Temporal convolutional networks for multi-person activity recognition using a 2D LIDAR

Fei Luo, Stefan Poslad, and Eliane Bodanese

***Abstract*—Motion trajectories contain rich information about human activities. We propose to use a 2D LIDAR to perform multiple people activity recognition simultaneously by classifying their trajectories. We clustered raw LIDAR data and classified the clusters into human and non-human classes in order to recognize humans in a scenario. For the clusters of humans, we implemented the Kalman Filter to track their trajectories which are further segmented and labelled with corresponding activities. We introduced spatial transformation and Gaussian noise for trajectory augmentation in order to overcome the problem of unbalanced classes and boost the performance of human activity recognition (HAR). Finally, we built two neural networks including a long short-term memory (LSTM) network and a temporal convolutional network (TCN) to classify trajectory samples into 15 activity classes collected from a kitchen. The proposed TCN achieved the best result of 99.49% in overall accuracy. In comparison, the TCN is slightly superior to the LSTM network. Both the TCN and the LSTM network outperform hidden Markov Model (HMM), dynamic time warping (DTW), and support vector machine (SVM) with a wide margin. Our approach achieves a higher activity recognition accuracy than the related work.**

***Index Terms*—Human activity recognition, trajectory classification, long short-term memory (LSTM), temporal convolutional network (TCN), localization, people tracking**

I. Introduction

HUMAN activity recognition (HAR) has a wide range of applications in surveillance, healthcare, smart home, intelligent control, etc. For example, doctors and dietitians can provide advice remotely for patients and customers based on their daily dietary activities [1]. People can use gesture recognition [2], [3] to contactlessly interact with electronic devices such as TV, computers, smart glasses, etc. Activity recognition can be used to detect abnormal behaviour patterns in many surveillance tasks [4], for instance, human falling detection can be very helpful in providing immediate medical assistance for the elderly [5].

A range of sensors have been applied for human activity recognition. Video-based activity recognition has been well developed. In [6], the authors extracted human skeletal outlines from videos to perform activity identification and gesture recognition. In [7], the authors derived a temporal representation of person-level actions from sport videos and combined the representation of individual people to recognize group activities. However, video-based systems can be affected by insufficient illumination and privacy concerns. Wearable inertial sensors, including accelerometer, gyroscope, and magnetometer have increasingly been applied to HAR as they are fairly ubiquitous, embedded in smart phones, smartwatches and sport bracelets. In [8], the authors used air-pressure sensors to measure the change of the air pressure resulted by muscular activities in order to recognize gestures. Heterogeneous wearable sensors have been fused in order to recognize complex human activities with more accurate results. In [9], the authors proposed a two-layer recognition framework to classify gym physical activities by fusing accelerometers and electrocardiograms. However, users may forget to wear wearable sensors [10] or feel uncomfortable to wear them [11]. Consequently, wireless sensing has been increasingly used instead. For example, Wi-Fi [12] and Radar [13] also have been used in HAR. In [14], the authors used Wi-Fi channel state information (CSI) to perform fall detection. In [15], the authors used the range-time-frequency information obtained from a Doppler Radar to perform subject localization and heartbeat detection. Compared to cameras and wearable sensors, Wi-Fi and Radar-based techniques are more suitable for HAR indoors, which has less interference than outdoors.

Location acquisition systems have been widely applied in our life and several techniques have been used, such as GPS, Radar, LIDAR, etc. These systems generate a large amount of location data, and in the past years, they have been applied in navigation, route planning, and mapping. Recently, more and more studies attempt to infer semantic information from the trajectories formed by the locations of people, in order to provide a higher level of services, for example, for personalized recommendation, smarter home or more intelligent human-like robot interaction. Although there are many localization techniques, none of these can be applied ubiquitously for all purposes. GPS hardly works indoors as its signals cannot penetrate walls. Radars may suffer interference from the ground surrounding and multi-path effects. Bluetooth and RFID (Radio-Frequency Identification) can perform indoor localization, however, due to their short-range sensing capability, they generally require larger deployments that may become difficult to maintain. As a surveying and mapping technique, LIDAR has been increasingly used in self-driving vehicles and robots due to its high localization accuracy, good real-time performance, and easy deployment. LIDAR also can be applied both indoors and outdoors. As 2D LIDARs are more affordable than 3D LIDARs, this paper proposes to use a 2D LIDAR to perform multiple people activity recognition indoors.

A trajectory collected by using a LIDAR is a sequence of coordinates. A trajectory is a time series that contains both

Fei Luo is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK. e-mail: f.luo@qmul.ac.uk.

This research is funded by a QMUL-CSC scholarship.

The author sequence of this paper follows the "first-last-author-emphasis" norm (FLAE).

spatial and temporal information. Recurrent neural networks (RNNs) are able to model long-term dependencies in time series by propagating information through their deterministic hidden state. As a variant of RNN, LSTM further uses three gates to regulate the flow of information in and out of each node, which overcomes the exploding and vanishing gradient problem that exists in conventional RNNs. TCNs leverage large receptive fields by stacking many dilated convolutions, allowing them to model even longer time scales up to the entire sequence length [16]. Both LSTM and TCNs have achieved outstanding results in many areas, such as sequential image classification, audio classification, language modelling, etc. In this paper, we built an LSTM network and a TCN to classify human trajectories into predefined activities and made a comparison between them.

It is more challenging to perform human activity recognition using wireless sensing techniques than using wearable sensors. For wearable sensor-based HAR, each person wears sensor devices, the sensor data is collected from each specific person. However, wireless sensing techniques including LIDAR, Radar, and Wi-Fi monitor a scenario that contains both humans and non-human objects such as walls, tables, animals, etc. It is necessary to differentiate humans from non-human objects before performing HAR. Another challenge is that multiple people can appear in the detection area of wireless sensors. For recognizing the activities of different people, it is necessary to track each person separately. LIDAR can achieve centimeter-level localization accuracy, by using this characteristic of LIDARs, it is possible to separate multiple people in the spatial dimension. LIDARs also have a very fast response time that makes them suitable for real-time tracking. These reasons drove us to choose a 2D LIDAR to perform human activity recognition.

In this paper, we used a 2D LIDAR to track multiple people simultaneously and collect their trajectories for activity recognition. By using a very short sliding window (2.5s) for trajectory segmentation, our approach is able to achieve nearly real-time human activity recognition. We applied spatial transformation and Gaussian noise for trajectory augmentation in order to overcome the problem of unbalanced classes. Finally, we built two neural networks, an LSTM network and a TCN, and compared their performance. Our proposed approach is able to deliver a very high classification accuracy of indoor activities in nearly real-time, which is essential to make feasible real-time monitoring applications.

The remainder of this paper is organized as follows. Section II presents the related work for human activity recognition from three aspects: sensors, machine learning, trajectory-based HAR. Section III illustrates the detailed methodologies including data processing, point clustering, multi-user tracking, trajectory segmentation and augmentation, and our proposed LSTM and TCN networks. In Section IV, we implement our method in an indoor scenario. We evaluate our proposed approach and present the results in Section V. Finally, we draw conclusions in Section VI.

