Neurocomputing 194 (2016) 279–287

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep Neural Networks for wireless localization in indoor and outdoor environments

Wei Zhang [a,1], Kan Liu [a,*,1], Weidong Zhang [a], Youmei Zhang [a], Jason Gu [b]

[a] School of Control Science and Engineering, Shandong University, China
[b] Department of Electrical and Computer Engineering, Dalhousie University, Canada

ARTICLE INFO

Article history:
Received 2 September 2015
Received in revised form
8 December 2015
Accepted 18 February 2016
Available online 6 March 2016

Keywords:
Wireless positioning
Deep Neural Networks (DNNs)
Hidden Markov model (HMM)
Deep Learning
Stacked Denoising Autoencoder (SDA)

ABSTRACT

In this paper, we propose a wireless positioning method based on Deep Learning. To deal with the variant and unpredictable wireless signals, the positioning is casted in a four-layer Deep Neural Network (DNN) structure pre-trained by Stacked Denoising Autoencoder (SDA) that is capable of learning reliable features from a large set of noisy samples and avoids hand-engineering. Also, to maintain the temporal coherence, a Hidden Markov Model (HMM)-based fine localizer is introduced to smooth the initial positioning estimate obtained by the DNN-based coarse localizer. The data required for the experiments is collected from the real world in different periods to meet the actual environment. Experimental results indicate that the proposed system leads to substantial improvement on localization accuracy in coping with the turbulent wireless signals.

1. Introduction

The problem of position estimation, including indoors and outdoors, has enjoyed massive attention in industry and academia. Especially, the successful application of Global Positioning System (GPS) enable people to travel around the world freely. However, GPS is sensitive to occlusion and cannot work in indoor environments. By measuring the intensity of the received signal, positioning with the wireless signal like Wi-Fi makes interior localization possible.

Measurement-based wireless positioning system is to infer the position based on Time of Arrival (TOA) [1] or Time Difference of Arrival (TDOA) [2] of the signals. However, general wireless signal receivers are not capable to measure round-trip time or angle. It needs additional devices which makes this system impractical in applications. In contrast, the fingerprint-based approaches [3,4] do not need any special device and thus are more feasible. For example, the proposed fingerprint method consists of two stages as illustrated in Fig. 1. At the offline stage, the Received Signal Strength Indications (RSSIs) from all Access Points (APs) are collected with known positions, referred as Reference Points (RPs), to build a fingerprint database about the environment. So each RP has a fingerprint characterized by its position and the captured RSSIs from all APs at that location. At the positioning stage, the currently captured RSSIs are matched with those of the RPs and the position is determined using the positions of several best fitted RPs.

The pioneering fingerprint-based positioning system was proposed in [3] which relies on K Nearest Neighbors (KNN) to find the best matches from the fingerprint database. Bayesian-based filtering methods were then proposed in [5–8] to enhance the positioning robustness. Later, Support Vector Machine (SVM) [4] and Compressive Sensing (CS) [9,10] were incorporated to correlate the RSSIs with the samples of the fingerprint database. Transfer learning was also introduced by Pan et al. [11] to reduce the burden in fingerprint database collection. Recently, Li et al. [12] proposed a shunting Short Term Memory based method to solve the localization problem in dynamic environments. A multi-hop method was proposed by De et al. [13] to target a accurate solution of the localization problem.

However, the major problem for accurate fingerprint-based localization comes from the variation of RSSIs due to the fluctuating nature of wireless signal, such as multipath fading and attenuation by static or dynamic objects like walls or moving people. Fig. 2 shows the variation of RSSIs from three APs at a fixed position over time. It appears that the RSSIs change frequently over a wide range. Moreover, it is necessary to collect more RPs for accurate positioning especially when the environment turns to be large, thus the built fingerprint database tends to be tremendous. Consequently, the challenge in wireless positioning is how to

* Corresponding author.
E-mail address: sakuraxiafan@gmail.com (K. Liu).
[1] The first two authors contributed equally.

http://dx.doi.org/10.1016/j.neucom.2016.02.055
0925-2312/

extract reliable features and find good mapping function from massive RPs with widely fluctuating RSSI signals. The aforementioned methods essentially belong to shallow learning architectures which have limited modeling and representational power when dealing with such big and noisy data problem.

To extract complex structure and build internal representation from such rich data, human information processing mechanisms suggest the Deep Learning architecture with multiple layers of nonlinear processing stages [14]. Deep Learning simulates the hierarchical structure of human brain, processing data from low level to high-level and gradually producing more and more semantic concepts. Deep Neural Networks (DNNs) have been applied to tackle such kinds of problems with notable success, beating state-of-the-art techniques in certain areas such as vision [15–18], audio [19–21] and robotics [22,23]. However, the question

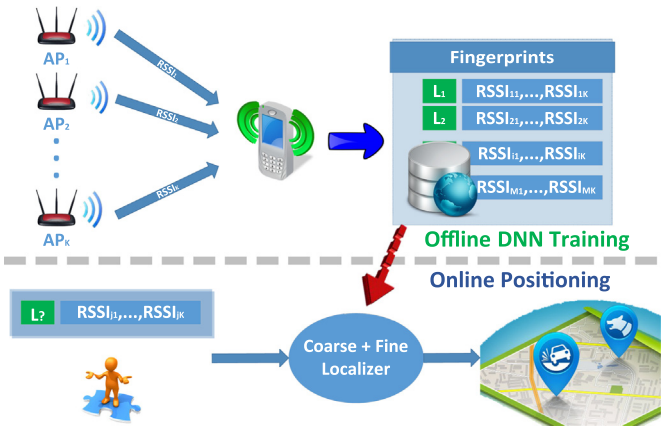


Fig. 1. Illustration of the proposed fingerprint-based Wi-Fi positioning system.

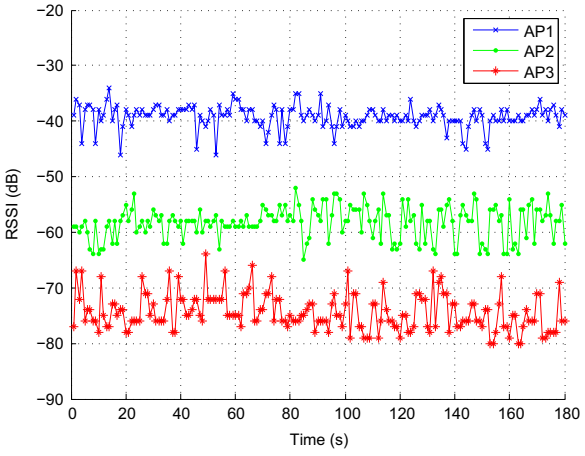


Fig. 2. The variation of RSSIs from three APs at a fixed position over time.

of applying DNNs to Wi-Fi localization has largely remained unanswered.

In this paper, we attempt to cast a light on this question and present a novel Wi-Fi positioning method based on Deep Learning. More specifically, we conduct the position estimation through feature learning and show how to cast it in DNN settings. A four-layer DNN architecture is built to extract features from massive widely fluctuating Wi-Fi data. The training is conducted with Stacked Denoising Autoencoder (SDA) for stable initialization and Backpropagation (BP) for global fine-tuning. There are several advantages for this formulation. First, the DNN learns reliable high-level features automatically from a large set of widely fluctuating RSSI samples and avoids hand-engineering. Second, this structure is capable of learning useful features directly from both unlabeled and labeled data. Third, the DNN-based estimator is more robust as it predicts position utilizing the extracted high-level features. Moreover, the structure has the advantage in handling massive RSSI samples since the prediction at positioning stage does not rely on any searching in sample space but requires only the forward evaluation of a trained feed-forward neural network.

