Cover Letter: Imaging using Point-Cloud Data Generated from a mmWave Radar

Akash Deep Singh[a,1], **Ziqi Wang**[a], **and Qiong Hu**[b]

[a]akashdeepsingh@ucla.edu (405228294), wangzq312@ucla.edu (905229998), Networked and Embedded Systems Laboratory, UCLA ECE; [b]qiong.j.hu@gmail.com (405065032), The Laboratory for Embedded Machines and Ubiquitous Robots, UCLA ECE

This manuscript was compiled on December 14, 2019

This cover letter talks about the ideas present in the paper (that follows it) and how they tie to computational imaging. We also discuss work that has been done as an extension to the paper and some future directions.

The paper titled "RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-wave Radar" has been published and presented at Mobicom 2019 (Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems) which took place on October 25, 2019.

1. Overview (Paper Abstract)

Accurate human activity recognition (HAR) is the key to enable emerging context-aware applications that require an understanding and identification of human behavior, e.g., monitoring disabled or elderly people who live alone. Traditionally, HAR has been implemented either through ambient sensors, e.g., cameras, or through wearable devices, e.g., a smartwatch, with an inertial measurement unit (IMU). The ambient sensing approach is typically more generalizable for different environments as this does not require every user to have a wearable device. However, utilizing a camera in privacy-sensitive areas such as a home may capture superfluous ambient information that a user may not feel comfortable sharing. Radars have been proposed as an alternative modality for coarse-grained activity recognition that captures a minimal subset of the ambient information using micro-Doppler spectrograms. However, training fine-grained, accurate activity classifiers is a challenge as low-cost millimeter-wave (mmWave) radar systems produce sparse and non-uniform point clouds. In this paper, we propose RadHAR, a framework that performs accurate HAR using sparse and non-uniform point clouds. RadHAR utilizes a sliding time window to accumulate point clouds from a mmWave radar and generate a voxelized representation that acts as input to our classifiers. We evaluate RadHAR using a low-cost, commercial, off-the-shelf radar to get sparse point clouds which are less visually compromising. We evaluate and demonstrate our system on a collected human activity dataset with 5 different activities. We compare the accuracy of various classifiers on the dataset and find that the best performing deep learning classifier achieves an accuracy of 90.47%. Our evaluation shows the efficacy of using mmWave radar for accurate HAR detection and we enumerate future research directions in this space.

2. Drawing Parallels to Computational Imaging

In this section, we discuss how the paper relates to the concepts we have studied in the Computational Imaging course.

A. Point cloud images as an input to the neural network. The data obtained from the milli-meter wave (mmWave) radar is in the form of point clouds. The number of points in each frame captured by the mmWave radar varies, increasing the complexity of constructing a neural network architecture that can process this data as is. Neural Networks require their input to be of uniform size. To overcome the non-uniformity in number of points in each frame, we converted the point clouds into 2D (pixels) and 3D (voxels) images. These images were given as an input to the neural network. The dimensions of the 3D images (which performed the best) were 10x32x32 (depth=10). This made the input of constant size irrespective points of the number of points in the frame. We decided these dimensions empirically by testing their performance. In our 3D image representation, the value of each voxel is the number of data points present within its boundaries. While having large number of voxels may represent underlying information well, it increases the data size by several orders of magnitudes.

B. Point cloud acquisition as a similar method of ToF camera. In addition to creating images from the point clouds, the paper (especially the mmWave hardware used) also shares a lot of similarities and employs common techniques with computational imaging.

To begin, in order to estimate the x,y,z coordinates of a point in space, the radar has to process the received signal. Two techniques are primarily used in order to find out these coordinates – time-of-flight sensing and angle-of-arrival estimation. In addition, the mmWave radar also uses Doppler effect to find out what the velocity of the target point is.

During lectures, we were introduced 3D depth sensing, and mmWave falls into the category of Time-of-flight(ToF) sensing. Similar to Lidar, mmWave sensing also suffer from "the curse of light speed". Even with the help of a very high data sampling rate and fast ADCs, it is still very difficult to get a precise ToF measurement. Thus, the point cloud acquired with mmWave radar also suffers from a coarse resolution, and further makes inferencing very challenging. The range-resolution of such EM wave is given by

$$r = \frac{c}{2B} \quad [1]$$

where c is the speed of light and B is the bandwidth. As indicated in (1), with a frequency of 77-81GHz and a bandwidth of 4GHz, a 3.75cm localization accuracy can be achieved theoretically. The difference between mmWave and Lidar is that instead of using AMCW (Amplitude Modulated Continuous Wave), mmWave uses FMCW (Frequency Modulated

Authors declare no conflict of interest with the material in this manuscript.

[1]To whom correspondence should be addressed. E-mail: akashdeepsingh@ucla.edu

Continuous Wave)where the distance of the target is inferred based on the frequency difference between probing waves and reflective waves. The sender will transmit a frequency chirp whose frequencies increase linearly overtime until reaching the maximum frequency. The reflected signal will be a time-delayed chirp signal. Thus at a certain point by calculating the Fast Fourier Transform (FFT) of the sending signal and the receiving signal, it is possible to get the frequency difference between those two. Since the slope (rate) of frequency change over time is known, we can estimate ToF with simple mathematics.

Meanwhile, the movements of the target object will lead to change in received frequencies due to Doppler effect. To estimate movement speed, the mmWave board takes out the response associated with multiple probing chirps. Instead of doing FFT on a single chirp response, mmWave radar will take out the data with the same amount of delay across chirp responses. It is easy to estimate the speed by looking at the frequency changes over time. By now we acquired a 2D matrix series – range(distance) and velocity over time. Finally, mmWave applies Multiple-input-multiple-output (MIMO) techniques, where three transmitting antennas and four receing antennas are used to form 12 TX-RX pairs. The three Tx are placed in a triangular way and the four Rx are placed as a linear array. Similar to the beam-forming scheme introduced in class, the Angle-of-Arrival (AoA) of receiving signals can be estimated with phase delay across Tx-Rx pairs. Combining AoA, ToF and time, we can get point clouds with (X,Y,Z,V), which indicated 3D location and velocity.*

3. Extension

As was introduced during the lecture, Adib et.al was able to capture a human image through wall with customized RF antenna arrays(2). His work shows great promise of using radio frequency for ToF imaging, whose potential is then unleashed by a lot of following work. As an extension to the paper, we tried to perform human tracking and imaging from mmWave point clouds. Unlike cameras, radars are point scatterers and hence not every part of the target is visible in every frame. However, if we observe a moving target for a period of time with continuous motion but not translate long-distance, we can estimate the size and shape of the target figure because as the number of observations increase, the chances of a point reflecting back to the radar increases. Hence, we can assume that if we observe a moving target for a period of time (time frame window), a majority of points present on the object reflect back to the radar at least once. And in our implementation, we empirically applied time frame window of 2 seconds.

