Design and Implementation of Non-intrusive Stationary Occupancy Count in Elevator with WiFi

Wei Shi(✉), Umer Tahir, Hui Zhang, and Jizhong Zhao

Xi'an Jiaotong University, Xi'an 710049, Shaanxi, People's Republic of China
weishi0103@sina.com

Abstract. Wi-Fi Sensing has shown huge progress in last few years. Multiple Input and Multiple Output (MIMO) has opened a gateway of new generation of sensing capabilities. This can also be used as a passive surveillance technology which is non-intrusive meaning it is not a nuisance as it is not need the subjects to carry any dedicated device. In this thesis, we present a way to count crowd in the elevator non-intrusively with 5 GHz Wi-Fi signals. For this purpose, Channel State Information (CSI) is collected from the commercially available off-the-shelf (COTS) Wi-Fi devices setup in an elevator. Our goal is to Analyze the CSI of every subcarrier frequency and then count the occupancy in it with the help of Convolutional Neural Network (CNN). After CSI data collection, we normalize the data with Savitzky Golay method. Each CSI subcarrier data of all the samples is made mean centered and then outliers are removed by applying Hampel Filter. The resultant wave is decimated and divided into 5 equal length segments representing the human presence recorded in 5 s. Continuous wavelet frequency representations are generated for all segments of every CSI subcarrier frequency waves. These frequency pattern images are then fed to the CNN model to generalize and classify what category of crowd they belong to. After training, the model can achieve the test accuracy of more than 90%.

Keywords: Wi-Fi Sensing · CSI · CWT · CNN

1 Introduction

Wi-Fi has become an essential part of our life. The wireless technology is growing and improving at an exponential rate. Today Wi-Fi is being used in desktop computers, laptops, mobile and many Internet of Things devices which are able to provide functionality because of it. With every passing year our technologies and electronic products are becoming tether less that is number of wires are being reduced. When signals from transceiver leaves for the receiving end it interacts with several objects in between. They each cause a specific variation in the radio frequency captured at the receiving end. This recorded variation can help us differentiate how many people are

This work is supported by NSFC Grants No. 61802299, 61772413, 61672424.

Q. Li et al. (Eds.): BROADNETS 2019, LNICST 303, pp. 3–19, 2019.
https://doi.org/10.1007/978-3-030-36442-7_1

standing in the elevator. Although human sensing has been done in the past but none has been done in an elevator so far with Wi-Fi using Channel State Information (CSI).

The main motivation behind this research is that Wi-Fi is relatively new area of research in terms of human sensing capabilities. Though a number of researches have been done in this regard, still much improvements and fine tuning is needed. As I have mentioned before, continuous development in human sensing based on radio signals is the driving force. Moreover, it should also be noted here that this is non-intrusive technique which is not an inconvenience to the monitored subjects as it serves its purpose at no additional cost to the people monitored. It also non-invasive in terms of privacy since the Wi-Fi signals research has not yet matured enough or due to technical limitations of radio signals it cannot record faces. However, recognition of different subjects using radio signals is also being researched upon in the world. Wi-Fi signals does not need the environment to be in proper lighting conditions. It does not make noise; it can work in the dark quietly as well. It serves as a passive tool for studying human detection that does not require Line of Sight to work unlike traditional methods. These are the primary motivation and the reasons why we chose to research in this area.

We have developed a way to count humans in an elevator. The techniques used were studied individually and combined in a sequence that yields more than 90% result accuracy. The first step is to collect the data samples. For this purpose, Wi-Fi CSI data is collected. Several post processing is involved before we feed the data samples to our CNN model. We record 7 full CSI samples for each human category. As we have a pair of receiving and transmitting antennas and we are considering only the receiving antennas for the task of recognizing number of humans, one category includes 14 CSI data samples. 57 sub-carrier level frequency waves in each of 14 CSI data samples. First data is normalized through Savitzky Golay Method. This smooths the data points and a trend becomes slightly visible and the noise is somewhat removed. We apply this to each sub-carrier level frequency wave samples for each instances of human category recorded. We then apply mean centering method to the CSI data. In order to reduce the frequency sampling rate, the data is segmented into 5 equal points this gives us 5 patterns in just one sub-carrier CSI wave. Segmented waves are converted into CWT image patterns. The segmentation allows us to achieve high accuracy rate as it increases the number of pattern images for our neural net for training.

2 Related Work

Vision Based approaches uses patterns to recognize such as face detection and human count. Human detection and count techniques for camera-based approaches are very mature by now and they are widely used in professional environments. Since humans are of different shape and sizes and can wear different style dresses, detection of humans through vision-based approaches become slightly more difficult. Only major challenge with vision-based approach is LOS. It can lose track of people when they leave the line of sight and when they arrive in the zone where the camera is incapable of detecting any features. Another challenge with this is proper illumination must be established before detecting humans. Camera-based approach may be disliked because it can record people's faces and features and sometimes it makes people uncomfortable. Prior consent, in public places may be a challenge in itself.

