ABSTRACT

THESIS PAPER: Real-Time Human Activity Recognition Based on Radar

STUDENT: Hanqing Guo

DEGREE: Master of Science

COLLEGE: Sciences and Humanities

DATE: May 2019

PAGES: 48

Most smart systems such as smart home and smart health respond to human’s locations and activities. However, traditional solutions either require wearable sensors or lead to leaking privacy. This work proposes a deep learning based ambient radar solution that is a real-time, privacy-preservative, and lightness resistant system. In this solution, we use a low-power, Frequency-Modulated Continuous Wave (FMCW) radar array to capture the reflected signals and then construct 3D image frames from the wireless radar signals. Then deep learning is applied to model and recognize various human activities from the wireless radar signals. This solution includes: 1) a data preprocessing mechanism to remove static background reflection, 2) a signal processing mechanism to transform received complex radar signals to a matrix containing spatial information, 3) a deep learning scheme to filter broken frames which are caused by the rough surface of human body, and 4) a deep recurrent neural network system to recognize human activities based on radar imaging results. This solution has been extensively evaluated in a research area for real-time human activity imaging that is recognizable for various activities. The results show that the solution is able to generate wireless imaging frame-by-frame compared to camera recorded video, and it can achieve over 86.7% accuracy in recognizing six different types of human activities based on the wireless radar imaging.
1 INTRODUCTION

With the development of Internet of Things and technology on smart home, human activity recognition becomes a hot research topic since it can be applied to many smart home applications, from human interaction to health care system. For example, smart devices need real-time human activity prediction and response to proper action plan; Robots is demanded to able to sense human activities in dark surroundings; Smart health system needs alert family members when it recognizes school-age children and elderly people fall in the bathroom alone. However, most of the proposed human activity recognition solutions are based on cameras, which makes users concerned privacy leaking problem [1, 2]. Hence, developing a human activity recognition system without computer vision techniques has been a very popular research task.

While traditional camera-based solutions result in privacy issues, researchers use wearable sensors to collect and process human motion data in many smart home scenarios. However, those wearable devices are inconvenient to users because users are required to remember to wear these sensors when wakeup, and to take it off when typing, washing hands or strenuous exercise [3]. Admittedly, the critical capability required in human activity recognition system is working consistently and wearable sensors cannot meet this requirement because the capture process would fail when users take off those wearable sensors, which means the results are unreliable during that time. Hence then, it is highly demanded to design a passive, non-invasive and real-time recognition system so that privacy protection and convenience can be guaranteed. Recent researches in this area apply Radio Frequency (RF) technology to smart systems. The benefits of using RF devices can be represented in two scenarios: 1) RF devices are able to “see” human in dark light condition; 2) RF signal is able to sense human activity through obstruction with proper signal processing and enough transmit power.

In this thesis, we propose a real-time human activity recognition system including two main functions: wirelessly imaging users’ activities and recognizing their activities. For human image capturing part, we propose a scheme of Human Image Capturing based on
FMCW Radar sensors, HICFR. This work uses a FMCW radar to sense the environment, then convert raw signals to 3D human images which contain spacial location information of the target in real-time. It has the following highlights: **First**, it uses antenna array combined with FMCW radar to transmit and receive directional beamforming to sense 3D environment, the transmit signal frequency change is between 3.3GHz to 10GHz, so it is able to detect precise distance change based on frequency difference. **Second**, it has data clean feature. The proposed calibration algorithm can record static environment response, then remove them from raw signals, thus only useful human motion responses can be reserved, which makes visualization results more clear. **Third**, it uses deep learning algorithm to process raw 3D images. It uses Convolutional Neural Network (CNN) to recognize whether the current frame is caused by irregular reflections, if true, filter them out so that the real-time captured frame is continuous. **Forth**, it is a very low power system, with average transmit power is below -40dbm/MHz, which means it has a minor effect on human body.

For recognizing part, we propose a scheme of Human Activity Recognition based on Radar Image and Recurrent Neural Network, HARNN. This work recognizes human activity in real-time and classifies them into six categories: *fall, stand-up, sit-to-stand, stand-to-sit, walk* and *still*. The contribution of this work includes: 1) It uses a unique dataset which comprised by radar images captured from capturing part. Compared with videos recorded by a camera, our solution uses RF to protect privacy. Compared with the signal processing approach which recognizes human activity only depends on signals transform, our design keeps all signal information and visualize it to a 3D video; 2) It uses a sequence of neural network (CNN) to extract features from captured images. Due to we are using images rather than signals to record human distance and position, the CNN can be easily applied to our data and it also saves computing resource compared with signal as data, which occurs to large fully connected layer and a massive number of neural; 3) It applys recurrent neural network (RNN) model to process continuous images. Due to those continuous images captured by radar are temporal series, the change of every frame has a potential relationship of activities;
4) It develops a video loader in the training process, and this video loader makes every video sample can be accessed with random sequential in every iteration, which contributes to faster convergence.

In the rest of this thesis, Section 2 reviews the literature solutions related to our work. Next, Section 3 describes the overview of system structure, includes system platform introduction, signal processing chains, surrounding reflection signals removal, and a brief introduction of human activity recognition. Then, Section 4 proposes the technical details of visualization system, it describes how to combine FMCW radar with an antenna array to collect 3D array which represents the power at specific spacial voxels, and how to utilize deep learning to remove bad reflection frames. Section 5 presents recognition algorithm that uses RNN to recognize human activity at each frame in real-time, followed by Section 6 evaluates the performance and feasibility of the whole system. Then Section 7 concludes this work and discuss possible changes which may achieve higher detection accuracy in the future.

2 RELATED WORK

Related works in this field include traditional camera based solutions, more recent RGB-D camera based solutions, sensor based solutions and RF based solutions. In this section, all of those solutions will be investigated and discussed in the smart home scenario.

2.1 Camera Based Solutions

Traditional camera based solutions always associate with computer vision techniques such as object detection, video classification, and human pose detection. All of the techniques mentioned above are based on cameras to collect the image or video data, so they are classified into camera based solutions. Since computer vision is a very broad field, we will describe those new approaches in detail as following subsection.
2.1.1 Real-time Object Detection