Another dominant challenge of performing human figure from the raw data comes from the noisy reflection signal of the environment, such walls, other furniture in the environment, and the ceiling. There are works on clustering and machine learning (3) to filter out the noises from the environment. We apply statistical approaches to solve the problem, track the target human position, and to estimate human postures.

We use three different filters on data points distributed in each dimension (X, Y, Z) based on their statistical characteristics, apply them in two different sorting orders, and implement a shift-and-add for each frame every two seconds, to estimate the target human figure.

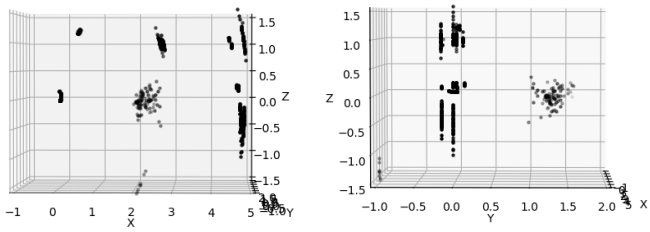

Fig. 1. Raw data from first 60 frames in the view of XZ (left) and YZ (right), respectively. The transparency of each data point represent its signal intensity.

A. Single-dimension filter. The distribution of data on X, Y, and Z dimension follow some certain different statistic characteristics, which will be illustrated respectively.

A.1. X. As seen from Fig. 1, the raw signal in X dimension mainly fall into three area: points on left edge and right edge are generated by the walls, and the clustered signals in the middle area come from the targeting human.

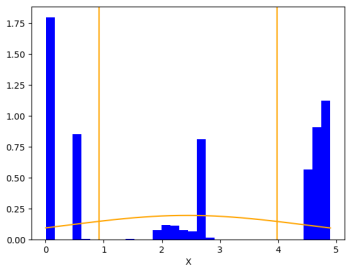

Fig. 2. The blue columns is the data histogram in X dimension, the yellow curve is the Gaussian fit curve, and rest two yellow lines are the bounds of the applied filter. Note that the data in middle area are the target signal.

Similar to the voxelization method in our published paper, where the whole space is divide into 10*32*32 voxels, we use 32 pillars to generate the histogram in Fig. 2 and observe the data distribution characteristics in X dimension.

In order to filter out the noises on the two edges, instead of use a harsh threshold on the edge (which may require manual adjustments for different system settings), we apply the Gaussian fit and FWHM filter, as seen in Algorithm 1.

And the resulting filter bounds are also shown in Fig. 2, which fall on the gaps between desired signal and the noises, as expected.

A.2. Y. In the right image of Fig. 1, we can see that the noise have strong intensity, large amount of number and are close to the left part of the environment, and the target signals distributed on the right.

Similar to the methods for X dimension, we also generated a histogram with 32 pillars, and tried to use a Gaussian fit curve to extract the target data in Y dimension, but as we can from Fig. 3, it seems no longer a proper approach.

However, if part of the noise has already been filtered, for example, after applying a filter on X and(/or) Z dimension

*We thank Yen-Chin Wang for helpful discussion on mmWave point cloud data acquisition using 3D FFT.

Algorithm 1 FWHM filter for X

1: Gaussian fit of the histogram
2: Find the peak point of the Gaussian curve p, where $p = (x_p, h_p)$
3: Find the minimal point to the left and right side of the peak on the Gaussian curve, l, r, where $l = (x_{min}, h_l)$, $r = (x_{max}, h_r)$
4: Find the left and right bound of the filter x_l, x_r, corresponding to the x position of h_{x_l} and h_{x_r},where $h_{x_l} = (h_p + h_l)/2$, $h_{x_r} = (h_p + h_r)/2$ on the Gaussian curve

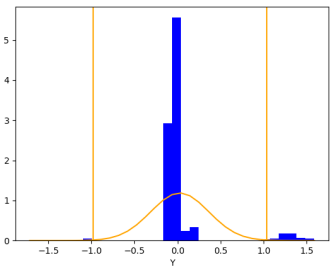

Fig. 3. The blue columns is the data histogram in Y dimension, the yellow curve is the Gaussian fit curve, and rest two yellow lines are the bounds of the applied filter. Note that the low data on the right is the target data, thus this filter only leave the noise remain.

as seen in Fig. 4, there could still be a Gaussian filter for Y dimension. The idea is that in any Gaussian curve, 99% of the information would be within the range of 3 standard deviation from the mean. Therefore, there is a high possibility that the data outside this range is a noise and can be neglected.

A.3. Z. As shown in Fig. 1, especially in the right YZ view, the target signal remain lower than a certain height, and those data above would be considered as noises, which is reasonable because in our implementation, the human cannot reach the ceiling no matter the motion.

Instead of applying a single Gaussian fit for signal in Z dimension, we use a function that is the summation of two Gaussians:

$$f(x) = p_0 * e^{-\frac{(x-p_2)^2}{2p_4^2}} + p_1 * e^{-\frac{(x-p_3)^2}{2p_5^2}}, \qquad [2]$$

where p_i $(i = 0, 1, 2, 3, 4, 5)$ are six parameters fit from the histogram.

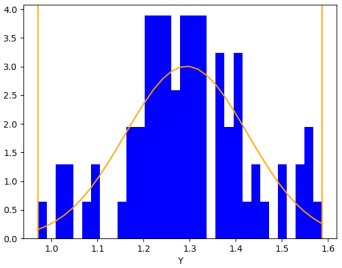

Fig. 4. The blue columns is the data histogram in Y dimension, the yellow curve is the Gaussian fit curve, and rest two yellow lines are the bounds of the applied filter.

Algorithm 2 Gaussian filter for Y

1: Gaussian fit of the histogram
2: Find $mean$, std of the Gaussian fit
3: Calculate the left and right bound of the filter y_l, y_r: $y_l = mean - 3 * std$, $y_r = mean + 3 * std$

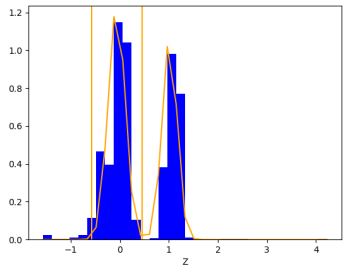

Fig. 5. The blue columns is the data histogram in Z dimension, the yellow curve is the fit curve, and rest two yellow lines are the bounds of the applied filter. Note that the data on the left is the target data, and right columns are the noise.