2.1 Vision/Camera Based Devices

Vision Based approaches [1–4] uses patterns to recognize such as face detection and human count. Human detection and count techniques for camera-based approaches are very mature by now and they are widely used in professional environments. Since humans are of different shape and sizes and can wear different style dresses, detection of humans through vision-based approaches become slightly more difficult. Only major challenge with vision-based approach is LOS. It can lose track of people when they leave the line of sight and when they arrive in the zone where the camera is incapable of detecting any features. Another challenge with this is proper illumination must be established before detecting humans. Camera-based approach may be disliked because it can record people's faces and features and sometimes it makes people uncomfortable. Prior consent, in public places may be a challenge in itself.

2.2 Non-camera Based Devices

Dedicated device-based approach includes wearable devices [5, 6], RFID tags [7–12], mobile phones [13] and other related sensors [14–16] such as smart watches in recent years are used to detect activity and count the crowd in the indoor environment. This technique uses some other resources to provide complete functionality such as Wi-Fi networks or Bluetooth connectivity. Radio frequency-based approaches is relatively new and developing form of detecting humans and their activity. It can be very sensitive to human activity and many scenarios have been under research for some time now including vital sign monitoring, indoor localization, human count and gestures recognition and sleep cycle monitoring. A special device UWB [17–21] radar have been used to count humans. It works on the signal propagation model as humans and other objects present between the transceiver and receiver will affect the propagation of signals. This technique is similar to that of Vision-based methods in terms of not requiring any of the monitored subjects to wear any wearable devices. For earlier work of detecting humans through Wi-Fi [22–29] have Received Signal Strength (RSS) [30–33]. Several research on localization has been done using RSS. It may be vaguely related to the radar but based on Wi-Fi. Our work also falls under radio frequencies as Wi-Fi basically works on radio signals. To tackle some of the problems researchers at MIT have used Hidden Markov Models to make up for human dissimilarities in structure and introduced some constraints with it on human motion variation. Then they map the responses received at the receiving end under a skeleton frame of reference which helps them to detect human activity as well as posture which also keep in view the human structural differentiations among each other. They call it Wi-Vi [34]. It works on similar principles that ultrasound, radar or LIDAR [36] works on. They went on to develop dedicated Wi-Fi systems that can detect sleep patterns, breathing patterns and tracking applications for human motion. Ultrasound and especially lasers are not affected by multi-path propagation.

3 Methodology

3.1 System Overview

In this section we will discuss the techniques used to project human counts through Wi-Fi CSI data. The architecture will be discussed through the flow chart and the overall flow will be explained. The very first task of our research is to gather CSI data through commodity COTS Wi-Fi hardware. The data is in raw form and must be passed through some data cleaning filters to remove outliers and then the output from pre-processing is fed to CNN. Figure 1 shows the whole flow of our architecture.

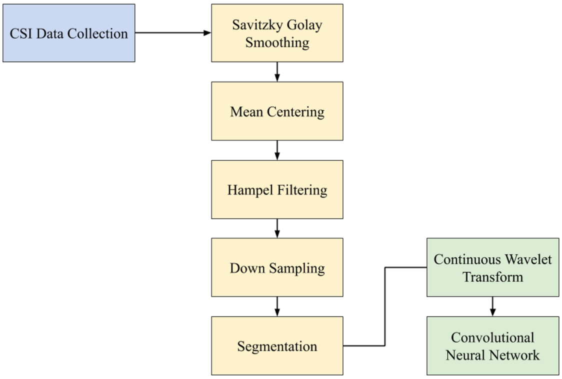

Fig. 1. Flow of our technique

3.2 CSI Data Collection

As obvious the first step is to setup the Wi-Fi devices in the elevator. We employ a pair of transmitters and receivers. In particular, we setup a Wi-Fi infrastructure, which includes two transmitters and two receivers. Then we selected to use the Intel Wireless Link 5300 NIC to collect the CSI data in the elevator, and the transmission rate is set as 1000 packets per second. One full sample is collected in 5 s that results in 5000 packets in 5 s at each of the receivers and they are recorded with the help of a software called Pico Scenes [35]. The Wi-Fi is configured to use 5 GHz frequency band. There are two transmitters and two receivers located in each corner of the lift.

3.3 CSI Data Pre-processing

CSI Data Smoothing

The first step after the data is recorded is to smooth it. We use Savitzky Golay Method to smooth our CSI sample data set. It is discussed briefly in the Sect. 2 of this document (Fig. 2).