II. Related work

HAR has been widely researched by using a diversity of sensing devices and methods. In this section, we will review some related work with respect to types of sensors, machine learning algorithms, and trajectory-based HAR.

A. Sensors

A variety of sensors have been applied in HAR, such as cameras, RFIDs, Wi-Fi, etc. Two kinds of techniques emerged from these types of sensors: Device-bound and Device-free techniques. Device-bound techniques require users to carry wearable sensors, such as electrocardiogram sensors, accelerometers, Bluetooth or Wi-Fi receivers. As many wearable sensors are attached on human body, they have great advantages in acquiring physiological characteristics [17] including body temperature, heart rate, and blood pressure, which are helpful for healthcare monitoring. Device-free techniques do not require users to wear any sensor. The sensing of activities is performed remotely by using, for example, cameras, Radars, LIDARs, etc. Device-free techniques have been attracting more and more attention as they are more convenient and require no effort or attachments from users. However, each sensor has its own advantages and limitations. Cameras are widely deployed for HAR because videos and images are visual forms that can easily be perceived by our eyes. However, most cameras suffer from insufficient illumination and have a narrow visual range that makes them unsuitable for larger areas. It is also privacy intrusive because people's faces may be exposed. RFID tags [18] and Bluetooth beacons [19] usually require a large deployment and maintenance because they are very short-range. They may need users to carry a RFID reader or a Bluetooth receiver to read the tags or beacons embedded in specific parts of the physical environment. Radars can provide micro-Doppler signatures generated from human activities [13]. Radars do not generate privacy concerns because personal identifiable features cannot be obtained with Radar detection, and they can penetrate walls, cloth, trees, etc. Wi-Fi devices can measure the CSI or the Received Signal Strength Indicator (RSSI) to perform activity recognition. They have similar merits as Radar devices. However, both Wi-Fi and Radar devices may present a weak anti-interference capability and suffer from multi-path effects. GPS and LIDAR have been used in human activity recognition through the analysis of patterns of location sequences of humans. As GPS is hardly received indoors, it is usually used outdoors within large and open spaces. LIDAR is a surveying method that measures the distance to a target by illuminating the target with pulsed laser light and measuring the reflected pulses. It has been widely applied for object localization, tracking, and mapping. Currently, few researchers have applied LIDAR to HAR due to its expensive cost.

B. Machine learning algorithms

Machine learning has been widely applied in human activity recognition to classify different activities. DTW is one of the most used algorithms in human activity recognition [20], [21].

It measures the similarity between two temporal sequences to find the optimum distance between time series [22]. As almost all data captured by sensors in HAR are time series, DTW has innate advantages in processing them. HMM is also favored in HAR [23], [24]. It is a Markov chain with both hidden and observable stochastic processes. In human activity recognition, observable components are the sensor signals while hidden elements are users' activities [25]. Other algorithms, such as SVM [26], kNN (*k*-nearest neighbour) [27], and Random Forest [28] also have been applied in HAR. Since deep learning has achieved very high performance in many fields, it is also increasingly applied into HAR. The authors in [29] built a Convolutional Neural Network (CNN) to classify RGB-D dataset in order to recognize human actions. Tang et al. [30] proposed a Coherence Constrained Graph LSTM to recognize group activity by modeling the relevant motions of individuals while suppressing the irrelevant motions. Guo et al. [31] proposed a feature fusion method, multiview Cauchy estimator feature embedding, to fuse the data of Kinect and inertial sensors for human action recognition. Lea et al. [32] used TCNs to build an Encoder-Decoder model for human action segmentation and recognition upon videos. Luo et al. [1] used a CNN to classify Radar frequency spectrograms for an indoor HAR comprising of fifteen different activities. Some of the above algorithms have also been used in trajectory-based human activity recognition as presented in Section II-C.

C. Trajectory-based HAR

Most trajectory-based human activity detection are achieved by using cameras. In [33], the authors built a DTW model to cluster similar trajectories in a video surveillance system for suspicious behavior detection. In [34], the authors proposed an LSTM model to perform trajectory and activity prediction from videos. In [35], the authors proposed to detect abnormal behaviours through trajectory analysis and anomaly modeling based upon a camera sensor network. In [36], the authors proposed a hierarchical Dirichlet process hidden Markov model (HDP-HMM) to model the transitions of different motions in video sequences. However, besides the shortcomings of small visual scope and privacy intrusion, trajectories extracted from a video are heavily dependent on the azimuth and inclination of the camera, and the coordinates of trajectories are hardly integrated into a global coordinate system [4]. GPS as the most popular outdoor localization technique also has been used for trajectory-based activity recognition. In [37], the authors built a hierarchical Dirichlet process HMM upon GPS trajectories to detect anomalies of human motion in dynamic traffic control. In [38], the authors proposed an automatic activity detection method using PoIs' (Points of Interest) spatial temporal attractiveness to identify activity-locations as well as durations from raw GPS trajectories. In [39], the authors proposed a cost sensitive approach for activity recognition from GPS-logs by measuring the importance of each activity from spatial and temporal perspectives. In [40], the authors modelled human mobility behavior using GPS trajectories to predict the purpose of a user's visit at a certain location. However, GPS requires users to carry a GPS receiver and it is very difficult to receive GPS signals indoors. Currently, there is little research in trajectory-based human activity recognition using LIDAR. In [41], the authors used a 2D laser to investigate interactions between persons and detect anomalies in three different environments: open layout laboratory, corridor, and an outdoor courtyard. Ma et al. [42] built a seq2seq model to model location-driven sequences of a user in order to recognize human activity in a kitchen. They also used a 2D LIDAR. However, they did not differentiate human targets and obstacles, and the system recognized activities of only one person, which is fine for single people occupancy, but not for a family home. In our research, we further classified human and non-human targets and performed continuous tracking to obtain trajectories of multiple people.

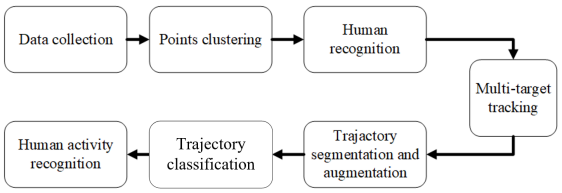


Fig. 1. Workflow of LIDAR-based human activity recognition

III. Methodology

The main steps of LIDAR-based human activity recognition are demonstrated in Fig. 1. Firstly, we used a 2D LIDAR to collect a set of points, which are reflected by walls, tables, humans, etc. This set of points was grouped into different clusters that represent different objects in a scene. We further performed human recognition upon these point clusters by using a Random Forest to classify geometric features extracted from each cluster. We used the Kalman Filter to perform continuous tracking of multiple people. Trajectories obtained from the tracking were further segmented and tagged with our predefined activity labels. Trajectory augmentation was implemented to enrich the samples and overcome the problem of unbalanced classes. Finally, we built an LSTM network and a TCN to perform trajectory classification. After training, both two networks are able to perform human activity recognition online. The details of each step are described in the following subsections.