So Eq. (2) allows the two peaks in Fig. 5 to distinguish between each other. The reason why a similar method of three Gaussian summation not used to fit and filter for data in X dimension is that, the number of the data in our histogram is only 32 (same as the column number), and may not be enough for a function of nine parameters to converge. The same converge problem sometimes can also occur for filter on Z, which would be even more critical in other extended applications as we will mention in the section of applications and additional results. The Algorithm 3 shows the pipeline to implement the filter.

Algorithm 3 Double-Gaussian filter for Z

1: Function fit of the histogram use Eq. (2)
2: Find $mean$, std of the left Gaussian curve: $mean = p_2$, $std = p_4$
3: Calculate the left and right bound of the filter z_l, z_r: $z_l = mean - 3 * std$, $z_r = mean + 3 * std$

B. Sorting order. There are two pipelines to apply these three individual filter methods, depending on whether the filter result in one dimension affects that in another dimension. We call it *Independent sorting* if the filter process is irrelevant and independent among three dimension, and otherwise *Dependent sorting*.

B.1. Independent sorting. In Algorithm 4, we only apply filter on X and Z dimension for the reason that has been explained in individual Y filtering. It is sufficient in most cases to only consider data distribution in X and Z dimension since the radar observation space is an rejection from the 3D real-world space to the 2D space, thus, the information of the depth is an additional but not necessary information for filtering.

B.2. Dependent sorting. In order to also make use of the information from Y dimension, we develop Algorithm 5 where the filter order follows: $Z \to X \to Y$.

The reason of the order is purely empirical, since the function fit of Z can effectively filter out most noises above the

Algorithm 4 Independent sorting

Find filter bounds in X: x_{min}, x_{max} using all the data
Find filter bounds in Z: z_{min}, z_{max} using all the data
for all signal point P **do**
    if $x_{min} \leqslant P_x \leqslant x_{max}$ and $z_{min} \leqslant P_z \leqslant z_{max}$ **then**
        P is targeting signal
    else
        P is noise

Algorithm 5 Dependent sorting

Find filter bounds in Z: z_{min}, z_{max} using all the data
Find the subset Z_{list} in which all the point P satisfy $z_{min} \leqslant P_z \leqslant z_{max}$
Find filter bounds in X: x_{min}, x_{max} using Z_{list}
Find the subset X_{list} in which all the point P satisfy $x_{min} \leqslant P_x \leqslant x_{max}$
Find filter bounds in y: y_{min}, y_{max} using X_{list}
for all signal point P in X_{list} **do**
    if $y_{min} \leqslant P_y \leqslant y_{max}$ **then**
        P is targeting signal
    else
        P is noise

targeting figure, and the filtering effect in X dimension comes the second, Y being the last.

In practical, with our data from walking experiments, most signal point in X_{list} could pass the Y filter, which also complies with the case of Independent sorting where Y information does not matter as all. But considering motions with big actions or quick transition in space, add a Y filter may make the filtered signal more clean.

Since Dependent sorting requires two extra loops to search among the data set to obtain subsets, we also concern about the computational complexity. With radar data of human motion in two to three minutes, the calculation and plotting time for independent sorting is about 5.7 seconds, and about 6.1 seconds for dependent sorting, in average.

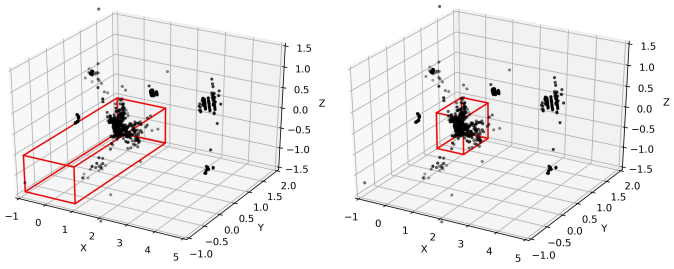

Fig. 6. Filtering results from Independent sorting (left) and Dependent sorting (right), generated from the 4-6 seconds in the experiment.

A comparison of the results from the two filtering pipelines are shown in Fig. 6. Empirically the results would be identical when the human figure is almost stationary, for example, in the first 2 seconds of the walking experiment, but with movements, the signal would spread in the space for the same period of time, therefore, dependent sorting would generously have a better performance than independent sorting.

C. Shift-and-add. As a common image processing methods, the approach of shift-and-add are typically applied when combining information in images captured at different time, space, or view-point etc.

Due to the sparsity of the point cloud generated by mmWave, we initially consider a time frame window of 2 seconds in the filtering process, which is a combination of 60 frames. Now that after the aforementioned sorting pipelines, the signals from the targeting figure have almost been determined. It is reasonable to consider the shift-and-add approach to retain extra spacial movement information due to the fact that each frame have a difference of 1/30 seconds in time.

The pipeline of shift-and-add is straightforward as described in Algorithm 6. The challenging problem would be how to properly define the center function that estimates the center points.

Algorithm 6 Shift and add

Define a function to estimate the center point among multiple data points
Find the center point of 60 frames *center*
for all frame **do**
    Find the center point of the single frame $center'$
    Shift the position of all the data points P in the frame in the same way that aligns $center'$ to *center*: $P' = P - (center' - center)$
    All the shifted data points P' form the new estimated targeting figure

There are variously ways to define the center-estimation function from our radar data. Since the data set contain the information of position, intensity and velocity, some reasonable approaches would be as follows:

1. Simple average of position
2. Intensity-weighted average position
3. Velocity-weighted average position
4. Combined intensity and velocity weighted average position
5. Position of the point with maximal intensity
6. Position of the point with maximal velocity
7. Position of the point with the best combined behavior of intensity and velocity

In general, among all the position in human body, the torso body part would have higher intensity, and the limbs would have higher velocity. In theory and anatomy, the human body is symmetric and the center would be somewhere around navel and chest, which lies inside of the torso. Therefore, we define the intensity-weighted average position of all points as the center point.

The comparison results from shift-and-add is presented in Fig. 7. Just as expected, the spread points due to time-dependent human motion becomes more concentrated after shift-and-add, and provide a more condensed estimation of the human figure.

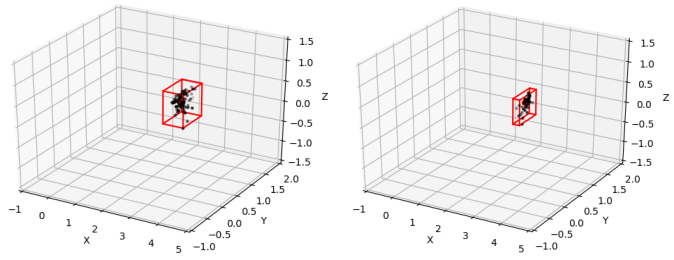

Fig. 7. Filtering results without shift-and-add (left) and with shift-and-add (right), generated from the first two seconds in the experiment.

4. Additional Results

After applying noise filters and shift-and-add, we are able to gain a knowledge of human performance as a cuboid representation and a center point from the raw sparse data cloud from radars, leading to several interesting applications.