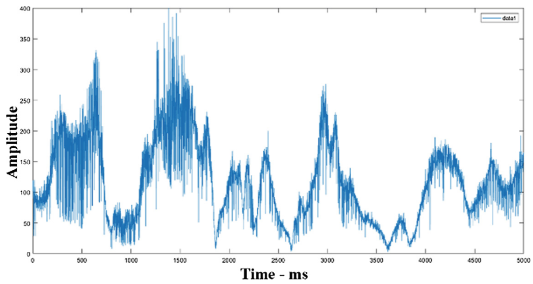

Fig. 2. One sub-carrier wave raw before smoothing

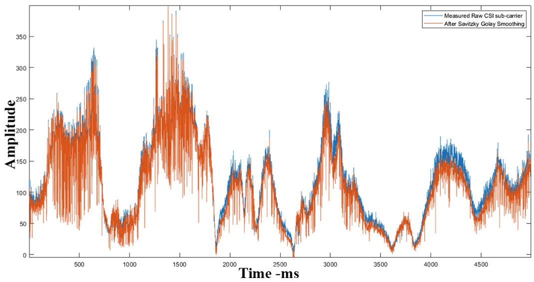

Fig. 3. After Savitzky Golay smoothing

As we can see in Fig. 3 the smoothing process is applied to the CSI data as the first thing in the data pre-processing. It works on same principle as a moving average. Local average is calculated by some window size and data is smoothed. After this smoothing is carried out the data is mean centered that is have a zero mean.

Mean Centering and Filtering

Mean Centering also refers to have a mean of zero. Centering in simple methodology means subtracting a constant from every value of a variable. Mean in simple words is average of the data that can be calculated [34]. Mean of any normal distribution is not zero. However, we can first normalize the data so that it is mean centered and have one standard deviation, as shown in Fig. 4.

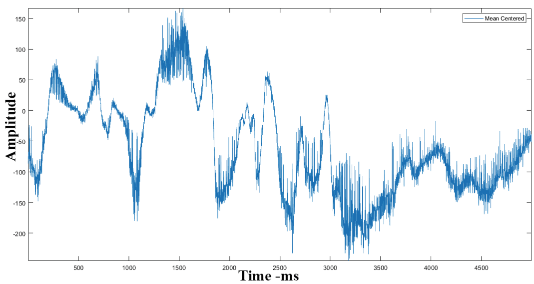

Fig. 4. Mean centered after smoothing

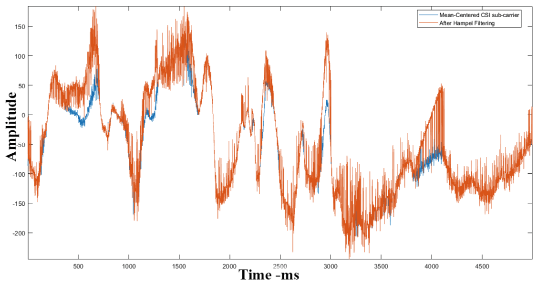

Fig. 5. Applying Hampel filtering with window of standard deviation 2

The Hampel filter is a filter that exchanges the middle value in the data window of size pre-decided with the median or standard deviation if it is too far from it. As shown in Fig. 5, Hampel filter looks like to add a specific noise that is only specific to one category of sub-carriers which gives the whole data some recognizable patterns and this is confirmed by CWT images.

Down Sampling

Our initial data have up to 5000 data points in 5 s which means the frequency is 1000 packets per second. We can decimate the frequency to be 200 data points per second. It will lower some complexity but the data will maintain its trend. In digital signal processing, down sampling and decimation are terms related with the process of downsampling. When down sampling is done on a sequence of time-series of a signal or other continuous function, it results in an estimate of the sequence that would have been a result of sampling the signal at a smaller sampling rate thus the overall trend of the signal will be there but in lower frequency. The wave is smaller in frequency but still have the same shape as before. As shown in Fig. 6, this technique involves basically translating the signal into a lower frequency rate and produces an approximation of the original signal as it would have been in a lower frequency.

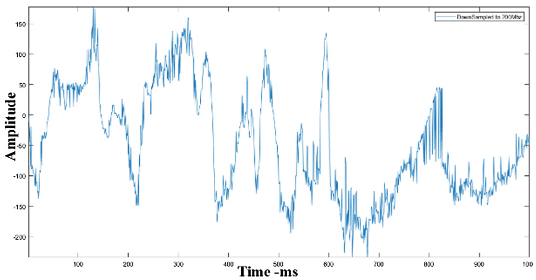

Fig. 6. Down sampling to 200 MHz, Time stretches 1000 ms instead of 5000 ms

3.4 Continuous Wavelet Transform (CWT)

Continuous wavelet transforms are a way of representing a time-series signal into a scalogram pattern based on the wavelet patterns to be noted in the series. The wavelet function we use called analytical morse which is also known as analytic morse parameter in CWT function in MATLAB.

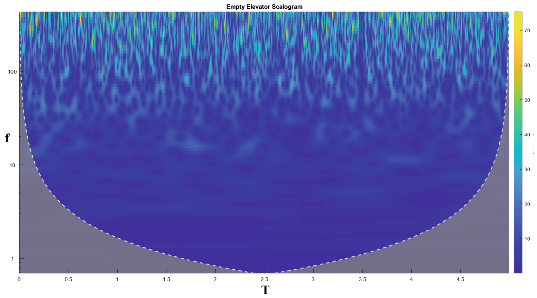

Fig. 7. CWT of empty elevator in 5 s

Figure 7 shows one full sub-carrier wave up to 5 s when the elevator is empty. We do not expect much variation in this instance. Figure 8 shows the segmented CWT of one sub-carrier when five people are in the elevator and as you can see the segmentation really result in similar CWT patterns in only one second window.