Previous object detection approaches have performed remarkable results on multiple large open-source datasets such as PASCAL VOC, Open Images, ILSVRC and COCO [4, 5, 6, 7]. Compared to traditional image classification, object detection task is more challenging because it should not only classify objects in the current frame but also localize all objects. Also, solutions need guarantee satisfied detecting speed and accuracy. The first paper which applies convolutional neural network (CNN) to object detection field published in 2014 [8]. The research group proposes a Region with CNN features (R-CNN) method which selects about 2K possible regions for one image and computes CNN features for every selected region, hence classifying regions using class-specific linear SVMs. Even though R-CNN achieves state-of-art accuracy at that time, the shortcoming of R-CNN is obviously exposed because it wastes time to propose regions and compute classification score repeatedly in every image. To improve R-CNN performance, researchers of Microsoft develop a more efficient framework named Spatial Pyramid Pooling (SPP) one year later [9]. They create several innovations based on R-CNN to accelerate the training and detecting speed. Kaiming He et al. propose SPP-net to solve fixed-size input problem, the SPP-net also optimize feature extraction process in R-CNN, different from computing every proposed region repeatedly in R-CNN, SPP-net extract feature map from the entire image only once and pool features in arbitrary regions to generate fixed-length representation for detector training. The SPP solution is 24-102× faster than R-CNN method in VOC 2007 dataset. The author of original R-CNN also propose a Fast R-CNN framework in 2015 [10], which integrates region proposal phase, CNN extracts feature phase and classifier phase to an end-to-end network, they get 213× faster at test-time and achieve higher mAP(mean Average Precision) on VOC 2012 dataset. They continue their effort on Fast R-CNN work and develop a Faster R-CNN solution, in which they introduce RPNs (Region Proposal Networks) and anchors to accelerate region proposal process, the faster R-CNN system has a frame rate of 5fps on a GPU and achieve state-of-art object detection accuracy on VOC 2007 (73.2% mAP) and 2012 (70.4% mAP).
More recently, researchers in the University of Washington propose a You Only Look Once (YOLO) scheme, which solves object detection problem with a simpler and higher efficient way compares to CNN’s family’s solution [12]. Instead of proposing possible regions in CNN’s solution, YOLO defines anchors and grid to process the entire input image and result with 155fps and 78.6% mAP. Moreover, Redmon also improves his YOLO performance on detection task at small objects, the new models YOLOv2 [13] and YOLOv3 [14] achieves an excellent tradeoff between speed and accuracy, it runs significantly faster than other detection methods with comparable performance.

Figure 1: Implementation Demo of YOLOv3
2.1.2 Action Recognition

Action recognition and video classification is another track to achieve human activity recognition. Compared to object detection technology, action recognition focuses more on action detection rather than object localization and classification. During the past five years, more and more models and schemes are developed to recognize action or activity in video source based on deep convolutional networks and recurrent networks [15, 16, 17, 18, 19, 20]. Before 2014, while the deep learning has not fully developed into video recognition, the most advanced human action recognition architecture proposed from a France Research Institute [21, 22] named dense trajectories (DT) and improved dense trajectories (iDT). They extract feature trajectories in the video frame by frame and generate dense trajectories. Then they design descriptors to encode the trajectory information based on human motion and achieve a state-of-art action classification accuracy. With the extending application of deep learning, a two-stream (spatial stream and temporal stream) convolutional networks is proposed to improve action detection accuracy [15]. They depose the video into spatial and temporal components, and design two deep convolutional networks to process a single frame and associated multi-frame optical flow. Then give action recognition confidence score on every frame. The two-stream solution result in competitive accuracy compares to the state-of-art result on benchmarks experiments. A year later, researchers from China extend former DT and iDT work to deep convolutional techniques which named trajectory-Pooled Deep-Convolutional Descriptors (TDDs) [17]. Instead of only using hand-crafted trajectory features, they aggregate deep-learned features as well as trajectory features to obtain effective descriptors. Compared to hand-crafted features, TDDs is able to learn high discriminative features, this advantage help TDD outperform superior result on HMDB51 [23] and UCF101 [24] database. Inspired by former two-stream and TDD solution, researchers at University of Maryland Joe et al. propose a method with two innovations 1) aggregate features of video clips and optical flow; 2) design feature pooling architecture and long-short term memory (LSTM) network to predict motion in video [18]. Their best performance
on UCF-101 datasets achieves 88.6% accuracy. Meanwhile, LSTM and recurrent convolutional networks are widely investigated in video description task, a research group of UT Austin propose a Long-term Recurrent Convolutional Network (LRCNs) to solve activity recognition, image description and video description problem [20]. Their design takes advantage of the flexibility and temporal-related property of the recurrent network, and achieve 28.8% accuracy on TACoS [25] dataset, which has outperforming the best-reported number of 26.9% by [18]. Besides recurrent convolutional networks idea, researchers from Facebook AI group also investigate using 3D convolutional network to learn spatiotemporal features, which named Convolutional 3D (C3D) [19]. Compares to 2D convolutional network, 3D convolutional network takes all frames concatenated as an entire input, thus it has the ability to model temporal information better. Their result shows C3D outperforms all previous best-reported method before 2015 on different datasets. More recently, action detection task has been extended to detect actions in untrimmed videos, which means the model should be able to understand where is the start and end timestamp. In 2017, a research group of The Chinese University of Hong Kong proposes a structured segment network (SSN) to detect each action instance via a structured temporal pyramid, and their solution works perfectly on untrimmed videos [16]. Basically, they design two classifiers on the top of the pyramid, one is work for action detection, the other one detects the completeness of the current action. This novel design allows the framework not only recognize actions but also localize the start and end frames of the action. Their work outperforms previous methods on two challenging dataset THUMOS14 [26] and ActivityNet [27].

2.1.3 Pose Estimation

Besides human localization, activity recognition, human 2D pose estimation is also a computer vision task which contributes to visualize human skeleton, hence capture human actions by skeleton changes. The common approach design person detector and perform single-person pose estimation for each detection[28]. Since then, many innovative techniques emerge
to leverage existing deep learning techniques for single-person pose estimation\cite{29, 30, 31} and achieve state of art performance. However, those solutions only fit the single-person scenario, as for multi-person scenario, researchers from CMU present an efficient method for multiperson pose estimation with state-of-the-art accuracy on multiple public benchmarks. Compare with the existing top-down approach which detect person first and then draw a human skeleton, their bottom-up solution overcome problems such as person detector fails, and runtime issues. They use CNN to extract part confidence maps and part affinity fields, since the detect multiple parts through a CNN, so that runtime does not increase dramatically compare to single-person detect task. Then they introduce bipartite matching to parse part association information and combine those confident part together to draw a skeleton of one person. Further research on human pose estimation focuses on 3D human pose

Figure 2: Implementation Demo of 2D Pose - Stage1

Figure 3: Implementation Demo of 2D Pose - Stage2
estimation[32]. Instead of building a human 2D skeleton in previous work, researchers of the fair lab present a Dense pose estimation which aims to map all human pixels of an RGB image to the 3D surface of the human body. To achieve 3D pose estimation, they introduce a DensePose-COCO dataset, which is a large-scale ground-truth dataset containing manually annotated image-to-surface correspondences for 50K images.

Figure 4: Implementation Demo of 3D Pose - Original

Figure 5: Implementation Demo of 3D Pose - Result
2.2 RGB-D Based Solutions

Compared to the traditional camera, RGB-D (Red, Green, Blue, and Depth) camera can capture both RGB video as well as depth information, which has significantly helpful in human activity recognition research [33]. Based on RGB-D images resource, many algorithms have been proposed [34, 35, 36, 37, 38] to solve human activity recognition task. Sung et al. used RGB-D images to detect and track human motions, those images with depth information were properly processed to achieve capture purpose [34, 37]. Jalal’s team proposed a solution, which uses translation and scaling invariant features with depth videos to recognize human logging activity [35]. More recently, Kinect was widely used to collect human motion data because of its abundant APIs; researchers trained machine models to do image segmentation for Kinect real-time video and capture coarse human outlines and motions [36, 38].

