Published in final edited form as:

Proc Int Conf Embed Netw Sens Syst. 2016 November ; 2016: 164–175. doi:10.1145/2994551.2994569.

Deep Learning for RFID-Based Activity Recognition

Xinyu Li[1], **Yanyi Zhang**[1], **Ivan Marsic**[1], **Aleksandra Sarcevic**[2], and **Randall S. Burd**[3]

[1]Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA

[2]College of Computing and Informatics, Drexel University, Philadelphia, PA, USA

[3]Division of Trauma and Burn Surgery, Children's National Medical Center, Washington, D.C., USA

Abstract

We present a system for activity recognition from passive RFID data using a deep convolutional neural network. We directly feed the RFID data into a deep convolutional neural network for activity recognition instead of selecting features and using a cascade structure that first detects object use from RFID data followed by predicting the activity. Because our system treats activity recognition as a multi-class classification problem, it is scalable for applications with large number of activity classes. We tested our system using RFID data collected in a trauma room, including 14 hours of RFID data from 16 actual trauma resuscitations. Our system outperformed existing systems developed for activity recognition and achieved similar performance with process-phase detection as systems that require wearable sensors or manually-generated input. We also analyzed the strengths and limitations of our current deep learning architecture for activity recognition from RFID data.

Keywords

Activity recognition; process phase detection; passive RFID; deep learning; convolutional neural network

1. INTRODUCTION

Passive RFID has been proposed for many real-world applications due to its advantages of size, cost and maintenance-free operation. It has been used for activity recognition with some success, but RFID-based activity recognition remains a challenge due to the limited information provided by sensed RFID data. Researchers have used it for activity recognition in some application scenarios where other sensors are unsuitable, such as medical applications, where camera-based solutions are limited by privacy concerns and wearable-sensor-based solutions may be inconvenient to wear and interfere with work. RFID based systems, however, have failed to achieve high accuracy of activity recognition in fast-paced and crowded environments. Two key challenges for RFID-based activity recognition are: the noise in received signal strength (RSS) that cannot be filtered out, and the absence of a direct link between the raw RSS values and human activity—an abstract concept.

Similar challenges exist in computer vision for large-scale image classification and in speech recognition for voice-to-text conversion in noisy environments. Deep learning [1] introduced in those fields has achieved high levels of performance [2,3,4]. The main difference between deep learning and traditional machine learning algorithms is that instead of manual feature selection and defining the rules for making correct predictions, deep learning is able to learn the “right” features from large datasets and use them for this purpose. Consistent with the views of others [5], we believe that deep learning has the potential to be successful for mobile sensing. In this paper, we apply deep learning to the problem of activity recognition in a fast-paced real-world environment using only passive RFID.

We present a deep-learning architecture that uses only RFID data for detection of process phases and activities during trauma resuscitation. The resuscitation process has five consecutive phases: pre-arrival, patient arrival, primary survey, secondary survey, and post-secondary survey. Each phase consists of several activities—the specific low-level tasks performed by care providers that may or may not use medical objects. We define an activity as the interval during which one or more objects are used explicitly for patient care, which excludes the preparatory or cleanup manipulation of these objects [6]. We chose trauma resuscitation as our application domain for two reasons. First, this complex work setting is prone to errors and inefficiencies and is in need of decision support. Activity recognition is an essential building block to enable the development of this type of system. Using computer vision is not preferred for privacy concerns and active wearable sensors are not feasible because the user must remember to wear them, they may interfere with work, and they require maintenance, such as battery charging. From a research perspective, RFID-based activity recognition has treated activity recognition as a binary classification problem where a specialized classifier decides whether or not an activity of a particular type is occurring. These types of systems, however, may not be scalable to a large number of activities. In addition, the common approach for activity recognition involves two steps: first detect the use of objects associated with specific activities by detecting human-to-object-interaction from sensor data, and then recognize activities based on used objects [6]. The predication errors made by the system in the first step will be cascaded into second step and impair the final prediction result.

Our approach for activity recognition uses passive RFID sensing. The RFID tags need to be strategically placed on objects of interest. Various features have been proposed and classifiers tested for RFID systems in different application settings [6,7], which makes it unfeasible to compare their relative efficiency. As a result, feature and classifier selection for RFID data is often arbitrary. Our research demonstrates a novel way for activity recognition from RFID data without using manufactured features. To perform process-phase detection and activity recognition from RFID data, we treated the process-phase and activity recognition as a multi-class classification problem instead of extracting manufactured features and cascading object-use detection with activity prediction. We implemented a deep convolutional neural network with three convolutional layers and three fully-connected layers totaling 8.7M weights. The network was developed with a Microsoft Azure cloud computing platform [8] and locally with Google TensorFlow [9]. We trained this network with RFID data collected during 16 actual trauma resuscitations in a trauma center. Different networks were trained for process-phase detection and for activity recognition. The experimental results showed that our system achieved performance on phase detection comparable to the system that used manually-generated log of executed tasks as input to phase recognition [10]. Our system recognized 10 common medical activities directly from RFID data with *F*-score 18% greater than an existing RFID-based system in the same application scenario [6]. To our knowledge, we are the first to apply deep learning with RFID sensing for activity recognition in complex teamwork. This paper contributes:

A deep learning model for activity recognition in complex teamwork based on passive RFID: We developed a system for complex activity recognition from RFID data using a deep convolutional neural network. Unlike existing systems that rely on manufactured features and a cascade structure with object-use detection followed by activity recognition, our system works directly with RFID data and performs multiclass classification of activities or process phases. It outperformed our recent system that used manufactured RFID features for activity recognition with 18% greater *F*-score [6].

Analysis of features that formed in convolutional layers of the network: We visualized and analyzed the neuron activations in each convolutional layer as has been done in image analysis [11]. We observed that our system is able to find very specific RFID features for process-phase detection and activity recognition. We confirmed our conclusion by replacing the input points that our network considered important with 0's, that sharply decreased the performance. Based on this analysis, we also identified potential future improvements of deep learning with RFID data.

System evaluation on actual trauma resuscitation data: We trained and tested the deep learning network using real-world data, recorded during actual trauma resuscitations in a trauma center, for both activity recognition and high-level process-phase detection.

The rest of this paper is organized as follows. Section 2 reviews state-of-the-art implementations of activity recognition and relevant applications of deep learning. Section 3 describes the design and implementation of our deep convolutional neural network. Section 4 presents our experimental results for process-phase detection and activity recognition. Section 5 visualizes and analyzes the activation maps of convolutional layers and the causes of prediction errors. Section 6 concludes the paper with a discussion of limitations of our work and future directions.

2. RELATED WORK

Due to their unique advantages (small, cheap and battery free), passive RFID tags have been used in applications where other sensors are not suitable or have failed. These applications include detection of human-object interaction [12,7], people and object tracking [13] and more complex problems such as activity recognition. RFID was used for activity recognition in a kitchen setting [14], but only as secondary to a vision system because the received radio signal was subject to noise and interference caused by moving people and other objects. RFID was also used as the primary system for process-phase detection with wearable RFID antennas and other sensors [15]. The system was able to achieve satisfactory performance for phase recognition, but wearing the antennas requires user participation and may interfere with work in fast-paced medical settings. Recent research demonstrated that the status of object manipulation can be estimated using passive RFID tags and fixed antennas based on manufactured features extracted from received signal strength indicator (RSSI) [16]. The use of specialized objects to perform complex activities provided the basis for activity recognition [6]. Challenges remain because the recorded RFID data contain noise and variance due to environmental changes, such as people moving in the room. The noise and variance in RFID data compromise the representativeness of manufactured features and in turn impact activity recognition results.

Similar challenges exist in other applications, such as image recognition and speech recognition, where large part of input data are inessential (e.g., redundant pixels, background noise), requiring the classifier to be insensitive to those variations. Earlier research tried to accomplish the complex tasks such as object recognition or activity recognition by using manufactured features, or building a hierarchical model with several layers of classifiers to extract low-level features for final decision making [17,18]. The use of deep learning in recent years has led to great leaps in many fields, from image classification [2] to speech recognition [19].