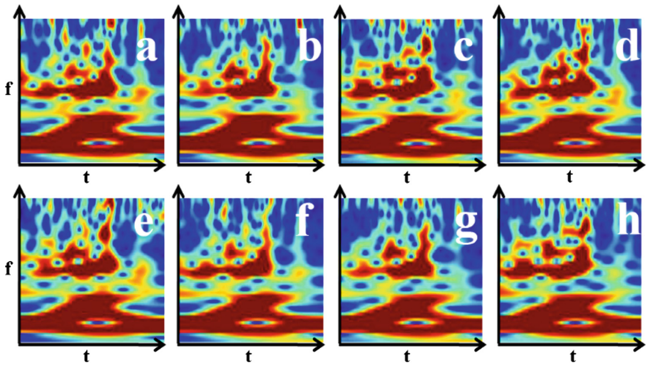

Fig. 8. (a–e) Represent one complete sub-carrier wave. (a) first segment representing 1st second, (b) representing 2nd second, (c) representing 3rd second and so on.

As we mentioned in segmentation and mean centering and filtering that CWT results in similar patterns for most of its segmented parts which proves that 1-s window can be useful.

3.5 CNN for Feature Extraction and Classification

We use deep learning for solving our classification problem of human occupancy. For this purpose, we make a neural net in MATLAB using its built in Deep learning toolkit and Alexnet. A Convolutional Neural Network (CNN) is also a deep Learning model in which we input an image of small size which is our CSI CWT pattern, importance in terms of learnable weights and biases are assigned and revised based on the cross-entropy loss function (Fig. 9).

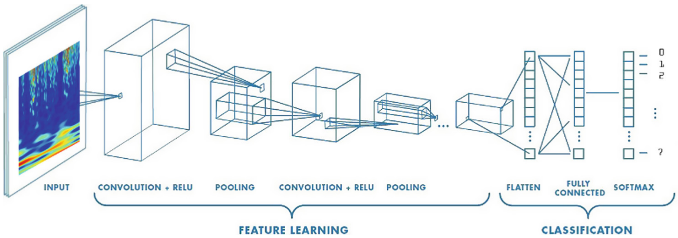

Fig. 9. Convolutional neural net overview

As shown in Fig. 8, It has eight fully connected layers as we have 8 categories to classify. 0 means when the elevator is empty and 7 when elevator is occupied by 7 people. The goal of our Convolution neural net operation is to extract the high-level characteristics such as edges, from the input image. We completely rely on CNN model to extract features automatically unlike other machine learning models where we have to extract features manually through PCA and ICA models of feature extraction and reduction. It not only detects edges but gradient orientation as well. With more added layers, the architecture tries to learn High-Level features as well, giving us a network, which has the whole idea and somewhat understanding of our continuous wavelet transform image patterns dataset. The first part of the neural net is the convolutional layer. As explained earlier, this part detects edges and other related features from the image based on a kernel which is also decided by the model. This kernel works as a filter moves to the right from left on the image. It move on by hops to the beginning starting from the left side of the CWT pattern with the same hop value and then keep repeating this process of hoping until the entire image is traversed. Thus, detectable features are detected. As our image is an RGB image all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output. The next is the Pooling layer which is responsible for reducing the span size of the Convolved Feature. This process also reduces dimensionality of our image patterns making it easier to compute them. Moreover, it is also helpful for extracting prominent features which may be rotational and share same position in each image, thus maintaining the process of effectively training of the model. The model is using Max Pooling as default that returns the maximum value from the part of the image that is covered by the Kernel at that iteration. Max Pooling also reduce noise internally. After this, comes the fully connected layers. Fully-Connected layer is usually easy way of learning combination of features.

3.6 Algorithm

After data collection is done, CSI is passed through several pre-processing techniques in MATLAB in the following order. First Taking out each sub-carrier of every sample one by one then de-noising and taking mean of zero as follows:

```
for i = 1: 57 all of matrix rows i.e. each sub-carrier as 57 sub-carrier in one sample
  Smooth signal data (Signal_data, sgolay) with Savitzky Golay method
 Signal_data m×n = Signal_data m×n − mean_m×n (Mean Centering)
  Hampel(Signal_data ,σ)  (Hampel Filtering)
end for i
```

After denoising, Down-sampling the sub-carriers to 200 Hz from 1000 Hz. After down-sampling the sub-carriers are segmented into 5 equal parts representing 1 s of window each. Then storing all segmented sub-carriers in one big matrix with labels.

```
for j = 1: 57 all of matrix rows i.e. each sub-carrier
  Down sample the frequency Signal_data by factor of 5
  Segment the Signal_data into 5 equal parts
  Concatenate the each Signal_data to form one Big matrix
 signal_data = ∪[Y_1, Y_2, Y_3, …, Y_j] , where Y ∈ CSI
end for j
Make labels for each Signal_data_m×n, (0,1,2…7)
```