A. Trajectory estimation. By connecting all the center points of every two seconds in the complete walking experiment, we can find out the motion trajectory as seen in Fig. 8.

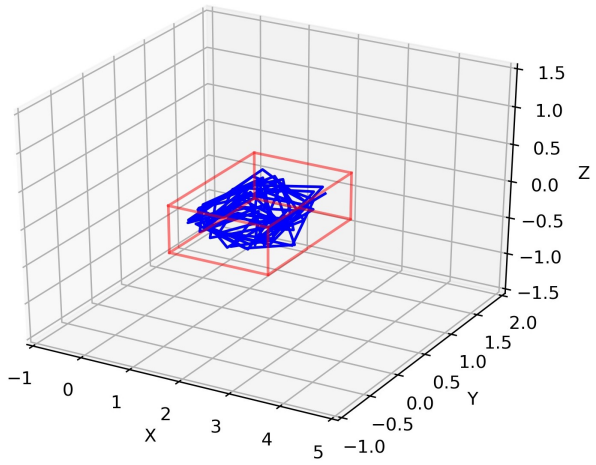

Fig. 8. Estimated trajectory of walking experiment in 178 seconds. The red box indicates the maximal boundary of the trajectory.

From the estimation, the maximal range of the trajectory is a 2m*1.6m area, while in experimental setting the moving space is 3m*3m. Considering other physical constraints in the area, the estimated result can be quite close to the ground true of the motion range.

The trajectory seems random, which also complies with the fact that in the experiment, the human just randomly walked around in the area.

The estimated trajectory have a range of 0.25m in Z dimension. In the research (4) of trajectory of human body mass during walking, the vertical range of the mass center is about 1% of the height, which in our case, the ground truth would be 0.19m. The two values are quite close, showing that the trajectory estimation is nice and accurate.

B. Figure size estimation. From the cuboid representation of the human figure, we may also want to estimate the figure size. As seen in Fig. 9, the size of the cubes varies dramatically in all dimensions among different time steps.

From the total 178 seconds of the walking experiments, the average size of the figure in X-Y dimension is 0.63m*0.67m, the average height is 0.97m, and the maximal height of all cubes is 2.23m. The ground truth of the human in the experiments is 1.86m, so the figure size estimation still has large errors.

The main reason of the error is the sparsity of the point cloud from the radars. Even if each cube is a estimation result from 60 frames in two seconds, there are still large amount of the space that has not been detected, thus the estimated human figure may not cover the whole body. And in the cases where the environmental noises are not completely wiped off, the figure would be a lot larger than the real human body figure. Therefore, it is not yet effective to use the results from this work to estimate figure size.

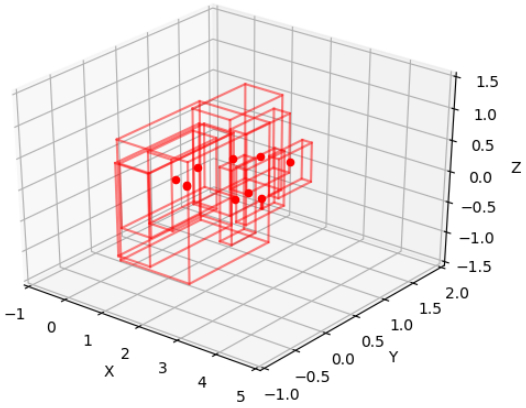

Fig. 9. Cuboid representation of human figure in 20 seconds

C. Figure skeleton. Relating to the work in our published paper, we would like to use the result of this work to extract human performance in a stick model referred by Fig. 10, so that the human motion may be effectively classified.

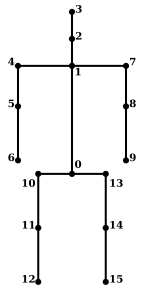

Fig. 10. Human frame used as a reference

The present cuboid representation does not offer efficient posture information, so we consider the human figure as a stationary avatar, whose ratio of each body part is determined purely by the height of the figure, and the orientation is set to be in X dimension. An example of such skeleton representation from first 10 seconds of the data is illustrated in Fig. 11. There could be more future works on developing the human figure frame on the basis of this work.

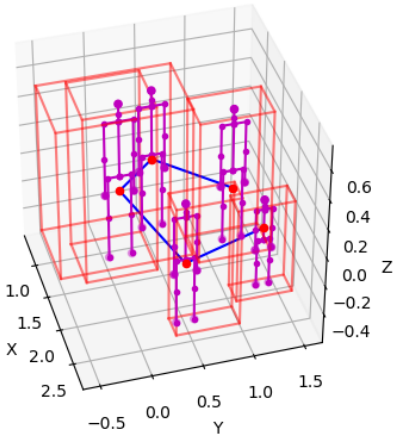

Fig. 11. Human frame and trajectory, generated from first 10 seconds of the walking experiments

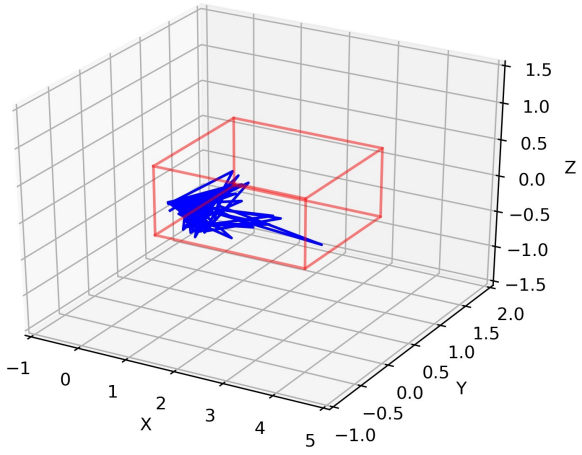

Fig. 12. Estimated trajectory of a different walking experiment

D. Transferable to different datasets.

D.1. More walking experiments. Since our algorithms and approaches with noise filters and shift-and-add are developed purely from one dataset in one experiment, we also would like to check its transfer ability by applying to more walking and other motion experiments.

We did not primarily know or assume the data distribution of this different walking experiment, and did not adjust any parameter in our filters or shift-and-add, the estimated trajectory result in Fig. 12 is still acceptable in practical, with a few exceptional points in each dimension.

The estimated vertical range is 0.74m, two times larger than the result from the original dataset, suggesting that there are more environmental noises involved in the estimated figure and the filter may not function as well as that is on the original dataset.

One of the possible improvement is the filter for Z dimension, where the fitting function may not converge if the initial value of mean and standard deviation is too far away from the fitted value. Those initial values are generated empirically from the original dataset, which could be replaced by a learning algorithm.

