Human Activity Recognition Using Deep Learning

Abhinay Gupta Somisetty
Master's in CSC
California State University, Sacramento
asomisetty@csus.edu

Hemanth Rachakonda
Master's in CSC
California State University, Sacramento
hemanthrachakonda@csus.edu

***Abstract*—**

Machine learning is becoming a helping hand in many domains and so it has also gained some fame in many areas. Human activity recognition has vast application scope in the areas like smart home, Smart security solutions, Health care systems and in human-computer Interaction. Human actions in a a WiFi enabled environment lead to distortion which is vast usage in activity recognition. Many applications like video gaming, hand signal recognition for training deaf and dumb, to monitor elderly people at home etc. We implemented convolutional neural network to predict and classify the human actions. The dataset consists of Channel State Information of 1116 series and 52 sub carriers per each series and total of 192 samples. We have used Unet model to train the data and used the best model to predict and classify the actions. Though activities can be captured using cameras but with issues and concerns with respect to privacy WiFi devices are very useful to overcome this issue. This paper introduces how the WiFi data is predicted and classified using UNET model and how the metrics are calculated. Our evaluation results prove that the results achieved are finer than the previous models.

Keywords: Human actions, WiFi, CSI, CNN, UNET, Classification.

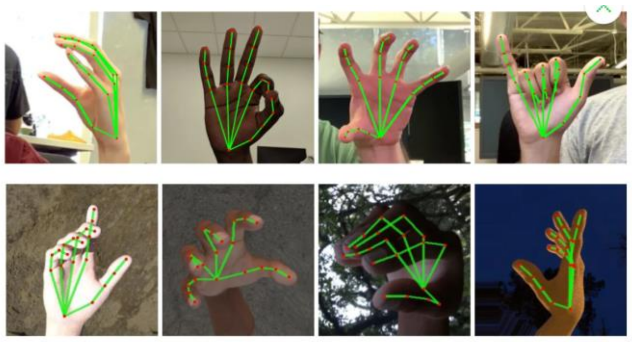
I. Introduction

Using WiFi devices for human action recognition is one of the widely used technique in indoor localization. In this project we use dataset which is collected from Wifi devices [2]. Tough Cameras can provide high resolution videos as a dataset we have some backdrops like low light videos and have privacy concern. To overcome the above limitations wifi based activity recognition have been proposed. We use this dataset and evaluate using deep learning models to achieve better accuracy. We have reviewed couple of papers as reference and also reviewing other related work A glimpse of how data is collected can be seen in fig-2. The model which we have introduced in this paper is used to increase the sensing ability of the WiFi sensors in classifying the human actions.

Sample level action recognition introduced in this paper, have two tasks. We have used convolutional neural network model to classify the human action classification as well as the human action prediction. Convolutional neural networks play a very important role in in any classification problems. Deep learning models has number of layers which dig deep into the input data and extract the accurate features and train the model to classify the data. We have used different metrics like Accuracy, Precision, Recall and F1 score [8].

For any classification problem these metrics are crucial to evaluate the model. Like any deep learning model, we have trained our model on the train data and test data and evaluated on the test data. We have discussed more about the dataset in section III and implementation of the model in the section IV. Results section has the various results achieved. Also, in results section we have compared our Unet model results with the pre-trained model that is discussed in [1].

We have used PyTorch [5] framework to implement our CNN model using Pycharm IDE and other torch libraries. Our model is trained on NVIDIA GeForce MX230 GPU.



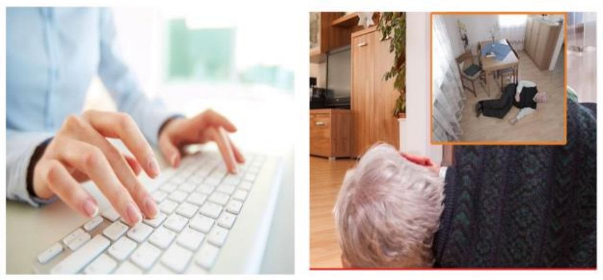


Figure -1 a) The top figure shows sign recognition which contribute in hand sign language. b) Using key stroke recognition, we can predict the mood of the person. c) Shows elderly person falling ground. Using activity recognition, we can detect this.