After other pre-processing tasks are done, the output sub-carriers from segmentation is wavelet transformed with labels to their category.

```
for k = 1: N, where N is the number of rows in the Signal_data
  Wavelet Transform, Time-frequency Representation (Signal_data_m×n) of each row
  Read Labels and store respective CWT pattern in relevant directory
end for k
```

After wavelet transforms are done, we randomly select 85% of wavelets for training and the remainder of wavelets are left for test and classification of what category they belong to. Then we launch the CNN model and input data to it.

```
Select training and testing dataset with labels
Load deep learning library ALEXNET
Define learning rate, batch size and fully connected lay-
ers
Run training process over selected CSI sub-carrier pat-
terns with ground truth

After the neural net has been trained on 85% wavelets
then it is time to test and classify
Classify test data with learned attributes
Output Accuracy and Confusion Matrix
end.
```

4 Evaluation

In this section, we determine the usability of the methodology proposed in this study. First data is passed through some pre-processing steps and converted into wavelet transforms. Working with the limited data, we had to segment the time-series signal data of all the waveforms collected in the data. After the data has been converted, it was fed to a convolutional neural net. As we rely over CNN to find features for us, it does exactly that and after a successful run of CNN over training dataset of CWT patterns. Our set of CWT test data is passed to the trained-net to classify. The accuracy is based on the true predicted labels.

4.1 Accuracy and Confusion Matrices

In the first training experiment, all of the sub-carrier segmented CWT were divided into 85% training and 15% testing data set. This is done randomly and the algorithm decides which images to send to training and testing automatically. After the training is done the classification process yields a confusion matrix as shown in Table 1. The accuracy is decided as follows:

$$Accuracy_1 = \frac{True\ Predictions}{Total\ number\ of\ tested\ images} \times 100 = 97.85\%$$

As the related sub-carrier waves are co-related and thus it can predict classifications of wavelet transforms with fairly good accuracy. Next, we try to judge the whole sub-carrier set of one sample with just assisting it with few of the fragments from that class. For instance, selecting one full CSI sample that have 57 sub-carrier wave data. In that data we select most of the sub-carriers CWT for one sample to send to test dataset and only few of the segments from that sample were sent to the training dataset. After this data was trained it yielded the accuracy of 91.98% and its respective confusion matrix is shown in Table 2. This introduction of fragments to the training was done because of the smaller size of the whole dataset. However, this proves that the method is viable for bigger dataset and can run classifications on unseen dataset.

Table 1. Confusion matrix$_1$

People	0	1	2	3	4	5	6	7
0	100	0	0	0	0	0	0	0
1	1	96	1	1	0	0	1	0
2	0	1	97	1	0	0	0	1
3	0	1	0	97	1	1	0	0
4	0	0	0	0	99	0	1	0
5	0	1	0	1	0	98	1	0
6	0	0	0	0	1	0	98	1
7	0	0	0	1	0	1	1	97

Table 2. Confusion matrix$_2$

People	0	1	2	3	4	5	6	7
0	100	0	0	0	0	0	0	0
1	0	97	1	0	0	1	0	0
2	0	14	72	2	4	1	5	2
3	0	2	1	93	1	3	0	0
4	0	2	0	1	88	2	3	3
5	0	0	0	0	0	98	1	0
6	0	0	0	0	1	0	97	2
7	0	0	0	5	1	1	3	90

4.2 Training Process and Iteration Results

The process of training that yields the Confusion Matrix$_2$ is shown in the Table 3. The whole process took approximately 21 h. The training was done on a single CPU. The batch size was chosen to be 20 image patterns per iteration and the learning rate was 1.0e−4. The table consists of number of epochs, number of iterations, mini-batch accuracy and loss and base learning rate which is mentioned. One epoch of time is completed when all the of CWT training images are iterated in mini-batches once. The mini-batch accuracy and loss is decided over the batch size of per iteration which is as mentioned 20. First it tries to learn the features and predict it. This learning process is performed for each batch and that decides its accuracy and loss. As you can see the learning loss is reducing that points to the information that the model is learning. Consequently, the accuracy is increasing. This mini-batch accuracy is not the accuracy of classification of the test data but only depends upon the batch of images that iteration.