A. Point clustering

The raw data of a LIDAR is a sequence of polar coordinates and each coordinate is noted as (r, θ) where r is the distance to LIDAR and θ is the angle. For facilitating subsequent processing, we transformed these coordinates from a polar coordinate system to a plane coordinate system. Then the data at each timestep can be described as a sequence $S = \{p_1, p_2, ..., p_n\}$, where n is the total number of points.

In order to differentiate objects, it is necessary to perform clustering to group the points in S into different clusters. We applied an algorithm called density-based spatial clustering algorithm (DBSCAN) to perform the clustering. DBSCAN is based on a threshold for a number of neighbors, $minPts$,

within a radius ε. A point with the neighbor count greater than or equal to $minPts$ within ε, is identified as a core point. A border point has the number of neighbors that is less than $minPts$ but it belongs to the ε-neighborhood of other points. If a point is neither a core nor a border point, then it is called a noise point. In order to understand how DBSCAN works, three terms are defined:

Direct density reachable: a point A is directly density reachable from another point B if A is in the ε-the neighborhood of B and B is a core point.
Density reachable: a point A is density reachable from B if there are a set of core points leading from B to A.
Density connected: two points A and B are density connected if there is a core point C, such that both A and B are density reachable from C.

The DBSCAN can be abstracted into the following steps:

1) Find all neighbor points within ε of every point, and each point with more than $minPts$ neighbors within ε are marked as a core point.
2) For each core point if it is not already assigned to a cluster, create a new cluster. Find recursively all its density connected points and assign them to the same cluster as the core point.
3) Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are marked as noise.

Compared to other clustering algorithms, DBSCAN does not need to specify the number of clusters, it is able to discover arbitrarily shaped clusters, and it is robust to noise [43].

B. Human recognition

The clusters obtained after point clustering represent many different objects. It is necessary to recognize which cluster represents human in order to track its trajectory. Clusters that represent humans have different geometric characteristics from clusters of non-human targets. Geometric feature can be defined in terms of linearity, circularity, angularity, etc. As Angus et al. did in [44], we extracted 15 geometric features (shown in Fig. 2) from each cluster. Based on these features, we implemented a Random Forest (RF) to classify human and non-human targets. RF is a tree-based ensemble classifier which is a meta estimator that fits several decision trees on various subcategories of data. This algorithm is used for both regression and classification. In general, the number of trees determines the robustness of the forest. We collected positive samples (human) from a clear and open space and negative samples (non-human) from various indoor spaces without people. The total number of samples was 12000 consisting of 6000 positive samples and 6000 negative samples. One benefit of random forest is that it can measure the confidence level in the classification by aggregating the classification output of each individual tree. Thus, rather than just using the predicted labels, we considered the confidence level generated from the RF classifier to eliminate the noise and increase reliability.

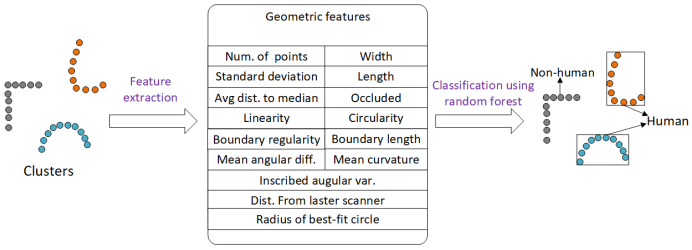


Fig. 2. Human recognition

C. Multiple people tracking

In order to obtain the trajectories of human clusters, it is necessary to perform continuous target tracking. Kalman filter is one of the most popular tracking algorithms due to its efficiency and accuracy [45]. It is an optimal recursive data processing, which uses a series of measurements observed over time to produce an estimate of the desired variables and finds the optimal state with the smallest possible variance error. It contains two steps: prediction and correction. In the prediction step, the state is predicted with the dynamic model, while in the correction step, the state is corrected with the observation model.

The process and measurement equations for the Kalman filter are given as follows:

$$\begin{aligned} x_k &= Ax_{k-1} + Bu_k + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \tag{1}$$

where k is the discrete time, x_k is the state vector, z_k is the observation vector, A and H are the transition matrix and the observation matrix respectively. B_k is the control-input model which is applied to the control vector u_k. w_{k-1} and v_k are Gaussian random variables with zero mean, so their probability distributions are $p(w) \sim N(0, Q), p(v) \sim N(0, R)$ where the covariance matrix Q and R are referred to as transition noise covariance matrix and observation noise covariance matrix.

The prediction stage for the Kalman filter is as follows:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{2}$$

a priori estimate of state $\hat{x}_k^-$ and covariance error P_k^- is obtained for the next time step k

The correction stage for the Kalman filter is as follows:

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (1 - K_k H)P_k^- \end{aligned} \tag{3}$$

K_k is the Kalman gain which is computed by above equations. After that a posterior state estimate x_k^- and a posterior error estimate P_k is computed by the measurement z_k. The prediction and correction equations are calculated recursively with the previous posterior estimates to predict new prior estimates.

In a tracking system, the state vector is $X = [x, y, v_x, v_y, a_x, a_y]^T$, where (x, y), (v_x, v_y) and (a_x, a_y) represent position, velocity and acceleration, respectively. The observation vector is $Z = (x', y')$.

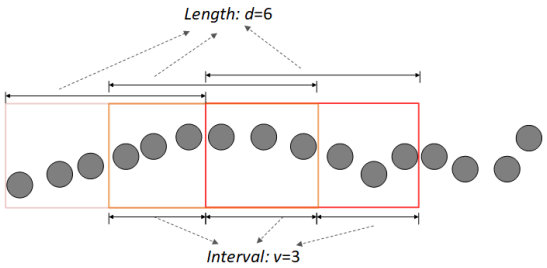


Fig. 3. Trajectory segmentation

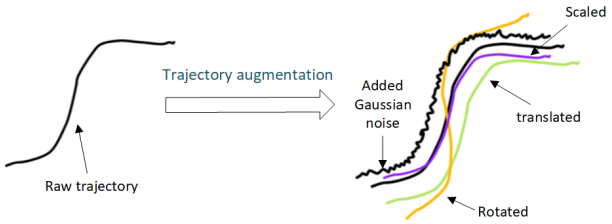


Fig. 4. Trajectory augmentation

D. Trajectory segmentation and augmentation

A trajectory can be the result of the location displacement when one or a sequence of activities is performed. For differentiating activities, we split a trajectory into segments that each segment represents one activity. Several ways have been used to perform trajectory segmentation, such as distance-based segmentation, time-based segmentation, and number of points-based segmentation. They split a trajectory into segments with the same distance, time, and number of points, respectively. In our research, we implemented the number of points-based segmentation. We used a sliding window with the length of d to perform segmentation at the interval of v. For instance, Fig. 3 shows a sliding window with the length of 6 (location points) and the sliding interval of 3 (location points).