Besides, temporal coherence plays an important role in localization. By assuming the positioning result at the current time instant to be similar to the ones observed at previous time instants, the estimation process should be regularized to make the result stable. In this work, a fine localizer is introduced based on Hidden Markov Model (HMM) to further refine the initial positioning estimate obtained from the DNN-based coarse localizer. The proposed positioning system is summarized in Fig. 3. For the offline DNN training, a four-layer DNN-based coarse localizer is trained to extract reliable features from massive noisy RSSI samples of a pre-built fingerprint dataset. For the online positioning, the pre-processed RSSI readings are fed into the coarse localizer to get a probabilistic position estimate, and then refined to produce the final position with a HMM-based fine localizer.

2. Related work

Several papers have addressed how to estimate the location using Wi-Fi signal or radio frequency identification (RFID) in the past years. Comparing with the RFID-based positioning systems [24,25], Wi-Fi is considered more suitable in the complex environment.

Measurement-based methods, which measure Time of Arrival (TOA) [1], Time Difference of Arrival (TDOA) [2] or use other methods [26–28] to calculate the location based on trigonometry. Unlike the measurement-based methods, fingerprint-based methods can be easily implemented without any additional hardware.

Bahl and Padmanabhan [3] relied on KNN to find matches from the fingerprint database, which is sensitive to the Wi-Fi noise. To enhance the positioning robustness, Bayesian-based filtering methods were proposed in [5–8]. However, the adoption of filter

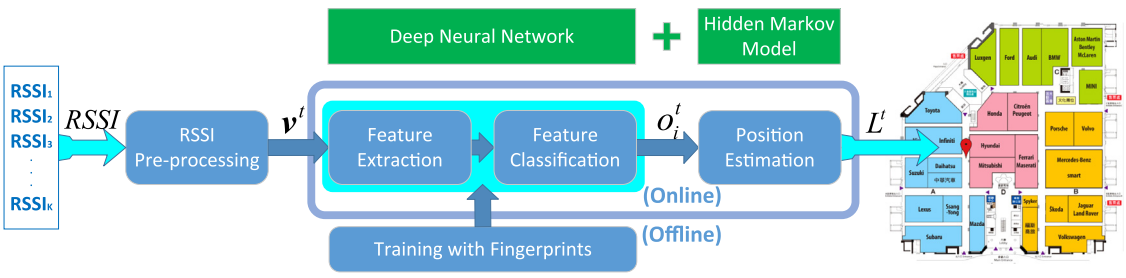


Fig. 3. Structure of the proposed cascade Wi-Fi positioning system. For the offline DNN training, a four-layer DNN-based coarse localizer is trained to extract reliable features from massive noisy RSSI samples of a pre-built fingerprint dataset. For the online positioning, the pre-processed RSSI readings are fed into the coarse localizer to get a probabilistic position estimate, and then smoothed with a HMM-based fine localizer to obtain the final result.

influences the traceability of positioning system which has certain requirements for real-time localization and it is hard to reach a compromise between the robustness and traceability. Wu et al. [4] proposed a SVM-based positioning system which formulates the localization task as a classification problem. Inspired by the adaptability, a shunting STM (Short Term Memory) based method was proposed by Li et al. [12] to solve the localization problem in dynamic environments. De et al. [13] proposed a multi-hop method to target a accurate solution of the localization problem.

The CS-based positioning method [29,10] regard the fingerprint database as a dictionary and reconstruct the position with the weight coefficients learned from all dictionary columns. Transfer learning was introduced by Pan et al. [11] to reduce the burden in fingerprint database collection. There also exists some methods like [30–32] which attempted to utilize shallow neural networks to fingerprint-based positioning problem. However, similar to the aforementioned algorithms, the shallow architecture limits the fitting ability and feature extraction performance is susceptible to the signal fluctuation. Some other work [33] proposed to improve the accuracy of fingerprint method by introducing user intervention.

In this work, a deep learning structure is built to extract features from massive widely fluctuating Wi-Fi data, and performs probabilistic position estimate automatically. Besides, an HMM-based fine localizer is introduced to formulate the relationship between adjacent position states and relieve the estimate variation intuitively. The integration of DNN and HMM first appeared in [21] for the speech recognition task, where restricted Boltzmann machine (RBM) is used to pre-train the model. In this work, to deal with the widely fluctuating wireless signals, SDA is advocated and investigated to reconstruct clean features that reveal higher level representations for the noisy inputs.

3. The proposed Wi-Fi positioning system

We consider a typical Wi-Fi positioning scenario of K connected APs and a mobile device equipped with a wireless adapter card taking RSSI measurements from all APs in the environment. As shown in Fig. 3, the localization problem in this paper is to estimate the position $\boldsymbol{L}^t$ of the mobile device from the RSSI observations as follows:

$$\boldsymbol{L}^t = F(\boldsymbol{v}^t) \tag{1}$$

where $\boldsymbol{L}^t$ stands for the position of the mobile device at the time instant t. $\boldsymbol{v}^t \in \mathbb{R}^K$ is the pre-processed RSSI vector at the time instant t of which the jth element r_j^t represents the observed RSSI value of the jth AP. The RSSI measurement interval is set to 1 s according to the frequency of the RSSIs. The function F, which is learned from a pre-built fingerprint database, represents the nonlinear mapping between the RSSI values and the position of the mobile device.

Particularly, a grid-based representation of the localization area is adopted where each cell with its central coordinates corresponds to a position in the localization area. The built fingerprint database is a $M \times (K+1)$ matrix of M vectors $\{\tilde{l}_i, \tilde{\boldsymbol{v}}_i, i = 1, \ldots, M\}$, where M is the size of the database. Each vector consists of a one-dimensional label $\tilde{l}_i$ of discrete position and the K-dimensional RSSI vector $\tilde{\boldsymbol{v}}_i$ observed from corresponding grid at the offline stage. The RSSI vector $\tilde{\boldsymbol{v}}_i$ contains K elements $\tilde{v}_{ij}$ $(j = 1, \ldots, K)$, and each represents the observed RSSI value of the jth AP. As depicted in Fig. 3, the proposed positioning system consists of the following three parts:

- *RSSI pre-processing*: The raw value of a AP's RSSI is rectified between -100 dBm and -30 dBm. The greater the value is, the less the distance between the AP and the observer will be. The task of pre-precessing is to normalize the raw RSSI readings at time instant t to the range between 0 and 1 based on $\boldsymbol{v}^t = (\boldsymbol{RSSI} + 100)/70$.
- *Offline DNN training*: The DNN-based coarse localizer is trained on the pre-processed RSSI readings of the fingerprint database. For the optimization, SDA is first trained to initialize the DNN with both labeled and unlabeled fingerprints which are obtained with known or unknown positions. BP is then performed to fine-tune the model with only labeled fingerprints.
- *Online positioning*: The online positioning consists of DNN-based coarse localizer and HMM-based fine localizer.
  - *Coarse positioning*: With the trained DNN, a more discriminative and concise representation of the RSSI readings can be obtained. The outputs of the coarse localizer are the probabilities of all the grid-based positions.
  - *Fine positioning*: Further refinement can be achieved by taking the temporal coherence into account and maintain a smooth transition between two adjacent positioning estimates. A HMM-based localizer is introduced to refine the coarse positioning estimate.