Table 3. Training process (over single CPU)

Epoch	Iteration	Time elapsed (hh: mm:ss)	Mini-batch accuracy	Mini-batch loss	Base learning rate
1	1	00:00:07	10.00%	4.2973	1.0e−04
1	50	00:04:28	20.00%	1.8777	1.0e−04
1	100	00:07:56	30.00%	1.9363	1.0e−04
1	200	00:15:49	35.00%	1.7443	1.0e−04
1	500	00:39:08	45.00%	1.6446	1.0e−04
1	1000	01:17:55	60.00%	1.3516	1.0e−04
2	1400	01:46:21	70.00%	0.6726	1.0e−04
2	2000	02:20:55	80.00%	0.5751	1.0e−04
2	2500	02:49:06	90.00%	0.2439	1.0e−04
3	3000	03:17:46	90.00%	0.2363	1.0e−04
3	3500	03:46:12	85.00%	0.4313	1.0e−04
3	4000	04:14:21	100.00%	0.0478	1.0e−04
4	4500	04:43:41	95.00%	0.0960	1.0e−04
4	5000	05:18:26	95.00%	0.1244	1.0e−04
4	5500	05:53:02	100.00%	0.0863	1.0e−04
5	6000	06:27:41	95.00%	0.0911	1.0e−04
5	6500	06:58:00	100.00%	0.0396	1.0e−04
6	7000	07:30:31	100.00%	0.0486	1.0e−04
6	7500	08:04:35	100.00%	0.0368	1.0e−04
6	8000	08:39:18	100.00%	0.0055	1.0e−04
7	8500	09:14:31	100.00%	0.0119	1.0e−04
7	9000	09:48:05	100.00%	0.0102	1.0e−04
7	9500	10:22:13	100.00%	0.0188	1.0e−04
8	10000	10:53:54	100.00%	0.0375	1.0e−04

(*continued*)

Table 3. (*continued*)

Epoch	Iteration	Time elapsed (hh: mm:ss)	Mini-batch accuracy	Mini-batch loss	Base learning rate
8	10500	11:29:57	100.00%	0.0118	1.0e−04
8	11000	12:07:11	100.00%	0.0102	1.0e−04
9	11500	12:40:55	100.00%	0.0078	1.0e−04
9	12000	13:13:29	100.00%	0.0137	1.0e−04
9	12500	13:41:36	100.00%	0.0018	1.0e−04
10	13000	14:09:54	100.00%	0.0006	1.0e−04
10	13500	14:37:51	100.00%	0.0037	1.0e−04
11	14000	15:05:45	100.00%	0.0007	1.0e−04
11	14500	15:33:42	100.00%	0.0084	1.0e−04
11	15000	16:01:30	100.00%	0.0084	1.0e−04
12	15500	16:29:22	100.00%	0.0051	1.0e−04
12	16000	16:57:12	100.00%	0.0044	1.0e−04
12	16500	17:25:01	100.00%	0.0001	1.0e−04
13	17000	17:52:50	100.00%	0.0014	1.0e−04
13	17500	18:20:40	100.00%	0.0101	1.0e−04
13	18000	18:48:30	100.00%	0.0006	1.0e−04
14	18500	19:16:20	100.00%	0.0109	1.0e−04
14	19000	19:44:24	100.00%	0.0008	1.0e−04
15	19500	20:12:53	100.00%	0.0258	1.0e−04
15	20000	20:41:15	100.00%	0.0012	1.0e−04
15	20500	21:09:39	100.00%	0.0139	1.0e−04
15	20850	21:29:33	100.00%	0.0031	1.0e−04

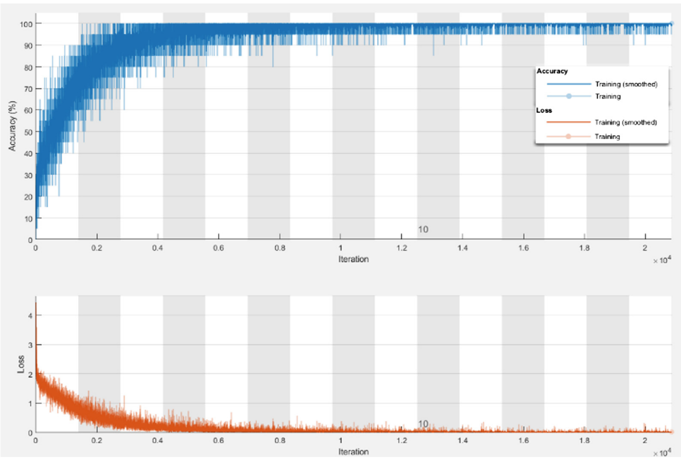

Fig. 10. Training process graphical

Figure 10 shows the full graph of the learning and training process of the CNN. The process took a long time to complete over a single CPU. Although the accuracy shows 100% but it is also an approximation of the learning process as there is still some loss in the learning process. There was room for more improvement but we approximate that the loss was nearly zero and for saving time, the training came to a conclusive end with 15 repetitions of learning of each and every CWT patterns involving 20850 iterations in total having 1390 iterations per epoch.

4.3 Experiment with Composition of Sub-carriers

In this experiment, the sub-carrier signals for one sample consisting of 57 sub-carriers at each antenna of one category were combined by taking average of 57 sub-carriers and outputting only one wave. The problem for us in this technique is that it drastically reduces size of our dataset which is already small to begin with. Consider the Fig. 4.2 and it has 57 sub-carrier information from one antenna.