Deep learning has revolutionized image classification and speech recognition. It is reasonable to expect similar success in pervasive computing [5]. Earlier research showed that deep learning can be applied to data from mobile phone sensors or an accelerometer for recognition of person's simple physical activities [20,21]. No system has yet been developed that combines deep learning and RFID for activity recognition during complex work, such as patient care, instead of simple physical activities like sitting or standing. We developed a deep learning system in a setting similar to one we previously studied [6] and achieved better performance on medical activity recognition compared with existing research.

3. DATA COLLECTION

3.1 Automated RFID Recording

This study has been approved by the Children's National Medical Center Institutional Review Board. We installed the hardware for RFID data collection and system activation control in an actual trauma room. The RFID data were collected with two Impinj R420 (8 ports) readers, set to record RFID data in maxmiller mode and dual target search mode [22]. Because trauma events occur without warning, we could not keep the system continuously recording. We developed a fully automated system that is activated at the start of each resuscitation and keeps recording RFID data from all tags while the resuscitation is in progress. We set up a Kinect V2 sensor to monitor the number of people in the room. The RFID system will be activated to record data when more than two people are in the room and stops when no people are in the room (Fig. 1). To recognize 10 medical activities (Table 1), we tagged 11 types of medical objects following existing tagging strategies [6]. Because the blood-pressure (BP) cuff was tagged on the inside and outside, we counted it as two different object types, resulting in a total of 12 types of medical objects. The system recorded the received signal strength (RSS) from tags during 16 actual trauma resuscitations in this format: [*Timestamp, Tag ID, RSS, Reader Name, Port Number*].

Attributes of RFID signal other than RSS, such as Doppler shift and phase angle, have been used for human-object interaction detection or people tracking [13,23]. Our experience and that of others [23] has shown that Doppler shift measured by Inpinj R420 reader API is not accurate enough for our purposes. The U.S. government regulation requires that RFID readers perform frequency hopping, which affects the phase-angle measurements. Based on our experiments, the phase angle measured by the Impinj reader will have around 2.68 rad standard deviation for a stationary tag, which makes it unsuitable for classification.

3.2 Pre-processing

Because of multiple instances of tracked objects and variable readout success rate, the recorded data needed to be preprocessed. We preprocessed the data in three steps:

1. **Object name lookup:** Many objects of the same type may be in the monitored area, such as four thermometers in our trauma room. We tagged all instances of an object type to ensure that all the objects used during trauma resuscitations are tracked, which resulted in about 50 tags in the trauma room. Not all the tags were visible to antennas all the time, because some objects were kept inside a cabinet or shelf. We maintained a lookup table mapping tag IDs to the tagged objects. Before further processing the data, we replaced each tag ID with the name of its associated object type. All instances of the same object type were given the same object name, so that each RSS data entry represented one of 12 object types. The RSS data from multiple instances of the same object were combined during averaging in the following step.

2. **Regularization of RSS data:** Because the number of successful readings by each antenna varies over time for each tag, the recorded time series had to be regularized to a constant sampling rate. The sampling rate was determined based on the minimum achieved reading rate of the tags. For our study, we used 1 second as the sample time because the number of readings per second for tags in the trauma room were greater than one if the tags were not occluded by people or other objects.

   The output of regularization is an $I \times J$ matrix (we call it an “antenna-object frame”), where I rows represent I tagged object types and J columns represent J antennas installed in the room. The element (i,j) is the *averaged RSS* collected during one second for object type i by antenna j. We had $I = 12$ types of objects and $J = 8$ antennas. The regularization process for every second generated a 12×8 matrix. We put a zero if no data was received by an antenna for a given point. Note that the RSS value has a physical meaning, where “0” means the received signal strength is 1 mW. Because in our implementation the distance between tags and antennas is at least 2m, the actual received signal strength is much lower than 1 mW, so it is safe to use “0” to indicate that no data were received.

3. **Stacking antenna-object frames:** The final step is to stack the antenna-object frames over time (Fig. 2). The pre-processed RFID data forms a 3D matrix with T layers of antenna-object frames, where T is the total time (in seconds) of recorded RFID data from all executions of the process. In our case, $T = 50{,}000$ sec, collected during 16 resuscitations.

4. DEEP LEARNING MODEL

4.1 Neural Network Structure

Several types of deep learning models have been proposed for different application and sensor types. Examples include the Convolutional Neural Network (CNN), widely used for image classification [13] and recently for speech recognition [24], the Deep Neural Network (DNN), used for speech recognition and audio sensing [25], and a multimodal structure used for audiovisual speech recognition [26].

Our choice of network structure was driven by the nature of our RFID data. The RFID signal received from a single tag by one reader antenna is a one-dimensional series, similar to a speech signal, for which both the DNN and CNN have been used. Our RFID data were collected by several antennas from multiple tags on same or different objects, which resulted in two additional dimensions: the receiving antenna and object/tag ID. We chose CNN over DNN because we wanted to process data from all tags together to capture potential concurrent object uses. CNN better handles high-dimensional input by representing it as a high-dimensional matrix. In addition, DNN is ineffective at learning features and requires as input extracted features rather than raw data. Given that it is hard to optimize manually selected features, a poor selection will lead to poor performance. CNN can generate useful features via its learnable filters, so it can directly accept RSS data as input. We implemented the CNN with three convolutional layers, followed by three fully-connected layers and a softmax layer for output (Fig. 3). We used the CNN with rectified linear units (ReLUs) because such units train several times faster than traditional *tank* units [2,25]. We recently implemented a modified DNN structure using a similar dataset, but directly using high-dimensional RSS in the input layer remains a challenge [27].

Unlike CNN structures for other applications, we designed the input and convolutional layers to reflect the structure of RFID data collected with multiple antennas. We next describe the building blocks of our CNN (Fig. 3).

4.2 Input Layer

The input layer prepares the input data for the convolutional network. The input data needs to be represented differently for different applications. For image classification, the input layer is often a single gray-scale image or three gray-scale images (for red, green and blue channels of color images). For speech recognition, the input layer is often constructed as a time-frequency feature map. In general, RFID data in the input layer has three dimensions that represent the objects, the antennas, and the observation time window. Unlike speech from a single microphone, RFID data are recorded by multiple antennas and have an extra dimension of space. Unlike stationary images, RFID data are related in three dimensions: spatially across antennas, temporally over time, and semantically over tagged objects that are manipulated concurrently in one or several parallel activities. Some similarity exists with video data processing where image frames are temporally related and pixels in each frame are spatially related [28].

Our input layer is formed in two steps: first stacking t antenna-object frames collected over t seconds, and second rotating the 3-D matrix to make the object and time as the first two dimensions (Fig. 4), and antenna as the third dimension. The time window *t* is determined by the duration of the shortest activity. A window shorter than the shortest activity minimizes the chance that multiple activities will be represented in it. In our problem domain, some activities take a short time, such as evaluating the ears, which on average lasts 10 seconds and has the lowest average duration. Based on Nyquist theorem, we chose t=5 to ensure that the time window is just 50% shorter than the shortest activities.

The convolution operation sums the contributions from different planes in the input layer and uses ReLU as:

$$h_j = \max\left(0, \sum_{k=1}^{K} h_k * w_{kj}\right)$$

where h_j is the j^{th} plane of output data from each convolutional layer, h_k is the k^{th} plane of input data which has K planes in total and w_{kj} is the k^{th} plane of kernel j. We used the number of object types and the time value as the first two dimensions of input layer, and the number of antennas as the third dimension. The first two dimensions represent the RSS from each object over a time window, which for stationary objects should appear flat when visualized as a gray-scale image (Fig. 4). If an object is manipulated, the RSS of its tag should be very different from stationary state in the visualization. This arrangement also ensures that each convolutional operation is performed on the data collected by all antennas, which makes our network structure applicable to scenarios with different number of antennas and antenna arrangements.

4.3 Convolutional Layers

The convolutional layers with sets of learnable filters are the core building blocks of convolutional neural networks, and the pooling layers implement the input data down-sampling. Several parameters need to be determined for constructing the convolutional layers [29,2]. For each convolutional layer, the size of the convolutional kernel decides the shape and number of feature maps used in convolution operation. No analytical procedure is available to determine the optimal number of convolutional layers for a given application. The most suitable network structure is usually determined empirically.

