FALL DETECTION FOR IOT SMALL SENSORS

Ho Ko, Yueqi Liu, Jiaxin Xie

h25ko@uwaterloo.ca y2926liu@uwaterloo.ca j83xie@uwaterloo.ca

ABSTRACT

Falling accidentally or suddenly in the elderly often cause injures, sometimes even result fatally especially if not having the corresponding treatment in time. In recent years, there are numerous researches successfully detect fall incidents, while these solutions might be viable in a well established lab environment, they are usually really expensive to set up and requires a lot of power. It is essential to explore a cost-efficient solution, easier to deploy, lower power consumption rate, and within a reasonable cost. To this end, this paper proposes two system frameworks and detection models for the elderly fall detection, and analyzes the feasibility of the models implemented on ESP32-cam. In terms of system architecture, the distributed system mainly analyzes the monitoring screen by local IoT devices, while in the centralized system,the centralized machine is responsible for the analysis process which has cheaper computing power. There are several steps during the experiment parts: choose the data set, data preparation, training the model. And we pick the different falling videos to do the evaluation, the second model is more satisfying.

1. INTRODUCTION

Population ageing is a global phenomenon: Virtually every country in the world is experiencing growth in the size and proportion of older persons in their population, which poses challenges to medical insurance. According to a report prepared by the United Nations in 2019 [1], there were 703 million persons aged 65 years or over in the world in 2019. The number of older persons is projected to double to 1.5 billion in 2050. According to the Centers for Disease Control and Prevention, one out of three older people falls each year[2]. The consequences of falls for the elderly are very serious, possibly leading to fractures, brain injuries, high blood pressure and other diseases, which threaten the life and health of the elderly. If prevention solutions are not invested in the immediate future, the number of injuries caused by falls will be double in 2030 due to the increasing portion of old people[3]. At the same time, falling behavior can easily lead to psychological trauma to the elderly. After the elderly fell, it was difficult for them to raise their voices to call for help and they were prone to shock. If there is no help and company for a long time, they are prone to depression and resistance to social interactions and daily actions.

How to design a reasonable and accurate fall detection system based on the life characteristics of the elderly counts a great deal.To detect the fall incidents of elder people and inform the medical staff or family members immediately, the intelligent fall detection system plays an indispensable role. The inertial measurement unit in the wearable device[4]is able to detect body angle changes, whereas power supply requirements and wearing instructions of the sensors may be too complicated for a stay-alone elderly. The alternative solution is by placing sensors in the environment for fall detection such as deep-field camera and ultrasonic/infrared sensors[5], which is usually not affordable for ordinary consumers. Utilizing conventional camera is one of the best low cost fall detection scheme, but it had always been challenging to map out the distance features and other recognition problems from a 2D data. Recent development in Machine Learning allow this approach to be not impossible anymore[6]. In this project, we will implement RNN and CNN related models to detect falling incidents, and reducing the model allowing it to be fully functionable in cheap IoT devices.

2. RELATED RESEARCH

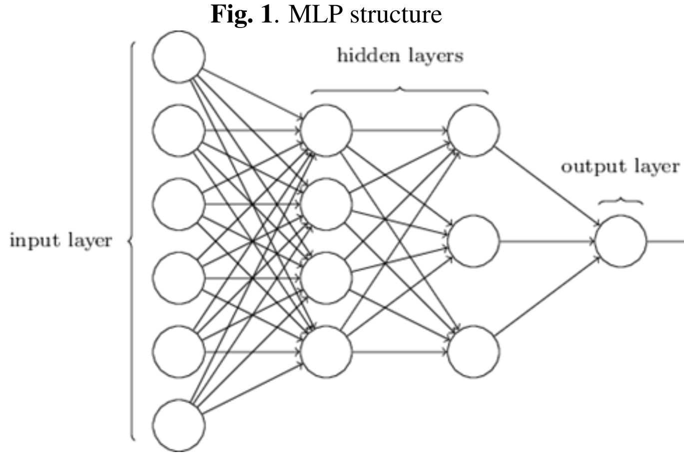
2.1. Machine Learning Model

With rapid development in the field of Machine Learning, image detection has made great progress. As deep neural networks have demonstrated impressive performance in this area. There are three following types of deep neural network are popularly used today: Multi-layer perceptrons (MLPs), Convolutional Neural Networks (CNN) and Recurrent Neural network (RNN).

2.1.1. MLP

For Multi-layer perceptron (MLP), a fully connected neuron network (Fig. 1), The model parameters increase rapidly with the growth of network depth. Each node is connected to every other node from the previous layer to the next layer, it has a huge amount of parameters that need to update and calculate. As a result, the model training will be difficult and lose the ability to generalize and even cause overfitting. Another common problem is that multi-layer perceptrons are not translation invariant. For example, If a flower is on the left-top of the first input picture and another flower is on the right-bottom of the second input picture, the MLPs will always try to correct itself that flower will always show in that place of the picture. Hence, MLPs are not the best idea to deal with image detection.