In our daily life, the frequency and duration of different activities are different. For example, the activity of 'having a meal' are mainly performed 3 times a day but the activity like 'washing hands' can be performed many times, while the time consumed by 'having a meal' is much longer than 'washing hands'. This leads the problem of data sparsity that some activities have many samples but others have few. To overcome this problem, we applied trajectory augmentation by using spatial transformation. In machine learning, data augmentation is used to create more training samples through different ways of processing or a combination of multiple processing upon the original samples. Data augmentation can increase the number of samples and boost the performance of deep learning. In this paper, we applied spatial transformation upon the raw trajectories to perform trajectory augmentation because the trajectories are spatial data. The methods are described as follows:

1) Translation: all points are translated to new positions by adding offsets T_x and T_y to x and y, respectively.

$$[x^{'} \quad y^{'} \quad 1] = [x \quad y \quad 1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} \tag{4}$$

2) Rotation: all points in the 2D plane are rotated around the origin through the counterclockwise angle θ.

$$[x^{'} \quad y^{'} \quad 1] = [x \quad y \quad 1] \cdot \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

3) Scale: all points are scaled by applying the scale factors S_x and S_y to the x and y coordinates,respectively.

$$[x^{'} \quad y^{'} \quad 1] = [x \quad y \quad 1] \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

In addition to spatial transformation, we also added Gaussian noise to the raw trajectories. The process of trajectory augmentation is shown in Fig. 4. It is worthy to note that it is required to control the range of spatial transformation to avoid changing the semantic information of trajectories.

E. Trajectory classification

1) Long short-term memory (LSTM): LSTM is a variant of RNNs that have been successfully applied in sequential learning. In RNNs, the gradient increase rapidly or decays exponentially over time. For overcoming this 'exploding and vanishing gradient' problem, LSTM uses a few gates to control the passing of the information along the sequence and thus can learn long-range dependencies. A typical LSTM unit contains an input gate i_t, a forget gate f_t, a cell c_t, an output gate o_t and an output response h_t. The input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. The recursive computation of an LSTM unit is

$$\begin{aligned} i_t &= \delta(W_{xi}x_{x_t} + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \\ f_t &= \delta(W_{xf}x_{x_t} + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\ o_t &= \delta(W_{xo}x_{x_t} + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o), \\ h_t &= \delta \odot tanh(c_t) \end{aligned} \tag{7}$$

where $\odot$ denotes element-wise product, $\delta(x)$ is the sigmoid function defined as $\delta(x) = 1/(1 + e^{-x})$, $W_{\alpha\beta}$ is the weight matrix between α and β (e.g., W_{xi} is the weight matrix from the input x_t to the input gates i_t), and b_β denotes the bias term of β with $\beta \in i, f, c, o$.

As shown in Fig. 5, we built an LSTM network that consists of two LSTM layers and one fully-connected layer. Segmented trajectories were input into the network. As each point in a trajectory has two values (x, y), the input shape of the LSTM network is $(t, 2)$ where t is the length of a trajectory. Both two LSTM layers have 120 units and the fully-connected layer has 200 hidden units. The size of the output layer is determined by the number of activities that we are going to recognize.

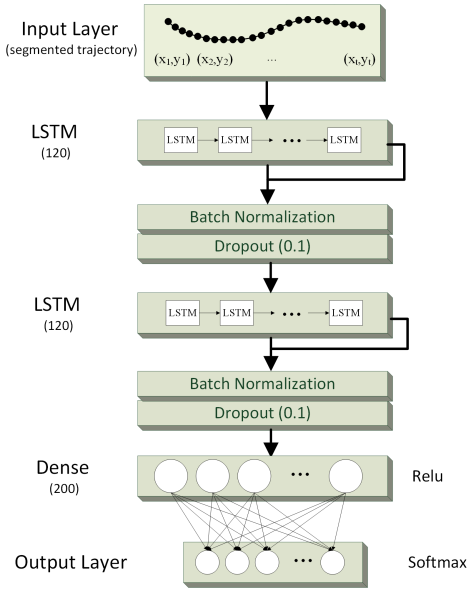


Fig. 5. The structure of our proposed LSTM network

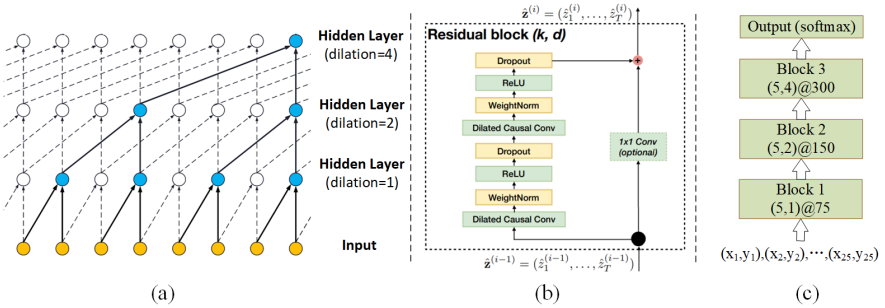


Fig. 6. The architecture of our proposed TCN

2) Temporal Convolutional Network (TCN): TCN is a class of time-series neural network models that capture long-range patterns using a hierarchy of temporal convolutional filters [32]. TCNs integrate dilated casual convolutions and the residual block structure to expand the receptive field and increase the depth. In dilated casual convolutions, an output at time t is convolved only with the elements from time t and earlier in the previous layer by using dilated convolution to enable an exponentially large receptive field [46]. For a 1-D sequence input $x \in \mathbb{R}^n$ and a filter $f : 0, f : \{0, \cdots, k-1\} \rightarrow \mathbb{R}$, the dilated convolution operation F on element s of the sequence is defined as [46]

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{8}$$

where d is the dilation factor, k is the filter size, and $s - d \cdot i$ accounts for the direction of the past. With dilated casual convolutions, the receptive field of the TCN can be increased by using larger filter sizes k and increasing the dilation factor d. Fig. 6(a) presents a dilated causal convolution with dilation factors $d = 1, 2, 4$ and filter size $k = 2$.

A TCN consists of several residual blocks. Each residual block contains a series of transformations τ. As shown in 6(b), a residual block consists of two sets of layers including a dilated casual convolution layer, a weight normalization layer, a ReLU activation layer, and a dropout layer. The output of each residual block is the sum of the output of these transformations and the input x:

$$o = Activation(x + \tau(x)) \tag{9}$$

This is also called skip connection. Residual blocks avoid the vanishing gradient problem by carrying gradient throughout the extent of a very deep network.

In this work, we built a TCN that consists of three residual blocks ('Block 1', 'Block 2', and 'Block 3') as shown in Fig. 6(c). After hyperparameter tuning, the kernel sizes for these three blocks were set as 5, the dilation rates of them were 1, 2, and 4 respectively. 'Block 1' has 75 filters, 'Block 2' has 150 filters, and 'Block 3' has 300 filters.

In training process, both the LSTM network and the TCN network were trained to minimize an objective function in terms of the parameters of the network. For activity recognition, let C be the number of activities, the following cross-entropy loss function is often used:

$$E_y(y^{'}) = -\sum_{i=1}^{N} y_i \cdot log(y_i^{'}) \tag{10}$$

where E is the loss function evaluated over N samples, y_i is the original label of the i_{th} sample and $y_i^{'}$ is the class score maps of the sample i calculated using a *softmax* activation function:

$$y_j = exp(x_j)/(\sum_{c=1}^{C} exp(x_c)) \tag{11}$$

where y is the softmax score and x is the output layer containing unnormalized class scores.