We chose to have 3 convolutional layers in our network, with odd-number kernel sizes: 3×3×32, 3×3×64 and 3χ3×128, and with stride 1, which have been shown as efficient in the VGG net [30]. Because the input data had a small dimension (12 objects × 5 one-second frames) in each antenna plane, zero padding was added to perform a wide-type convolution in order to maintain the size of each output plane the same as the input plane. The number of feature maps in each kernel and the number of convolutional layers was determined empirically. We used 5,000 seconds of data from 50,000 seconds of total available RFID data as training data for classifying the five resuscitation phases. The number of feature maps in each convolutional layer was determined by a script looping through the powers of 2 from 16 to 256 and choosing the combination of kernel sizes for convolutional layers that performed best on detecting the five resuscitation phases. We reasoned that 3 convolutional layers will also provide the best tradeoff for activity recognition, because phase detection and activity recognition use the same RFID data as input data.

More convolutional layers generally yield better performance, but the performance gain diminishes. We only tested the CNN with 1 to 4 convolutional layers, because the network with 5 layers has over 30M weights which was not feasible for our hardware. The results (Table 2) show only a small gain in precision, recall and *F*-Score when using four convolutional layers but a large difference in memory cost (around 2 times). We concluded that using 3 convolutional layers has the best tradeoff between the computational resources and performance gain.

We did not use pooling layers, which in other applications have been used to extract low-level, shift-invariant features, and to reduce the data dimensionality for computational efficiency. Unlike images, which normally contain redundant pixels unimportant for the classification, the raw RFID data matrix has very little redundancy. Pooling with a minimum window (2×2) would only leave one fourth of the pooled data which would distort the spatial and temporal relationships of the RFID data. A new pooling strategy with learnable weights was recently proposed [24], which will be tested in our future work.

4.4 Fully Connected Layers

No more than two fully-connected layers have commonly been used to avoid overfitting [2,24]. Our experiments showed that in our domain 3 fully connected layers work better than 2 layers. This finding is due to the orders-of-magnitude dimensionality reduction between the neurons in the last convolutional layer (7680) and the output layer (5 for process phases and 10 for activities).

4.5 Model Training

We trained two CNNs to detect 5 process phases and 10 resuscitation activities, respectively, using preprocessed RFID data (Fig. 2) from 16 trauma resuscitations. The label (one of 5 process phases or 10 activities) for each second of data was manually generated by medical experts from video review of the corresponding trauma resuscitations. The 16 resuscitations provided a total of 50,000 seconds of data. Due to the great variability of the resuscitation process, the duration of each activity is unpredictable, and some of the 10 activities were not well represented, unlike the 5 resuscitation phases which were all well represented. Given the unbalanced dataset, randomly selecting the number of samples would not guarantee sufficient data for all activity classes during training and testing. As suggested [6], we selected a percentage of data from each class for training and used the remainder for testing.

Overfitting was a concern because process-phase detection is a relatively small multi-class classification problem with only 5 classes, compared with other CNN applications, such as image classification with thousands of classes [2]. We took two steps to avoid model overfitting. First, we applied the “dropout” in fully-connected layers, which is widely used in CNNs to avoid overfitting during model training [31]. Second, we implemented the cross-validation and set the system to stop training when the cross-validation error starts to increase. We initialized the learning rate at 0.01 and adjusted it based on the ADAM optimization (Adam Optimizer in TensorFlow) [32],

We implemented our CNN using Microsoft Azure cloud service and locally with Google TensorFlow [33]. Both frameworks achieved similar performance and allow users to manually define the CNN with #Net or Python. The advantage of Azure is that it allows the user simultaneously run several CNN training processes with different data or network parameters and easily compare their performance. The training process is faster in Azure compared to training with computers using a Core i5 CPU. On the other hand, in Azure the trained weights are not accessible to the user. The TensorFlow runs locally, though the training speed depends on hardware and it is impractical simultaneously to train several models on a single computer. All the trained weights, however, are accessible, which makes TensorFlow suitable for model analysis. Because of these features, we used Azure for CNN model design and TensorFlow for experimental evaluation.

5. EXPERIMENTAL RESULTS

5.1 Detection of Process Phases

We first applied deep learning for detection of five phases of resuscitation: pre-arrival (PA), patient arrival (A), primary survey (P), secondary survey (S) and post-secondary survey (PS). The phase detection is considered challenging because process phase is a high-level concept, usually defined using lower-level concepts, such as used objects or constituent activities. We preprocessed all recorded RFID data and randomly selected 5000 seconds from each phase as training data and used the remaining data for testing. In this way, less than 50% of total data was used for training. We trained our deep convolutional neural network (Fig. 3) using TensorFlow platform and stopped the training when the cross-validation error remained constant for one epoch. The system achieved the average accuracy of 72.03 % for detection of the five phases (Table 3).

We compared the performance of our deep learning system with commonly used classifiers: one-vs-all SVM, one-vs-all logistic regression, Random Forest and Bayesian Net, using previously introduced features [6], on the same data set. We treated phase detection as a multi-class classification problem and considered the detection of each process phase as a binary classification problem. We used the common metrics of *F*-score, Informedness, Markedness [34] and Matthew Correlation Coefficient (MCC) [35]. The results (Fig. 5) show that our convolutional neural network achieved best performance and a 15% performance gain over random forest, the second-best classifier.

Among the few published works on process-phase detection, we chose three representative systems [15,10,36] and compared our deep learning system with them (Table 4). Our system achieved a similar accuracy as a system based on wearable sensors [15]. The advantage of our system is that it uses the data collected with fixed antennas that do not require human involvement, which is more practical for time and safety-critical medical applications. Our system is also advantageous compared to the system that takes as input a manually-generated activity log [10], or the system that takes input directly from medical equipment sets [36]. Although all systems achieved comparable performance (Table 4), this comparison is not direct because our system is designed for a different problem domain than others [10,36]. Given that our system does not require any human involvement to generate an activity log or machine-signal log from medical equipment for process-phase detection, it is easier to generalize to other similar application scenarios.

5.2 Activity Recognition in Trauma Resuscitation

Compared with detecting phases of a process, medical activity recognition is considered more important, because of its fundamental role in building decision-support systems or other artificial intelligence systems that help improve patient care and outcomes. Unlike systems designed for recognizing simple physical activities of individuals, such as sitting, standing, or sleeping [37,20,21], recognizing complex teamwork activities is significantly more challenging. We trained our convolutional neural network (Fig. 3) with preprocessed RFID data for 11 medical activities (10 shown in Table 1 and “other” as a catch-all activity). We could not split the training and testing data as we did for process-phase detection, where we used 5000 RSS samples from each class for training data and the rest for testing, because we had very limited data for brief activities, such as evaluation of patient’s ear. Using the same number of instances for training each activity could cause bias. We randomly selected 40% of data in each class for training and the remaining 60% data for testing. Our system achieved average accuracy of 80.40% for recognition of 11 activities (Table 5).

Unlike some other activity recognition systems that trained independent binary classifiers for different activities [14,6], our system treats activity recognition as a multi-class classification problem and can scale up if additional activities need to be recognized. To avoid the evaluation bias caused by different training and testing sets, we fed the same training and testing sets into traditional classifiers. We compared the performance of our convolutional neural network to traditional classifiers using evaluation metrics introduced above. Our network still performed best compared with all other classifiers (Fig. 6). It achieved about 10% higher *F*-score compared with random forest, which was the second best classifier, and around 30% higher *F*-score compared with all other classifiers. The same performance gain held for other evaluation metrics.

To demonstrate the advantage of our deep learning system in medical applications, we first compared this system with our previous system for resuscitation activity recognition from passive RFID that uses a cascade model with manufactured features such as visible antenna combination, the Spearman rank correlation coefficient, and other features [6]. The RFID data used for evaluating our deep learning and that we previously used [6] were collected in same environment with real-patients, using different RFID readers (Impinj vs. Alien). Our deep learning achieved 30% higher *F*-score, and MCC and double informedness scores compared with the system in [6] using sensor data as input. Even when our previous system [6] used ground truth of object-use as input to the classifier (instead of object-use detected from sensor data), our deep learning still achieved better performance in *F*-score, informedness and MCC (Fig. 7). This comparison shows the power of deep learning to process the noisy RFID data and the potential of deep learning model applied to RFID-based applications.

We also compared the performance of our deep-learning system using RFID data for medical activity recognition with several state-of-the-art recognition systems for real-world application scenarios [14,38,39] (Table 6). Although these systems were implemented for different environments, a comparison shows that our deep learning was able to achieve performance similar to vision-based systems for activity recognition. As was the case with other applications [2,4], our deep learning-based activity recognition system also achieved better performance compared to systems that used traditional classifiers.