3.1. DNN structure and training

The proposed coarse localizer is a DNN-based probabilistic estimator which consists of four layers. Fig. 4(a) shows the proposed network with a $K-200-200-200-N$ structure, where the number of the output units N is the total number of discrete squared positions. The input of the DNN-based coarse localizer is the pre-processed K-dimensional vector $\boldsymbol{v}^t = \{v_i^t, i = 1, \ldots, K\}$. The

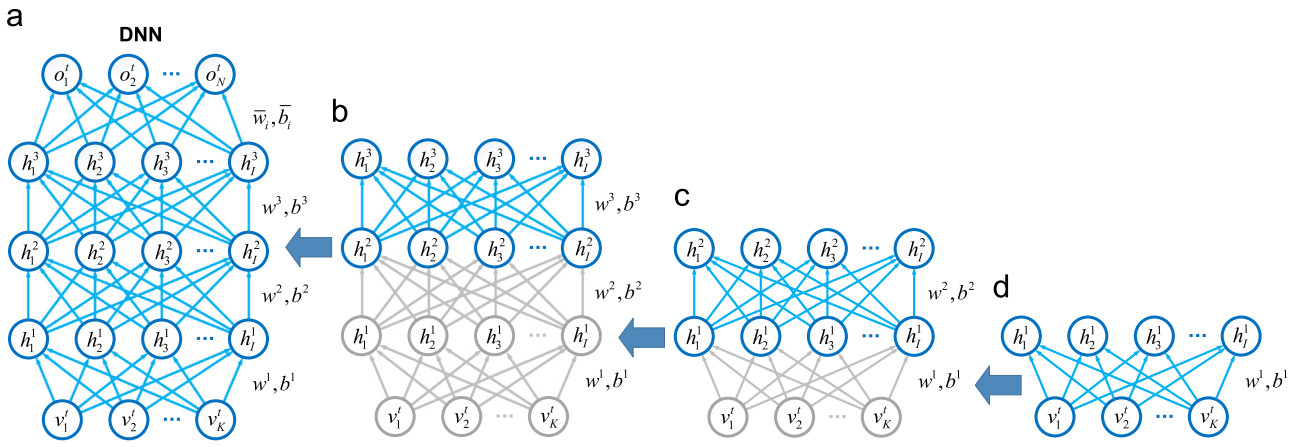


Fig. 4. The proposed DNN structure.

labeled fingerprints consist of the K ordered RSSI readings observed from known positions while the unlabeled fingerprints are from unknown positions. The network contains three hidden layers with 200 hidden units $\{h_i^j, i=1,\ldots,200, j=1,\ldots,3\}$ each layer. The number of the hidden layers 3 and the number of the hidden units 200 here are experimental values for this particular task. A softmax regression layer is added on the top of the hidden layers with N-dimensional outputs $\{o_i^t, i=1,\ldots,N\}$, i.e., the probability estimates of all discrete positions at time t.

Although shallow neural networks can be well trained with the BP method, the difficult optimization of going beyond one or two hidden layers for DNN has long prevented obtaining an expected result. This situation has been changed with the successful approach proposed by Hinton et al. [34], in which an unsupervised RBM training is performed with contrastive divergence layer-by-layer before the global BP fine-tuning. This method relies on the energy model with the likelihood derived criteria and avoids getting stuck in the kind of poor solutions one typically reaches with random initializations.

As a competitive alternative to RBM, Denoising Autoencoder (DA) [35] is a stochastic version of the autoencoder which reconstructs the input into a partially destroyed and corrupted version. Moreover, DA can be stacked to construct a deep network called Stacked Denoising Autoencoder (SDA) [36], in which the output of DA in the lower layer is used as the input of the higher layer. Unlike the contrastive divergence training of RBM, SDA uses denoising as a training criterion to learn stable and robust representation from corrupted and noisy inputs. The representations extracted layer by layer using a purely unsupervised local demonising criterion proved to make the subsequent classification better and easier. Hence, SDA is advocated to pre-train the proposed DNN model before BP global tuning.

As shown in Fig. 5, some parts of the DA input $\boldsymbol{v}$ are randomly selected and forced to be zero to generate the corrupted input $\boldsymbol{v}'$ in the pre-training phase of the coarse localizer, in order to remove the effect of random noise in RSSIs. The hidden representation $\boldsymbol{h}$ of the DA is then obtained from $\boldsymbol{v}'$ to calculate the reconstructed input $\boldsymbol{u}$, which can be formulated as

$$\boldsymbol{h} = \frac{1}{1+\exp(-\boldsymbol{w}\boldsymbol{v}' - \boldsymbol{b})} \tag{2}$$

$$\boldsymbol{u} = \frac{1}{1+\exp(-\boldsymbol{w}'\boldsymbol{h} - \boldsymbol{b}')} \tag{3}$$

where $\boldsymbol{w}$ and $\boldsymbol{w}'$ are connection weights between the visible layer and hidden layer. $\boldsymbol{b}$ denotes the hidden units biases while $\boldsymbol{b}'$ denotes the visible units biases. $\boldsymbol{h}$ represents the hidden units. The training of DA is to minimize the reconstruction error between the $\boldsymbol{v}$ and the $\boldsymbol{u}$. Note that the input $\boldsymbol{v}$ of DA can be any hidden layer of the DNN.

As depicted in the last three subgraphs of Fig. 4, the SDA is learnt in a greedy layer-wise fashion from all the labeled and unlabeled fingerprints of the database using stochastic gradient descent to minimize the squared error between the uncorrupted inputs and reconstructed outputs. The corruption step is a masking noise process, i.e. each active input has a probability (e.g., 0.1) to be set to 0.

Once the SDA has been trained, their parameters describe multiple levels of representation for the RSSIs and can be used to initialize the DNN for the global BP optimization. Particularly, we employ the stochastic gradient descent method for the BP fine-tuning with the labeled fingerprints from the database.

To prevent overfitting when training the DNN localizer with the fingerprint database, dropout [37] is employed and some feature detectors of each layer (e.g., 5%) are randomly omitted during BP training.

3.2. Online positioning

The online positioning phrase consists of two steps: the coarse positioning based on DNN and fine positioning based on HMM. For the coarse positioning, DNN is employed to extract the wireless signal features from $\boldsymbol{v}^t$ and calculate the probability of the current position at each grid. The extraction is a process of extracting RSSI features between two adjacent layers in a bottom-up manner as follows:

$$\boldsymbol{h}^1 = 1/(1+\exp(-\boldsymbol{w}^1\boldsymbol{v}^t - \boldsymbol{b}^1)) \tag{4}$$

$$\boldsymbol{h}^2 = 1/(1+\exp(-\boldsymbol{w}^2\boldsymbol{h}^1 - \boldsymbol{b}^2)) \tag{5}$$

$$\boldsymbol{h}^3 = 1/(1+\exp(-\boldsymbol{w}^3\boldsymbol{h}^2 - \boldsymbol{b}^3)) \tag{6}$$

where $\boldsymbol{w}^j$ and $\boldsymbol{b}^j$ are the trained weights and biases of the hidden layer $\boldsymbol{h}^j$, respectively.