We also tried composition of sub-carriers by superimposing all of them over each other in one sample and then some cause constructive interference and some destructive and the resultant were segmented and fed to the CNN. The results were not very good and they were even lower to an extent which was not worth mentioning. So, we tried with taking an average of each data point in time of every sub-carrier signal in one sample and then do the segmentation over it after which it shows promise as may be a good solution in some scenario.

Now when we take its mean for each point in time the whole resultant signal becomes like in the Fig. 7 after mean of zero (Fig. 11).

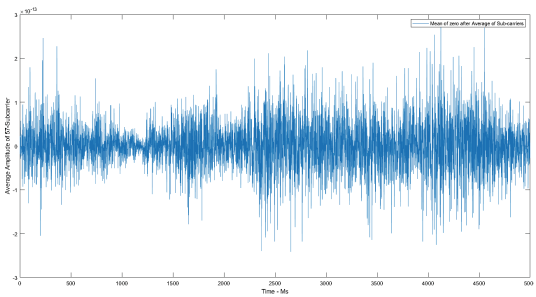

Fig. 11. Mean centered average of sub-carriers

After averaging all of the samples, we are left with only 14 samples each antenna for each category. When the signal segmentation is done for one category, we are left with only 70 CWT patterns from one antenna which is not enough as the person standing changes his posture will change the wave and it will not be correlated to the rest of the samples. For getting more posture and variability information from all of the samples we need more data collection. This method may result in good accuracy if it is done to a large dataset. However, this is yet to be tested and part of our future work.

With what little dataset we had we checked accuracy of this experiment. It yields 51.78% accuracy. The confusion Matrix for this experiment is given below in Table 4.

Table 4. Confusion matrix

People	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	43	29	0	0	14	0	14	0
1	14	86	0	0	0	0	0	0
2	14	14	29	14	0	0	29	14
3	0	29	0	43	0	0	29	0
4	0	0	14	0	43	14	0	29
5	0	29	0	0	0	43	14	14
6	14	14	0	0	0	14	57	14
7	14	0	0	0	0	0	14	71

This is why we treat every sub-carrier wave of every sample from all the categories as features of their respective class. This gives us more information and makes up for all possible variability caused by the multipath propagation of different signals.

5 Conclusion

We have developed a way to judge a signal at sub-carrier level to classify human occupancy in the elevator. The proposed technique involves a combination of easily available commercial off the shelf Wi-Fi hardware. We set it up in the elevator and configure to use 5 GHz band for increased sensitivity. The use of 5 GHz gives us a greater number of sub-carrier channel state information and those we have used as features. After pre-processing, all the sub-carrier segments are wavelet transformed and then they are fed to Alexnet a CNN model for feature extraction and classification. We rely on the capabilities of CNN for feature extraction automatically and after learning over a set of training data of wavelet transforms of sub-carriers of different labels. A set of unseen test wavelet transforms are given to the neural net to classify them to their true categories.

We experimented with compositing all the sub-carrier signals before segmenting and wavelet transforms of each sample with mean of their respective data points in time and some features are lost in this process and it reduces the data size and for a small dataset the accuracy will be decreased.

That is why each sub-carrier level information is taken as a possible feature for its class because of the small size of dataset and it gives us a good point to start. The model predicts the sub-carrier wavelet transforms with more than 97% accuracy and when we feed a complete sample of unseen patterns for respective categories with only few segments of them as training, the model predicts them with more than 91% accuracy. This proves that when training over a large dataset of several hundred samples of CSI of each category it will yield a good percentage of accuracy even for unseen samples.

References

1. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection. In: ICPR 2008. IEEE (2008)
2. Nichols, J.D., et al.: Multi-scale occupancy estimation and modelling using multiple detection methods. J. Appl. Ecol. **45**(5), 1321–1329 (2008)
3. Lin, S.-F., Chen, J.-Y., Chao, H.-X.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum. **31**(6), 645–654 (2001)
4. Kim, M., Kim, W., Kim, C.: Estimating the number of people in crowded scenes. In: Proceedings of the IS&T/SPIE Electronic Imaging, p. 78820L (2011)
5. Weppner, J., Lukowicz, P.: Bluetooth based collaborative crowd density estimation with mobile phones. In: PerCom 2013. IEEE (2013)
6. Schauer, L., Werner, M., Marcus, P.: Estimating crowd densities and pedestrian flows using Wi-Fi and Bluetooth. In: Mobiquitous 2014. ICST (2014)
7. Wang, J., Katabi, D.: Dude, where's my card?: RFID positioning that works with multipath and NOLS. In: Proceedings of ACM SIGCOMM (2013)
8. Wang, J., Vasisht, D., Katabi, D.: RF-IDraw: virtual touch screen in the air using RF signals. In: Proceedings of ACM SIGCOMM (2015)
9. Wang, J., Xiong, J., Jiang, H., Chen, X., Fang, D.: D-Watch: embracing "Bad" multipaths for device-free localization with COTS RFID devices, pp. 253–266 (2016)
10. Wei, T., Zhang, X.: Gyro in the air: tracking 3D orientation of batteryless internet-of-things. In: Proceedings of ACM MobiCom, pp. 55–68 (2016)
11. Yang, L., Chen, Y., Li, X.-Y., Xiao, C., Li, M., Liu, Y.: Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices. In: Proceedings of ACM MobiCom (2014)
12. Ding, H., et al.: Human object estimation via backscattered radio frequency signal. In: INFOCOM 2015. IEEE (2015)
13. Wirz, M., Franke, T., Roggen, D., Mitleton-Kelly, E., Lukowicz, P., Troster, G.: Probing crowd density through smartphones in city-scale mass gatherings. EPJ Data Sci. **2**(1), 1 (2013)
14. Lam, K.P., et al.: Occupancy detection through an extensive environmental sensor network in an open-plan office building. IBPSA Build. Simul. **145**, 1452–1459 (2009)