5.3 Generalizability Experiments with Recognition of Laboratory Activities

To demonstrate generalizability of our activity recognition system beyond the trauma room application, we performed additional experiments in a typical laboratory room with staged activities. We identified six activities that commonly take place in a research lab: programming (P, 1-2 people), eating (E, 2-3 people), lab-meeting (LM, 3-5 people), writing-on-whiteboard (WB, 1 person), reading (R, 1-2 people) and no-activity (NA, nobody). To demonstrate antenna configuration flexibility, we used a different antenna configuration with four antennas mounted on the ceiling 3 meters above the ground and attached to an Impinj R420 reader (Fig. 8). Seven types of objects were tagged with the same RFID tags we used in the trauma room: a book, a mouse, a keyboard, the surface of chairs, the surface of a dining table, the surface of a desktop and several marker pens used for writing (total 26 tags in the experimental area). With the agreement of our laboratory colleagues, we collected 10 hours of data (“No-activity” for 2.5 hours and 1.5 hours per each of the remaining five activities) using the same methods as we used in the trauma room. We applied the same CNN structure for model training and achieved the average recognition accuracy of 90.8% for the six lab activities (Table 7). This experiment showed that our activity recognition system works well with different antenna configurations and application environments. The CNN training process in a different application environment was straightforward and did not require manual feature selection or parameter tuning, thus making our system simple to train and easy to use.

6. DISCUSSION

6.1 Visualizing Deep Learning for RFID

To better understand how deep learning works in our context, we visualized the activation maps in each convolutional layer using the method proposed for image processing [11]. Unlike pixels, our data points arranged in the input layer do not represent spatial information. We then cannot expect that object shifts in space will result in shifted activations of some neurons as happens in visualization for videos [11]. We do expect that different neurons will fire for input data recorded during different process phases. To visualize our CNN, we fed 1,000 randomly selected RSS sample data for each of 5 resuscitation phases (5,000 samples in total) to the network trained for process-phase detection and visualized the activation maps by averaging the 1000 activation maps for each phase (Fig. 9).

We selected the CNN trained for process-phase prediction because a single resuscitation phase comprises several activities, and several medical objects are used in most phases. For this reason, only time-invariant features should be useful for process-phase detection. We found the following from the visualizations (Fig. 9):

The first convolutional layer appears to have extracted RSS features of different objects. Some of the feature maps appear to have extracted RSS at a certain time (see the neurons lighted in the third column in all activation maps in the top row, marked with ① in Fig. 9).

The second convolutional layer appears to sample the output from first convolutional layer at different times using different sampling masks. Example sampling masks can be seen in the visualized activation maps (middle row of Fig. 9), where some neurons fired frequently and others almost never did. Also the sampling patterns appear to complement each other. For example, two activation maps (marked with ② in Fig. 9), show complementary neurons firing. For this reason, different activation maps completely cover all combinations of objects over time, and there is no sampling bias.

Our most interesting findings are several very specific features in the third convolutional layer. For example, the neuron in sixth row of the same activation map in all process phases except the first one (Pre-Arrival) fired frequently (lighted dot marked with an arrow in map ③ in Fig. 9). This neuron represents the blood-pressure gauge stand, one of the 12 objects tagged for our study. By viewing the corresponding video recording, we found that the gauge stand was often placed against the wall of the trauma room (Fig. 10), where it was not well covered by RFID signal before the patient arrived (Pre-Arrival). When the patient arrived, the gauge stand was repositioned near the patient bed (Fig. 10) and the tag became better exposed to antennas throughout the resuscitation. This observation may explain why this neuron did not fire in the pre-arrival phase. The firing pattern of several other neurons can be explained with actual situations. For example, the neuron representing the cardiac monitor adapter only fired during the last three phases (marked with ④ in Fig. 9). By reviewing the videos, we found that the adapter was hanging on the tool mount when not in use, with its electrical cord wrapped around the tag (Fig. 10). The adapter was often used during last three phases, when the tag was well exposed to RFID signals. The neuron representing Bair Hugger Connector (marked with ⑤ in Fig. 9) was firing during the first two phases. By reviewing the videos, we found that during the last three phases people were staying around the head of the patient bed where the Bair Hugger connecter was located (Fig. 10), which was not the case during the first two phases. The connector is placed at around 50 cm above the ground which makes it easy to get blocked by people standing near it.

To confirm that our network found the features important for phase detection, we replaced the RSS data with 0 for the objects corresponding to neurons that strongly fired during some phases, as described in the third observation above. We also randomly selected 3 neurons corresponding to other objects and replaced their RFID data with 0 to compare the effect of these interventions on phase detection. The result showed 16.5% lower recall and 13% lower *F*-Score when the RFID data from objects “important” to our deep learning were zeroed, compared to the case when RFID data from three other randomly selected objects were zeroed.

Understanding the meaning of each activation map in real-world context remains a challenge, and not all activation maps can be easily explained by actual situations. Our convolutional network implements the so called “2.5D convolution”, where two-dimensional convolutions are summed up along the third dimension. In this way, the spatial relationship between different antenna coverages is overlooked. The 3D convolution was recently used for video classification and object recognition [3,40]. Our future work will test this method to better exploit spatial relationships in RFID data.

6.2 Error Analysis

Although our deep learning achieved strong performance on process-phase and activity recognition, we noticed that the prediction accuracy varied for different classes. For example, recognition accuracy for the patient-arrival phase was significantly lower than for other phases (Table 3). To understand the errors made using deep learning, we used the probability of a random guess for each class as the baseline. If the probability of classifier confusing two classes was greater than the baseline, we considered that the classifier worked inadequately for the given class.

With this type of rule, the baseline for prediction of 5 process phases is 20% and our network exceeded this baseline only in one case: it predicted 35% of time the patient-arrival as pre-arrival phase (Table 3, **boldface**). By reviewing the ground truth and discussing with medical experts, we found that no objects were used either because the patient had not yet arrived (pre-arrival phase) or the care has not yet started (patient-arrival phase). The RSS information from tagged objects remained stable during these two phases. This finding implied that deep learning confused the classes with similar input data.

Similar problem occurred with activity recognition. In this case, the baseline for recognition of 11 activities is 9.9%. The error rate above the baseline is in **boldface** in the confusion matrix (Table 5). The first finding was that the system had difficulty distinguishing oxygen preparation (BC) and warm sheet (EC) activities. By discussing with medical experts and reviewing the activity ground truth, we found that these two activities were performed simultaneously over 80% of time in the observed 16 resuscitations. Such overlapping activities generated very similar training data for both activities, which compromised the system performance. The problem of co-occurring activities is hard to solve with RFID sensing only, because both (or more) activities will be represented in the training data and cannot be segmented to train for each activity separately. A purely RFID-based deep learning system is not good at distinguishing these types of activities and needs to be complemented with other sensors.

In addition to BC and EC activities, our system also confused temperature measurement (EA) and blood pressure measurement (BP) activities with “other than listed (OT)”, resulting in a higher-than-baseline confusion probability (Table 5). This problem occurred because for some objects most of the manipulation time was *not* for task-performance purpose (40.8% of manipulation time for the thermometer and 92.93% of manipulation time for the BP Bulb was task-unrelated). The task-unrelated manipulations were labeled in ground truth as “other activity (OT)”, and we only had very limited training data for some short activities, such as EA or BP. Getting enough training data for these activities would require a great deal of manual coding work. The temperature measurement (EA) only takes around 10-20 seconds, which provided us with fifteen 5-second training samples. Collecting 500 training samples would require ground-truth coding of more than 30 resuscitations. For some short activities, even more cases would be required. The lack of training data for short activities explains why our system performed poorly for these activities. Unlike the previous deficiency, which is systemic and cannot be addressed by more training data, this deficiency can be addressed by acquiring sufficient training data.

6.3 Limitations and Extensions

A key limitation of our current system is that it relies only on relies on RFID sensing to capture activity information and making predictions. Some activities, such as palpation of the patient’s body, do not involve the use of physical objects that can be tagged. In addition, RFID technology does not work very well with metal objects or liquid containers, and objects in sterile packages can be tracked only until the packaging is discarded. Our continuing research involves the use of multimodal sensing for activity recognition. In particular, we are using the Kinect sensor with microphone array to capture depth images and ambient sound for more reliable and complete activity recognition.