The goal of coarse positioning is to calculate $P(\boldsymbol{L}^t = l_i | \boldsymbol{v}^t), (i=1,\ldots,N)$ which represents the probability of the device's position in l_i at time instant t given the current observations $\boldsymbol{v}^t$. l_i stands for the label of the ith gridded position. Hence, softmax regression is performed at the output layer o^t to evaluate the extracted high-level features and yield the probabilistic position estimates.

$$P(\boldsymbol{L}^t = l_i | \boldsymbol{v}^t) = o_i^t = \frac{\exp(-\overline{\boldsymbol{w}}_i\boldsymbol{h}^3 - \overline{b}_i)}{\sum_i \exp(-\overline{\boldsymbol{w}}_i\boldsymbol{h}^3 - \overline{b}_i)} \tag{7}$$

where $\overline{\boldsymbol{w}}_i$ denotes the connection weights between the third hidden layer $\boldsymbol{h}^3$ and the output layer o. $\overline{b}_i$ is the bias of the output layer. Thus we obtain the position estimate given the pre-processed RSSI values at time instant t.

Moreover, the position at the current time instant should be similar to the ones observed at previous time instants. This property motivates the adoption of a HMM-based fine localizer to further improve the coarse estimate and enforce a continuous temporal coherence of positioning result. In this paper, we rely on the first-order Markov assumption to govern the smooth transition between position estimates. As shown in Fig. 6, HMM indicates that the current state (position) $\boldsymbol{L}^t$ depends on its previous states. Specifically, considering the positioning time delay and the final accuracy, we associate the current observed RSSIs $\boldsymbol{v}^t$ with its six prior RSSIs $\boldsymbol{v}^{t-6},\ldots,\boldsymbol{v}^{t-1}$ and six posterior RSSIs $\boldsymbol{v}^{t+1},\ldots,\boldsymbol{v}^{t+6}$ to estimate the probability $P^t(\boldsymbol{L}^t = l_i)$ based on HMM with

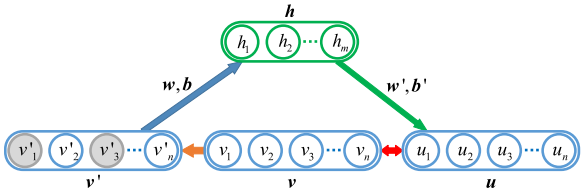


Fig. 5. An example of DA structure which reconstructs the input into a partially destroyed and corrupted version.

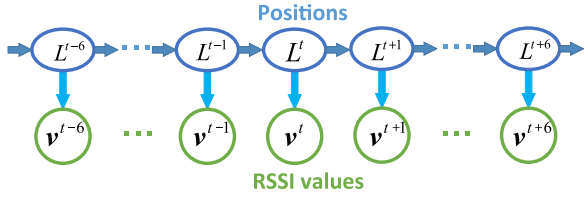


Fig. 6. The proposed HMM fine localizer to enforce smooth transition among adjacent states.

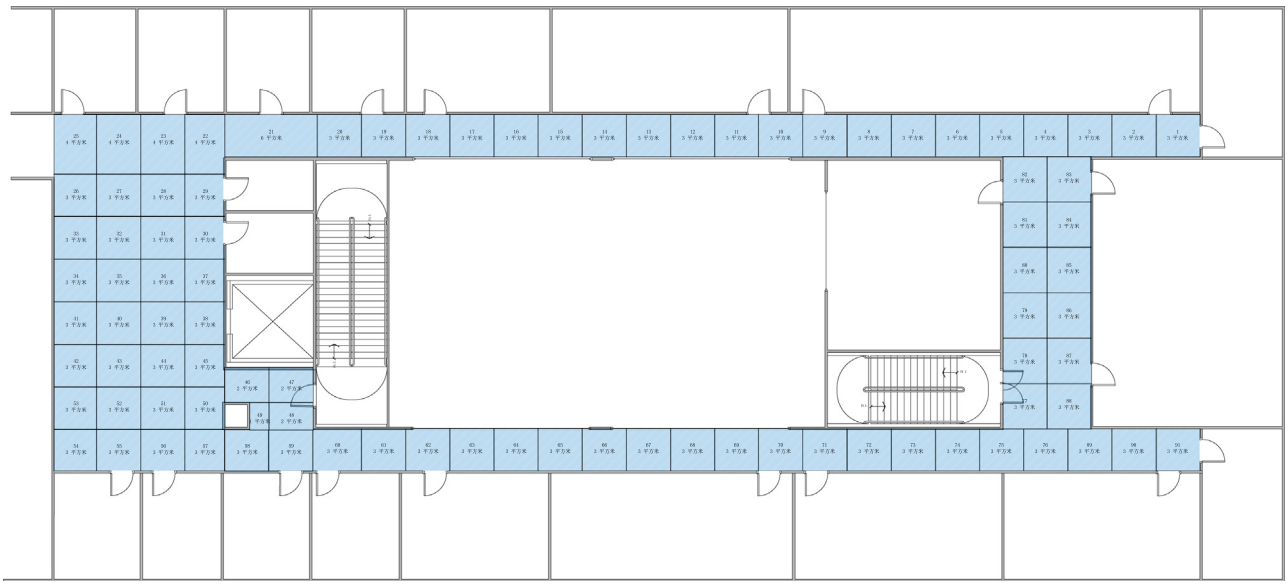


Fig. 7. Indoor environment. It is divided into 91 grids with 1.8 m × 1.8 m each, and 163 APs were observed.

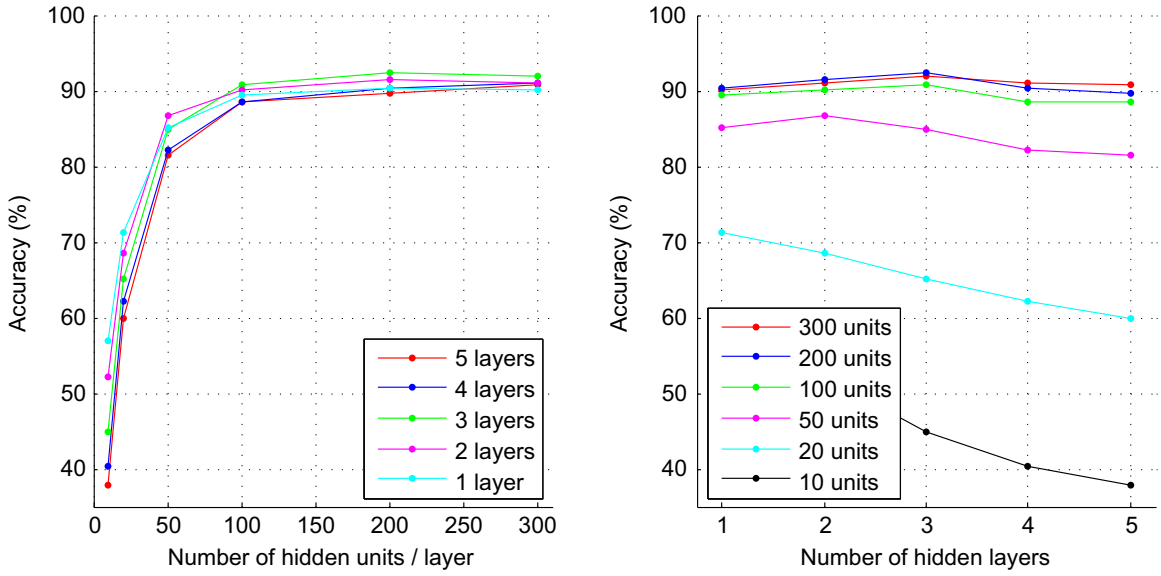


Fig. 8. Left: the indoor localization accuracy of the DNN-based coarse localizer with respect to the number of hidden units. Right: the indoor localization accuracy of the DNN-based coarse localizer with respect to the number of hidden layers.