Fig. 1. MLP structure



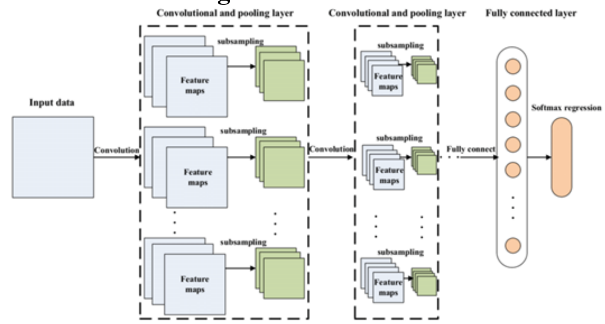
2.1.2. *CNN*

We need a way to reduce the weight number and leverage the spatial correction of the image features of pixels. By solving those problems above, we introduce a convolutional neural network (CNN). The main structure of CNN is involved Convolution layer, pooling layer after the convolution layer, and fully-connected layer(Fig. 2). Comparing with MLP, CNN moves a fully-connection structure to a sparse connection structure, and the weight value sharing between the neurons (Every node does not connect to every other node, a different group of neurons sharing the same weight values). This greatly reduces the pressure of parameter training and is less wasteful, which is easier to train than MLP and solves the problem of gradient descent. Another benefit of using CNNs is their ability to develop an internal representation of the two-dimensional image. This allows the model to learn the position and the scale-invariant structure in data, which is very important work for Image recognization. It considers the local connectivity since the filter is panned around (convolution) the whole image according to the filter size and the stride. Base on those benefits, CNN has become the main model dealing with image processing.

2.1.3. *RNN*

The signals of each layer of neurons can only travel up one layer in MLP and CNN, and the sample is processed independently at various times, while RNN can retain the previous value to some extent and inherit the characteristics of the

Fig. 2. MLP structure



past time series which makes RNN widely used in natural language processing, speech recognition, handwriting recognition and other scenarios. Although CNN is capable of capturing the local features of images, it is difficult to interact with information for a long time in videos of variable length. RNN's ability to extract information contained in continuous images gives it an advantage in the extraction of features in video images. Without supervision, the attention mechanism of RNN can be used to locate the area of each frame of the image to focus on the key areas of the video. Bin Zhao et al. [7] input the image characteristics of the video into the RNN model and introduce an attention mechanism to make it as a decoder to generate subtitles. In 2020 [8], Han Zhao et al. combined the advantages of different models and proposed a fusion attention model based on CNN and RNN to improve the recognition accuracy of human actions in videos.

2.1.4. *LSTM and GRU*

In RNN system, if the current system state is affected by the state of the system a long time ago, then the RNN system will have a long-term dependency problem, due to tendency of the gradient to disappear after multi-stage propagation. The derivative models of RNN, such as LSTM and GRU, can well solve the problem of long-term dependence. The structure of LSTM and GRU is shown in the Figure 3 and Figure 4.The core of these two models is the cell state, which is represented by a horizontal line running through the cell. These cell memories pass through each layer of the RNN network like a conveyor belt, conveying important state information.

LSTM has long-term memory and short-term memory, and it uses three gates to filter and determine the cell state.: forget gate, input gate, and output gate. The function of the forget gate is to use the sigmoid activation function to filter the current cell input and the previous cell input state. The function of the input gate is to activate the same input through the sigmoid and tanh functions respectively, and then obtain the summed output. There are two inputs to the output gate, one is the value of the forget gate, the input gate, and long-term memory activated by the tanh function, and the other is the filtered value of the current cell input and the previous cell state. With the continuous update of the cell state, important parts of historical information are retained, and unneeded information will be filtered. With the continuous update of the cell state, important parts of historical information are retained, and unneeded information will be filtered. There are some differences in structure between GRU and LSTM. The GRU model includes reset gate and update gate. It has no long-term memory and uses the cell output of this round directly as the input of the next cell.

Fig. 3. LSTM structure

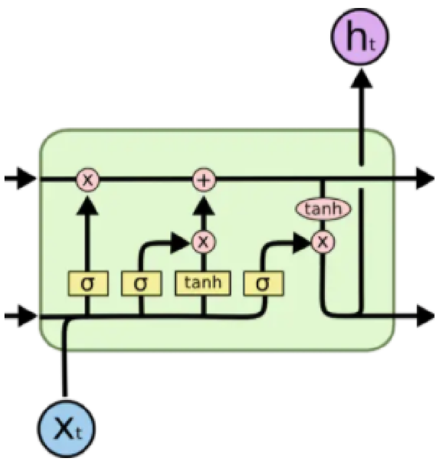
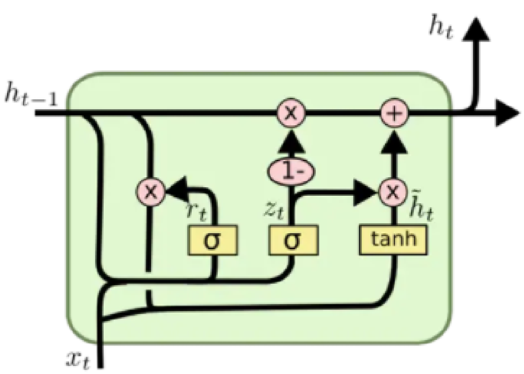