II. Related Work

Before implementing this model we have done research in human activity recognition, how Wifi devices are used to detect the gestures of humans [15]. Some of the gestures include push, kick, pull, strike, etc. They used WiSee which is the effects of Doppler shift to recognize these gestures. They achieved good and effective results. Another paper which introduces how fall detection is recognized using wifi frequency [14]. There is another paper which gives insight of CSI [6,13]. This paper is about the identification of face using wifi. They used channel state information to identify the activity of person and WiWho to identify the faces. They achieved efficient results. To better understand the calculation of CSI and adaptive mechanism [9]. To understand how the convolutional Neural networks for indoor localization.5. G. we referred the paper [12]. Various other work in human activities are based on the objects around them. This paper Illustrates human-object interaction [11]. They used triplets (human, verb, object) in that how humans will react according to that object and the corresponding verb is placed. The final output is to find the verb. This paper uses a fast R-CNN method [10].

III. APPROACH

A. *Dataset*

The dataset is extracted from source[2]. The data consists of train data and test data. These data consist of channel state information. Metadata summary is found in Table 1 and Table 2 below.

Metadata summary

Train_data_amp	CSI amplitude of training data: 1116x52x192. • CSI series: 1116 • Carriers (per each series): 52 • Samples: 192
Train_label_instance	Action labels, 1116x192 • CSI series: 1116 • Samples: 192 • Labels (0-6): background + 6 actions
Train_label_mask	Action labels, 1116x192 • CSI series: 1116 • Samples: 192 • Labels (0 or 1): binary
Train_label_time	Start and end time of 1116x2 • CSI series: 1116 • Start and End

Table 1: Showing the Train data summary

Test_data_amp	CSI amplitude of training data: 278x52x192. • CSI series: 278 • Carriers (per each series): 52 • Samples: 192
Test_label_instance	Action labels, 278x192 • CSI series: 278 • Samples: 192 • Labels (0-6): background + 6 actions
Test_label_mask	Action labels, 278x192 • CSI series: 278 • Samples: 192 • Labels (0 or 1): binary
Test_label_time	Start and end time of 278x2 • CSI series: 278 • Start and End

Table 2: Showing the Test data summary.

As it is mentioned that in both train and test data, label_instance has data labelled from 0-6 like mentioned below.

Action	Label
Background	0
UP	1
DOWN	2
LEFT	3
RIGHT	4
CIRCLE	5
CROSS	6

label_mask has binarily labelled data with 0-non or 1-Yes. Label_instance is used for action classification problem whereas label_mask is used for action prediction. All the actions are hand actions like shown in fig – 3. and for all the CSI series, the action start time and end time are annotated manually.

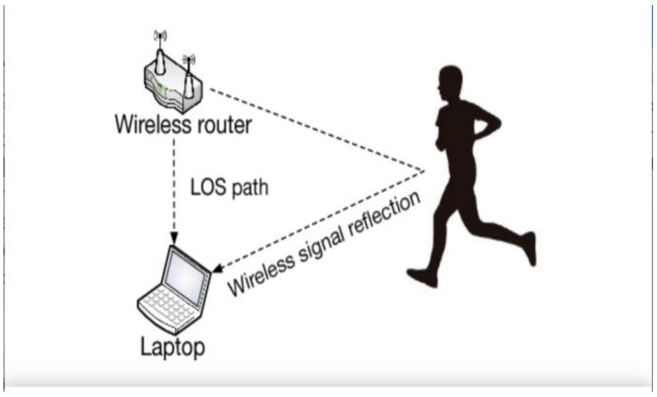


Figure – 2 How actions are captured using WiFi Signal.

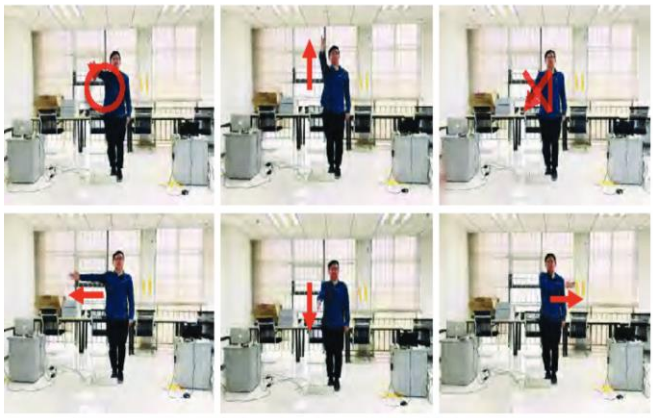


Figure – 3 A volunteer doing actions in a WiFi environment.