parameter $\boldsymbol{\theta}$ as follows:

$$P^t(\boldsymbol{L}^t = l_i) = P(\boldsymbol{L}^t = l_i | \boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6}, \boldsymbol{\theta}) = \frac{P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6}, \boldsymbol{L}^t = l_i | \boldsymbol{\theta})}{P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6} | \boldsymbol{\theta})}$$
$$= \frac{P^t_\alpha(\boldsymbol{L}^t = l_i) P^t_\beta(\boldsymbol{L}^t = l_i)}{\sum_{i=1}^{N} P^t_\alpha(\boldsymbol{L}^t = l_i) P^t_\beta(\boldsymbol{L}^t = l_i)} \quad (8)$$

where the product $P^t_\alpha(\boldsymbol{L}^t = l_i)P^t_\beta(\boldsymbol{L}^t = l_i)$ is the probability of the $\boldsymbol{L}^t = l_i$ with the observation sequence $\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6}$ which can be calculated by

$$P^t_\alpha(\boldsymbol{L}^t = l_i)P^t_\beta(\boldsymbol{L}^t = l_i) = P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6}, \boldsymbol{L}^t = l_i | \boldsymbol{\theta})$$
$$= P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^{t+6} | \boldsymbol{L}^t = l_i, \boldsymbol{\theta}) P(\boldsymbol{L}^t = l_i | \boldsymbol{\theta}) = P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^t | \boldsymbol{L}^t$$
$$= l_i, \boldsymbol{\theta}) P(\boldsymbol{L}^t = l_i | \boldsymbol{\theta}) P(\boldsymbol{v}^{t+1}, \ldots, \boldsymbol{v}^{t+6} | \boldsymbol{L}^t = l_i, \boldsymbol{\theta}) = P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^t, \boldsymbol{L}^t$$
$$= l_i | \boldsymbol{\theta}) P(\boldsymbol{v}^{t+1}, \ldots, \boldsymbol{v}^{t+6} | \boldsymbol{L}^t = l_i, \boldsymbol{\theta}). \quad (9)$$

In (9), $P^t_\alpha(\boldsymbol{L}^t = l_i)$ represents the probability of $\boldsymbol{L}^t$ in grid l_i with the observation sequence $\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^t$, while $P^t_\beta(\boldsymbol{L}^t = l_i)$ is the probability of the observation sequence $\boldsymbol{v}^{t+1}, \ldots, \boldsymbol{v}^{t+6}$ given $\boldsymbol{L}^t$ in grid l_i. We utilize the forward algorithm (10) and backward algorithm (11) to calculate the $P^t_\alpha(\boldsymbol{L}^t = l_i)$ and $P^t_\beta(\boldsymbol{L}^t = l_i)$ respectively.

$$P^\tau_\alpha(\boldsymbol{L}^\tau = l_i) = P(\boldsymbol{v}^{t-6}, \ldots, \boldsymbol{v}^\tau, \boldsymbol{L}^\tau = l_i | \boldsymbol{\theta}) = \left[\sum_{j=1}^{N} P^{\tau-1}_\alpha(j) A_{ji} \right] B_{i\tau},$$
$$\forall \tau \in \{t-5, t-4, \ldots, t\} \quad (10)$$

$$P^\tau_\beta(\boldsymbol{L}^\tau = l_i) = P(\boldsymbol{v}^{\tau+1}, \ldots, \boldsymbol{v}^{t+6} | \boldsymbol{L}^\tau = l_i, \boldsymbol{\theta}) = \sum_{j=1}^{N} P^{\tau+1}_\beta(j) A_{ij} B_{j(\tau+1)},$$
$$\forall \tau \in \{t, t+1, \ldots, t+5\} \quad (11)$$

where $\boldsymbol{\theta}$ consists of $\boldsymbol{l}$, $\boldsymbol{V}$, $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{\pi}$. $\boldsymbol{A}$ is the state transition matrix, and A_{ij} $(i,j = 1, \ldots, N)$ represents the probability of the device's position transition from l_i of current time instant to l_j at next time instant. The state transition matrix $\boldsymbol{A}$ can be determined by the layout of the location area. In our positioning system, the A_{ij} is set to 1 if l_j is physically adjacent to l_i and set to zero otherwise. Hence, the sum of the transition probability from l_i to each position equals

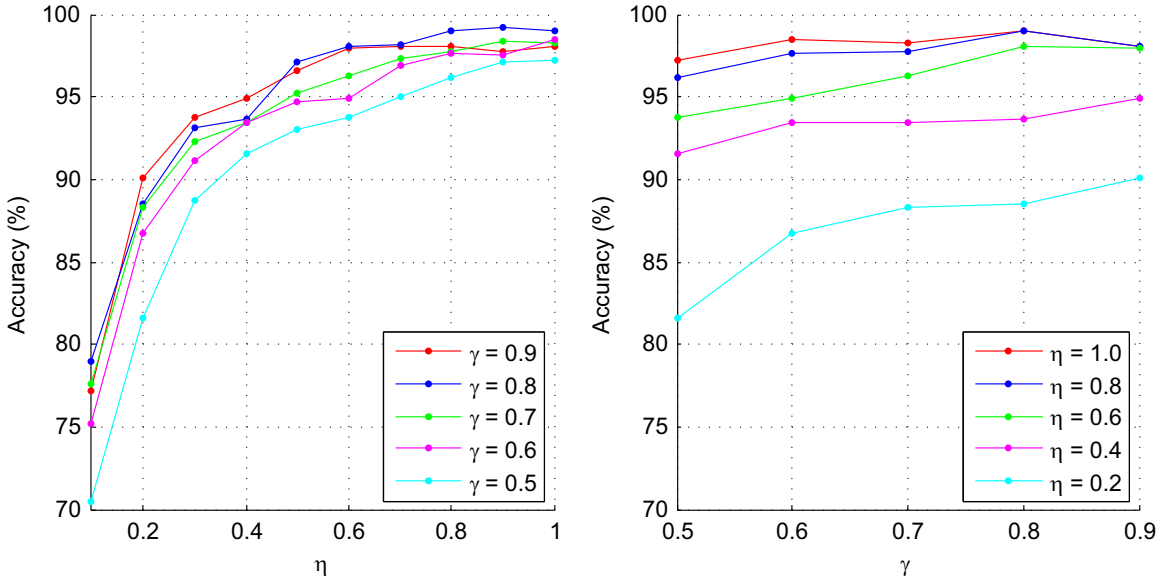


Fig. 9. Left: the indoor localization accuracy with respect to η. Right: the indoor localization accuracy with respect to γ.

Table 1
The indoor localization accuracy for various learning algorithms in terms of RMSE in meters with respect to different sample sizes in indoor experiments.