Fig. 4. GRU structure



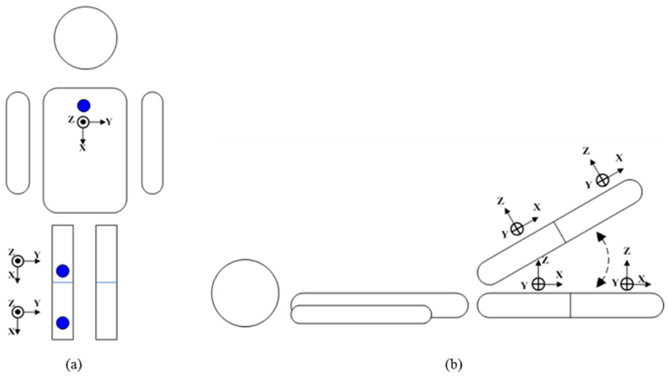
2.2. Portable Sensor and Other Sensor System Detects Falls

The data interaction and portability of the sensor system make it possible to detect the fall of the elderly. Arduino is a convenient and flexible open source electronic prototyping platform, which consists of a single-chip-based hardware platform and a set of Arduino IDE development environment. Because it can obtain a large amount of switch and sensor data, control different kinds of electronic components at the same time, and communicate with other programs running in the computer, Arduino is often used to develop interactive smart products. Jayashree et al.[9] built a portable fall recognition system for the elderly based on Arduino. By detecting the accelerometer and gyroscope data, it judged whether the tracked angle parameter has exceeded the set threshold. If it is judged that the elderly has fallen, the system will send an alarm message to the guardian. The characteristic of this system is that it is easy to install and carry, can be used in many scenes of the life of the elderly, and the cost is relatively low. However, since the sensor will produce zero-point drift error after power-on, and temperature drift will occur with the change of temperature, when the IMU is actually used, uncertainty and misleading factors are prone to appear. And since the elderly have limited knowledge about the power supply requirements of wearable devices, they may not be able to supply power to the system in time, leading to blind spots in the detection of elderly falls.

Another form of wearable devices uses the accelerometer that comes with the smart phone to detect the behavior of the elderly falling. With the development of information intelligence in the era, the popularity of smart phones among the population is gradually increase, and countless applications have been designed to facilitate people's lives. Sensors such as camera microphones, accelerometers, and gyroscopes embedded in smartphones increase the possibility of human-computer interaction. As the human body falls, the acceleration first decreases and then increases due to changes in body posture. In the data collection stage, Tran et al.[10] designed an Android program to detect the changing trend of the phone's accelerometer, and preset the human body to put the smartphone in the trouser pocket below the waist, allowing the young volunteers to fall down on the soft cushion and observe Acceleration change range. By observing the experimental data, the author sets the low and high thresholds for the determination of falling behavior, and defines that only when the acceleration reaches the low threshold and then reaches the high threshold can it be determined that the falling behavior has occurred. This system makes full use of the smart phone's sensors, but it suffers from great limitations in the experimental stage. The subjects of the experiment were young volunteers, which could not fully reflect the fall data of elderly users, and the experiment was carried out on soft cushions, without considering the error caused by the impact of the actual hard ground on the mobile phone and the sensor.

There are also a type of wearable sensor uses a three-axis accelerometer to measure the inclination of the body and is installed on the wrist or other body parts, or connected in shoes or clothing fabrics.[11, 12]. Besides, gyroscopes can estimate the rotational acceleration during failing.[11].

Fig. 5. Direction of tri-axial accelerometers [12]



Unfortunately, that wearable sensor may be disturbed by the external environment, like temperature and humidity, which will lead to false signals. Also, the mean user for wearable fall detection sensors are elderly, they may feel difficult to wear and disassemble frequently. Always wearing the sensor in the right places is also an important point to ensure accuracy, this may also be difficult for the elderly. Furthermore, people may feel uncomfortable wearing all those devices all day during walks or cooking, and they may even forget to change the battery of those devices.

Youngmin et al.[13] presented a real-time fall detection systems (FDS)which used smart phone accelerometers to detect the fall behavior of elderly users, and used Google 3D mapping service to track their location when the elderly fell, ensuring effective intervention in the fall of the elderly. The author analyzed the Signal Vector Magnitude (SVM) ,which reflects the overall changes in acceleration on the x-axis, y-axiss, and z-axis, without considering the change in the direction of acceleration, so that the elderly can put the mobile phone in any direction in the pocket of the clothes. However, if the mobile phone signal is poor or the mobile phone is out of power, this FDS can not do a good job.