IV. IMPLEMENTATION OF MODEL

In the implementation part we have used Temporal Unet model with the 5 both Up and Down layers to train the data set unlike the traditional Unet model [3] and evaluated the model. We have compared our results with the pre-trained Unet model with 4 layers [1]. Both the results will be discussed in the results section. Our Unet model is briefly discussed in the coming sections.

A. *UNET Model*

Now we use UNET model to train our dataset and evaluate. This CNN model is used to for classification of different classes as well as predicting the accuracy. The basic structure of our UNet model looks like in Fig-4. In the first glance, it looks like "U" shape. Unet architecture is symmetric and the architecture consist of two parts.

- Contracting path (Left path)
- Expansive path (Right path)

The contractive path consists of general convolutional process. Whereas expansive path consists of transposed 2d convolutional layers. This can also be considered as down sampling and up sampling. Now let us consider the architecture of the Unet.

a) Contracting path: Contracting path consists of number of process which constitutes of 2 convolutional layers. This process is seen in the Fig - 4. The channel change from 1 to 64. These convolutional layers reduce the size of input CSI series. Deeper the number of layers will help the model to extract more features and the contracting path can learn more features in the larger views which is most efficient in action recognition.

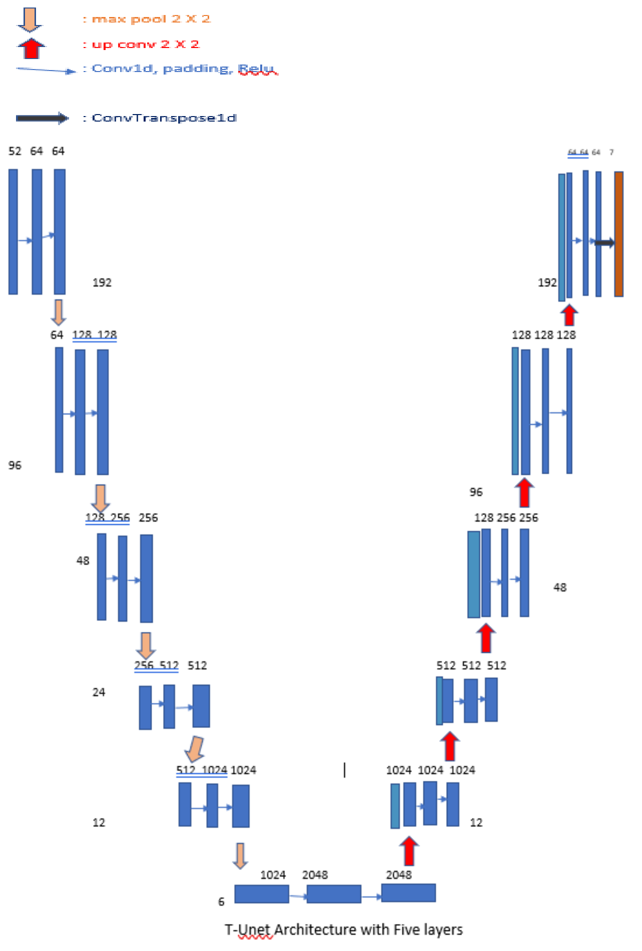


Figure 4: Unet architecture with Five layers.

b) Expansive path: The upsampling technique increases the size of the CSI. Basically, it does padding followed by some convolution operations.

These two parts are connected by links which has convolutional layers and to which transposed convolution is applied after this phase in expansion part.