15. Jiang, C., Masood, M.K., Soh, Y.C., Li, H.: Indoor occupancy estimation from carbon dioxide concentration. Energy Build. **131**, 132–141 (2016)
16. Wang, S., Burnett, J., Chong, H.: Experimental validation of CO2-based occupancy detection for demand-controlled ventilation. Indoor Built Environ. **8**(6), 377–391 (2000)
17. Depatla, S., Muralidharan, A., Mostofi, Y.: Occupancy estimation using only WiFi power measurements. IEEE J. Sel. Areas Commun. **33**(7), 1381–1393 (2015)
18. Choi, J.W., Quan, X., Cho, S.H.: Bi-directional passing people counting system based on IR-UWB radar sensors. IEEE Internet Things J. **5**, 512–522 (2017)
19. Mohammadmoradi, H., Yin, S., Gnawali, O.: Room occupancy estimation through WiFi, UWB, and light sensors mounted on doorways. In: Proceedings of the 2017 International Conference on Smart Digital Environment, pp. 27–34 (2017)
20. Lv, H., et al.: Multi-target human sensing via UWB bio-radar based on multiple antennas. In: Proceedings of the IEEE TENCON, pp. 1–4 (2013)
21. He, J., Arora, A.: A regression-based radar-mote system for people counting. In: Proceedings of the IEEE PerCom, pp. 95–102 (2014)
22. Abdelnasser, H., Youssef, M., Harras, K.A.: WiGest: a ubiquitous WiFi-based gesture recognition system. In: Proceedings of IEEE INFOCOM (2015)
23. Chen, B., Yenamandra, V., Srinivasan, K.: Tracking keystrokes using wireless signals. In: Proceedings of ACM MobiSys (2015)
24. Ding, H., et al.: RFIPad: enabling cost-efficient and device-free in-air handwriting using passive tags. In: Proceedings of IEEE ICDCS (2017)
25. Li, H., Yang, W., Wang, J., Xu, Y., Huang, L.: WiFinger: talk to your smart devices with finger-grained gesture. In: Proceedings of ACM UbiComp (2016)
26. Lien, J., et al.: Soli: ubiquitous gesture sensing with millimeter wave radar. ACM Trans. Graph. **35**(4), 142 (2016)
27. Pu, Q., Gupta, S., Gollakota, S., Patel, S.: Whole-home gesture recognition using wireless signals. In: Proceedings of ACM MobiCom (2013)
28. Wang, W., Liu, A.X., Shahzad, M., Ling, K., Lu, S.: Understanding and modeling of WiFi signal based human activity recognition. In: Proceedings of ACM MobiCom (2015)
29. Xi, W., et al.: Electronic frog eye: counting crowd using WiFi. In: INFOCOM 2014. IEEE (2014)
30. Xu, C., et al.: SCPL: indoor device-free multi-subject counting and localization using radio signal strength. In: IPSN 2013. IEEE (2013)
31. Zheng, Y., Zhou, Z., Liu, Y.: From RSSI to CSI: indoor localization via channel response. ACM Comput. Surv. **46**(2), 1–32 (2013)
32. Adib, F., Katabi, D.: See through walls with WiFi! ACM SIGCOMM Comput. Commun. Rev. **43**(4), 75–86 (2013)
33. Nakatani, T., Maekawa, T., Shirakawa, M., et al.: Estimating the physical distance between two locations with Wi-Fi received signal strength information using obstacle-aware approach. Proc. ACM on Interact. Mob. Wearable Ubiquit. Technol. **2**(3), 1–26 (2018)
34. Adib, F.: MIT. Wi-Vi: See Through Walls with Wi-Fi Signals [EB/OL], August 2013. http://people.csail.mit.edu/fadel/wivi/radar.png
35. GitLab: PicoScenes Installation [EB/OL], 11 August 2018. http://gitlab.com/wifisensing/PicoScenes-Setup/
36. Tamas, L., Lazea, G.: Pattern recognition and tracking dynamic objects with LIDAR. In: Robotics. VDE (2011)