Environmental elements are also a type of way to detect falling. For example, use a circular microphone to detect sound in the room, and distinguish if it is a falling signal. [14]; use fiber sensors or some other pressure sensors to capture the sudden weight change on the floor, and do falling distinguish by the floor pressure event[15]; Install two infrared sensors on different walls in the house to gain 3d image information, use k-NN classification method to detect falling action.[16, 17]. A common challenge for this method is hard to distinguish between human action and animal action. Usually, the detection accuracy can be interfered by the distance between the sensor and the humans, to ensure accuracy the distance can be no more than five meters away. So it is not practical and too expensive to install a large number of sensors in the house. Even for the wireless physical layers that use channel state at a low cost, detection will be interrupted by multiple people in the room, or if the furniture is pushed to the floor, which interferes with the mathematics used for monitoring only a single entity.

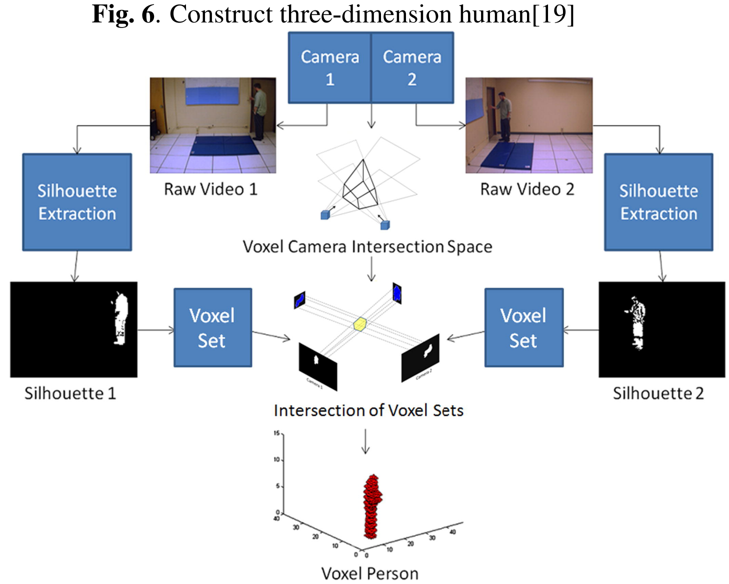
2.3. Image-based Detection with Machine Learning

With the rapid development in the field of deep learning, fall detection system based on computer vision has made great progress.

Another category for fall detection is image-based, which can be classified into a single camera, multi-camera, or images of 3D depth data. Descent events are detected by analyzing the volume distribution along the vertical axis, and an alarm will be triggered when a major part of this distribution gets unusually close to the floor at a predefined time, meaning that someone falls to the floor. This kind of device needs a very complex way to install and calibrate, which will cost a lot of time. Besides, it may also fail to distinguish the furniture falling and person falling[18].

For camera detection, using multi-cameras to monitor the same scene to construct a three-dimensional representation of a human body. Fuzzy logic and machine learning is used to predetermined human's action, by summaries the human works to figure out whether the human is falling or not[19].

Fig. 6. Construct three-dimension human[19]



The same challenge exists to detect the falls that are partially covered by an object such as furniture. So, many researchers have a focus on indirect fall detection base on inactivity time, spatially the time laying on the floor. However, this will send an error detection signal if the human is lying on the floor for rest or doing some floor exercises like yoga.

One published work by considering the less-consumption devices, choose to use Raspberry Pi 2 with regular USB camera and some other accessories (overall 91€), but the fall can only be detected when they are within 10 meters of the devices.[20]. Currently, even with the improved solution and frame rate of the upgraded Raspberry Pi 4, it is still difficult to monitor the entire house. Lastly, for the 3D depth data capture by Kinect devices, either a multi-camera of a single camera are used to detect[21], they may also fail to recognize the falling which happened behind the fog glass door (scenario: bathroom).

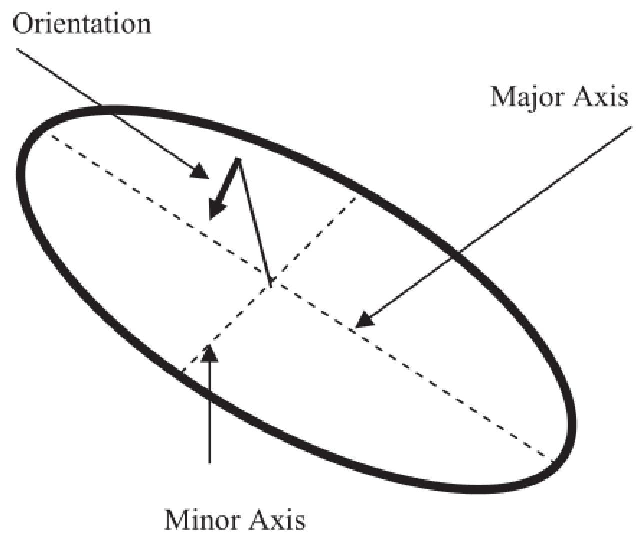
Nuttapong et al.[1] stated that by computing from Motion History Image (MHI) method which indicates the speed of human motion, C-Motion method is able to measure the change of human motion. That is to say, if the human body moves violently such as fast walking or running, the C-Motion method will return a high value. Moreover, the Standard Deviation of C-Motion can be used to indicate the rate of change of human motion, thus it can distinguish human falling behavior with high acceleration. Generally, the author used an IP camera to record indoor images, applied foreground segmentation to locate the position of the human body in the frame, and calculated the standard deviation of the C-Motion value and the orientation standard deviation of the ellipse to obtain the rate of change of motion and shape of the human body. Although MHI is easy to compute, when the screen is converted from 3D to 2D, it is prone to loss of information due to motion self-occlusion and motion over-writing, which leads to large errors in recognition. At the same time, MHI is dependent on the temporal duration value and is not suitable for processing variable-length action sequences.