Generalization is an important aspect of a classifier, and our system generalizes well for different attributes of trauma resuscitations. The cases we used for training and testing were performed by different trauma teams with patients having different injuries and health conditions. Unlike a model trained for image classification, a CNN model trained for RFID data cannot be directly used in a different environment with different antenna configuration or tagging strategy. For image analysis, a target appears similar regardless of the changing background or camera. On the other hand, the same activity captured in RFID data by different antenna configurations and tagging strategies may be rather different because the radio signal may experience very different conditions. As a result, the model has to be retained for different antenna configurations or tagging strategies. Input data that is less influenced by the hardware configuration, such as using the standard deviation of RSS instead of RSS values would partially solve this problem because standard deviation is lacking other information. RSS values depend on the distance between tag and reader antennas and the status of the tag (covered or exposed) while the standard deviation does not contain such information. Further investigation is needed to find the sensory input both robust to hardware configuration and representative enough to support activity recognition and better model generalization, which will be our future work.

It is challenging to achieve high precision in our application scenario because activates are relatively short (from 10 seconds to few minutes) compared with the entire trauma resuscitation (30 to 60 minutes). In addition to the 10 activities we monitored, tens of other activities occur during trauma resuscitation that we did not monitor, which we labeled as “other activity.” We did not manually remove their corresponding data from our testing set as has been done in other research [12,16], but these “extraneous” data may cause false predictions because in some cases an object may be used in the labeled activities as well as in “other activities.” Because $Precision = TP / (TP + FP)$, where TP denotes true positives and FP denotes false positives, if TP is smaller than FP then the precision is significantly influenced by the FP, and if TP is greater than FP the precision is significantly influenced by TP. For short activities (e.g. around 10 seconds for pupil examinations) a few seconds of error prediction will lead to significant precision drop compared with longer activities (e.g.

several minutes for cardiac lead placement). The problem is that during evaluation the resuscitation record is sliced into small segments and the prediction is performed for each segment independently, without preserving the continuity of activities. Another problem is a possible time offset between predictions and ground truth. To deal with inadequacies of traditional evaluation metrics, researches have designed a new set of metrics for activity recognition, such as frame metrics and 2SET metrics [41]. Implementing these evaluation metrics will be part of our future work.

A possible extension of our system involves tuning the prediction results based on domain-expert knowledge. For example, if several activities should follow certain sequential order, the final decision can be made based on both softmax score and constraints from expert knowledge. The challenge is that medical processes are complex and extracting precise and complete constrains from expert knowledge is often not feasible. Training the system to learn the features from RFID data and from ground-truth coding to generate useful constrains will be part of our future work.

7. CONCLUSION

This paper presents a deep learning system for complex teamwork activity recognition based on passive RFID data. Unlike existing systems that rely on manufactured features and a cascade of object-use detection followed by activity recognition, our system works directly with RFID data and produces multiclass classification of work activities or process phases. A deep learning approach generally supports scalable extension to include new activities without adding new classifiers. Using the data from actual trauma resuscitations, our deep learning achieved a 30% better *F*-score compared with existing research. Although our system achieved comparable performance to existing process-phase detection systems, our system has the advantages of not requiring special equipment or human cooperation with data acquisition. Our research demonstrated the feasibility of using passive RFID technology in fast-paced and privacy-sensitive complex application scenarios, and showed the power of deep learning for activity recognition based on passive RFID. We also analyzed the limitations of deep learning approach for RFID data and pointed to future improvements, particularly complementing RFID with other sensors and using multimodal deep learning.

ACKNOWLEDGMENTS

The authors are grateful to Jonathan Johannemann and Sen Yang for comments on the paper draft. This research was supported by the National Institutes of Health under Award Number R01LM011834.

9. REFERENCES

[1]. LeCun Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436–444. [PubMed: 26017442]

[2]. Krizhevsky Alex, Ilya Sutskever, and Hinton Geoffrey E.. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097–1105. 2012.

[3]. Karpathy Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725–1732. 2014.

[4]. Abdel-Hamid Ossama, Deng Li, and Yu Dong. "Exploring convolutional neural network structures and optimization techniques for speech recognition." In Interspeech, pp. 3366–3370. 2013.

[5]. Lane Nicholas D., and Petko Georgiev. "Can deep learning revolutionize mobile sensing?" In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, pp. 117–122. ACM, 2015.

[6]. Li Xinyu, Dongyang Yao, Xuechao Pan, Jonathan Johannaman, JaeWon Yang, Rachel Webman, Aleksandra Sarcevic, Ivan Marsic, and Burd Randall S.. "Activity Recognition for Medical Teamwork Based on Passive RFID." IEEE International Conference on RFID IEEE, 2016.

[7]. Philipose Matthai, Fishkin Kenneth P., Mike Perkowitz, Patterson Donald J., Dieter Fox, Henry Kautz, and Dirk Hahnel. "Inferring activities from interactions with objects." IEEE pervasive computing 3, no. 4 (2004): 50–57.

[8]. Mund Sumit. Microsoft Azure Machine Learning. Packt Publishing Ltd, 2015.

[9]. Abadi Martin, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Corrado Greg S. et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv: 1603.04467 (2016).

[10]. Forestier Germain, Laurent Riffaud, and Pierre Jannin. "Automatic phase prediction from low-level surgical activities." International journal of computer assisted radiology and surgery 10, no. 6 (2015): 833–841.

[11]. Yosinski Jason, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. "Understanding neural networks through deep visualization." arXiv preprint arXiv: 1506.06579 (2015).

[12]. Li Hanchuan, Ye Can, and Sample Alanson P. "IDSense: A human object interaction detection system based on passive UHF RFID." In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 2555–2564. ACM, 2015.

[13]. Yang Lei, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices." In Proceedings of the 20th annual international conference on Mobile computing and networking, pp. 237–248. ACM, 2014.

[14]. Wu Jianxin, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and Rehg James M.. "A scalable approach to activity recognition based on object use." In 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE, 2007.

[15]. Bardram Jakob E., Afsaneh Doryab, Jensen Rune M., Lange Poul M., Nielsen Kristian LG, and Petersen Søren T.. "Phase recognition during surgical procedures using embedded and body-worn sensors." In Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on, pp. 45–53. IEEE, 2011.

[16]. Parlak Siddika, Ivan Marsic, Aleksandra Sarcevic, Bajwa Waheed U., Waterhouse Lauren J., and Burd Randall S.. "Passive RFID for Object and Use Detection During Trauma Resuscitation." IEEE Transactions on Mobile Computing 15, no. 4 (2016): 924–937.

[17]. Felzenszwalb Pedro F., Girshick Ross B., David McAllester, and Deva Ramanan. "Object detection with discriminatively trained part-based models." IEEE transactions on pattern analysis and machine intelligence 32, no. 9 (2010): 1627–1645. [PubMed: 20634557]

[18]. Wu Chen, Amir Hossein Khalili, and Hamid Aghajan. "Multiview activity recognition in smart homes with spatio-temporal features." In Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, pp. 142–149. ACM, 2010.

[19]. Hinton Geoffrey, Li Deng, Dong Yu, Dahl George E., Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." IEEE Signal Processing Magazine 29, no. 6 (2012): 82–97.

[20]. Alsheikh Mohammad Abu, Dusit Niyato Lin Shaowei, Tan Hwee-Pink, and Zhu Han. "Mobile big data analytics using deep learning and apache spark." IEEE Network 30, no. 3 (2016): 22–29.

[21]. Chen Yuqing, and Yang Xue. "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer." In Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on, pp. 1488–1492. IEEE, 2015.

[22]. Speedway Revolution Quick Start Guide, available online: https://support.impinj.com/hc/en-us/articles/202755368-Speedway-Revolution-Quick-Start-Guide.

[23]. Han Jinsong, Han Ding, Chen Qian, Dan Ma, Wei Xi, Zhi Wang, Zhiping Jiang, and Longfei Shangguan. "Cbid: A customer behavior identification system using passive tags." In 2014 IEEE 22nd International Conference on Network Protocols, pp. 47–58. IEEE, 2014.

[24]. Abdel-Hamid Ossama, Li Deng, and Dong Yu. "Exploring convolutional neural network structures and optimization techniques for speech recognition." In Interspeech, pp. 3366–3370. 2013..

[25]. Lane Nicholas D., Petko Georgiev, and Lorena Qendro. "DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning." In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 283–294. ACM, 2015.

[26]. Ngiam Jiquan, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Ng Andrew Y.. "Multimodal deep learning." In Proceedings of the 28th international conference on machine learning (ICML-11), pp. 689–696. 2011.

[27]. Li Xinyu, Zhang Yanyi, et al. "Deep Neural Network for RFID Based Activity Recognition." Wireless of the Students, by the Students, and for the Students (S3) Workshop with MobiCom 2016, ACM, 2016.

[28]. Simonyan Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." In Advances in Neural Information Processing Systems, pp. 568–576. 2014.

[29]. LeCun Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86, no. 11 (1998): 2278–2324.

[30]. Simonyan Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv: 1409.1556 (2014).

[31]. Srivastava Nitish, Hinton Geoffrey E., Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15, no. 1 (2014): 1929–1958.

[32]. Kingma Diederik, and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv: 1412.6980 (2014).

[33]. Abadi Marfin, et al. "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015." Software available from tensorflow.org.

[34]. Powers David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).

[35]. Matthews Brian W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme." Biochimica et Biophysica Acta (BBA)-Protein Structure 405 2(1975): 442–451.

[36]. Blum Tobias, Nicolas Padoy, Hubertus Feußner, and Nassir Navab. "Modeling and online recognition of surgical phases using hidden markov models" In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 627–635. Springer Berlin Heidelberg, 2008.

[37]. Lara Oscar D., and Labrador Miguel A.. "A survey on human activity recognition using wearable sensors." IEEE Communications Surveys & Tutorials 15, no. 3 (2013): 1192–1209.

[38]. Deng Zhiwei, Mengyao Zhai, Lei Chen, Yuhao Liu, Srikanth Muralidharan, Mehrsan Javan Roshtkhari, and Greg Mori. "Deep structured models for group activity recognition." arXiv preprint arXiv: 1506.04191 (2015).

[39]. Deng Zhiwei, Arash Vahdat, Hexiang Hu, and Greg Mori. "Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition." arXiv preprint arXiv:1511.04196 (2015).

[40]. Maturana Daniel, and Sebastian Scherer. "Voxnet: A 3d convolutional neural network for real-time object recognition." In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 922–928. IEEE, 2015.

[41]. Ward Jamie A., Paul Lukowicz, and Gellersen Hans W.. "Performance metrics for activity recognition." ACM Transactions on Intelligent Systems and Technology (TIST) 2, no. 1 (2011): 6.

CCS Concepts

1.5.2 [Pattern Recognition]: Design Methodology–Classifier design & evaluation; C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems.

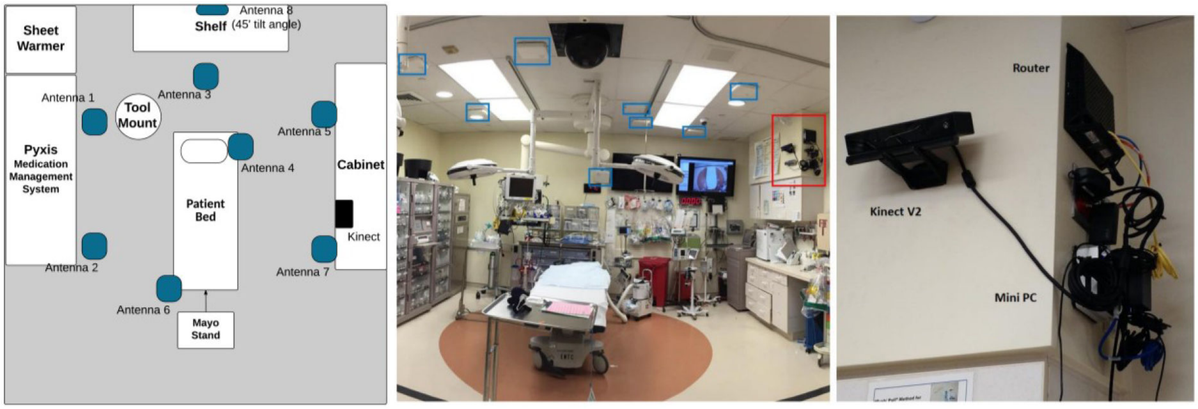

Fig. 1.

Left: Antennas 1 to 7 are mounted on the ceiling and facing down; Antenna 8 is mounted on the wall and facing 45° to the ground. Middle: A photo of the room with the antennas labeled with blue rectangles and the Kinect and Mini PC labeled with a red rectangle. Right: Zoom-in of the Kinect, router, and Mini PC.

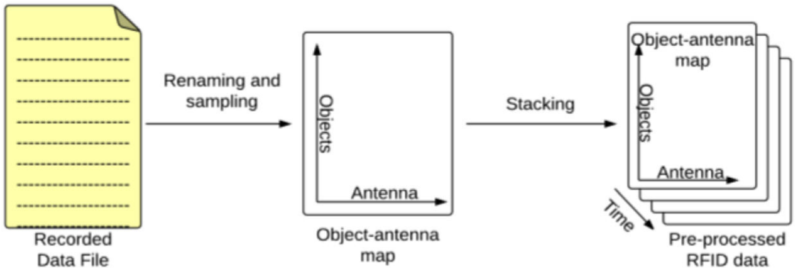

Fig. 2.
Our preprocessing procedure for RFID data.

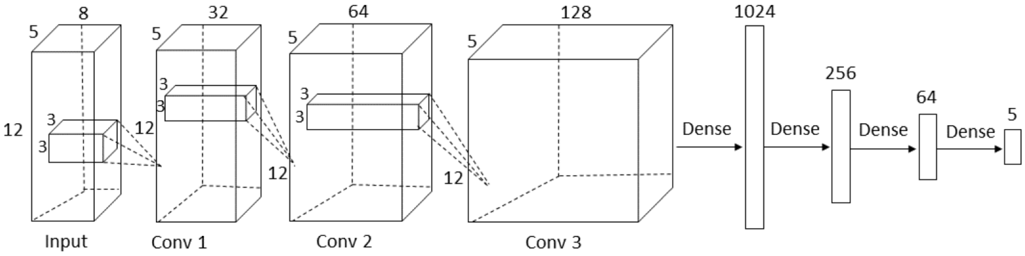

Fig. 3.

The convolutional neural network structure with 3 convolutional layers and 3 fully (dense) connected layers.

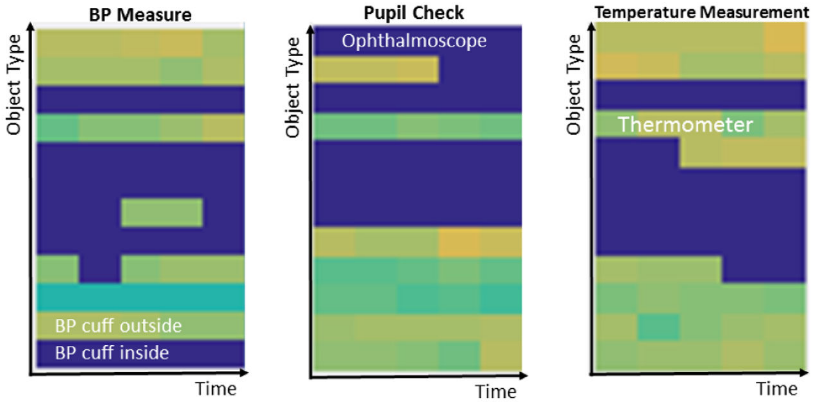

Fig. 4.

A visualization of selected RFID planes in the input layer. Each graphic highlights the objects used during a selected activity (labeled on top). Object types are along the vertical axis; time is along the horizontal axis. Note two "types" for the blood-pressure (BP) cuff, for inner and outer tags.