Sample sizes	20*N*	40*N*	60*N*	80*N*	100*N*
KNN	1.44	2.13	2.12	2.72	2.62
SVM	1.46	1.98	2.35	2.54	2.31
LLE	1.31	1.81	2.09	2.20	2.11
DNN(RBM)	1.01	0.806	1.23	0.992	1.07
DNN(SDA)	0.706	0.706	0.786	0.833	0.719
KNN+HMM	0.825	1.37	1.53	1.96	2.04
SVM+HMM	0.751	1.25	1.68	1.80	1.75
LLE+HMM	0.643	1.08	1.93	1.85	1.61
DNN(RBM)+HMM	0.526	0.452	0.386	0.448	0.439
DNN(SDA)+HMM	0.330	0.415	0.345	0.339	0.398

Table 2
Computational time of the indoor localization for various learning algorithms.

Algorithms	KNN	SVM	LLE	DNN(RBM)	DNN(SDA)
Testing time (s)	0.22	13.70	0.24	0.26	0.25

1. $\boldsymbol{B}$ is the observation matrix, and $B_{j\tau}$ $(j=1,\ldots,N, \tau=t-6,\ldots,t+6)$ represents the probability that $\boldsymbol{v}^t$ are observed on the assumption that the device locates at l_j, i.e. $B_{j\tau}=P(\boldsymbol{v}^\tau|\boldsymbol{L}^\tau=l_j)$. To determine $B_{j\tau}$, we use the a posteriori probability $P(\boldsymbol{L}^\tau=l_j|\boldsymbol{v}^\tau)$ obtained by the DNN-based coarse localizer at the time instant τ to calculate the a priori probability $P(\boldsymbol{v}^\tau|\boldsymbol{L}^\tau=l_j)$, which can be formulated based on Bayes' rule.

$$P(\boldsymbol{v}^\tau|\boldsymbol{L}^\tau=l_j)=\frac{P(\boldsymbol{v}^\tau)P(\boldsymbol{L}^\tau=l_j|\boldsymbol{v}^\tau)}{P(\boldsymbol{L}^\tau=l_j)} \tag{12}$$

$P(\boldsymbol{v}^\tau)$ represents the probability of observation $\boldsymbol{v}^\tau$, which is a determined value. $P(\boldsymbol{L}^\tau=l_j)$ denotes the a priori probability of $\boldsymbol{L}^\tau$ at l_j and is set to $1/N$. We can then obtain $P(\boldsymbol{v}^\tau|\boldsymbol{L}^\tau=l_j)=cP(\boldsymbol{L}^\tau=l_j|\boldsymbol{v}^\tau)$, where $c=P(\boldsymbol{v}^\tau)/P(\boldsymbol{L}^\tau=l_j)$ is a constant. Moreover, c makes no contribution to the final result as it appears in both the numerator and denominator of (8). $\boldsymbol{\pi}$ is the prior probability and its element represents the probability of the initial location at l_i which equals $1/N$ in our positioning system.

We finally determine the device's current position $\boldsymbol{L}^t$ as follows:

$$\boldsymbol{L}^t=\sum_{i=1}^{N}P^t(\boldsymbol{L}^t=l_i)\boldsymbol{L}_i \tag{13}$$

where the weights $P^t(\boldsymbol{L}^t=l_i)$ are obtained by the HMM-based fine localizer with the observation sequence $\boldsymbol{v}^{t-6},\ldots,\boldsymbol{v}^{t+6}$ mentioned above and $\boldsymbol{L}_i$ stands for the position of the gird l_i.

4. Experiments and results

Experiments were conducted in both indoor and outdoor environments divided into hundreds of squared grids, which have few people around. A person walks around and holds a Nexus 7 equipped with a wireless card that can receive RSSIs from surrounding APs. For each grid, the RSSI values are collected at four different positions and five samples are recorded at each position to deal with the fluctuation of Wi-Fi signals. Moreover, such collection process is repeated five times in different periods. Therefore, totally one hundred RSSI samples are collected at each grid. In the testing, the goal is to figure out the current location of the user given a collection of RSSIs and compare with the ground truth. Besides, we also have applied the Wi-Fi localization algorithm in some mobile devices like tablet. Visit http://vsislab.com/projects/WirelessPositioning for positioning demos with mobile devices.

4.1. Indoor localization

Fig. 7 shows one floor of a building interior, where the corridors are divided into $N=91$ squared grids with size of 1.8 m × 1.8 m each. A total of 100*N* samples were obtained with 163 APs observed. Thus the DNN structure consists of an input layer with 163 units, an output layer with 91 units and three hidden layers with 200 units each layer.

In the training stage, the SDA was trained by setting 10% of input pixels to 0 on every update and regularizing the sparsity of hidden units to be 0.05. When tuning whole network with BP, we randomly dropped off a set of hidden units with probability 5% at every update to prevent overfitting.

As shown in Fig. 8, we tested the performance of the proposed DNN-based localizer with respect to different numbers of hidden

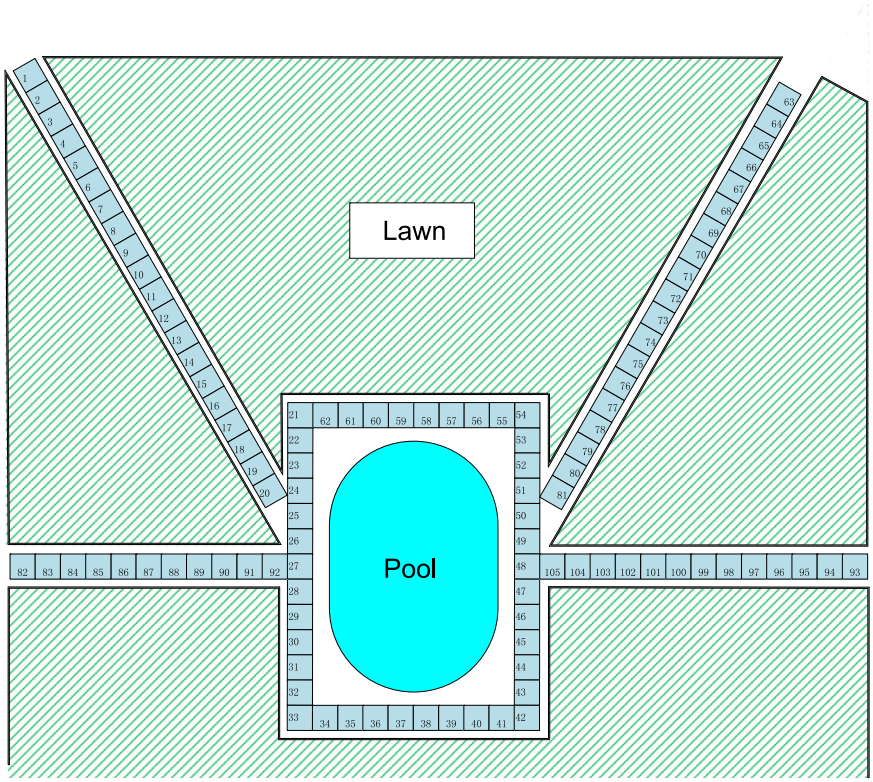


Fig. 10. An example of outdoor localization area. The outdoor garden was divided into 105 squared grids with 2 × 2 m each, and 359 APs were observed in the localization area.

Table 3
The outdoor localization accuracy for various learning algorithms in terms of RMSE in meters with respect to different sample sizes in outdoor experiments.