D.2. More different motions. Apart from walking, there are four more motions implemented in our published paper: boxing, jack, jump and squat. We tested our approaches on all of them and we would use the result with jumping as a representative for analysis.

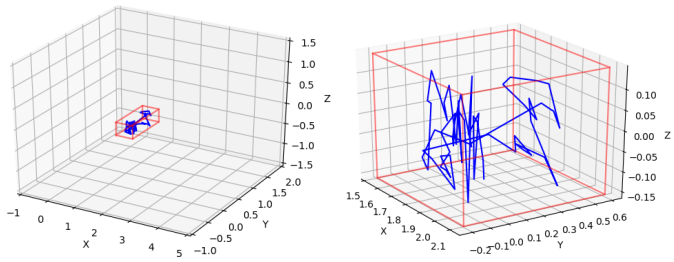

Fig. 13. Trajectory estimation for jumping in the view of the whole space (left) and a zoomed-up closer look of the details (right).

As presented in Fig. 13, the trajectory from jumping occupies a less proportion of the space compared to walking. One of the possible reasons might be the accuracy of the human figure estimation is higher for jumping compared to walking. A jumping figure occupies a larger proportional of the actual physical space, allowing more data points from the human being to be detected by the radars, thus the data cloud of jumping would be more condense than that of walking, leading to a more accurate and concentrated figure estimation, and then a shrunken trajectory space.

The vertical range of the trajectory is 0.44m, which would be a practical estimation to the ground truth.

The trajectory shape also aligns with the real situation. In the experiments, the person is asked to jump constantly and shift the position randomly. From the shape of the trajectory, the line has multiple overlapping in the left part of the image, suggesting it is a in-place jumping, and then randomly shift to a second place, vividly reflecting the real motion.

Overall, the method developed and introduced in this work on filter and shift-and-add have a satisfactory performance on human figure estimation and trajectory estimation, and is transferable to dealing with data from multiple different motions.

5. Future Work

Combining the capabilities mmWave radar with other types of sensors can result in some interesting capabilities. For instance, a camera is not able to see through fog whereas a radar can. On the other hand, a camera produces high quality, high resolution images whereas the images produced by a radar are coarse. If we combine these two modalities together, they can complement each other well and create a more robust imaging system.

1. Jacobi R, Aginsky A (year?) Choosing 60-ghz mmwave sensors over 24-ghz to enable ...
2. Adib F, Hsu CY, Mao H, Katabi D, Durand F (2015) Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)* 34(6):219.
3. Zhao P, et al. (2019) mid: Tracking and identifying people with millimeter wave radar in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*.
4. Jurcevic Lulic T, Muftic O (2002) Trajectory of the human body mass centre during walking at different speed in *DS 30: Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik.*

6. Attached Paper

Please turn over/ scroll to the next page to read the full paper.

RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-wave Radar

Akash Deep Singh[a,1], **Sandeep Singh Sandha**[a], **Luis Garcia**[a], **and Mani Srivastava**[a]

[a]Networked and Embedded Systems Laboratory (NESL), UCLA ECE and UCLA CS

This manuscript was compiled on December 14, 2019

Accurate human activity recognition (HAR) is the key to enable emerging context-aware applications that require an understanding and identification of human behavior, e.g., monitoring disabled or elderly people who live alone. Traditionally, HAR has been implemented either through ambient sensors, e.g., cameras, or through wearable devices, e.g., a smartwatch, with an inertial measurement unit (IMU). The ambient sensing approach is typically more generalizable for different environments as this does not require every user to have a wearable device. However, utilizing a camera in privacy-sensitive areas such as a home may capture superfluous ambient information that a user may not feel comfortable sharing. Radars have been proposed as an alternative modality for coarse-grained activity recognition that captures a minimal subset of the ambient information using micro-Doppler spectrograms. However, training fine-grained, accurate activity classifiers is a challenge as low-cost millimeter-wave (mmWave) radar systems produce sparse and non-uniform point clouds. In this paper, we propose , a framework that performs accurate HAR using sparse and non-uniform point clouds. utilizes a sliding time window to accumulate point clouds from a mmWave radar and generate a voxelized representation that acts as input to our classifiers. We evaluate using a low-cost, commercial, off-the-shelf radar to get sparse point clouds which are less visually compromising. We evaluate and demonstrate our system on a collected human activity dataset with 5 different activities. We compare the accuracy of various classifiers on the dataset and find that the best performing deep learning classifier achieves an accuracy of 90.47%. Our evaluation shows the efficacy of using mmWave radar for accurate HAR detection and we enumerate future research directions in this space.

1. Introduction

Although the recognition and monitoring of human activities can enable safety critical applications–e.g., monitoring disabled and or elderly people living-alone that may need medical attention (1, 2)–the emergence of low cost sensing capabilities have enabled ubiquitous possibilities of human activity recognition for everyday applications. Several context aware applications have recently emerged such as workout tracking and efficacy (3), and factory floor monitoring (4). Traditionally, human activity has been inferred either through ambient sensors (e.g., cameras) and/or wearable sensors (e.g., smartwatches with IMUs). Although wearables have proven to be effective approaches for human activity recognition, it is not practical to assume that all of the subjects in a space will use wearables that are compatible with the inference model. Ambient sensor approaches are robust to heterogeneous environments since they do not rely on users having a particular device. However, the sensor data from cameras carry a significant amount of ambient information that may be of concern for privacy-sensitive applications. For instance, there have been cases where cameras in the maternity ward of a hospital were used to spy upon female patients (5). In this context, cameras can be replaced by sensors whose data can provide a sufficient amount of ambient information to realize the same utility, e.g., if we only care about how many patients are in a room or whether there are a certain number of nurses in the ward.

Prior works have shown that less information-rich ambient sensors can effectively infer human activities while not exposing subjects to privacy risks using radio frequency signals. For instance, WiFall (6) showed how WiFi routers can be used to detect whether a human has fallen or not. However, the work is not robust beyond binary classification of two classes that are significantly different from each other. Generally, WiFi has a narrow band (when compared to the high bandwidth of a mmWave radar) and does not have sufficient range resolution to perform robust classification. Radars have begun to emerge as a popular modality for activity recognition(7) as they provide the advantage of operating in any lighting condition and work through a multitude of environmental conditions, e.g., fog and rain. Further, the emergence of millimeter-wave technology has enabled cost-effective distributed sensing applications.

Millimeter-wave (mmWave) technology operates in the frequency range of 30GHz and 300GHz. Since, antenna size is inversely proportional to frequency, the higher you go up in the frequency spectrum, the lower the size of antenna becomes. As a result, mmWave radars are compact in size. Also, we can pack a large number of antennas into a very small space which enables highly directional beam-forming ($\approx 1^{\circ}$ angular accuracy). Since these radars have a large bandwidth, they have a superior range resolution. Further, new low cost, off-the-shelf radars have led to an increase in the popularity of mmWave based sensing solutions. However, these devices are resource-constrained and, instead of providing raw data, their output is in the form of point clouds*. The number of points in each frame captured by the mmWave radar varies, increasing the complexity of constructing a neural network architecture that can process this data as is. Hence, several feature extraction and data pre-processing techniques have been proposed in previous works which convert this data into a format which is constant in size and can be given as input

*To get raw ADC data from these devices, you need to connect them with expensive hardware

Authors declare no conflict of interest with the material in this manuscript.

[1]To whom correspondence should be addressed. E-mail: akashdeepsingh@ucla.edu

to a neural network (8, 9).

In this paper, we propose , a framework for human activity recognition that utilizes point clouds generated from a mmWave radar. To account for the sparsity of the mmWave radar point clouds, leverages the notion that human activities typically last over a few seconds and accumulates point clouds over a sliding time window. Each point cloud is voxelized to overcome the non-uniformity of the data and is then fed into a set of classifiers. We collected a new HAR dataset consisting of point clouds using mmWave radar for 5 different classes of activities. We evaluated and compared the accuracy of various classifiers on the collected dataset. In our evaluation, the best performing deep learning classifier composed of a set of convolutional layers and long-short term memory layers can achieve an average test accuracy of 90.47%.

Contributions. Our contributions are summarized as follows.

- We propose , a framework that performs human activity recognition using a pre-processing pipeline for point clouds generated by mmWave radar.
- We evaluate different machine learning approaches for human activity detection using point cloud.
- We generate a new point cloud dataset for human activity detection and make it available open-source along with the data processing, classifier training and evaluation code, and pre-trained classifiers.

The rest of this paper is arranged as follows. In Section 2, we present the preliminary information required to understand the framework. We provide an overview of and its implementation in Section 3, and evaluate the classification approaches in Section 4. The related work is presented in Section 5, and we discuss and conclude in Section 6 and Section 7, respectively.

The source code and datasets of are available online at https://github.com/nesl/RadHAR.

2. Background

We provide the preliminary information necessary to understand the framework. We discuss the basics of mmWave radar physics.

A. Millimeter-wave Radar. Over the last several years, there has been a growth in low cost single chip radars that work in the mmWave range. One family of such popular devices are Texas Instruments' mmWave radar. These sensors output the point clouds that contain information like x,y,z positions of each point among other data.

Bandwidth and range resolution. Range resolution of a radar is its ability to distinguish between 2 targets present very close to each other. The range resolution and bandwidth are related as,

$$d_{res} = \frac{c}{2B} \quad [1]$$

where d_{res} is the range resolution in m, c is the speed of light in m/s and B is the bandwidth in Hz swept by the chirp of the radar. Hence, if we want a better range resolution, the bandwidth should be high. The maximum continuous bandwidth for the radar that we used is 4 GHz which corresponds to a range resolution of about 4 cm.

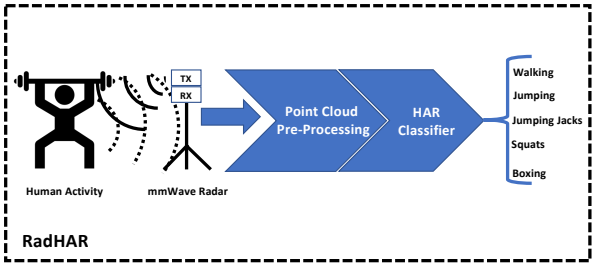

Fig. 1. framework overview.

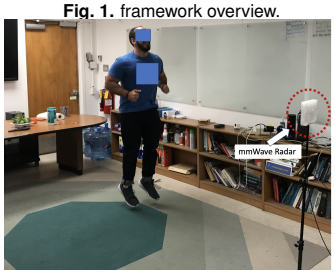

Fig. 2. Data collection setup.

3. Overview

The full pipeline of the framework is depicted in Figure 1. The framework first collects data from a mmWave radar that is monitoring a human. The point cloud data is pre-processed before being fed into a HAR classifier. We present an overview of each component in detail.

A. Data Collection & Pre-processing. We have used TI's IWR1443BOOST (10) radar to collect the new point cloud dataset called *MMActvity* (millimeter-wave activity) dataset. It is a FMCW (Frequency Modulated Continuous Wave) radar which uses a chirp signal. This radar works in the 76-GHz to 81-GHz frequency range. The radar includes four receiver and three transmitter antennas, which enable tracking multiple objects with their distance and angle information. This antenna design enables estimation of both azimuth and elevation angles, which enables object detection in a 3-D plane (11).

B. MMActvity Dataset. For data collection, the radar is mounted on a tripod stand at a height of $1.3m$. The data from the radar is sent to the laptop via ROS (Robot Operating System) messages over USB. The ROS node running on the laptop is developed and described in (8). To record and store the data, we use rosbag which another ROS package. Finally, we convert these rosbags into .txt files. These files are then used to create voxelized representation of the point clouds. The data collection and pre-processing pipeline is describe in Figure 4.

We have collected the data from two users[†]. The users perform 5 different activities in front of the radar as shown in Figure 2. These activities are: Walking, Jumping, Jumping Jacks, Squats and Boxing. The data is collected in a continuous periods of about 20 seconds for a subject performing the same activity. Some of the data files are longer than 20 seconds. In total, we have collected 93 minutes of data. The description of the dataset can be found in Table 1.

The captured point clouds contains spatial coordinates (x,y,z in meters) along with velocity in meters/second, range (distance of the point the from radar) in meters, intensity (dB)

[†]The data is collected from the authors and thus does not require approval from IRB.

Fig. 3. Data collection and storage pipeline

Activity	# of data files	Total duration (seconds)
Boxing	39	1115
Jumping Jacks	38	1062
Jumping	37	1045
Squats	39	1090
Walking	47	1269

Table 1. Details of the MMActivity dataset.

and bearing angle (degrees). The sampling rate of the radar is 30 frames per second.

C. Data Pre-processing. We divided collected data files into separate train and test files with 71.6 minutes data in train and 21.4 minutes data in test. To overcome the non-uniformity in number of points in each frame, we converted the point clouds into voxels of dimensions 10x32x32 (depth=10) which makes the input of constant size irrespective points of the number of points in the frame. We decided these dimensions empirically by testing their performance. In our voxel representation, the value of each voxel is the number of data points present within its boundaries. While having large number of voxels may represent underlying information well, it increases the data size by several orders of magnitudes.

Since activities are performed over a period of time, the time window from activities are generated in-order to capture the temporal dependencies. We create windows of 2 seconds (60 frames) having a sliding factor of 0.33 seconds (10 frames). The 2-second window was chosen based on the previous works in human activity recognition from multimodal timeseries datasets (12) and human identification using point clouds (9). Finally, we get 12097 samples in training and 3538 samples in testing. We use 20% of the training samples for validation. In the *time window voxelized representation*, each sample has a shape of $60 * 10 * 32 * 32$.

D. Classifiers. We evaluate different classifiers on the MMActivity dataset. We train Support Vector Machine (SVM), multilayer perceptron (MLP), Long Short-term Memory (LSTM) and convolution neural network (CNN) combined with LSTM. We compare the inference capability of these classifiers on the same train and test split of MMActivity dataset. These deep learning classifiers are generally adapted in a wide range of applications. LSTM and CNN combined with LSTM architectures are inspired from (9). Next, we explain the details of classifiers (data inputs, architectures and training details).

D.1. SVM Classifier. The input to the Support Vector Machine (SVM) classifier is generated by flattening the time window voxelized representation ($60 * 10 * 32 * 32$) and then applying the Principal Component Analysis (PCA) for dimensionality reduction. We used PCA to reduce the dimensions of data from 614400 ($60 * 10 * 32 * 32$) to 6000 which explained 80% of variance in data. SVM with RBF kernel was used.

D.2. MLP Classifier. It is composed of fully-connected layers and an output layer. We flatten the time window voxel representation ($60 * 10 * 32 * 32$) of the sample to create input size of 614400 dimensions for the MLP classifier. The MLP classifier has 4 fully connected layers followed by the output layer. We use dropout layers to avoid overfitting. It has 39.35 million trainable parameters.

D.3. Bi-directional LSTM Classifier. A bi-directional LSTM layer consists of two LSTM layers operating in parallel. The input to the first layer is provided as-is whereas the input is reverse copy of the data for the the second layer. As a result, a bi-directional LSTM layer preserves the information from both the future and the past. This network consists of the Bi-Directional LSTM layer followed by the 2 fully connected layers and an output layer. The input (60*10240) to the network is created by preserving the time dimensions (60) and flattening the spatial dimensions in the samples (10*32*32). We used Bi-Directional LSTM with size of 64 and 64 hidden units. The Bi-directional LSTM classifier has 5.29 million trainable parameters.

D.4. Time-distributed CNN + Bi-directional LSTM Classifier. Time-distributed CNN applies CNN layers to every temporal slice of the input data. The architecture of Time-distributed CNN + Bi-directional LSTM classifier consists of 3 time distributed convolutional modules (convolution layer + convolution layer + maxpooling layer) followed by the bi-directional LSTM layer and an output layer. Overall the network has 291k trainable parameters. This classifier is directly trained on the the input sample with its time and spatial dimensions.

Training and implementation. The classifiers were implemented using Sklearn and Keras. We use GridSearchCV function from sklearn to optimize the hyperparameters (C and gamma) of the SVM. Adam optimizer with a learning rate of 0.001 was used to train deep learning classifiers. The models with minimum loss on the validation data was saved after training for 30 epochs.

S.No	Classifier	Accuracy
1	SVM	63.74
2	MLP	80.34
3	Bi-directional LSTM	88.42
4	Time-distributed CNN+ Bi-directional LSTM	90.47

Table 2. Test accuracy of different activity recognition classifiers trained on the MMActivity Dataset.

We now evaluate the aforementioned approaches.

4. Evaluation

As shown in Table 2, the classifiers trained for the human activity recognition have different performance. The table reports average results of 5 different training sessions. The SVM classifier has poor performance with test accuracy of 63.74%. One reason might be the input to the SVM is not using the domain specific feature extraction approaches as used by Kim et al.(13) where they use a Doppler radar (2.4 GHz) and then convert the output into micro-Doppler signatures. All the three deep learning classifier are working directly on the time window voxel data. MLP classifier consists of the fully connected layers which doesn't assume anything about the input data and has test accuracy of 80.34%. Bi-directional LSTM classifier tries to learn the sequence of input data. The input data to the LSTM

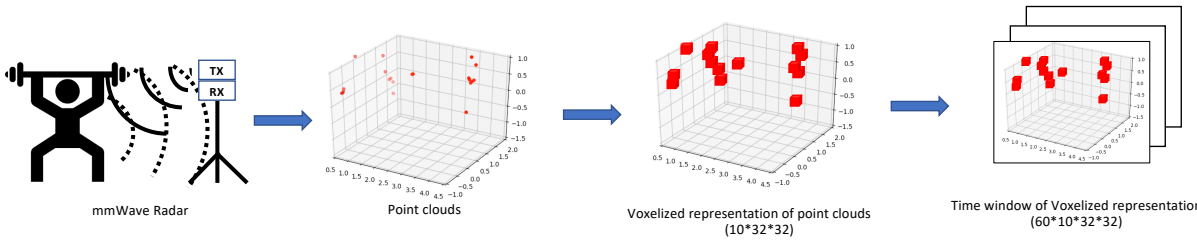

Fig. 4. Workflow of data preprocessing. The voxel size if 10*32*32. The time windows are generated by grouping 60 frames (2 second) together.

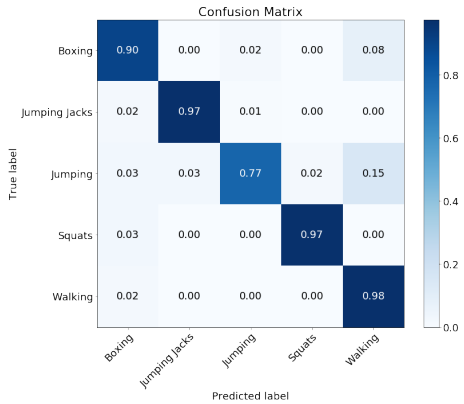

Fig. 5. Confusion matrix of time-distributed CNN + bi-directional LSTM classifier.

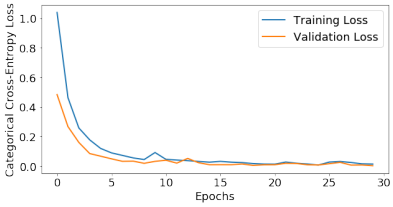

Fig. 6. Variation of training and validation loss of Time-distributed CNN + Bi-Directional LSTM

preserve the time component. Since human activities are performed over a duration and due to preserving time sequence for input to LSTM, the LSTM performance is significantly better then the MLP with test accuracy of 88.42%. The best performing classifier is Time-distributed CNN + Bi-directional LSTM which has test accuracy of 90.47%. Time-distributed CNN layers learn the spatial features from the data, as the point clouds are spatially distributed and the Bi-directional LSTM layers learn the time dependency for the activity windows. Our evaluation shows specialized spatial and temporal layers in deep learning architectures can result in boost in accuracy. The confusion matrix for one of the trained Time-distributed CNN+ Bi-directional LSTM classifier is shown in Figure 5. As seen from the figure, the activity of jumping and boxing is confused with the walking. The reason might be the similarities in the data for these activities. The variation of the training loss and the validation loss for Time-distributed CNN + Bi-directional LSTM classifier with the training epochs is shown in Figure 6.