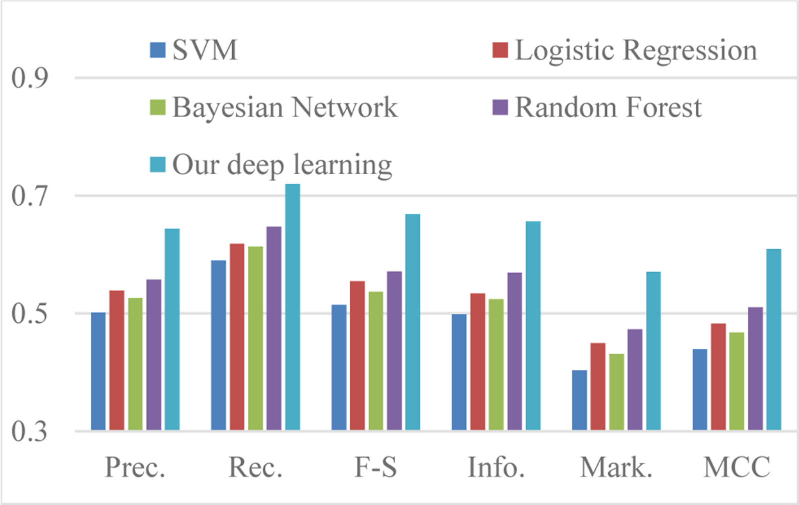

Fig. 5.

Performance comparison of using different classifiers for prediction of resuscitation phases. Performance metrics on the horizontal axis are introduced in the text.

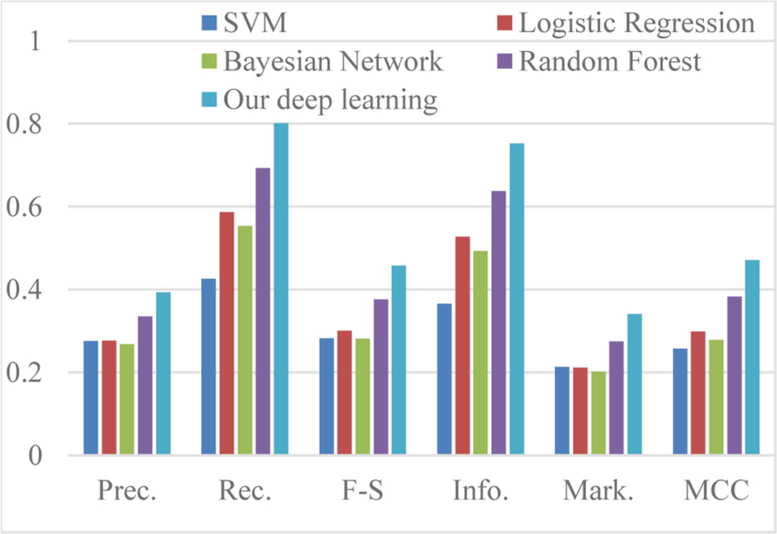

Fig. 6.

Comparison of results using different classifiers for resuscitation phase prediction.

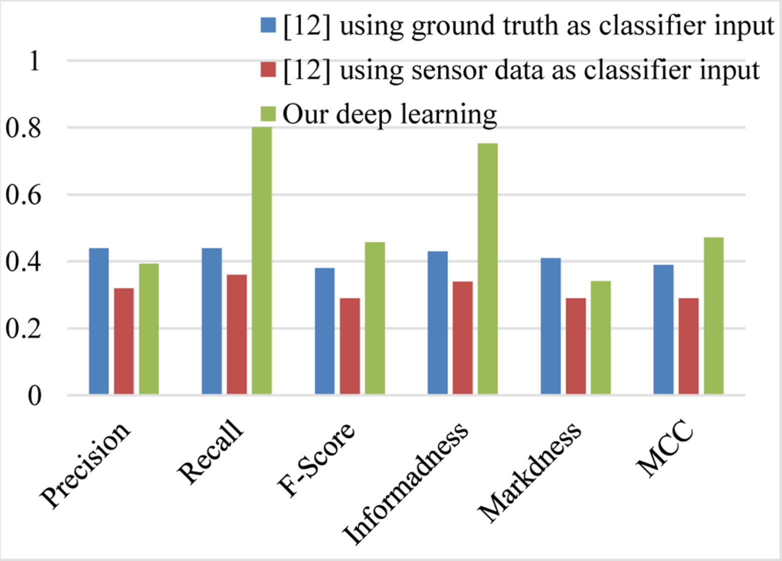

Fig. 7.

Comparison of results in [6] with our deep learning system in the same application environment.

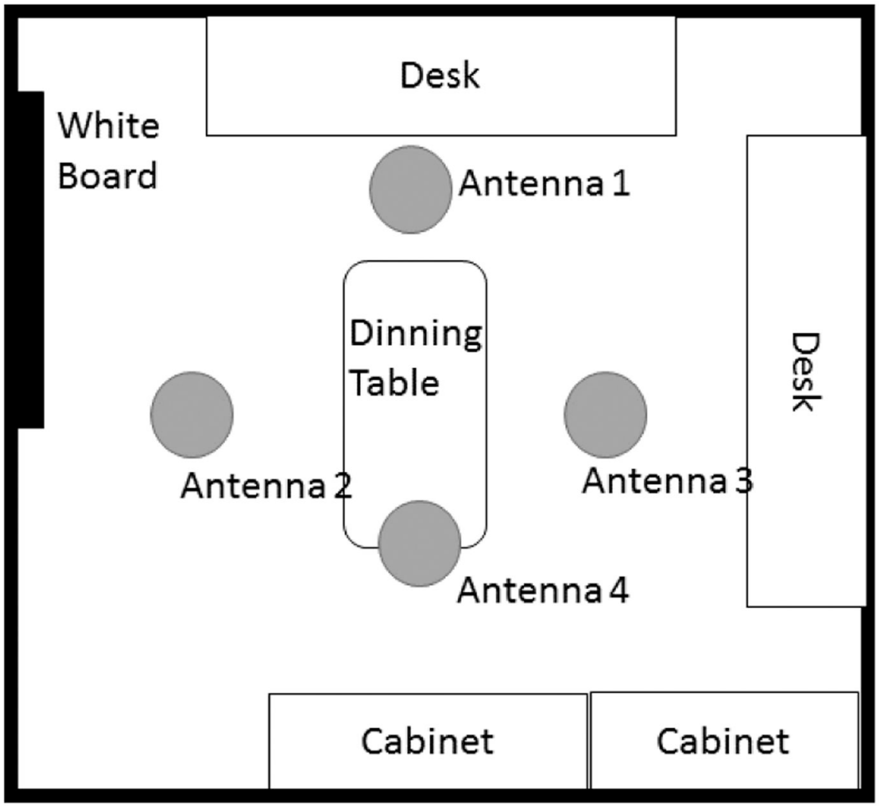

Fig. 8.

Room layout for a lab-activity recognition experiment.