In training process, we applied dropout to perform regularization that prevents overfitting. The term 'dropout' refers to dropping out a part of the units in a neural network. By avoiding training all nodes on all training data, dropout decreases overfitting. We initialized dropout with the rate of 0.1 for the LSTM network and 0.25 for the TCN. In the LSTM network, batch normalization was applied after each LSTM layer. Batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. Except for preventing overfitting, batch normalization also can accelerate the training. The optimizer used for both networks is Adam whose learning rate was initialized as 0.0003.

IV. Experimental Setup

In this section, we present the LIDAR and our experiments. The LIDAR we used is an UST-10LX as shown in Fig. 7(a). Its error in localization accuracy is within 40mm. It has a maximum detection distance of 30m and scan angle of 270°. Its angular resolution is 0.25°. More information about UST-10LX can be found in [47]. We made the experiments in a kitchen scenario located in an university campus. Activities in the kitchen are directly related to dietary health. Kitchen scene context-based activity recognition can be a useful method

(a) (b)

Fig. 7. (a) UST-10LX LIDAR, (b) Kitchen scenario

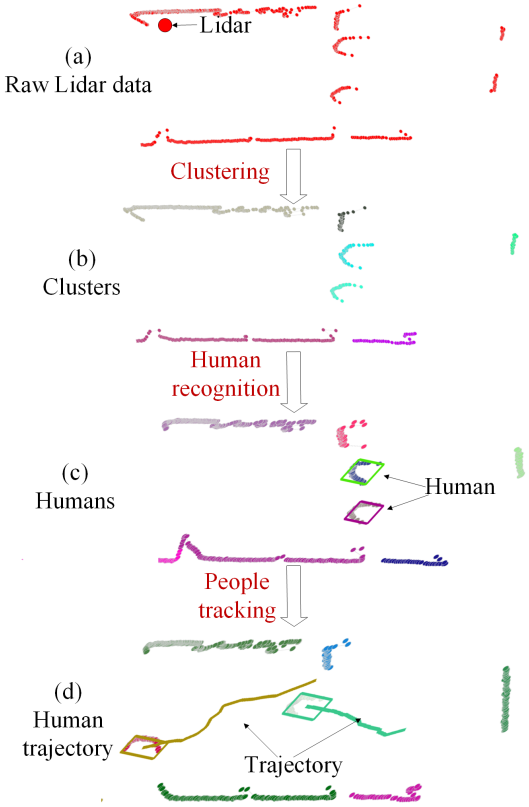


Fig. 8. LIDAR data processing

for diet controlling and dietary treatment, also helpful for developing a smart kitchen as a part of a smart home. Fig. 7(b) presents the bird‘s-eye view of the 3D model of the kitchen.

In our experiments, the LIDAR was placed at one meter high and close to the up-left corner of the kitchen. We used a sampling frequency of 10 Hz (10 scans per second) to collect data and further processed the collected data using the methods described in Section III as shown in Fig. 8. The raw Lidar data (Fig. 8(a)) consists of the points reflected from obstacles, walls, and humans. After clustering using DBSCAN, the points were clustered into different groups that are shown with different colors in Fig. 8(b). We performed human recognition by classifying 15 geometric features extracted from these clusters using Random Forest with 25 trees. Finally, we built a Kalman filter for each human target in order to obtain their trajectories.

We monitored the kitchen using the UST-10LX for a whole day and plotted a density map of human trajectories to present the spatial distribution of their daily activities. As shown in

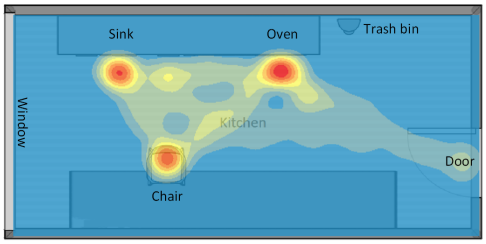


Fig. 9. Density map of human trajectory in the kitchen

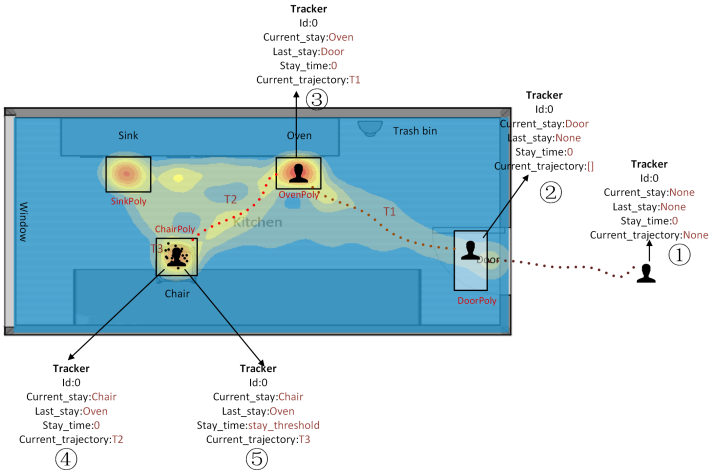


Fig. 10. Process of trajectory collection

Fig. 9, human trajectories are mainly concentrated around the sink, the oven, the chair, and the door in the kitchen. These four areas can be seen as four stay areas that people usually stop for some amount of time. For example, people usually stay around the sink to wash their cooking utensils, stay by the oven for cooking, and sit on the chair for eating. Based on this, we predefined 15 activities including: ‘get in’, ‘get out’, ‘from sink to door’, ‘from chair to sink’, ‘from oven to chair’, ‘from sink to chair’, ‘washing’, ‘from door to sink’, ‘cooking’, ‘from oven to door’, ‘sitting’, ‘from oven to sink’, ‘from chair to oven’, ‘from door to oven’, ‘from sink to oven’.

The trajectories of these activities (except ‘get in’ and ‘get out’) begin or end at the stay areas. Based on this, we defined a tracker for each person in the detection range. The properties of a tracker include ‘id’, ‘current stay’, ‘last stay’, ‘stay time’, and ‘current trajectory’ as shown in Fig. 10. The trajectory collection process for each person is:

1) Initialize a tracker for each person in the detection area and set the tracker’s ‘current stay’ and ‘last stay’ as none.
2) Update the tracker at each scanning frame of the LIDAR and judge whether the current location of the tracker is within a stay area. If it is within a certain stay area, then set its ‘current stay’ as the name of the stay area, and initialize its ’current trajectory’ as a point list to store the trajectory of the tracker.
3) When the tracker reaches a new stay area, then set its ’last stay’ as the value of ’current stay’ and its ’current stay’ as the name of this new stay area. Save its ’current trajectory’ as a trajectory labelled ‘from ‘last stay’ to

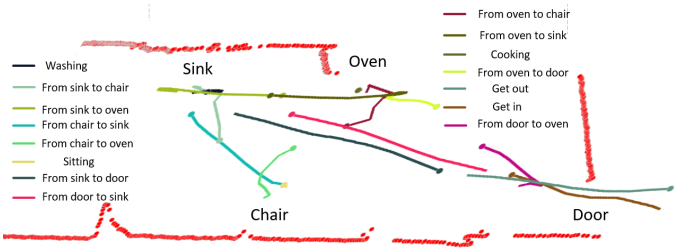


Fig. 11. Trajectory samples of human activity

'current stay", then clear its 'current trajectory' to empty for storing the next trajectory of the tracker.

4) If the tracker does not get to a new stay area but stays in the same stay area for some amount of time and the stay time is beyond the threshold (80 frames used in our work), then save its 'current trajectory' and label it with 'cooking' if the stay area is 'OvenPoly', 'sitting' if the stay area is 'ChairPoly', and 'washing' if the stay area is 'SinkPoly' as shown in Fig. 10. After this, set its 'current trajectory' as empty to store the next trajectory of the tracker.
5) Repeat step 3 and step 4 until the tracker leaves the detection range of the LIDAR.

The trajectories of 'get in' and 'get out' do not begin or end at any stay area. Therefore, we collected their trajectories manually. We started to record the trajectory of participants before they got in, and stopped to record after they got in, then saved the trajectories and labelled them as 'get in'. For 'get out', the collection process was similar.

Trajectories that begin or end at the stay areas were recorded and labelled automatically. The geometric relationship between trajectories and the stay areas assures that those trajectories were correctly labelled. For 'get in' and 'get out', they were recorded manually. Some of them may have been mislabelled but they could be filtered out by judging their directions, as the directions of 'get in' and 'get out' are opposite to each other. Besides, the collected trajectory can be visually checked.

We collected trajectories of these activities and segmented them using a sliding window with a length of 25 (location points) and the interval of 10 (location points), so each trajectory has 25 points. As the sampling frequency is 10 Hz, the length of the sliding windows also can be measured by time, in our case it is 2.5s. We choose this length because a longer trajectory may contain more than one activity and will lead to more latency, and a shorter trajectory does not have enough information related to a human activity. In the experiments, the number of people in the kitchen was varying and the maximum number of people was 4. The total time for monitoring was about 40 hours. In Fig. 11, we demonstrate one trajectory sample in our dataset for each activity with different colors. As it can be seen, the points of the trajectories of 'washing', 'cooking', and 'sitting' tend to cluster together because people do not move much when they perform these activities.

As shown in Fig. 12, the quantity of each type of activity

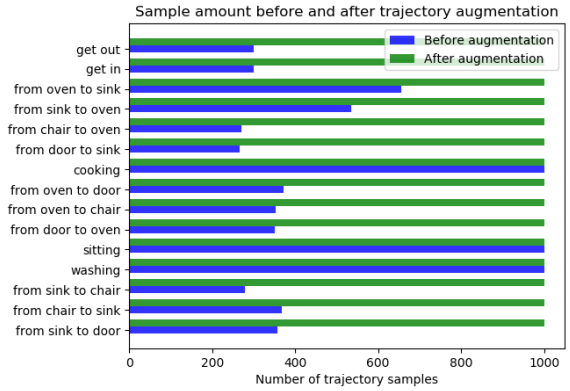


Fig. 12. Sample amount before and after trajectory augmentation

before augmentation is: 357 'from sink to door', 367 'from chair to sink', 279 'from sink to chair', 1000 'washing', 1000 'sitting', 350 'from door to oven', 353 'from oven to chair', 372 'from oven to door', 1000 'cooking', 267 'from door to sink', 270 'from chair to oven', 536 'from sink to oven', 656 'from oven to sink', 300 'get in', and 300 'get out'. Some activities (cooking, washing, sitting, etc.) have many samples while others (from sink to chair, from chair to oven, etc.) do not. We implemented trajectory augmentation to overcome this problem of unbalanced classes. After trajectory augmentation, we obtained 15000 samples in total and the number of samples per activity was 1000.

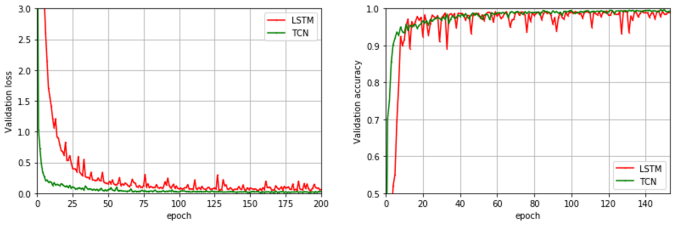
V. Evaluation and Results

In this section, we evaluated our proposed neural networks in three ways including evaluation on the test dataset, comparison with the baseline, and comparison with the related work.

A. Evaluation on the test dataset

We split the trajectory samples into two groups, 80% of the samples were used for training, and 20% for testing. During the training, we performed a hold-out validation [48], 15% of the sample data was randomly excluded from training. The testing dataset was not exposed to our networks in training. The achieved overall accuracies by the LSTM network and the TCN in testing were 99.39% and 99.49% respectively. The TCN performed slightly better than the LSTM. More importantly, the convergence speed of the TCN is significantly faster than the LSTM. As shown in Fig. 13, the lines of validation loss and accuracy of the TCN are smoother than those of the LSTM network. This is probably because TCN has a backpropagation path different from the temporal direction of the sequence that is beneficial to avoid the problem of exploding/vanishing gradients [46].

As shown in the normalized confusion matrix of Fig. 14, most of the activities are correctly classified with 100% or 99%. For 'from chair to sink', 1% samples have been misclassified into 'from chair to oven' and 1% samples have been misclassified into 'sitting'. It is because some trajectory samples from these classes begin from the chair in the kitchen and have some overlaps. The same situation also happened



(a) Validation loss (b) Validation accuracy

Fig. 13. Validation loss and accuracy of the proposed LSTM and TCN

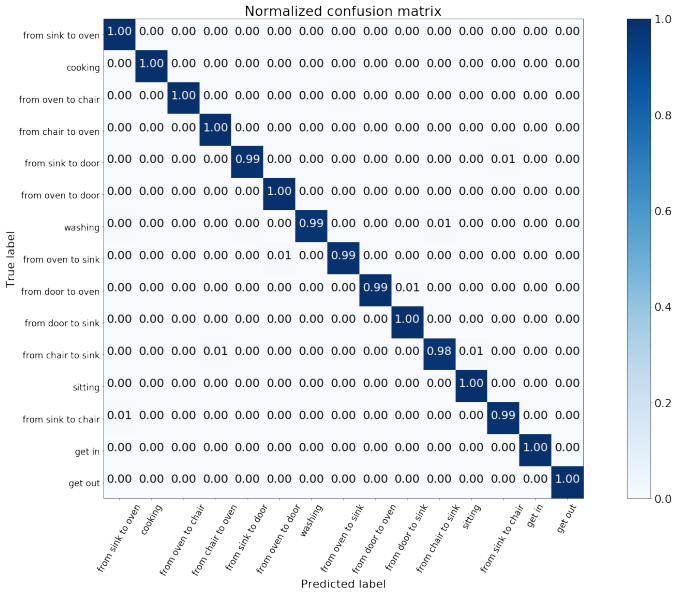


Fig. 14. Normalized confusion matrix of the TCN in testing

between 'from door to oven' and 'from door to sink'. 1% samples of 'from door to oven' have been misclassified into 'from door to sink'. Also, 1% samples of 'from sink to door' have been misclassified into 'from sink to chair'.

For validating the effect of trajectory augmentation, we compared the networks that trained on the samples after augmentation and those trained on the raw samples. As shown in Table I, after applying our proposed trajectory augmentation, the performance of the LSTM network has been increased about 1.7% in OA, Recall, and F1; and the performance of the TCN has been increased about 1.5% in all three metrics.

TABLE I
THE EFFECT OF TRAJECTORY AUGMENTATION

	OA	Recall	F1
TCN with trajectory augmentation	99.49%	99.53%	99.51%
TCN without trajectory augmentation	97.96%	97.93%	97.96%
LSTM with trajectory augmentation	99.39%	99.41%	99.39%
LSTM without trajectory augmentation	97.68%	97.79%	97.65%

B. Comparison with the baseline

As mentioned in Section II, both HMM and DTW have been frequently applied to trajectory-based activity recognition in the related work. SVM is a widely used method in classification problems. We used these three algorithms as the baseline to make a comparison with our networks.

The SVM implemented in our work is an RBF (Radial Basis Function) SVM. It has two hyperparameters $(C, gamma)$. Cross-validation was used to tune the hyperparameters. Given a hyperparameter space $C : [1, 30]$, $gamma : [0.1, 1.0E - 5]$, a pair of parameters was selected from the hyperparameter space by the cross-validation in each training and validation iteration. The optimal parameters selected for SVM in this work were $C : 4.8$, $gamma : 10E - 4.5$.

The hyperparameter of an HMM is the number of hidden components. We searched for the number of hidden components from 0 to 20. It was found that 5 is the optimal number of hidden components.

DTW usually works with kNN to perform classification. The implementation in our work combined kNN and DTW by replacing the Euclidean distance measurement with the DTW distance measurement. The hyperparameter of kNN was the number of neighbours k and that of DTW was the *max warping window*. After tuning with cross-validation, the optimal hyperparameters were k : 1, *max warping window*: 18.

As shown in Table II, both the LSTM network and the TCN are superior to these three algorithms in all three metrics, The TCN achieved the best results: 99.49% in OA, 99.53% in Recall, and 99.51% in F1.

TABLE II
COMPARISON OF THE CLASSIFICATION MODELS

	OA	Recall	F1
TCN	99.49%	99.53%	99.51%
LSTM	99.39%	99.41%	99.39%
SVM	95.88%	95.76%	96.69%
HMM	85%	86.7%	85.5%
DTW	90.9%	91.56%	91.71%

C. Comparison with the related work

In [42], the authors also used a 2D LIDAR to perform human activity recognition. They developed a seq2seq model to perform the classification of 17 activities and achieved 88% overall accuracy. We cannot perform a completely direct comparison between the work in [42] and ours. The layout of the kitchen and the LIDAR used in [42] are different from ours. The structure of the seq2seq model implemented in [42] is unclear as the way of trajectory collection and the number of their trajectory samples. However, we still can argue that our approach is able to perform multiple people tracking and it achieved very good accuracy in nearly real-time human activity recognition. The work in [42] did not consider the interference from non-human objects and it can only be used to recognize the activities of a single person. A trajectory can be generated by several activities. In [42], it is not clear how the authors decided the number of activities that the seq2seq model output. It is more suitable to represent each activity by using a small trajectory segment, which also can improve real-time capability in activity recognition.

D. Analysis

From above evaluations, it can be seen that deep learning has great performance in trajectory-based activity classifications in the comparison with traditional algorithms. Trajectory

augmentation is an effective way to improve the classification accuracy when the number of trajectories is small and imbalanced. The TCN performed better than the LSTM in trajectory classification. Because the length of the trajectory samples is only 25 (point), the capability of TCNs in learning long-range temporal patterns was not fully released. The high accuracy achieved in our work infers that indoor trajectory-based human activity recognition is applicable. Of course, it also has relationship with the layout of indoor scenarios. For a more complex indoor layout, the performance of activity recognition probably will decline.

VI. Conclusion and Future Works

In this paper, we proposed to perform human activity recognition by using a 2D LIDAR. We used the DBSCAN to cluster the LIDAR points and perform human and non-human classification based upon 15 geometric features of the clusters. We further applied the Kalman filter to perform multiple people tracking and obtain their trajectories. We used a sliding window to perform trajectory segmentation. For overcoming the problem of unbalanced classes, we proposed trajectory augmentation by using spatial transformation and adding Gaussian noise. Finally, we built two networks (an LSTM and a TCN) to perform HAR upon the trajectory samples. Our proposed TCN achieved a very good result of 99.49% overall accuracy, which is slightly superior to the LSTM and much higher than the results achieved by SVM, HMM, and DTW. We compared our work to state-of-the-art approach for human activity recognition using LIDAR. Our proposed networks outperform the state-of-the-art approach.

With the coming age of Internet of things, various electronic sensors are being connected wirelessly. One future research direction is on fusing the data acquired from different sensors in order to achieve more efficient and accurate human activity recognition [49]. In modern life, people switch between different scenarios such as homes, offices, shops, etc. Therefore, sensor fusion including wearables sensors (accelerometers, GPS receivers, gyroscopic sensors, etc.) and ambient sensors (RFID, WiFi, camera, etc.) in those scenarios is necessary to achieve ubiquitous HAR. Another research direction is on exploring unsupervised and semi-supervised learning with a smaller amount of training data in order to make HAR more applicable in real scenarios. Finally, it is important to protect users' privacy while performing HAR. Current HAR systems need to upload sensor data to servers or clouds, the sensor data could be illegally used resulting in intrusion of users' privacy. One solution would be to deploy HAR system on clients without uploading the sensor data and another one would be performing sensor data encryption before uploading without compromising performance.

In our future work, we will attempt to overcome the problem of unobstructed Line of Sight (LOS). For object detection and tracking, it is necessary to ensure there is no obstruction between the LIDAR and targets. With a single LIDAR, the moving targets are easily blocked by obstacles or other moving targets that obstruct their LOS. For example, in our research, a person can block the LOS of another person frequently, this leads to a broken and uncontinuous trajectory. For overcoming this problem, we will use multiple LIDARs to perform human detection and tracking from a multi-perspective. A trajectory-based human activity recognition relies on trajectory pattern analysis. However, some micro-activity such as eating, drinking, calling, etc., cannot be recognized by only human trajectory. While wearable sensors or Radars are able to recognize them. So we will attempt to combine LIDARs with wearable sensors or Radars to perform both micro-activity and macro-activity recognition simultaneously.