Sample sizes	20*N*	40*N*	60*N*	80*N*	100*N*
KNN	3.52	5.82	7.15	8.55	8.42
SVM	3.06	5.56	6.45	6.21	6.52
LLE	1.42	4.56	5.20	5.31	5.39
DNN(RBM)	5.42	3.55	3.29	4.16	3.67
DNN(SDA)	1.63	2.53	2.53	2.92	2.34
KNN+HMM	0.608	1.51	1.61	2.72	2.94
SVM+HMM	0.644	0.800	1.39	1.48	1.97
LLE+HMM	0.430	0.908	1.03	1.36	1.30
DNN(RBM)+HMM	0.507	0.286	0.375	0.472	0.388
DNN(SDA)+HMM	0.377	0.316	0.360	0.406	0.368

units and hidden layers. It is observed that, as the number of hidden units increases, generally better localization results can be obtained. However, this is not true for testing on the number of hidden layers, and the performance becomes even worse when the network goes deeper. This is probably because excessive layers would make the gradients difficult to propagate if without enough training samples. It is observed from Fig. 8 that the best localization results are obtained with three hidden layers and 200 units per hidden layer. Hence, we choose a four-layer DNN to build the localizer in this work.

Fig. 9 shows influence of the size of training samples and labeled samples to the localization accuracy. γ denotes the rate of the training samples in the whole dataset which reflects the influence of the training set. The influence of the labeled set is reflected by η which denotes the rate of the labeled samples in the training set. It is not surprised that the more samples are labeled, the better localization performance is obtained, which suggests us to choose $\eta = 1$ for our task. Also, more training samples normally can improve the localization accuracy and the results suggest that $\gamma =$ 0.8 works well. Hence, the number of training samples is four times (0.8:0.2) that of the testing ones in the following experiments.

To test the scalability of the proposed method, the total 100*N* fingerprints are randomly sampled to form additional four sample sets with different sizes such as 20*N*, 40*N*, 60*N* and 80*N*. We tested the performance of the proposed method in terms of root mean square error (RMSE) in meters with different sizes of sample sets, and compared with some existing methods such as KNN [3], SVM [4], Locally Linear Embedding (LLE) [38], RBM+BP [39], as well as their combinations with HMM.

As shown in Table 1, the proposed deep network generally maintains good performance for all conditions as the sample size grows. It outperforms KNN, SVM and LLE methods significantly especially when the sample size goes to big. This is because the shallow-structured methods have limited modeling capability and cannot perform well to extract reliable features from a large set of widely fluctuating RSSI samples. The DNN-based coarse localizer, on the other hand, learns reliable high-level features automatically from the large set of fluctuating RSSI samples which avoids hand-engineering. Therefore it is more capable of learning useful features and thus more robust when estimating positions with these features.

In fact, both KNN and LLE localizers aim to determine the current position using that of best-fitted RPs. But the results show that this may not produce reliable estimation especially when the fingerprint database goes to big. This is probably because of the variation of RSSIs due to the fluctuating nature of wireless signal. Finding the best-fitted RPs is difficult and susceptible to the signal fluctuation. In contrast, the proposed DNN localizer maintains good performance for all sizes, which demonstrates its superior capability of learning stable features from noisy data.

In addition, the computational time of different localization methods was compared in Table 2 under the same condition. Apparently, KNN, LLE and the proposed DNN-based localizers cost similar computational time, while SVM needs much more time for

the localization task. The major computation of the proposed method lies in the network training stage. The prediction at the positioning stage of DNN is relatively straight-forward. Compared to the conventional learning-based methods, the proposed localizer does not rely on any searching in the sample space but only requires the forward evaluation of the trained neural network which is normally computational efficient.

Moreover, by integrating the sequential information via the proposed HMM-based fine localizer, further improvements of positioning accuracy can be obtained. The final localization results proved to be more stable and accurate after the fine positioning based on HMM.

Besides, we tested the DNN by using RBM to pre-train the network inspired by [21], which also produced satisfactory results. Apparently, SDA yields significantly better results than the others, which demonstrates its superior capability of learning stable high-level features from noisy data.

4.2. Outdoor localization

As shown in Fig. 10, the outdoor experiments were conducted in a garden of the campus where all trails are divided $N=105$ squared grids with the size of $2\,\mathrm{m}\times 2\,\mathrm{m}$ and 359 APs were observed. Similarly, the DNN structure is composed of three hidden layers with 200 hidden units each layer. The input and output layers have 359 and 105 units, respectively.

Table 3 shows the localization results of the proposed algorithm as well as existing methods. Similar to the indoor results, the proposed DNN method produces low positioning errors and does not vary too much as the sample size grows. This demonstrates that Deep Learning shows considerable superiority over the other shallow methods in dealing with such large and noisy data problem. It also shows that SDA performs much better than RBM in DNN training for wireless positioning.

As shown in Tables 1 and 3, compared to the indoor environment, the performance of the coarse localizer (without enforcing temporal coherence, via HMM) in outdoor environment is slightly worse. This is probably due to the following reasons: First, the positioning environment is larger and needs a bigger fingerprint database; Second, since the distance between the receiver and APs increases in the open outdoor environment, the multipath fading and attenuation by static or dynamic objects become worse. Also, the denser distributed APs inside buildings make the variation of the radio more distinctive and easily to be detected, so the localizer can get better results than that in the outdoor environment. However, after integrating the temporal information via the HMM-based fine localizer, the proposed method can produce a competitive satisfactory localization result for the outdoor experiments.

5. Conclusions

In this paper, we have developed a Deep Learning scheme for Wi-Fi based localization, which is rarely reported in the literature. At the offline stage, a four-layer DNN structure is trained to extract features from massive fluctuating Wi-Fi signals and build fingerprints. At the online positioning stage, the proposed DNN-based localizer gives a coarse estimate about the position of the target. Then, HMM is integrated to further refine the positioning results based on the sequential information. The recent advances in Deep Learning enable us to employ modern training techniques to boost positioning performance. Various results in both indoor and outdoor environments demonstrated that the proposed method achieves state of the art performance on positioning with Wi-Fi signals.

Acknowledgments

This work was supported by the NSFC Grant nos. 61203253, 61573222 and 61233014, Major Research Program of Shandong Province 2015ZDXX0801A02, Research Found of Outstanding Young Scientist Award of Shandong Province BS2013DX023, Open Program of Jiangsu Key Laboratory of 3D Printing Equipment and Manufacturing 3DL201502, and Program of Key Lab of ICSP MOE China.