2.3 Sensor Based Solutions

Sensor based solution can be divided into two categories: wearable sensors and off-the-shelf sensors. The former one usually uses wearable bands, watch, or even cell phone to collect human motion data. And the latter solution always considers off-the-shelf sensors such as ultrasonic devices, infrared sensor and beamforming devices.

2.3.1 Wearable Sensor Based Solutions

Accelerometer and Gyroscope [39] are the most common sensors applied to collect human motion data. Because most cell phones are embedded with accelerometer and gyroscope sensor, there is an increasing number of applications recognize human activity based on cell phone [40, 41, 42]. Besides that, Inertial measurement unit (IMU) sensor which combines accelerometer and gyroscope sensor is also widely used to wearable devices [43]. Those sensors can collect linear acceleration, rotation angle, the angular velocity of targets so that human who wears sensors can be collected motion data. Based on raw data collected by those sensors, researchers proposed various algorithms to recognize human activities [44, 45, 46, 47].
Zhang et al. designed physical features based on physical parameters of human motion, then find the most critical physical features for human activities, thus to improve the recognition accuracy [44, 45]. Years later, Ferhat et al. investigated k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Gaussian Mixture Models (GMM) classification techniques to process raw data and found the best scheme to recognize human activities [46]. Recently, researchers from the UK explored how to use deep, convolutional and recurrent models to detect human activities [47]. Researchers in China Jun et al. proposed a novel probabilistic algorithm to learn compact features provided by raw data from both camera-based and wearable sensor-based system. Their experiments on three public datasets achieve competitive performance in terms of accuracy as well as efficiency [48].

2.4 RF Based Solutions

Radar sensors and RF devices usually used for military or wireless communication purpose. However, it has been recently considered for smart home applications because of its data confidentiality, and the performance does not depend on lighting conditions. Recent researches either based on FMCW radar [49, 50, 51, 52] or used off-the-shelf devices [53, 54, 55]. A research group of Massachusetts Institute of Technology (MIT) Adib et al. designed MIMO antenna sensor with FMCW technique to detect human move [49] and even capture human figure through a wall [51, 50]. Off-the-shelf devices such as ultrasonic sensor or walabot [56] were also investigated feasibility to recognize human activities [53, 54, 55]. Avrahami et al. proposed a human activity recognition scheme based on 2D heat maps generated by walabot, while Zhu et al. [54] applied traditional signal processing algorithms to filter and cluster raw data thus recognize human actions. Both of them achieve higher than 80% accuracy in their research tasks.
3 SYSTEM DESIGN OVERVIEW

3.1 Radar Image Capture System Overview

As we mentioned before, our system includes two main functions: capturing users images and recognizing their activities. To enable real-time human image capturing with ambient radar in smart home scenario, we propose a solution: Human Image Capturing based on FMCW Radar sensors (HICFR) system. HICFR scans 3D surroundings reflections with FMCW chirps and 2D antenna array. While FMCW chirps are used to compute the direct distance from the detected object to receive antenna, and the 2D antenna array is placed to identify spatial directions. It emits FMCW chirps to scan 3D volume of surroundings, then received signals are processed to remove environment fix reflections,. After that, we calculate the reflection powers of any scanned voxel and construct it to 3D images, then a Deep Neural Network (DNN) based filter algorithm is designed to address noncontinuous reflection frames problem, thus capture real-time human activities.

3.2 Human Activity Recognition System Overview

As for recognizing part, we propose a Human Activity Recognition based on Radar Image and Recurrent Neural Network (HARRIRN) system. This work recognizes human activity in real-time and classifies them into seven categories: fall, stand up still, walk, sit-to-stand, stand-to-sit and jump. HARRIRN takes continuous radar images and videos as system input. For every input frame of a video, we pass the frame to our CNN to extract position features. Then the extracted features of each frame are considered as input for RNN and LSTM cell because those features are extracted from the continuous frame, thus they contain temporal information of the activity duration, and RNN is suitable for processing temporal sequence. In this way, the output of the last RNN cell gives confidence score of specific activities, we calculate the difference or loss between predict confidence with ground truth label, then back-forward the loss and update RNN cell parameters to optimize our neural
network model. The contribution of this work includes: 1) It uses a unique dataset which comprised by radar image captured from capturing part. Compared with videos recorded by a camera, our solution uses RF to protect privacy. Compared with the signal processing approach which recognize human activity only depends on signals transform, our design keeps all signal information and visualize it to a 3D video. 2) It uses a sequence of neural network (CNN) to extract features from captured images. Due to we are using images rather than signals to record human distance and position, the CNN can be easily applied to our data and it also saves computing resource compared with the solution uses signal sample as training data, which occurs to large fully connected layer and a massive number of neural. 3) It applies recurrent neural network (RNN) and long short term memory (LSTM) algorithm to process continuous images because those continuous images captured by radar is temporal series, the change of every frame has a potential relationship of activities. 4) It develops a video loader in the training process, and this video loader makes every video sample can be accessed with random sequential in every iteration, which contributes to faster convergence.

3.3 Sensing Platform

Since HICFR requires to emit FMCW chirps and collect received signals by 2D antenna array, there is an off-the-shelf radar sensor called Walabot [56] meets our requirement. Walabot has compact size and low-cost feature with a board size of $72mm \times 140mm$ and the average power is lower than -41dbm/MHz. The frequency range of FMCW chirp emitted by Walabot is 3.3GHz-10GHz, which is good enough to detect direct distance within 10 meters range based on the gradient of FMCW chirp. It also contains 18 pair of antenna, which are arranged to 2D antenna array.

Figure 6 shows internal antenna array of Walabot. Walabot emits FMCW chirps to scan $\phi$ in horizontal direction and $\theta$ in vertical direction. Then it communicates to HICFR with USB port to send raw signals for further processing. The scanned area of walabot can be present as figure 7 below:
Where $\theta$ is the elevation angle to detect the height of human, and $\phi$ is Wide angle to capture the width of the human. $R$ is FMCW signals travel distance from transmit antenna to humans head, and $R'$ is the hypotenuse of a triangle whose angle is $\theta$ and hypotenuse $R$ rotate $\phi$ degree, the scan range is the sector which triangle passed. In our case, $\theta$ is from $-45^\circ$ to $45^\circ$ and $\phi$ is from $-90^\circ$ to $90^\circ$. The direct travel distance $R$ can be calculated by FMCW properly with formula (1) and figure 8 as below:

$$R = \frac{c|\Delta t|}{2} = \frac{c|\Delta f|}{2(df/dt)}$$  \hspace{1cm} (1)
While $\Delta t$ is signal travel time from transmit antenna to object and reflect back to receive antenna, and $\Delta f$ is frequency difference of transmit and receive signals. $df/dt$ is the slope of transmit or echo frequency chirp and $c$ is the speed of light. To simplify description, Equation (1) and formula 8 are not considering Doppler frequency shift effect.

### 3.4 Complete System Overview

The complete activity recognition system contains 6 phases. 1) Data collection and Calibration; 2) Coarse Visualization; 3) Fine Visualization; 4) Video dataset construction; 5) Train Activity recognition network and 6) make it to real-time application.

As shown in figure 9, environment sensing period is running ahead of detecting. First, $HICFR$ emits FMCW chirps and records static background reflections, when $HICFR$ starts to do object detection task, its 2D antenna array collects raw signals. Second phase is designed to convert signals to images. Since $HICFR$ scans 3D surroundings with parameters $R, \theta$ and $\phi$, the raw received signals then being processed to represent power of every spatial points with different $R, \theta$ and $\phi$, namely voxel in scanned area. Then $HICFR$ subtracts recorded background reflections power, and removes environment fix reflection to get pure power information of changed objects, which is a 3D matrix $M$ with dimension of...
(sizeX, sizeY, sizeZ), where sizeX, sizeY, sizeZ can be computed by equations (2).

\[ sizeX = \frac{\text{range}(R)}{\text{res}(R)} \]
\[ sizeY = \frac{\text{range}(\theta)}{\text{res}(\theta)} \]
\[ sizeZ = \frac{\text{range}(\phi)}{\text{res}(\phi)} \]  

In the equations above, \( \text{range}(\cdot) \) is detect range of parameters, while \( \text{res}(\cdot) \) is designated parameters’ sampling interval. Next, we propose a novel solution to achieve coarse-to-fine visualization. Because human body acts as an uneven reflector rather than a scatterer, thus some signals reflect back directly to the antenna array, while other signals are deflected from other path or even away from receive antenna, in this case, constructed 3D images may contain some ambiguous results. The third phase addresses this issue by using Machine Learning algorithm. We collect dataset for regular reflection and deflected reflection images, then train a Deep Neural Network (DNN) which contains Convolutional Layer, Pooling layer and Linear layer to recognize them. The trained DNN is placed to the main program loop thus eliminates ambiguous frames from the real-time stream.

In further processing, we introduce our Human Activity Recognition based on Radar Image and Recurrent Neural Network (HARNN) system. HARNN takes real-time streams to two sub-paths: One is going to video dataset construction phase, and the other connects to the real-time application phase directly. The stream is considered as a real-time test dataset to achieve real-time activity recognition. For the video dataset construction phase, we collect the radar image videos when we walk, run, jump, sit and stand, fall and stand up. Those videos are split based on activities and classified into several categories with different activity labels, which means each video has an activity label, and frames in the video contain temporal information of an activity. Since many frames represent one activity, and the input frames have a temporal relationship, we choose the Recurrent Neural Network (RNN) to learn and predict human behaviors based on radar videos. Finally, given a continuous radar video and a trained RNN, our HARNN system can recognize human activity in real-time.
Figure 9: Flow Graph of HICFR
As shown above, the key challenges and main contributions of HICFR and HARNN are:
1) Compute reflection powers of every voxel with $R, \theta$ and $\phi$ based on the received complex signals of antenna array, 2) Construct 2D/3D images with known 3D power matrix, 3) Address ambiguous images issue which caused by signal deflection with Deep Neural Network and achieve real-time filtering scheme, 4) Build a labeled radar video dataset, each video represent a human activity, 5) Train a RNN to learn temporal information of radar video frames and 6) Integrate RNN and real-time stream together to achieve real-time activity recognition.

4 VISUALIZATION SYSTEM DESIGN

In this section, we dive in the technical detail of HICFR. Since walabot antenna array collects RF-signals, which is complex signals, they can be represented by amplitude and phase as follows:

$$s_t = A_t e^{-j2\pi \frac{r}{\lambda} t}$$  \hspace{1cm} (3)

Where $s_t$ is signals received at $t$ moment. $A_t$ is amplitude of signal at time $t$, $r$ is travel distance of signal and $\lambda$ is signals’ wavelength. Since received phase has linear function with travel distance, so $2\pi \frac{r}{\lambda} t$ is the signal phase when it reach to receive antenna at moment $t$.

Revisit to equation 3, due to the receiver is an antenna array, so $s_t$ should have more complex format to specify the signal is received by which receive antenna, note as $s_{n,t}$, where $n$ is $n_{th}$ antenna number, and $s_{n,t}$ is signals received by antenna $n$ at moment $t$.

Another parameter needs to be clarified is $r$. Since human body is a surface rather than a point, it reflects signals from different directions to all antennas, the received signals at moment $t$ of one antenna contains more than one points’ reflection, thus $r$ varies from multiple reflect points. Figure 10a shows when antenna array scans human body, his left hand $p_1$ reflects to antenna $a_1, a_2, a_3, a_4$ as blue dot line, his right hand $p_2$ reflects to antenna
array as red dot line. Based on above description, equation 4 is designated as follow:

\[
s_{n,t} = \sum_{k=1}^{K} A_n t e^{-j2\pi \frac{r_{n,k}}{\lambda} t}
\]

\( r_{n,k} = \text{travel}(p_k, a_n) \)

Suppose \( p_k \) is \( k_{th} \) points on the detected object, then \( K \) is number of points being scanned, \( r_{n,k} \) is signal travel from \( p_k \) to antenna \( a_n \).

### 4.1 Compute Voxel Power

**Power of Direction:** Based on equation 3 and 4, the problem can be declared as: known signals \( s_{n,t} \) received by antenna \( a_n \) at moment \( t \), then compute reflection power of every scanned points. Because both angles and distance property can be reflected to phase of received signal. More specifically, the power of specific angle \( \phi, \theta \) can be referred by antenna array property, while the power of specific distance \( r \) can be calculated by FMCW feature. Revisit to figure 10a and change antenna array panel to a plane figure, antenna \( a_1, a_2, a_3, a_4 \) receive reflection from \( p_k \), the coming direction of beam is \( \phi \) as shown in both figure 10b and figure 7. While \( \alpha_1, \alpha_2, \alpha_3, \alpha_4 \) are angles between antenna to \( p_k \), and \( d \) is distance between
two antennas. Thus power of direction $\phi$ can be presented as $P(\phi)$ in equation 5:

$$P(\phi) = \left| \sum_{n=1}^{N} s_{n,t} e^{-j2\pi \frac{nd\sin\phi}{\lambda}} \right|$$

(5)

Where $N$ is how many antennas in the dimension. Because $s_{n,t}$ travel different distance for each antenna, and the difference can be represented by $nd \sin \phi$ as depicting with light blue color. Thus their phase change of antenna $n$ is $2\pi \frac{nd\sin\phi}{\lambda}$, $\lambda$ is signal wavelength.

**Power of Distance:** The travel distance of signals also related to the direct distance from point $p_k$ to antenna $a_n$. Frequency Modulated Continuous Wave measures reflection depth by calculating frequency shift between transmit and receive chirp. Equation 1 shows the FMCW feature. We define $v$ is slope of frequency chirp versus time, where $v$ is equal to $df/dt$ in figure 8. So the power of distance $r_k$ can be calculated by phase change of $s_{t,n}$ as shown below in equation 6:

$$P(r_k) = \left| \sum_{n=1}^{N} \sum_{t=1}^{T} s_{n,t} e^{-j2\pi \frac{vt}{c}r_{n,k}} \right|$$

(6)

where $r_k$ is signal travel distance from point $k$. $T$ is the duration of each chirp. Because $f = vt$ and $r/c = t_{travel}$, we can easily get the phase change is $2\pi ft_{travel}$, thus power of $r_k$ is summation over duration $T$ and total antenna number $N$.

**Power of Voxel:** Since HICFR scans 3D surroundings, reconsider situation shown at figure 10b, where $p_k$ on same panel of antenna. However, points in 3D volume need three parameters to locate, either with $(r, \theta, \phi)$ in spherical coordinate system or $(x, y, z)$ in cartesian coordinate system. We choose spherical coordinate system because the power of $\theta$ and $\phi$ can be calculated based on our 2D antenna array. Figure 11 shows how it works:

The 2D antenna array is on $X-Y$ panel, where blocks is antenna. $d_x$ and $d_y$ are distance between two antenna in two dimensions. $Y-Z$ panel is the dimension drawn in figure 10b, and $d_y, \phi$ is $d, \phi$ in equation 5, while $\theta$ is elevation angle from $Y-Z$ panel to $R$. In 3D figure, $R$ is mapping to $Y-Z$ pannel as $YZ$ with $\cos \theta$, and it is mapping to $X-Z$ panel.
as $XZ$ with $\cos \phi$, where $\phi$ is wide angle from $YZ$ to $Z$ axis. Thus the distance change at $Y - Z$ panel for each antenna is $\cos \theta \ast nd_y \ast \sin \phi$ as blue line, that change at $X - Z$ panel is $\cos \phi \ast md_x \ast \sin \theta$ as light blue line shows.

\[
P(r_k, \theta, \phi) = \left| \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{t=1}^{T} s_{n,m,t} e^{-j2\pi \frac{v_r}{c} t} e^{j \frac{2\pi}{X} \cos \theta (nd_y \sin \phi + md_x \cos \phi)} \right|
\]  

(7)

Since distance change represents phase change of signals, then we can calculate power of any voxel by equation 7. $s_{n,m,t}$ is the signal received by receive antenna $n$ from transmit antenna $m$ at time $t$. 

Figure 11: 3D Voxel Power Description
4.2 Construct 3D Image

**Remove Background Reflection:** To get rid of environment reflections such as desks or walls, *HICFR* starts a sensing process before capture humans, name as calibration. Since background reflection is static and the reflection power is fixed, so that calibration sensing, calculating and recording the background reflection power of any voxel, after that, when *HICFR* starts human image capturing task, it subtracts the static background reflection power from the real-time reflection power. We need to make sure there is no human enters the lab during calibration period.

**Construct 2D/3D Image:** Once *HICFR* calculates the power of every voxel and removes background reflection power, it gets a 3D matrix $M$ with the dimension of $(sizeX, sizeY, sizeZ)$, where $sizeX, sizeY, sizeZ$ can be referred from equation 2. Since a 2D image is related to either $(R, \theta)$, $(R, \phi)$ or even $(\theta, \phi)$. To make 2D image has a clear meaning, we choose to construct 2D image with distance and wide angle $(R, \phi)$. At first, we find the highest power from $M$, suppose the highest reflection power is from point at $(R_a, \theta_b, \phi_c)$, then $M(R_a, \theta_b, \phi_c)$
is the highest value in $M$, and $M(R, \theta_b, \phi)$ is a 2D array because parameter $\theta$ is fixed as $\theta_b$. Thus we draw a 2D heatmap image based on $M(R, \theta_b, \phi)$, where the color shows reflection power intensity, the darker color means the higher reflection power at $(R, \theta_b, \phi)$. Figure 12a shows 2D image capturing scenario and its corresponding heatmap. As can be seen from figure 12a, the range of $\phi$ is from $-60^\circ$ to $60^\circ$, where $\phi$ is the angle from dash blue line to human, in this case, dash blue line is the base line in the middle of Walabot, thus $\phi$ is wide angle from base line to object. While 2D image only depicts the highest power layer of fixed $\theta$, 3D image shows more information about object width, height and location. Figure 12b shows how to construct 2D images to 3D images. HICFR uses marching cubes algorithm to draw vertices and faces of stacked images, then it uses normalized filter to remove low power points. It is very clear to see that human is at a shorter direct distance to radar, and the height of human is greater than the chairs in 3D vision.
4.3 Filter Reflection

Another challenge of real-time human image capturing is signal deviation. Since human body is not a plane surface, especially when human moves, the surface of body is extremely deformed. As a result, while our antenna array transmits signals and scans human body, only signals that close to normal surface are reflected back toward the antennas. Other signals may be deviated from another routes and back to receiver, which makes our antenna “misunderstand” the real distance and angle from object. This scenario is shown in figure 14:

\[
\begin{align*}
\text{walabot} & \\
\text{r}_1 & \theta_1 \phi_1 \\
\text{r}_2 & \theta_2 \phi_2 \\
\text{r}_3 & \theta_3 \phi_3 \\
\text{d}_1 & \text{d}_2 \\
\text{r}_3 &= \text{d}_1 + \text{d}_2
\end{align*}
\]

Figure 14: Signal Deviation

In this case, the distance between human chest and leg is not quite large, however, signals transmitted from antenna array travel to human leg and deviate it’s coming route, thus receive antenna gets signals from \( r_3 \theta_3 \phi_3 \), where \( r_3 = d_1 + d_2 \), so that antenna “misunderstand” human leg position with wrong distance and angles, and it results in deformed 3D shape. To address this issue, we design a Deep Neural Network to recognize whether current 3D figure is deformed or not, and remove them from image capture stream. DNN Recognition:

We use transfer learning technique to solve this problem more efficiently. Due to it’s a image processing problem, the proper Deep Neural Network should have Convolutional layers to reduce the amount of possible parameters and calculation. Based on that, we choose \textbf{resNet18} to classify our 3D image. In this work, our contribution is 1) Collect regular and ambiguous images as training dataset, 2) Change the network structure of resNet18 to make
the DNN converge faster, 3) Load pre-trained DNN parameters to filter bad frames.

We collect training dataset from real human activities, while one person walks around in the lab, we construct 3D images and concatenate them as 3D videos. Then we classify them manually into 2 categories: regular frames and ambiguous frames. Figure 13 shows samples of dataset, while figure 13a shows regular 3D reflection power and 13b has ambiguous images. As can be seen from the dataset, the regular frames show human 3D position very clear, and the ambiguous frames always “misunderstand” location of some part of human body.

**Change resNet18 Structure:** The first step to apply transfer learning is changing the last Fully Connect(FC) layer, the last FC layer dimension of normal resNet18 is (512, 1000), which means *in feature* to FC layer is 512 and output 1000 features. The 1000 *out feature* usually feed into *softmax* functions to be classified into 1000 categories. In our design, we only have 2 categories: regular and ambiguous. Then we change the dimension of last FC layer to be (512, 2). The second change of original resNet18 is changing the pooling layer before FC layer. Resnet18 uses Average Pooling layer to compress features to 512, but Average Pooling sometimes cannot extract good features because it takes all into count and results an average value. Since our dataset images have strong edges, and **Max Pooling** extracts the most important or extreme features. So we change the pooling layer to the same size of Max Pooling layer and compare the different convergence of them.

**Load DNN:** In our *HRCIF* design, pre-trained parameters of DNN are loaded in main process. Every frame captured by radar is fed into DNN, then it is recognized and labeled by our DNN with either “regular” or “bad” labels. Next, if the coming frame has “bad” label, the main process filters it and holds the previous frame as current visualization stream result. Since the load and detect process are extremely fast, thus our *HRCIF* can achieve real-time filtering and streaming.
5 HUMAN ACTIVITY RECOGNITION SYSTEM DESIGN

In this section, we dive into technical detail of our Human Activity Recognition based on Radar Image and Recurrent Neural Network (HARNN) system. As mentioned at Section 3, the main purpose of HARNN system is to recognize human activity based on the collected radar image. The whole HARNN system can be divided into three steps: 1) construct labeled video dataset for training phase; 2) Create a suitable neural network model to learn activity features based on labeled training data; 3) Make a prediction in real-time when given real-time radar image stream.

5.1 Video Dataset Construction

Video dataset construction is the prerequisite preparation for HARNN system. The construction process includes two steps: 1) collect continuous videos for any activity; 2) crop and cut fix length frames and set labels for them. Figure 15 below shows the process for dataset construction. At the very beginning, our script generates real-time stacked 3D images as shown in figure 12b. Those continuous images generated by radar and program then be passed to our DNN filter as discussed in Section 4.3. The trained DNN which is considered as our bad reflection filter is placed in the middle of figure 15 as a red block. Our
filtering strategy is when encountering bad frames, then hold the previous frame as current frame until next regular frame is found, the technical detail of loading DNN to recognize bad or regular frame can be found in Section 4.3 with load DNN part. The output of our DNN filter can be considered as continuous radar 3D frames which are able to represent human activities. To gather enough training data, we run our collecting script for a long time with joint activities, then save those videos to our data pool. In our second step of dataset construction, we cut and crop the videos in hand to make sure every activity has fix length of the frame to represent, and label the fixed length frames with its corresponding activity, those activity frames then classified into different categories with their label as the directory name. Note that we have two categories related to “fall” activity: fall and stand up. To be able to stand up is a critical point to determine whether the elder people need help. Once this step complete, we have six categories which have been labeled as sit-to-stand, stand-to-sit, still, walk, fall, and stand up. In each category, it contains a lot of fixed length radar 3D frames which represent their corresponding activity. The reason for crop and trim them as fixed length is we need feed fixed length frames to our Recurrent Neural Network (RNN) as input.

5.2 Activity Recognition Network

The activity recognition network composed of two networks: 1) Convolution Neural Network (CNN) and 2) Recurrent Neural Network (RNN). While CNN is applied to extract image features and RNN is used for predicting human activities. Figure 16 below shows the typical network structure of CNN. As can be seen from figure 16, a classic CNN contains several layers: input layers, convolutional layers, pooling layers, fully connected layers and output layer, those layers work together to achieve image classification task. At the very beginning, given a human image to CNN, it defines convolutional kernels as the white square block and compute convolutional result from the original image, then pooling layer is used to extract the max value or average value of the square region to compress feature maps. Those process combined together to get abstract features of the original image. The output of convolu-
tional layers and pooling layers contains highly abstract information, which called the final feature map. This final feature map is then flattened to 1-Dimension vector for further fully connected layers processing, where fully connected layers compute suitable weight and bias and shrink vector length to be available types of the classification task. Then the given shrunk 1-Dimension vector is further processed with normalization and softmax function and finally result in the probability of each class. The benefit of using the convolutional kernel and convolution compute is saving compute resource and accelerate compute speed because convolutional kernel itself represent the weight parameters, the computation happens when sliding the kernel and each convolutional compute only depends on kernel size.

Suppose input image size is (224*224*3), and kernel size is (7*7*3), the number of kernel is 10, to traverse the whole image, the kernel needs move 218*218 times when padding is 0 and stride is 1 and output size is (218*218*10)and multiply computation running times is 218*218*7*7*3*10. However, when using a typical neural network, we need to flatten the input image as vector length of (224*224*3), and the vector length is a number of neurals at the first layer, to output size as (218*218*10), the weight matrix needs to do multiple computations for (224*224*3*218*218*10) times. This number is far greater than using CNN.
solution. It can be also proved that higher abstract information result in less computation waste. In our case, for every frame, we only take CNN as an information extractor rather than classifier because it’s not an image classification task. Our CNN procedure ends at the first fully connect layers as marked as the red rectangle in figure 16. Given a 3D radar image, our CNN extracts the abstract information which called feature map. Then our HARNN system takes those feature maps and moves on to the RNN system. Figure 17 shows the processing procedure of RNN, we use $fp$ to represent the final feature map, where $fp[t]$ means feature map of 3D radar image at $t$ moment. We also introduce $K$ as fix length of frames, so $fp[t - K]$ is feature map at $K$ frames ahead moment $t$. Equation 8 shows the detail of RNN. When given input $(fp[t - K], fp[t - K + 1], ..., fp[t])$, RNN produces $y[t]$ as predict human activity at moment $t$. Different from using DNN to predict activity, RNN considers temporal information of activity. For each frame of a complete activity, every frame has almost equal contribution to the output, however, RNN model trying to understand the sequential information between given frames, which means it predicts result based on the order of input.