Timed Motion History Image (tMHI) is a time-coded blur point format, which can be used for motion segmentation and cover the same MHI area at different capture rates. Suad et al.[22] proposed a human fall detection model based on motion information, projected histograms and shape direction changes. After segmenting the motion detection target by background subtraction,the author used tMHI to analyze whether the motion object has a large motion change in a given time, analyzed the direction change of the fitted ellipse to determine the morphological change of the human body, and finally detected the change of the projection histogram feature Circumstance, thereby judging whether the human body falls. In this process, if the standard deviation of the difference between horizontal and vertical histograms is higher than the set threshold, and large motion information is detected, it can be determined that the human has fallen.

After background subtraction, the ellipse is used to fit the contour of the human body to obtain the shape change of the human body. After a fall, the ratio of the length of the long and short axis of the ellipse and the direction angle of the ellipse will change, which can be used as a judgment factor[23], as shown in the Figure 6. Furthermore, by calculating the horizontal and vertical projection histograms of the foreground row by row and column by column[24], the horizontal and vertical projection histograms can be obtained and the maximum value of the two projection histograms can be found. By calculating the standard deviation between the maximum values, the behavior of falling can be distinguished from other states of the human body.

Researchers analyzing the physical of falling that there are some parameters like speed, acceleration or the head and leg movement, were well correlated with falling. People can use machine learning to identify falls by extracting these parameters to enhance the semantic-based feature of the image.

Fig. 7. Ellipse Feature[23]



3. METHODOLOGY

The main goal for this project is to help detect and report incident that happened to old people for medical attention as soon as possible. The traditional solution to this problem mainly depends on the small devices that have a gyroscope sensor for detecting fall down incident by observing the abnormal acceleration data. It is not only inconvenient for the old person to wear the device 24-7 a week, the detection accuracy also varies a lot depending on the relative location of the device and the person's daily activities.

Fortunately, due to the development of chip manufactures, the cost of small IoT devices are getting significantly low, letting its application in various field become feasible. Here in this project, we explored the possibility of setting up mini-camera in the surrounding area for a easier usage, simpler deployment and more accurate fall detection system for helping the elderly people.

We proposed two system architecture and two variation models to the fall detection task, targeting to be compatible and possible to implement with ESP32-cam. These devices cost under 9 USD each, cheap enough to be affordable to general public.

3.1. System Architecture

Our two system architecture solutions are the distributed version and the centralized one. The first one mainly finishes all the analysis process in the local camera, where as the later one processes the final analysis at a centralized machine.

3.1.1. Distributed Analysis

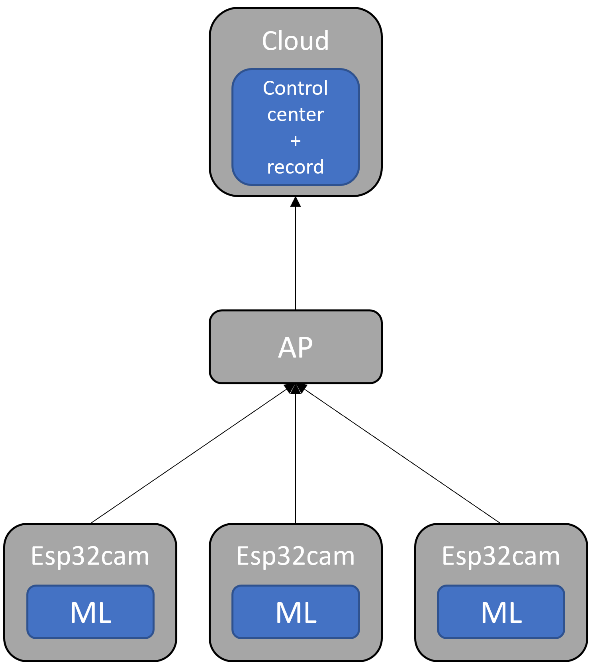


Fig. 8. Distributed approach

This is a straight forward approach, every IoT device is doing its work individually. Each of them will analysis the real-time footage via a trained ML model, report and upload the footage to the cloud whenever any incident is detected. After information is uploaded to the cloud, pipelines for emergency service and system for noticing relatives can be activated for in-time support.