Acknowledgment

The authors would like to thank Dr. Kristou Mehrez and Hokuyo Automatic company to provide the LIDAR for our research.

References

[1] F. Luo, E. Bodanese, and S. Poslad, "Kitchen activity detection for healthcare using a low-power radar-enabled sensor network." Institute of Electrical and Electronics Engineers, 2019.

[2] A. Rafii, S. B. Gokturk, C. Tomasi, and F. Sürücü, "Gesture recognition system using depth perceptive sensors," Mar. 26 2019, uS Patent App. 10/242,255.

[3] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 851–860.

[4] X. Song, X. Shao, R. Shibasaki, H. Zhao, J. Cui, and H. Zha, "A novel laser-based system: Fully online detection of abnormal activity via an unsupervised method," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 1317–1322.

[5] P. Vallabh and R. Malekian, "Fall detection monitoring systems: a comprehensive review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 6, pp. 1809–1833, 2018.

[6] J. C. Nunez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Velez, "Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition," *Pattern Recognition*, vol. 76, pp. 80–94, 2018.

[7] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1971–1980.

[8] P.-G. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 485–494, 2015.

[9] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1384–1393, 2018.

[10] P. C. Shih, K. Han, E. S. Poole, M. B. Rosson, and J. M. Carroll, "Use and adoption challenges of wearable activity trackers," *IConference 2015 Proceedings*, 2015.

[11] J. Huberty, D. K. Ehlers, J. Kurka, B. Ainsworth, and M. Buman, "Feasibility of three wearable sensors for 24 hour monitoring in middle-aged women," *BMC women's health*, vol. 15, no. 1, p. 55, 2015.

[12] Z. Ma, B. Wu, and S. Poslad, "A wifi rssi ranking fingerprint positioning system and its application to indoor activities of daily living recognition," *International Journal of Distributed Sensor Networks*, vol. 15, no. 4, p. 1550147719837916, 2019.

[13] F. Luo, S. Poslad, and E. Bodanese, "Human activity detection and coarse localization outdoors using micro-doppler signatures," *IEEE Sensors Journal*, 2019.

[14] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2016.

[15] L. Ren, Y. S. Koo, H. Wang, Y. Wang, Q. Liu, and A. E. Fathy, "Noncontact multiple heartbeats detection and subject localization using uwb impulse doppler radar," *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 10, pp. 690–692, 2015.

[16] E. Aksan and O. Hilliges, "Stcn: Stochastic temporal convolutional networks," *arXiv preprint arXiv:1902.06568*, 2019.

[17] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE sensors journal*, vol. 15, no. 3, pp. 1321–1330, 2014.

[18] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with rfid-based sensors," in *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 2009, pp. 51–60.

[19] M. Sridharan, J. Bigham, P. M. Campbell, C. Phillips, and E. Bodanese, "Inferring micro-activities through wearable sensing for adl recognition of home-care patients," *IEEE journal of biomedical and health informatics*, 2019.

[20] W. Xi, D. Huang, K. Zhao, Y. Yan, Y. Cai, R. Ma, and D. Chen, "Device-free human activity recognition using csi," in *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition*. ACM, 2015, pp. 31–36.

[21] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. IEEE, 2011, pp. 1–5.

[22] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[23] S. Kamal, A. Jalal, and D. Kim, "Depth images-based human detection, tracking and activity recognition using spatiotemporal features and modified hmm," *J. Electr. Eng. Technol*, vol. 11, no. 3, pp. 1921–1926, 2016.

[24] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st annual international conference on mobile computing and networking*. ACM, 2015, pp. 65–76.

[25] C. A. Ronao and S.-B. Cho, "Recognizing human activities from smartphone sensors using hierarchical continuous hidden markov models," *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, p. 1550147716683687, 2017.

[26] Z. Chen, Q. Zhu, Y. C. Soh, and L. Zhang, "Robust human activity recognition using smartphone sensors via ct-pca and online svm," *IEEE paul2015effectiveTransactions on Industrial Informatics*, vol. 13, no. 6, pp. 3070–3080, 2017.

[27] P. Paul and T. George, "An effective approach for human activity recognition on smartphone," in *2015 Ieee International Conference on Engineering and Technology (Icetech)*. IEEE, 2015, pp. 1–3.

[28] S. Balli, E. A. Sağbaş, and M. Peker, "Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm," *Measurement and Control*, vol. 52, no. 1-2, pp. 37–45, 2019.

[29] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.

[30] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph lstm for group activity recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[31] Y. Guo, D. Tao, W. Liu, and J. Cheng, "Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 4, pp. 617–627, 2016.

[32] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156–165.

[33] K. L. Y. Siang and S. W. Khor, "Path clustering using dynamic time warping technique," in *2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT)*, vol. 1. IEEE, 2012, pp. 449–452.

[34] J. Liang, L. Jiang, J. C. Niebles, A. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," *arXiv preprint arXiv:1902.03748*, 2019.

[35] Y. Wang, D. Wang, and F. Chen, "Abnormal behavior detection using trajectory analysis in camera sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, p. 839045, 2013.

[36] Q. Gao and S. Sun, "Trajectory-based human activity recognition with hierarchical dirichlet process hidden markov models," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, 2013, pp. 456–460.

[37] T. Fuse and K. Kamiya, "Statistical anomaly detection in human dynamics monitoring using a hierarchical dirichlet process hidden markov model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3083–3092, 2017.

[38] L. Huang, Q. Li, and Y. Yue, "Activity identification from gps trajectories using spatial temporal pois' attractiveness," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on location based social networks*. ACM, 2010, pp. 27–30.

[39] W. Huang, M. Li, W. Hu, G. Song, X. Xing, and K. Xie, "Cost sensitive gps-based activity recognition," in *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*. IEEE, 2013, pp. 962–966.

[40] H. Martin, D. Bucher, E. Suel, P. Zhao, F. Perez-Cruz, and M. Raubal, "Graph convolutional neural networks for human activity purpose imputation from gps-based trajectory data," 2018.

[41] A. Panangadan, M. Matarić, and G. S. Sukhatme, "Tracking and modeling of human activity using laser rangefinders," *International Journal of Social Robotics*, vol. 2, no. 1, pp. 95–107, 2010.

[42] Z. Ma, J. Bigham, S. Poslad, B. Wu, X. Zhang, and E. Bodanese, "Device-free, activity during daily life, recognition using a low-cost lidar," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[43] P. H. Ahmad and S. Dang, "Performance evaluation of clustering algorithm using different datasets," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 3, pp. 167–173, 2015.

[44] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 726–733.

[45] R. J. Meinhold and N. D. Singpurwalla, "Understanding the kalman filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, 1983.

[46] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[47] Hokuyo, "Scanning laser range finder smart-urg mini ust-10lx (uust003) specification," *Online available*, 2015.

[48] S. Yadav and S. Shukla, "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification," in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE, 2016, pp. 78–83.

[49] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang, and Z. Zhao, "An overview of data fusion techniques for internet of things enabled physical activity recognition and measure," *Information Fusion*, 2019.