Voxelized representation with velocities. All the evaluation presented above used the voxel representation, where the value of each voxel is the number of data points present within its boundaries. We also evaluated with the voxelized representation where the value of each voxel is the sum of the velocity of all the points present within its boundaries. The Time-distributed CNN + Bi-directional LSTM classifier trained on velocity voxel representation also had the similar performance as shown in Table 2. We now discuss the works directly related to the framework.

5. Related Work

Human activity recognition is widely explored using various sensing modalities. Researchers have used sensors like cameras and inertial measuring units (12, 14, 15), sound (12, 16) and WiFi (6). However, optical sensors like cameras capture a significantly large amount of information and sensors like inertial measuring units need to be present on the user body.

Micro-Doppler spectrograms using radars for human activity recognition have been studied in detail over the last decade (7, 13, 17). In (13), the authors use a Doppler radar to collect data of 12 subjects performing 7 different activities. They create micro-Doppler spectrograms from this data and extract 6 features from it to train an SVM classifier. Unlike our work, the radar used here is in the S-band (2-4 GHz).

Recently, researchers have exploited low-cost single-chip mmWave radar systems for person identification and tracking (9) and human activity recognition (8). In (8), the authors convert the point cloud data into micro-Doppler spectrograms before using a CNN to classify it. In (9), the authors use voxelized representation of point clouds for human identification using a LSTM and a CNN + LSTM classifier. Our deep learning classifier architectures are inspired from (9), however, we are targeting a different problem of human activity recognition.

In this work, we show that the time window voxel representation of the sparse points clouds can be used for human activity recognition. We evaluate multiple classifiers and achieve test accuracies as high as 90% percent for the deep learning classifier based on the convolutional layers and long-short term memory layer. Our evaluation shows that deep learning approaches can achieve comparable performance to the previous domain specific feature extraction approaches like used by Kim et al.(13).

6. Discussion and Future Research

We now discuss the limitations of our approach and enumerate future research directions.

Spatial and temporal dependencies in point clouds. As shown in Table 2, MLP classifier has poor performance. The reason might be due to fact the fully connected layers in MLP classifier makes no spatial and temporal assumption about the data. On the other hand, Time-distributed CNN + Bi-directional LSTM classifier assumes spatial and temporal dependency in the data and hence performs better.

Limitations of voxelized representation. Voxels result in significant increase in the required memory and computa-

tion. This can be seen in dimensionality of each input sample (60*10*32*32 = 614400), which has to be processed by the deep learning classifier. This begs the need for neural networks which are trainable directly on point clouds. One such network is the PointNet (18) which can be used for applications like object classification, part segmentation and scene semantic parsing.

7. Conclusion

We presented framework for HAR using the time window voxel representation of sparse mmWave radar point clouds. Our evaluation of the classifier shows that deep learning classifiers can be directly trained on the time window voxel representation and can achieve test accuracy greater than 90%. The classical machine learning approaches require domain specific feature extraction and shows poor performance on the voxels. Deep learning classifiers are able to learn the feature extraction transformation by directly training on the voxels. The classifiers which are designed to handle the spatial and temporal dependencies in data perform better the fully connected deep learning classifier.

We also presented a new MMActivity dataset of point clouds along with source code and pre-trained models which are available open-source.

The research reported in this paper was sponsored in part by the National Science Foundation (NSF) under award #CNS-1329755, by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and by the Army Research Laboratory (ARL) under Cooperative Agreement W911NF-17-2-0196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ARL, DARPA, NSF, SRC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

1. Attal F, et al. (2015) Physical human activity recognition using wearable sensors. *Sensors* 15(12):31314–31338.
2. Ni B, Wang G, Moulin P (2011) Rgbd-hudaact: A color-depth video database for human daily activity recognition in *2011 IEEE international conference on computer vision workshops (ICCV workshops)*. (IEEE), pp. 1147–1153.
3. Shen C, Ho BJ, Srivastava M (2017) Milift: Efficient smartwatch-based workout tracking using automatic segmentation. *IEEE Transactions on Mobile Computing* 17(7):1609–1622.
4. Hu J, Lewis FL, Gan OP, Phua GH, Aw LL (2014) Discrete-event shop-floor monitoring system in rfid-enabled manufacturing. *IEEE Transactions on Industrial Electronics* 61(12):7083–7091.
5. Bonifield J (2019) Cameras secretly recorded women in california hospital delivery rooms.
6. Wang Y, Wu K, Ni LM (2016) Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing* 16(2):581–594.
7. Çağlıyan B, Gürbüz SZ (2015) Micro-doppler-based human activity classification using the mote-scale bumblebee radar. *IEEE Geoscience and Remote Sensing Letters* 12(10):2135–2139.
8. Zhang R, Cao S (2018) Real-time human motion behavior detection via cnn using mmwave radar. *IEEE Sensors Letters* 3(2):1–4.
9. Zhao P, et al. (2019) mid: Tracking and identifying people with millimeter wave radar in *International Conference on Distributed Computing in Sensor Systems (DCOSS)*.
10. Instruments T (2019) Iwr1443 single-chip 76-ghz to 81-ghz mmwave sensor evaluation module iwr1443boost (active). Accessed: 2019-07-05.
11. Instruments T (2018) Iwr1443boost evaluation module user's guide. *http://www.ti.com/lit/ug/swru518c/swru518c.pdf*. Accessed: 2019-07-05.
12. Xing T, Sandha SS, Balaji B, Chakraborty S, Srivastava M (2018) Enabling edge devices that learn from each other: Cross modal training for activity recognition in *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*. (ACM), pp. 37–42.
13. Kim Y, Ling H (2009) Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing* 47(5):1328–1337.
14. Chen C, Jafari R, Kehtarnavaz N (2015) Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor in *2015 IEEE International conference on image processing (ICIP)*. (IEEE), pp. 168–172.
15. Sun X, Qiu L, Wu Y, Cao G (2017) Actdetector: Detecting daily activities using smartwatches in *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. (IEEE), pp. 1–9.
16. Zhan Y, Kuroda T (2014) Wearable sensor-based human activity recognition from environmental background sounds. *Journal of Ambient Intelligence and Humanized Computing* 5(1):77–89.
17. Fairchild DP, Narayanan RM (2016) Multistatic micro-doppler radar for determining target orientation and activity classification. *IEEE Transactions on Aerospace and Electronic Systems* 52(1):512–521.
18. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: Deep learning on point sets for 3d classification and segmentation in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 652–660.