Figure 17: RNN structure

Revisit equation 8, the output of $k_{th}$ unit is denoted as $h_k$. There are three kinds of weight matrices, $W_{xh}$, $W_{hh}$ and $W_{hy}$ respectively. While $W_{xh}$ associates with input $fp$, $W_{hh}$ relates to output of recurrent unit $h_k$, and $W_{hy}$ is key factor of output sample $y[t]$. In the middle of two recurrent units, an activation function $\sigma_h$ enables non-linearity between previous
output and present input. The other activation function, denote as $\sigma_y$, usually set as $tanh$ function to limit the output between $+1$ to $-1$. To describe the forward propagation of the RNN model, we take $fp[t - K]$ and $h_0$ as initial input. At the very beginning, $fp[t - K]$ is fed to the first recurrent unit, and then multiply its weight matrix $W_{xh}$, meanwhile, weight matrix $W_{hh}$ combines with $h_0$ and form the initial previous output. The sum of $W_{hh}h_0$, $W_{xh}fp[t - K + 1]$ and bias $b_h$ pass through activation function $\sigma_h$ and produce $h_1$. This process repeatedly happens at every recurrent unit and finally ends at $K_{th}$ layer, where $y[t]$ is computed by $h_K$, $W_{hy}$, $b_y$ and activation function $\sigma_y$. The output $y[t]$ is a vector which contains the probability of every activity at moment $t$, we update the network’s parameters by computing the difference between prediction and real label, hence improve model accuracy after several training iterations.

$$h_1 = \sigma_h(W_{hh}h_0 + W_{xh}fp[t - K + 1] + b_h)$$
$$h_k = \sigma_h(W_{hh}h_{k-1} + W_{xh}fp[t - K + k] + b_h)$$
$$y[t] = \sigma_y(W_{hy}h_k + b_y)$$

\(8\)

### 5.3 Real-time Application

In this section, we introduce our real-time human activity recognition mechanism. Similar to the load DNN mechanism mentioned in Section 4.3, the real-time application based on our trained RNN model. In figure 18 shown below, suppose our trained RNN takes four frames as input, which means $K$ in equation 8 is equal to 4. As discussed before, the output of HRCIF system is continuous radar frames without bad reflection, and those frames generated in real-time. That means to predict current activity, we need to retrieve previous frames and put them together as the input of RNN. For example, to predict human activity at moment $t = 4$, we need save previous three frames into a buffer, which represent by blue dotted block, this block is also considered as our RNN system, the output of this block $y[4]$ is the vector
contains probability of all activities. With time passing, the block dotted block is a sliding window which always predicts current activity probability, so that our application achieves real-time loading frames and giving a possible prediction.

![Sliding Window Diagram](image)

Figure 18: Real-time Mechanism

6 PERFORMANCE EVALUATION

6.1 Performance Of HICFR System

In this section, we implement our 3D image capturing design as well as DNN classification scheme. The 3D image parameters are set based on our lab size, with direct distance from 0 to 200 centimeter, wide angle Phi(φ) from $-60^\circ$ to $60^\circ$, and elevation angle Theta(θ) from $-20^\circ$ to $25^\circ$.

![Human Real-time Image Capturing](image)

Figure 19: Human Real-time Image Capturing

Hyperparameters of DNN: The DNN is trained with mini-batch strategy to make it
converge more smoothly. We use CrossEntropyLoss as loss function shown in equation 9. Where $x$ is output of DNN, whose dimension is $(\text{minibatchsize}, 2)$, and $label$ is labels for one minibatch data with dimension $(\text{minibatchsize}, 1)$. We use SGD optimizer to update parameters with learning rate $= 0.01$ and momentum $= 0.9$, and a lr scheduler is applied to adjust learning rate with stepsize $= 7$ and gamma $= 0.1$. Then we compare the running loss and accuracy of each iteration in figure 20. Note that running loss and accuracy will be cleared after one epoch.

$$loss(x, \text{label}) = -\log\left(\frac{e^{x[\text{label}]}}{\sum_j e^{x[j]}}\right)$$ (9)

Figure 20a shows the original performance of resnet18 and figure 20b is our DNN result. It is clear to see our DNN converges faster and has less strong vibration compare to original resnet18.

The whole process of HRCIF results in figure 19. The first row records real human motions, the second row is the results before filtering, and the third row is a final result of detecting human. At the very beginning, the human is standing on the right of radar with a wide angle of $60^\circ$, where the cube in row 2 and 3 stand around $\phi = 60^\circ$ and $R = 110cm$. With human moves close to radar from frame 1 – 3, our captured images show wide angle $\phi$ and direct distance $R$ are decreasing gradually. While the human move away from the radar,
the wide angle $\phi$ and direct distance $R$ are increasing. During this time, the frame ahead of the last frame is “bad frame”, so our DNN detects and recognizes the “misunderstanding”, thus hold the previous frame to the current one.

6.2 Performance Of HARNN System

In this section, we implement HARNN system and evaluate our result in different scenarios. Revisit section 6.2, there are three main processes to implement HARNN system: 1) Construct dataset 2) Train RNN and 3) Embed RNN to a real-time application.

6.2.1 Video Dataset Construction

For each activity type, we collect more than 20 radar videos, each of them is longer than ten minutes, our HICFR system produces one frame every second, so the recorded video has a frame rate as one frame per second, which means we have overall $20 \times 10 \times 60 = 12000$ frames in total. Since those frames are recorded in real-time, it contains different activities in one video, for example, when record activity “fall”, our candidate also walk a little bit to have different fall down angle. However, our RNN model demands pure activity frames which a sequential frame only contains exactly one activity. Hence, we need the hand-craft operation to cut and crop our original videos into fixed length and each of the fixed length frames has only one ground truth label. In our case, we crop our video by the length of 3, 5, 8 frames. For each crop standard, we collect 93 samples for fall, 79 samples for stand-up, 93 samples for sit-to-stand, 82 samples for stand-to-sit, 96 samples for walk and 81 samples for still. The total training dataset contains $3 \times \text{sum(samples)} = 1572$ activity samples, across from six different categories with three different crop standard.

6.2.2 Activity Recognition Network

In section 6.2, we introduce a CNN and RNN system to achieve activity recognition. Figure 21 shows the main flowgraph of HARNN system. It can be clearly seen that our candidate
Figure 21: Human Real-time Activity Recognition

falls gradually from the bottle row which marked as ground truth. Our second row from the bottle called the output of HICFR is our training samples for the fall activity, the sample length is five, means the $K$ mentioned in section 5.2 is equal to five. From the first radar image to the last one, it records the position changes of the candidate during the fall activity. The first frame shows human stands on wide angle 60° and direct distance 150 cm, and the purple volume going down along with wide angle dimension since the individual move to the centering direction of radar from frame 2 to frame 3. With our falling on the ground, our last two frames show that the detected human position in elevation angle dimension is very low, and that visualization result indicates someone falls to the ground. However, to make our system understand what activity shown in the five frames, we need RNN model to train our system. Before this RNN model, we introduce a CNN model to extract abstract features of those frames, the CNN model is similar to the CNN model applied in HICFR system, we also
choose the most lightweight and common ResNet called ResNet18. But the different part
is that we revise the last fully connected layer and remove the softmax layer. The original
resNet18 has 512 flatten features before fully connected layer, the new fully connected layer
applied in our CNN produce 64 features to extract highly abstract information. Then we
take the 64 features of every frame as feature map ($fp$) to feed into our RNN model.