This will require real-time analysis computational power in each and every sensors, which in this case ESP32 is fortunately able to run some trained simple machine learning model or else fps might need to be decreased, reducing the accuracy of detecting the moment of the incident. But in general, this is a simple architecture and allows user to easily setup or even expand the system.

The downside of this approach is that when more tasks are being pursued, highly complicated or even multiple models might be needed for the analysis process, resulting in way higher computational requirement and increasing the cost by a large margin. Also, footage other than the critical moments might not be able to be stored or accessed, since uploading all the footage to the cloud seems to be a bit hideous both for security and network traffic reasons.

3.1.2. Centralized Analysis

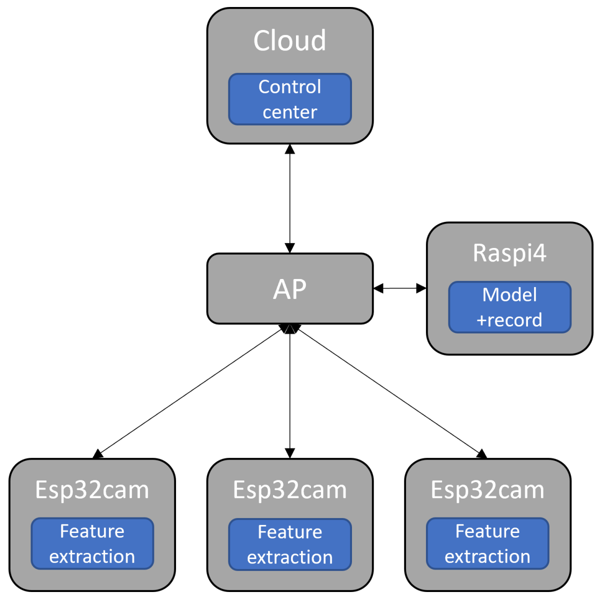


Fig. 9. Centralized approach

This approach added an intermediate level machine in between the sensors and the cloud, sharing part of the analysis task from the lowest level IoT devices. So the low level sensor only have to do feature extraction whereas the actual analysis is done in the centralized machine. For example(Fig. 6), the Raspberry Pi 4(Raspi4) is responsible for analysing all the extracted data from each sensors and determine if any fall accident occurred.

Do notice that the network connection in this architecture is bi-directional, not like the distributed approach(Fig. 5). Sensors are not only responsible for feature extraction, but also storing recent footage, letting the intermediate machine be able to request for the corresponding video or even do further analysis. The reason not to transfer all the footage to intermediate machine directly is to ease the network traffic in the local network, as if that is the case, all sensors will be sending streaming data to that machine, which will case network bandwidth problem and also require significantly higher computational power to analyse raw data.

This system design have way cheaper computational power need for the sensors and more appropriate task management ability, allowing the possibility of extending the tasks far beyond just fall detection. It may even optimize the computational resources one step further comparing to the previous approach by only analysing the extracted features from camera that have possibility in locating a human. For example, if there is one sensor in living room, hallway and in bedroom, it will be impossible for the camera in living room to detect a fall incident if the camera in the bedroom detected the only person. It do have an obvious drawback, which is that the system is not straightforward to setup, and might need extra support to maintain the system.

3.2. Analysis Model

The analysis model consist of two part, the feature extraction part and the analysis part. The main objective for feature extraction is to extract the critical information from each frame, for example the location and position of the person in the picture. And for analysis, it is mainly looking into the correlation between frames or aka time variant. In the following part, we introduce two variant of feature extraction methods.

3.2.1. CNN

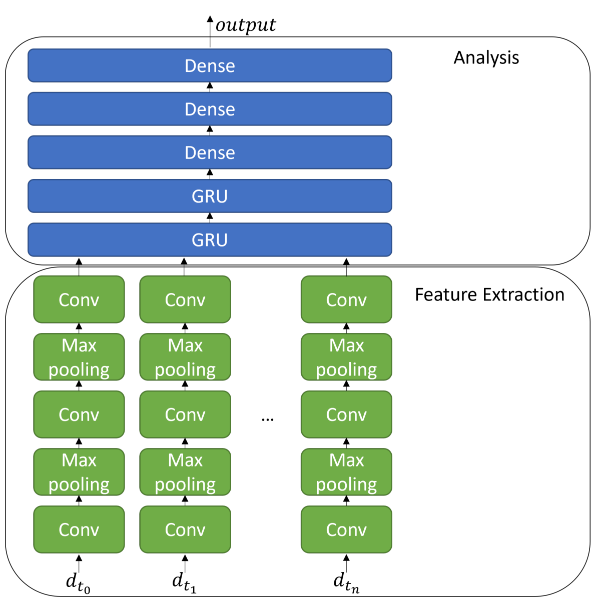


Fig. 10. CNN+GRU+MLP

As shown in Fig. 7, the feature extraction part is composed of convolution and max pooling layers, allowing the model to recognize what pattern within the scene is the most critical information. Theoretically, it will automatically adjust to focus on the details of the human's properties during the training process. Drop out layer can be applied after the convolution layers for training a more sparse representative of the extracted data. As sparse data is compressible, this will further optimize the whole system if using the centralized architecture.