B. *Approach*

Our T-Unet model has five layers both up and down unlike the traditional model. We have implemented T-Unet model using ReLu activation function and other convolutional neural network functions to train our model and evaluate the data. ReLu activation function can be used in the neural networks with many layers as it overcomes the vanished gradient problem. Any convolutional layer in a neural network implements batch normalization technique for training deep models which standardizes the inputs for each mini batch in a layer. For any classification problem criterion is very important. We used torch.nn.CrossEntropyLoss for this classification. It is very useful while training a classification problem with n classes. Input parameters for this function are size_average which is None.

Criterion= nn.CrossEntropyLoss(size_average=False).cuda()

Then we used adam optimizer function[7].We also tried executing the SGD optimizer but the results were not so favorable than Adam. Parameters for the optimizer are iterable unet parameters and learning rate. Learning rate is hyper param which is used in any optimizer in the CNN for training the neural networks with range between 0-1. For training the model we have used different learning rates like 0.1,0.05 and 0.005. We training accuracy has been consistently improving after each epoch when lr=0.005.

optimizer = torch.optim.Adam(unet.parameters(), lr=0.005)

The above-mentioned criterion and optimizer are passed to scheduler algorithm which decreases the learning rate of each param by gamma value specified which is 0.5 in our case. This is reduced once per each milestone which is another param specified. Milestones are nothing but epoch indices in ascending order.

Scheduler= torch.optim.lr_scheduler.MultiStepLR (optimizer, milestones=[20,40,60,80,100,120,140,160,180,200,220,240,260,280,300], gamma=0.5)

Thus, T-Unet model is trained using the various CNN functions to train the model along with the 200 epochs and the batch size of 128 and as mentioned with different learning rates and optimizers. But on a final note we have used lr=0.005 and Adam optimizer with the lr decay of gamma=0.5 for every 20 epochs. So, when we compare different parameters and layers and functions, we achieved best results using the above discussed. The results are compared and shown in the results section. Accuracy is calculated for each epoch and learning curves are plotted between train and test accuracy for each epoch for both prediction and classification. These graphs can be seen in fig-5 and fig-6.

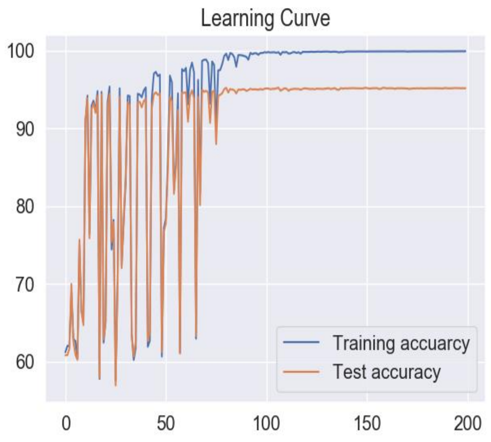


Figure – 5 Train vs Test accuracy learning curve for action prediction.

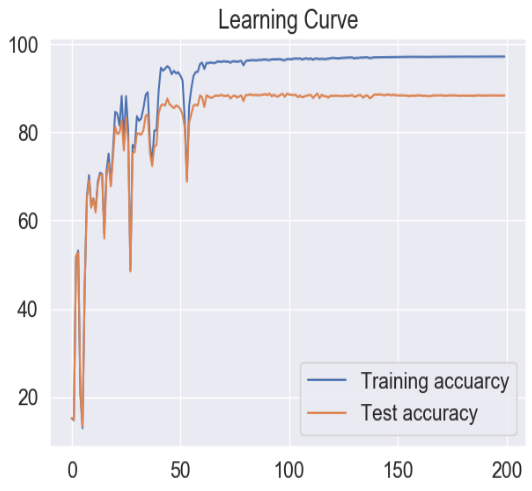


Figure – 6 Train vs Test learning curve for action classification.

From the above curves we see that although we have used learning reducing function on each epoch but after almost 140th epoch the learning rate is stable. The best trained model weights are saved in a pickle file and used to evaluate.

C. Evaluation

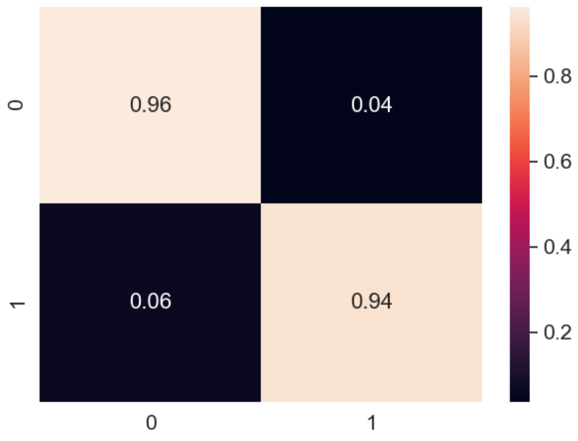
By using the above mentioned Unet model we will train our model and evaluate the model. For action prediction we use train/test_label_mask by loading the trained model (pickle file) and evaluate on the test data. We calculate recall, precision, F1-score and accuracy for classifying the test data. These metrics are derived by:

$$\text{Recall} = \frac{True\ positive}{True\ Positive+False\ negative}$$

$$\text{Precision} = \frac{True\ positive}{True\ Positive+False\ positive}$$

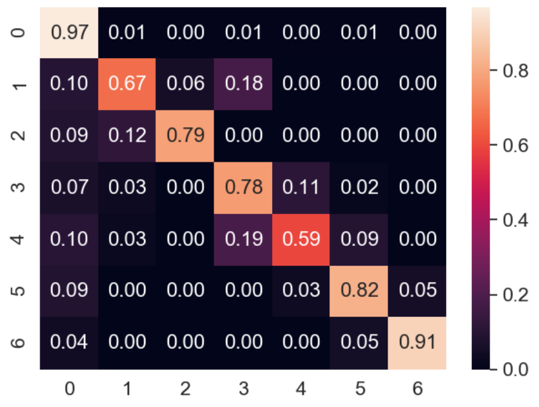
$$\text{Accuracy}= \frac{True\ positive+True\ negative}{True\ Positive+False\ poitive+True\ negative+False\ negative+}$$

In this process we draw the confusion matrix between the two labels non and action in the data. The matrix is seen in fig - 7. Similarly, we will train our model train/test_label_instance which has different action for classification. The confusion matrix for the classification model can be seen in fig - 8.



	0	1
0	0.96	0.04
1	0.06	0.94

Figure – 7 Confusion matrix for prediction the action.



	0	1	2	3	4	5	6
0	0.97	0.01	0.00	0.01	0.00	0.01	0.00
1	0.10	0.67	0.06	0.18	0.00	0.00	0.00
2	0.09	0.12	0.79	0.00	0.00	0.00	0.00
3	0.07	0.03	0.00	0.78	0.11	0.02	0.00
4	0.10	0.03	0.00	0.19	0.59	0.09	0.00
5	0.09	0.00	0.00	0.00	0.03	0.82	0.05
6	0.04	0.00	0.00	0.00	0.00	0.05	0.91

Figure – 8 Confusion matrix for classifying the various actions.

From the above fig 8 we can say that our model classified few classes very well and few actions like 4-right is slightly underperformed. Almost 20% of the action 4 is classified as action 3. But on an average this model has performed better than another model [1]. The fig - 9 shows the confusion matrix of the pre trained model and the results are compared in the coming section.

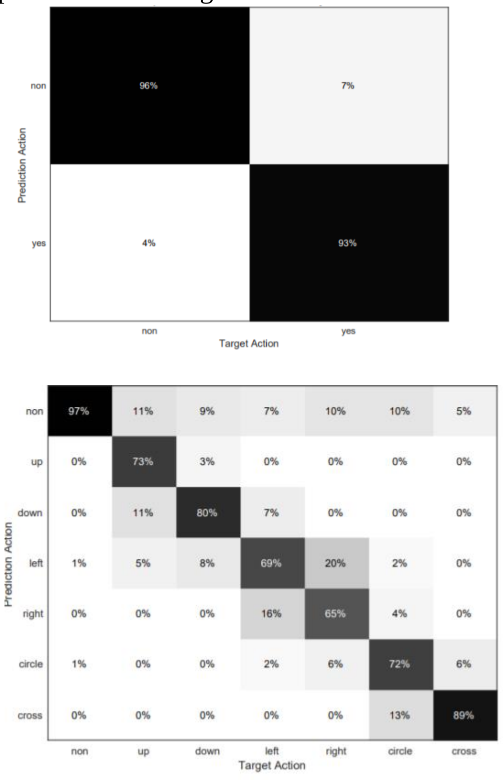


Figure – 9 Pre trained model a) action prediction b) action classification.[1]

From the above confusion matrix we can see that in the task sample action detection, the model has performed better than the pre trained model.