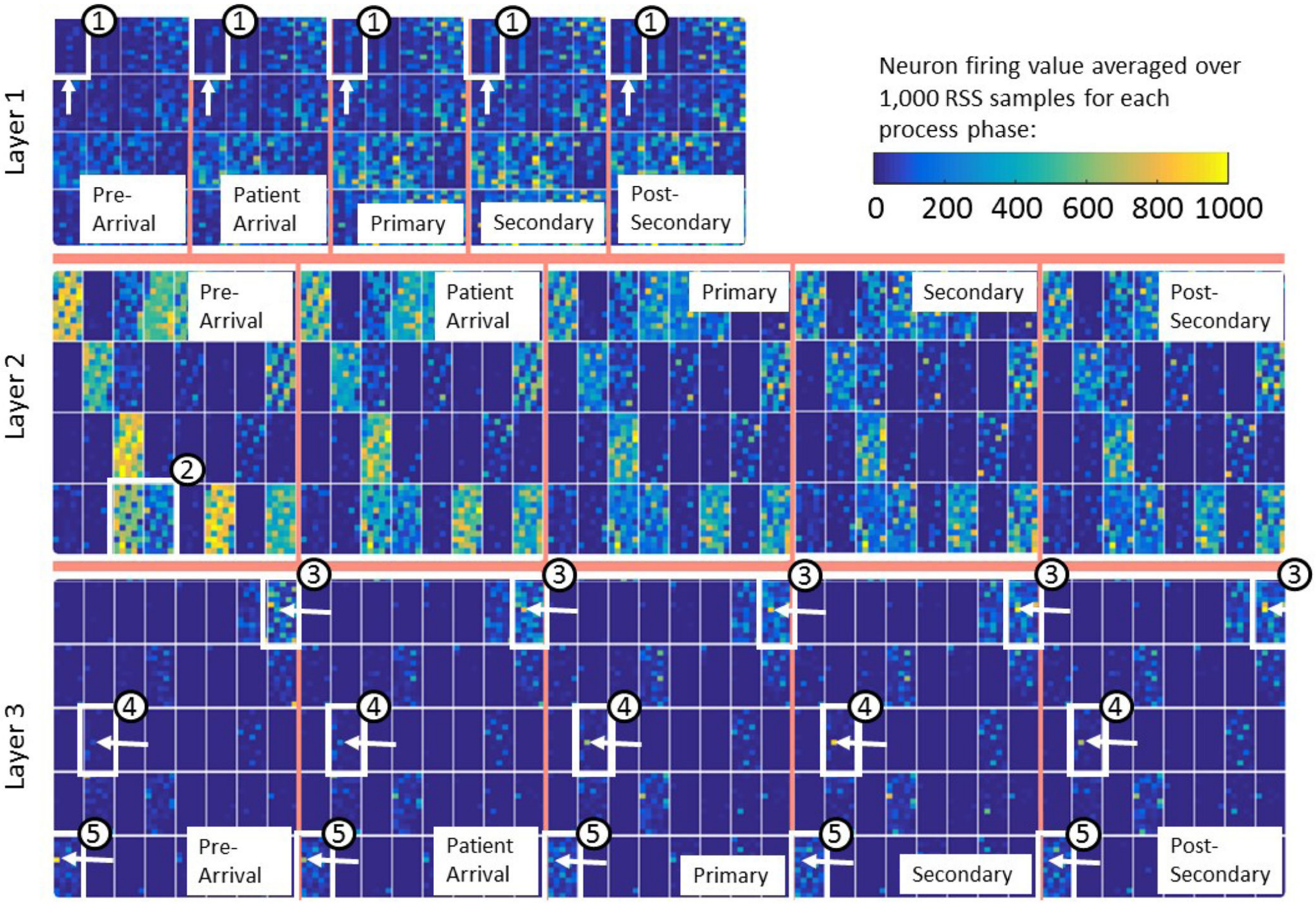

Fig. 9.

Example activation maps for selected planes from the three convolutional layers during process-phase detection. The five groups of activation maps in each of the three layers (separated by thick vertical lines) correspond to five resuscitation phases, as labeled in the images. (Please view the color version for better visibility.)

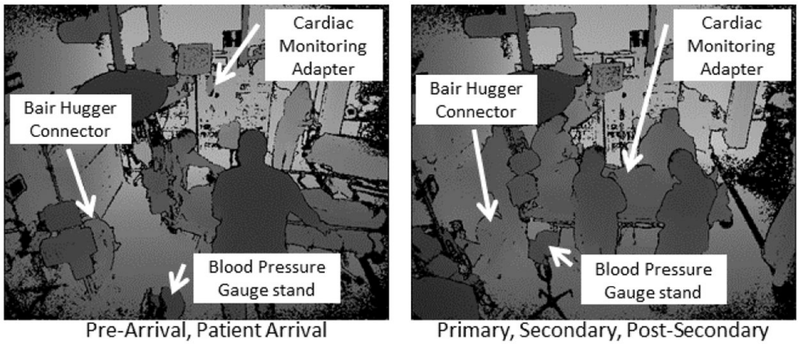
Cardiac Monitoring Adapter
Bair Hugger Connector
Blood Pressure Gauge stand
Pre-Arrival, Patient Arrival
Cardiac Monitoring Adapter
Bair Hugger Connector
Blood Pressure Gauge stand
Primary, Secondary, Post-Secondary

Fig. 10.

The positions of tagged objects during different process phases. The images were captured with Kinect depth sensor.

Table 1.

Activities used in this paper and their medical code.

Activity	Code	Activity	Code
Pulse Ox Placement	BA	Ear Exam	EAR
Oxygen Preparation	BC	Warm Sheet	EC
Blood Pressure Measurement	BP	Mouth Exam	M
Cardiac Lead Placement	CA	Nose Exam	N
Temperature Measurement	EA	Pupils Exam	PU

Table 2.

Comparison of performance and memory cost on process-phase detection for a neural network with different number of convolutional layers.

Num. of conv. layers →	**1**	**2**	**3**	**4**
Precision	0.55	0.61	0.64	0.65
Recall	0.66	0.69	0.72	0.71
F-Score	0.57	0.63	0.66	0.66
# of weights (millions)	2.2	4.3	8.7	18.6
Memory required for training (GB)	4.5	9.2	18.75	38

Table 3.

Confusion matrix for 5 resuscitation phases.

	PA	**A**	**P**	**S**	**PS**
PA	84.86%	4.85%	1.82%	3.58%	4.89%
A	**35.02%**	60.97%	1.79%	1.58%	0.63%
P	15.00%	7.47%	63.91%	10.33%	3.29%
S	8.01%	1.64%	6.53%	76.90%	6.91%
PS	9.74%	0.35%	2.67%	13.73%	73.51%

Table 4.

Comparison of performance of our deep learning network and existing systems for medical phase detection.

Process phase detection system	Acc.	Pre.	Rec.	F-S
Automatic phase detection from low-level surgical activities [10]	n/a	0.75	0.74	0.74
Modeling and online recognition of surgical phases using hidden Markov models [36]	83%	n/a	n/a	n/a
Phase recognition during surgical procedures using embedded and body-worn sensors [15]	77%	n/a	n/a	n/a
Our deep learning network	72%	0.63	0.70	0.65

Table 5.

Confusion matrix for recognition of 11 resuscitation activities. “OT” for activity *other* than selected 10 activities.

	OT	BA	BC	BP	CA	EA	EAR	EC	M	N	PU
OT	72.76%	0.66%	4.32%	2.56%	4.87%	5.64%	1.44%	7.06%	0.03%	0.05%	0.60%
BA	**13.22%**	85.90%	0.44%	0.00%	0.00%	0.00%	0.00%	0.44%	0.00%	0.00%	0.00%
BC	7.60%	0.79%	54.08%	4.62%	2.20%	5.38%	6.30%	**17.80%**	0.42%	0.08%	0.73%
BP	**20.33%**	2.40%	2.59%	64.14%	9.06%	0.00%	0.00%	0.18%	0.00%	0.00%	1.29%
CA	7.06%	0.00%	0.00%	0.00%	92.94%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
EA	**12.89%**	0.00%	2.22%	0.00%	0.00%	80.67%	3.33%	0.89%	0.00%	0.00%	0.00%
EAR	2.43%	0.00%	0.00%	0.00%	0.00%	0.00%	97.57%	0.00%	0.00%	0.00%	0.00%
EC	9.95%	0.90%	**21.24%**	2.16%	1.16%	3.66%	3.90%	56.82%	0.02%	0.02%	0.18%
M	6.15%	3.08%	**13.85%**	3.08%	0.00%	7.69%	0.00%	3.08%	63.08%	0.00%	0.00%
N	3.92%	1.96%	0.00%	0.00%	5.88%	5.88%	1.96%	0.00%	0.00%	76.47%	3.92%
PU	8.59%	0.00%	**10.10%**	4.04%	3.03%	5.56%	3.54%	4.55%	0.51%	0.51%	59.60%

Table 6.

Comparison of different activity recognition systems used in real-world scenarios.

Activity Recognition System	Input Source	Approach	Accuracy of Activity Recognition
A Scalable Approach to Activity Recognition based on Object Use [14]	Video data and RFID	Dynamic Bayesian Network	60.84% average accuracy for 16 single-person daily-life activities, without using domain knowledge
Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition [39]	Images	Recurrent Neural Networks	81.2% average accuracy for 5 real-life group activities
Deep Structured Models for Group Activity Recognition [38]	Images	Convolutional Neural Network	80.6% average accuracy for 5 real-life group activities
Our deep learning network	RFID	Convolutional Neural Network	80.2% average accuracy for 10 group activities during actual resuscitations

Table 7.

Confusion matrix for 6 lab activities.

	P	E	LM	WB	R	NA
P	99%	0%	0%	0%	1%	0%
E	1%	92%	0%	7%	0%	0%
LM	0%	9%	91%	0%	0%	0%
WB	11%	0%	0%	85%	0%	4%
R	2%	0%	0%	0%	78%	20%
NA	0%	0%	0%	0%	0%	100%