3.2.2. Optical-flow

Other than using the fully automated machine learning model for feature extraction, it is also reliable to use the Gunnar-Farneback optical flow algorithm for the feature extraction part. Since the goal is to detect if any falling event is happening, we are most interested in the moving object, which can be indicated and labeled by this method. Also, this is especially useful if the environment is mostly static, which is very common in most of the homes. Since not much object is moving around in normal case, it will be sparse data, and compression before transmitting the data to the intermediate machine for analysis will be possible.(if using the centralized architecture)

4. EVALUATION AND EXPERIMENTS

We implemented two variation models, one with only CNN for feature extraction and the other one is optical-flow combined with CNN. Both variation uses the same analysis model(GRU(128)x2+MLP(64/16/1)).

4.1. IMVIA Fall Dataset

One of the biggest challenge is to collect enough data for training purposes, for this project, we decided to use the fall dataset generated by the ImViA laboratoire in University of Burgundy. It includes 191 labeled 320x240 videos in 25fps which took place in 4 different scenes. We know that it is not a large dataset, but considering the time and computational resources we have, this size is appropriate.

4.2. Data preparation and training model

First, to enlarge the sample for training the model, we slice each videos into sections of length equal 1 second(25 frames), for example, generating 200 samples from a 9 second video. Via this, we have generated over 200 GB of data to train.

After the data preparation process, we insert the data into the first variation of the model, similar to the structure shown in Fig. 10. At last we modified and added another layer of max pooling and convolution between the feature extraction and analysis section. With $d_{t_0}, d_{t_1}, \ldots, d_{t_25}$ corresponding to frames within each 1 second slice, we trained the model aiming to determine if the falling incident is happening within the corresponding time frame.

Similarly, we preprocessed the data by applying the Gunnar-Farneback algorithm optical-flow algorithm and feed the processed data into the second variation model, which the details can be found in Fig. 11.

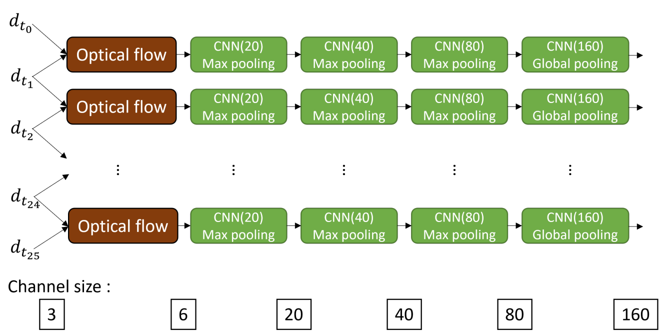


Fig. 11. feature extraction of the second model

4.3. Result

For evaluation, we picked the falling videos that were recorded in a totally different scene, one that is not included in the training data.

The result of the first model is acceptable, which we suppose it can be improved by running more epochs, but due to time limitation, this is the result of training after 2 hours. Even this version is be able to apply to real world scenario with appropriate threshold to determine the falling incident. For example, according to the result in Fig. 12, the threshold can be set to 0.326 for fall detection.

And the result of the second model is satisfying, which we confirmed that it nearly perfectly mapped out when the falling incident is happening.

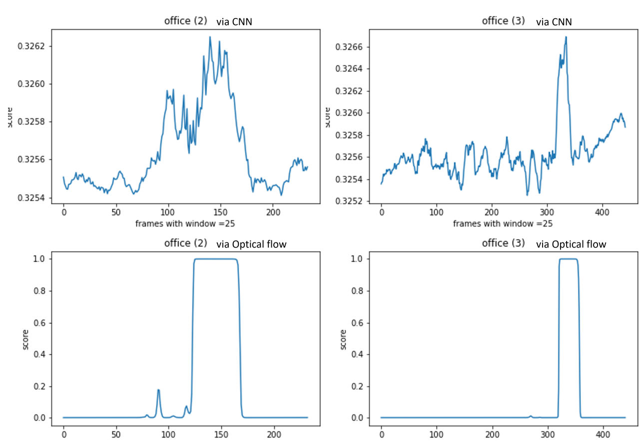


Fig. 12. Comparison of two models with two examples

5. CONCLUSION

During the high demand of fall detection in a private home for elderly people, we want to build a model which could detect based on low-quality videos and could be implemented on mini IoT devices in the next step. The specification of the IoT device needs to be cheap and low consumption in energy and computational wise, so it is a challenging task which no existing solution can perfectly solve this problem. This project is aiming to purpose a solution with possible system architecture with compatible optimized ML models, letting low-cost smart health IoT device for fall detection is achievable and available in the near future.