**Training strategy:** The training strategy for the HARNN system is unique since it
contains two kinds of neural network. We use pre-trained resNet18 to extract features, and
this period does not update any parameters of resNet18. In this way, we can guarantee the
CNN extract exactly same features for same input frame during the whole training process,
that means whatever the difference between prediction and label, it does not change the
original weights, kernels and bias parameters of CNN model. In other words, the training
process is not end-to-end, we fix the CNN parameters and only train RNN, the backward
function also only works on parameter updating of RNN model.

**Video loader:** When training our network, the training dataset need to shuffle data
to serve the purpose of reducing variance and making sure that models remain general and
overfit less. The shuffling process demands that load dataset in a random and different order.
For images loading task, pytorch provides built-in functions to shuffle dataset. However, it
doesn’t work on video dataset. In this case, our implementation design a shuffle mechanism.
For the beginning of each iteration, we create a list of dictionary named $l_1$ to store all training
videos, with the key is video name and value is video label. The dictionary data structure is
convenient to retrieve training data label. Then we randomly create a list $l_2$ with the same
length of $l_1$, but the value in each cell represents the order of indexing video. Then we can
generate $l_3$ which is shuffled dataset based on $l_1$ and $l_2$. For example,

\[
l_1 = [\{"walk\_1" : "walk"\}, \{"walk\_2" : "walk"\}, \{"fall\_1" : "fall"\}, \{"still\_1" : "still"\}]
l_2 = [3, 0, 1, 2]
l_3 = [\{"walk\_2" : "walk"\}, \{"fall\_1" : "fall"\}, \{"still\_1" : "still"\}, \{"walk\_1" : "walk"\}]
\]

In this way, we can achieve the shuffling purpose and load the video as the order shown in
\[ l_3. \]

**Training hypo-parameters:** The hypo-parameters includes \textit{learningrate}_α, momentum \( \beta \) and initial \( h_0 \). We choose SGD optimizer to update parameters with \textit{learning rate} = 0.01 and \textit{momentum} = 0.9, and a \textit{lr scheduler} is applied to adjust learning rate with \textit{steps} = 7 and \textit{gamma} = 0.1. The loss function is same as equation 9, and the only difference is the number of labels, since our RNN predict probabilities of six different activities, and the former DNN only work on distinguish “regular” and “bad” frame. We trained our 1572 videos for 100 iterations, and the result analysis is placed to next sub-section.

### 6.2.3 Result Analysis

We use GTX 1080 Ti GPU with cuda acceleration technique to train our RNN for 100 epochs, the running time is approximately 2 hours. For each epoch, our program iterates all training data at that length standard. We trained 3 different RNNs with different input length as 3, 5, 8 respectively. The table below shows the training accuracy result of different RNN models.

<table>
<thead>
<tr>
<th>Activity</th>
<th>frame length=3</th>
<th>frame length=5</th>
<th>frame length=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>fall</td>
<td>33.3%</td>
<td>89.2%</td>
<td>91.4%</td>
</tr>
<tr>
<td>stand-up</td>
<td>46.8%</td>
<td>91.1%</td>
<td>97.4%</td>
</tr>
<tr>
<td>sit-to-stand</td>
<td>40.8%</td>
<td>78.4%</td>
<td>38.8%</td>
</tr>
<tr>
<td>stand-to-sit</td>
<td>58.5%</td>
<td>80.4%</td>
<td>42.7%</td>
</tr>
<tr>
<td>walk</td>
<td>45.8%</td>
<td>82.3%</td>
<td>45.8%</td>
</tr>
<tr>
<td>still</td>
<td>96.2%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>overall</td>
<td>52.3%</td>
<td><strong>86.7%</strong></td>
<td>68.5%</td>
</tr>
</tbody>
</table>

As can be seen from above, the first RNN which takes 3 frames as input has the worst prediction accuracy, the overall accuracy is less than 60%. The reason for this issue is 3 frames
is too short to contain all information about an activity. From the human perspective, we need at least 3 continuous frames which indicate human stay on the ground to determine that is a fall activity. The second RNN model performs the best result among those three RNNs, it achieves 86.7% accuracy overall. As for the third RNN, it has the best performance for fall and stand-up activity detection, however, the longer frames contain more joint information, which will distract the faster activity prediction, e.g., sit-to-stand, stand-to-sit. Besides that, we also compare the divergence of prediction and ground truth, then draw confusion matrix as shown in figure 22(a)(b)(c). The confusion matrix shows the number of correct recognition and wrong recognition. For example, the first row and first column of figure 22(a) is 31, means that there are 31 samples with true label as fall that recognized to activity fall; the first row and second column of figure 22(a) is 9, means that there are 31 samples with true label as fall that recognized to activity stand-up. It turns out RNN with frame\textit{length} = 3 is likely to misunderstand between fall and stand-to-sit, stand-up and sit-to-stand, walk and stand-to-sit. RNN with frame\textit{length} = 8 works pretty well at first two categories. But for some quick movement and activity, it confuses to understand the correct information, e.g., wrong prediction between sit-to-stand, stand-to-sit and walk. Our best RNN model comes from the one with frame\textit{length} = 5. But there also exists some prediction flaws, as some sit-to-stand and stand-to-sit videos are recognized as the walk. The potential reason can be found in the next part. For loss convergence evaluation, the training result of three different RNNs can be found in figure 22(e)(f)(g). For the first RNN, it cannot converge well for one hundred epochs. The second RNN with frame\textit{length} = 8 experience violent vibration and finally converge at very low loss. Compared with the former two RNNs, our best RNN with frame\textit{length} = 5 converge very quick and stay stable during the rest of epochs. Our final result is shown in figure 23. We list all six categories’ activity and shows the ground truth versus 3D frames. The visualization result represent human activity pretty well, and our RNN can predict human activity based on those frames. With carefully analyze the visualization result, we found the potential reason for RNN misunderstanding
Figure 22: Accuracy vs Loss Convergence
walk and sit-to-stand and stand-to-sit. Revisit figure 19, we hold the previous frame when encounter “bad reflection” frame, however, if the gap between those two frames is pretty large, it can be considered as a sit-to-stand activity because the most obvious feature of sit-to-stand is position changes in sudden. An example of this scenario can be found in the third row of figure 23. When human have sit-to-stand transition activity, the purple volume has extremely position change between frame 3 and frame 4. And if our collected walk video is discontinuous, it can also find the extremely position change in walk dataset.

7 Conclusion

In this work, we describe two novel approaches to visualize users location, HICFR, and recognize the human activity, HARNN through commodity off-the-shelf radar sensors. Through the technique feature of FMCW radar, our HICFR achieves illustrating human 3D image in real-time. The collected radar images are then put to HARNN system to recognize human activity. We collect 1572 video samples in total to train three RNNs, the best RNN has state-of-art performance compared with the rest of RNNs. Our overall accuracy reaches to 86.7% to classify into six categories based on radar images. Further work will focus on fine recognition design and implementation towards problem mentioned in section 6.2.3, the possible design is instead of holding the previous frame, we design an interpolate algorithm to produce a new frame between the gap of two frames. We will evaluate this idea in further research.
Figure 23: Overview dataset
Bibliography