V. Results

Below we will discuss the results of the different models and compare the results of the pretrained model [1] to the model we have discussed in section IV.

A. Prediction Results of T-Unet model

Metric	**Dataset**	
	Training	***Test***
Accuracy	99.81	95.23

Table-3: Prediction of Unet model with five layers.

B. Classification Results of T-Unet model

Metric	***Training***	***Test***
Precision	97.64	88.97
Recall	97.52	89.11
F1 score	97.54	89.08
Accuracy	96.95	89.11

Table-4: Classification of Unet model with five layers

C. Prediction Results of Pretrained model

Metric	***Training***	***Test***
Accuracy	N/A	95.09

Table-5: Prediction of pre-trained model.

D. Classification Results of Pre-trained model

Metric	***Training***	***Test***
Accuracy	N/A	88.60

Table-6: Classification of pre trained model.

From the above results we can clearly say that our model performed slightly better that the pre-trained model [1] in terms of accuracy.

VI. Conclusion

Like mentioned Machine learning play an important role in the modern-day society. To extract the features digging deep into layers, deep learning models helps for feature extraction in any classification problem. Human activity recognition is another such classification problem which has two task human activity prediction and human activity classification. Using Unet model with five layers which is discussed in the implementation section has improved the performance than the pre-trained model. Along, with the addition of the layer, changing the params resulted in the better performance by achieving accuracy of 89.11% without over fitting of the data.

VII. Future Work

As the future work, we would like to collect more data using WIFi CSI and train our model on various datasets. Also, implementing other deep learning models like Densenet and by using ensemble techniques in deep learning we can ensemble both the models. In most of the cases ensemble models give better results even on large datasets. This would definitely a great work and contribution in this area.

VIII. References

[1] Wang, Fei and Song, Yunpeng and Zhang, Jimuyang and Han, Jinsong and Huang, Dong, “Temporal Unet: Sample Level Human Action Recognition using WiFi”, 2019.

[2] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, “Joint activity recognition and indoor localization,” arXiv preprint arXiv:1904.04964, 2019

[3] Jeremy Zang, “UNet — Line by Line Explanation,” 2019

[4] Jason Brownie, “How do convolutional layers work in deep learning neural networks.”, 2020

[5] Pytorch,,“Torch.NN”, 2019.

https://pytorch.org/docs/stable/nn.html

[6] X. Wang, L. Gao, S. Mao, and S. Pandey, “Csi-based fingerprinting for indoor localization: A deep learning approach,” IEEE Transactions on Vehicular Technology, vol. 66, no. 1, pp. 763–776, 2017.

[7] PyTorch, “TORCH OPTIM”, 2020

[8] Purva Huligol, “Accuracy vs F1-score” 2019. https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2

[9] J Palacios, D Steinmetzer, A Loch, M Hollick, J Widmer, “Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11 ad Devices”, 2018

[10] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 1, pp. 221–231, 2013.

[11] G. Gkioxari, R. Girshick, P. Dollár, and K. He, “Detecting and recognizing human-object interactions, in Proceedings of,the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8359–8367.

[12] X. Wang, X. Wang, and S. Mao, “Cifi: Deep convolutional neural networks for indoor localization with 5 ghz wi-fi,” in 2017 IEEE International Conference on Communications (ICC). IEEE, 2017, pp. 1–6.

[13] Y. Zeng, P. H. Pathak, and P. Mohapatra, “Wiwho: Wifi-based person identification in smart spaces,” in Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN). IEEE, 2016, p. 4.

[14] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, “Falldefi: Ubiquitous fall detection using commodity wi-fi devices,” Proceedings of the Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), vol. 1, no. 4, p. 155, 2018.

[15] Q. Pu, S. Gupta, S.Gollakota, and S. Patel, “Whole-home gesture,recognition using wireless signals,” in Proceedings of the 19th annual international conference on Mobile computing & networking. ACM, 2013, pp. 27–38.