References

[1] M. Ciurana, F. Barceló, S. Cugno, Indoor tracking in wlan location with toa measurements, in: Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access, ACM, Torremolinos, Spain, 2006, pp. 121–125.
[2] L. Cheng, C.-D. Wu, Y.-Z. Zhang, Indoor robot localization based on wireless sensor networks, IEEE Trans. Consum. Electron. 57 (3) (2011) 1099–1104.
[3] P. Bahl, V.N. Padmanabhan, Radar: An in-building rf-based user location and tracking system, in: INFOCOM 2000, Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, vol. 2, IEEE, Tel Aviv, 2000, pp. 775–784.
[4] C.-L. Wu, L.-C. Fu, F.-L. Lian, Wlan location determination in e-home via support vector classification, in: 2004 IEEE International Conference on Networking, Sensing and Control, vol. 2, IEEE, 2004, pp. 1026–1031.
[5] I. Guvenc, C.T. Abdallah, R. Jordan, O. Dedeoglu, Enhancements to rss based indoor tracking systems using Kalman filters, in: GSPx & International Signal Processing Conference, 2003, pp. 91–102.
[6] D. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello, Bayesian filtering for location estimation, IEEE Pervasive Comput. 3 (2003) 24–33.
[7] J. Hightower, G. Borriello, Particle filters for location estimation in ubiquitous computing: a case study, in: UbiComp 2004: Ubiquitous Computing, Springer, 2004, pp. 88–106.
[8] A.S. Paul, E.A. Wan, Wi-fi based indoor localization and tracking using sigma-point Kalman filtering methods, in: Position, Location and Navigation Symposium, 2008 IEEE/ION, IEEE, Monterey, CA, 2008, pp. 646–659.
[9] C. Feng, W.S.A. Au, S. Valaee, Z. Tan, Received-signal-strength-based indoor positioning using compressive sensing, IEEE Trans. Mob. Comput. 11 (12) (2012) 1983–1993.
[10] S. Nikitaki, P. Tsakalides, Localization in wireless networks via spatial sparsity, in: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), IEEE, Pacific Grove, CA, 2010, pp. 236–239.
[11] S.J. Pan, V.W. Zheng, Q. Yang, D.H. Hu, Transfer learning for wifi-based indoor localization, in: Proceedings of Workshop on Transfer Learning for Complex Task of the 23rd Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence, 2008.
[12] Y. Li, S. Li, Y. Ge, A biologically inspired solution to simultaneous localization and consistent mapping in dynamic environments, Neurocomputing 104 (2013) 170–179.
[13] A.O. De Sá, N. Nedjah, L. De Macedo Mourelle, Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms, Neurocomputing 172 (2016) 322–336.
[14] L. Deng, Three classes of deep learning architectures and their applications: a tutorial survey, APSIPA Trans. Signal Inf. Process 2012, http://dx.doi.org/10.1017/atsip.2013.9.
[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[16] Q.V. Le, Building high-level features using large scale unsupervised learning, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Vancouver, BC, 2013, pp. 8595–8598.
[17] H. Zhang, F. Zhou, W. Zhang, X. Yuan, Z. Chen, Real-time action recognition based on a modified deep belief network model, in: 2014 IEEE International Conference on Information and Automation (ICIA), IEEE, Hailar, China, 2014, pp. 225–228.
[18] W. Zhang, Y. Zhang, L. Ma, J. Guan, S. Gong, Multimodal learning for facial expression recognition, Pattern Recognit. 48 (10) (2015) 3191–3202.
[19] A.-R. Mohamed, G.E. Dahl, G. Hinton, Acoustic modeling using deep belief networks, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 14–22.
[20] H. Lee, P. Pham, Y. Largman, A.Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, in: Advances in Neural Information Processing Systems, 2009, pp. 1096–1104.
[21] G.E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 30–42.
[22] P. Sermanet, R. Hadsell, M. Scoffier, U. Muller, Y. LeCun, Mapping and planning under uncertainty in mobile robots with long-range perception, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, IROS 2008, IEEE, Nice, France, 2008, pp. 2525–2530.

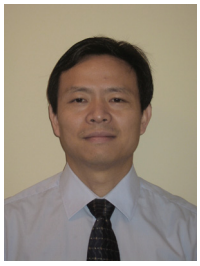
[23] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, Y. LeCun, Learning long-range vision for autonomous off-road driving, J. Field Robot. 26 (2) (2009) 120–144.
[24] Z. Chen, C. Wang, Modeling rfid signal distribution based on neural network combined with continuous ant colony optimization, Neurocomputing 123 (2014) 354–361.
[25] Z. Yang, P. Zhang, L. Chen, Rfid-enabled indoor positioning method for a real-time manufacturing execution system using os-elm, Neurocomputing 174 (2016) 121–133, http://dx.doi.org/10.1016/j.neucom.2015.05.12.
[26] N. Chang, R. Rashidzadeh, M. Ahmadi, Robust indoor positioning using differential wi-fi access points, IEEE Trans. Consum. Electron. 56 (3) (2010) 1860–1867.
[27] B. Zhang, F. Yu, Lswd: localization scheme for wireless sensor networks using directional antenna, IEEE Trans. Consum. Electron. 56 (4) (2010) 2208–2216.
[28] C.-H. Lim, Y. Wan, B.-P. Ng, C.-M.S. See, A real-time indoor wifi localization system utilizing smart antennas, IEEE Trans. Consum. Electron. 53 (2) (2007) 618–622.
[29] C. Feng, W. S.A. Au, S. Valaee, Z. Tan, Orientation-aware indoor localization using affinity propagation and compressive sensing, in: 2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, Aruba, Dutch Antilles, 2009, pp. 261–264.
[30] R. Battiti, A. Villani, T. Le Nhat, Neural network models for intelligent networks: deriving the location from signal patterns, Proc. AINS 2002, UCLA, 2002.
[31] K. Derr, M. Manic, Wireless based object tracking based on neural networks, in: 3rd IEEE Conference on Industrial Electronics and Applications, 2008, ICIEA 2008, IEEE, Singapore, 2008, pp. 308–313.
[32] G. Ding, Z. Tan, J. Zhang, L. Zhang, Fingerprinting localization based on affinity propagation clustering and artificial neural networks, in: 2013 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, Shanghai, China, 2013, pp. 2317–2322.
[33] Z. Yang, C. Wu, Y. Liu, Locating in fingerprint space: wireless indoor localization with little human intervention, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, ACM, Istanbul, Turkey, 2012, pp. 269–280.
[34] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[35] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1096–1103.
[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[37] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580.
[38] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[39] W. Zhang, K. Liu, W. Zhang, Y. Zhang, J.-F. Gu, Wi-fi positioning based on deep learning, in: 2014 IEEE International Conference on Information and Automation (ICIA), IEEE, Hailar, China, 2014, pp. 1176–1179.

Wei Zhang (S'06-M'11) received the Ph.D. degree in Electronic Engineering from The Chinese University of Hong Kong in 2010. He is currently an Associate Professor of the School of Control Science and Engineering at Shandong University, China. His research interests include multimedia, computer vision, artificial intelligence, and robotics.

Kan Liu (S'13) received B.S. degree in automatic control from Huazhong University of Science and Technology, Wuhan, China, in 2011. He is pursing the M.S. degree in pattern recognition and intelligent system from Shandong University, Jinan, China. His research interests include wireless sensor networks and pattern recognition

Weidong Zhang (S'13) received B.S.degree in biomedical engineering from Zhejiang University, Hangzhou, China, in 2012. He is pursing the M.S. degree in biomedical engineering from Shandong University, Jinan, China. His research interests include computer vision and machine learning.

Youmei Zhang (S'13) received B.S. degree in Measurement and Control Technology and Instrumentation from Shandong University, Jinan, China, in 2013. She is pursing the M.S. degree pattern recognition from Shandong University, Jinan, China. Her research interests include facial expression recognition and abnormal behavior analysis.



Jason Gu was born in Jiangsu, P.R. China. He received a B.Sc. degree in 1992 from the Department of Electrical Engineering & Information Science (Special Class for the Young Gifted 1987–1990), University of Science and Technology of China, M.Sc. degree in 1995 from Biomedical and Instrumentation Engineering, Jiaotong University (Shanghai), China and Ph.D. degree from the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada.