quality	Original video	The success rate of detecting poor-quality-video at least 60%	
2	Dealing with video in extreme light conditions	A video record in extreme lighting condition, e.g. night	The success rate of detecting dark-video at least 60%	
3	Extract appropriate twenty frames	A complete video	Twenty frames containing the entire action	
4	Rotate the diverse postures to upright positions	Image from different viewpoints	Target with normalized position	
5	Extract correct skeleton points	Image of the target	Image with 16 correct skeleton points	

Table 3.8 Accessibility tests table for detection

ID	Description	Input	Excepted Output	P/F
1	Capture of the fall action of single target	A video with only one fallen target	The system can extract clips that indicates the fall action	
2	Capture of the fall action of multi-targets	A video with several targets	The system can extract clips that indicates the fall action	
3	Obstacle test	A video which contained obstacle	The system can detect targets whose occlusion does not exceed 50%.	
4	Trigger alert	Signal of detecting the falling motion	Send the alerts in 10 secs	
5	Manual verification	A part of video that triggers the alert	The alert is correctly triggered	
6	Identification rate for single fall	A video containing fall action	The identification rate of detecting is higher than 70%	
7	Algorithm time-consuming	Run one-time complete system	The system can detect a fall action in 10s	
8	Stability of running	Running system	The system could steadily run for 24 hours	
9	Reset the status after a detection	The signal of finishing detection	The system can successfully restart detection	
10	Sequential fall	Continuous surveillance videos	The system can continuously detect fall event	

11	Save for 3 months	The data and source material	The system can store the data for 3 months	
12	Detection of surrounding people	A video contains group behavior	The system can detect fall event more accurately with the help of group behavior	
13	The system can send accurate geographic location	A video material from specific location	The system will return the location of video shot	

Implementation Plans

Task List

The table above shows the task list of our project from week 13 to week 16. Contents in this table include the ‘Weeks’, ‘Task’ items and describe how we arrange the work in each week. The tasks were divided equally among each group member, and we finished the project together.

Table 4.1 Task list

Weeks	Task
13	Come up with project ideas
	Reading materials
	Choose an idea under the guidance of tutor
	Create task list
	Learn about Git
	Make notes for report
14	Sketch out project features
	Produce principles
	Make implementation rules
	Describe software requirements
	Make notes for report
15	Design use case
	Plan acceptance tests
	Milestones
	Deliverables
	Activity networks
	Gantt chart
	Check grammar
	Make notes for report
16	Check and modify report
	Format the document
	Add content and cover sheet
	Learn image preprocessing
	Learn features extraction
	Finish report

Active Network

The following chart is an active network, which represents the different critical stages of the project development. A total amount of 24 weeks is allocated for planning and design, 120 for implementation, and 16 for project testing. The parameter “T” in this chart indicates one person working on for 1 week or equivalent time.

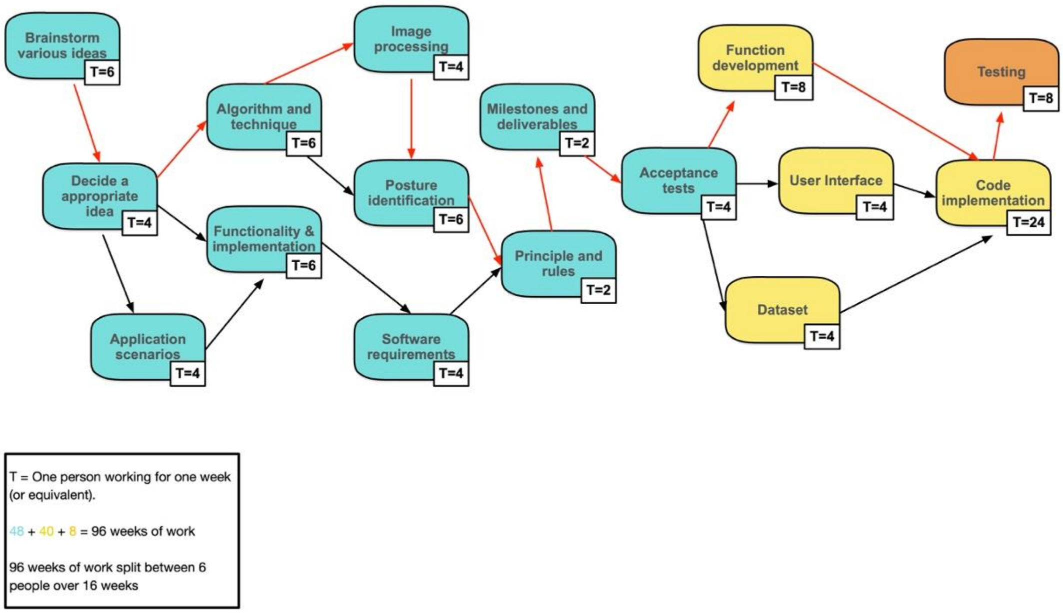


Figure 4.1 Active network

Gantt Chart

This Gantt chart below displays our project separated into several unique key tasks. Each has given planned time and duration time that this task has lasted. Some tasks like code implementation and User interface development can be carried out simultaneously.

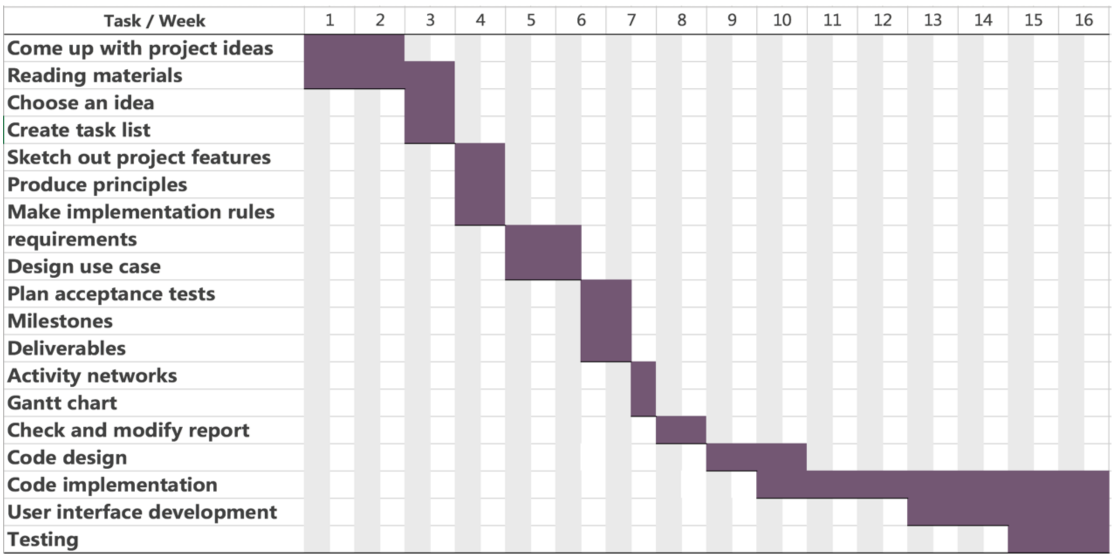


Figure 4.2 Gantt chart

Milestones

There are many important milestones in the process of implementing the fall detecting system that describe the mean functions of the system. The table below illustrates these milestones which contain the start date, end date and duration. What's more, the table also shows the dependencies of internal contents and the acceptance of test procedure.

Table 4.2 Milestones table

Task	Description	Internal dependencies	Acceptance test	Start date	End date	Duration
1	Extract 20 frames which reflect consecutive action.		P3, D7	25-Feb	27-Feb	2 days
2	Implement optical flow algorithm to determine the target.	T1		28-Feb	1-Mar	2 days
3	Extract skeleton point of the target.	T1, T2	P5	2-Mar	6-Mar	4 days
4	Train the CNN on the chosen database. For this purpose, we use optical flow images of consecutive frames to teach the network how to detect different actions.	T3		7-Mar	14-Mar	7 days
5	We apply transfer learning by reusing the network weights and fine-tuning the classification layers.	T4		15-Mar	18-Mar	3 days
6	Build classifier to test whether the target fall or not.	T4	D1	19-Mar	22-Mar	3 days
7	Add preprocessing to video under extreme illumination condition and test.	T1	P2	23-Mar	27-Mar	4 days
8	Add preprocessing to videos taken from different angles.	T1	P4	28-Mar	1-Apr	4 days
9	Optimize Fall Detection for Occluded Targets.	T2	D3	2-Apr	5-Apr	3 days
10	Add the detection function for multi-target.	T2	D2	6-Apr	9-Apr	3 days
11	Add the function of assistant judgment through pedestrian response.		D12	10-Apr	12-Apr	2 days
12	Implement User Interface and alarm Functions.		D4, D5, D13	13-Apr	14-Apr	1 day

13	Testing.	T1-T12	P1-P5, D1-D13	15-Apr	19-Apr	4 days
14	Fully runnable.	T1-T13		20-Apr	21-Apr	1 day

Deliverables

Based on our milestones table, we set up a deliverables table as below.

Table 4.3 Deliverables table

Description	Deadline
Extract skeleton point (single target)	06-March-2019
Classifier (single-target fall action)	23-March-2019
Preprocess (extreme illumination condition)	27-March-2019
Preprocess (videos from different angles)	01-April-2019
Classifier (multi-target falls action)	09-April-2019
Final project	21-April-2019

Appendices

Reference List

[1] Kwolek, B. , & Kepski, M. (2014). Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine, 117*(3), 489-501.

[2] Apichet, Y., Rasmequan, S. (2017). Adaptive directional bounding box from RGB-D information for improving fall detection. *Journal of Visual Communication & Image Representation.* Vol 49, 257-273.

[3] Vallejo, M., Isaza, V. & Lopez, J. (2013). Artifcial Neural Networks as an alternative approach to traditional fall detection methods. *Proceedings of the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC*, pp. 1648-1651.

[4] Musci, M., De Martini, D., Blago, N., Facchinetti, T. & Piastra, M. (2018). Online fall detection using recurrent neural networks.

[5] Wang, S., Chen, L., Zhou, Z., Sun, X. & Dong, J. (2015). Human fall detection in surveillance video based on PCANet. *Multimedia Tools and Applications.* 75. 10.1007/s11042-015-2698-y.

[6] Zerrouki, N. , Harrou, F. , Houacine, A. , & Sun, Y. . (2017). Fall detection using supervised machine learning algorithms: A comparative study. International Conference on Modelling. IEEE.

[7] Oquab, M., Bottou, L., Laptev, I. & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 1717–1724, IEEE, Columbus, Ohio, USA.

[8] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos.

[9] Hamprecht, F., Schnorr, C. & Ahne, B. (2007). A duality based Approach for Realtime TV-L1 Optical Flow. *Pattern Recognition*, pp. 214–223.