6. REFERENCES

[1] Nuttapong Worrakulpanit and P. Samanpiboon, "Human fall detection using standard deviation of c-motion method," 2014.

[2] Centers for Disease Control and Prevention. Important Facts about Falls., ," .

[3] Medrano C. Plaza I. Igual, R., "Challenges, issues and trends in fall detection systems," 2013.

[4] Umar Asif, Benjamin Mashford, Stefan Von Cavallar, Shivanthan Yohanandan, Subhrajit Roy, Jianbin Tang, and Stefan Harrer, "Privacy preserving human fall detection using video data," in *Machine Learning for Health Workshop*. PMLR, 2020, pp. 39–51.

[5] Francy Shu and Jeff Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Scientific reports*, vol. 11, no. 1, pp. 2471–2471, 2021.

[6] Gangireddy Prabhakar Reddy and M Kalaiselvi Geetha, "Video based fall detection using deep convolutional neural network," *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 2, pp. 5542–5551, 2020.

[7] Bin Zhao, Xuelong Li, and Xiaoqiang Lu, "Cam-rnn: Co-attention model based rnn for video captioning," *IEEE transactions on image processing*, vol. 28, no. 11, pp. 5552–5565, 2019.

[8] Han Zhao and Xinyu Jin, "Human action recognition based on improved fusion attention cnn and rnn," in *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, 2020, pp. 108–112.

[9] N. Jayashree, B. Pavithra, S. Monika, R. Charanya, and M. Priya, "Arduino based iot device for septuagenarian fall detection," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–4.

[10] Tran Tri Dang, Hai Truong, and Tran Khanh Dang, "Automatic fall detection using smartphone acceleration sensor," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 12, 2016.

[11] Rosaria Rucco, Antonietta Sorriso, Marianna Liparoti, Giampaolo Ferraioli, Pierpaolo Sorrentino, Michele Ambrosanio, and Fabio Baselice, "Type and location of wearable sensors for monitoring falls during static and dynamic tasks in healthy elderly: a review," *Sensors*, vol. 18, no. 5, pp. 1613, 2018.

[12] Kun-Hui Chen, Po-Chao Chen, Kai-Chun Liu, and Chia-Tai Chan, "Wearable sensor-based rehabilitation exercise assessment for knee osteoarthritis," *Sensors*, vol. 15, no. 2, pp. 4193–4211, 2015.

[13] Young-Min Lee, Hongjin Yeh, Ki-Hyung Kim, and Okkyung Choi, "A real-time fall detection system based on the acceleration sensor of smartphone," *International Journal of Engineering Business Management*, vol. 10, 2018.

[14] Yun Li, KC Ho, and Mihail Popescu, "A microphone array system for automatic fall detection," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 5, pp. 1291–1301, 2012.

[15] Guodong Feng, Jiechao Mai, Zhen Ban, Xuemei Guo, and Guoli Wang, "Floor pressure imaging for fall detection with fiber-optic sensors," *IEEE Pervasive Computing*, vol. 15, no. 2, pp. 40–47, 2016.

[16] Wei-Han Chen and Hsi-Pin Ma, "A fall detection system based on infrared array sensors with tracking capability for the elderly at home," in *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*. IEEE, 2015, pp. 428–434.

[17] Danielsen Asbjørn and Torresen Jim, "Recognizing bedside events using thermal and ultrasonic readings," *Sensors*, vol. 17, no. 6, pp. 1342, 2017.

[18] Edouard Auvinet, Franck Multon, Alain Saint-Arnaud, Jacqueline Rousseau, and Jean Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution," *IEEE transactions on information technology in biomedicine*, vol. 15, no. 2, pp. 290–300, 2010.

[19] Derek Anderson, Robert H Luke, James M Keller, Marjorie Skubic, Marilyn Rantz, and Myra Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Computer vision and image understanding*, vol. 113, no. 1, pp. 80–89, 2009.

[20] Koldo De Miguel, Alberto Brunete, Miguel Hernando, and Ernesto Gambao, "Home camera-based fall detection system for the elderly," *Sensors*, vol. 17, no. 12, pp. 2864, 2017.

[21] Lei Yang, Yanyun Ren, and Wenqiang Zhang, "3d depth image analysis for indoor fall detection of elderly people," *Digital Communications and Networks*, vol. 2, no. 1, pp. 24–34, 2016.

[22] Suad Albawendi, Ahmad Lotfi, Heather Powell, and Kofi Appiah, "Video based fall detection using features of motion, shape and histogram," in *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, New York, NY, USA, 2018, PETRA '18, p. 529–536, Association for Computing Machinery.

[23] Kishnaprasad G. Gunale and Prachi Mukherji, "Fall detection using k-nearest neighbor classification for patient monitoring," in *2015 International Conference on Information Processing (ICIP)*, 2015, pp. 520–524.

[24] Homa Foroughi, Alireza Rezvanian, and Amirhossien Paziraee, "Robust fall detection using human shape and multi-class support vector machine," in *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, 2008, pp. 413–420.