공학석사 학위논문

Categorization을 이용한 WiFi 기반 저복잡도 행동 인식 기법

WiFi-Based Low-Complexity Gesture Recognition using Categorization

2022년 2월

서울대학교 대학원

컴퓨터공학과

김 지 수

Categorization을 이용한 WiFi 기반 저복잡도 행동 인식 기법

WiFi-Based Low-Complexity Gesture Recognition using Categorization

지도교수 전 화 숙

이 논문을 공학석사 학위논문으로 제출함

2021년 12월

서울대학교 대학원

컴퓨터공학과

김 지 수

김지수의 석사 학위논문을 인준함

2021년 12월

위 원 장 문병로 (인)

부위원장 전화숙 (인)

위 원 권태경 (인)

Abstract

WiFi-Based Low-Complexity Gesture Recognition using Categorization

Kim Ji Soo

Department of Computer Science and Engineering

The Graduate School

Seoul National University

As smart homes and augmented reality (AR) become popular, the convenient human-computer interaction (HCI) methods are also attracting attention. Among them, many researchers have paid attention to gesture recognition that is simple and intuitive for humans. Camera-based and sensor-based gesture recognition have been very successful, but have limitations including privacy issues and inconvenience. On the other hand, WiFi-based gesture recognition using channel state information (CSI) does not have these limitations. However, since the WiFi signal is noisy, Deep learning (DL) models have been commonly utilized to improve the gesture recognition performance. DL models require large training data, large memory, and high computational complexity, resulting in long latencies that disrupt real-time systems. To solve this problem, support vector machines (SVMs) that require less computation and memory than

powerful deep learning models can be utilized. However, the SVM shows poor performance when there are many target classes. In this paper, we propose a categorization method that can divide ten gestures into four categories. Since only two or three target gestures belong to each category, a traditional machine learning model like support vector machine (SVM) can achieve high accuracy while requiring less computation and memory consumption than the DL models. According to the experimental results, when using the SVM alone, the accuracy is about 58%. However, when used with categorization, it can improve up to 90%. Furthermore, the gesture recognition performance of the DL models can also be improved by combining the proposed categorization method if the hardware has sufficient memory and computational complexity.

keywords : WiFi, Channel State Information, Gesture Recognition, Categorization, Low complexity
Student Number **: 2020-24710**

Contents

List of Figures

List of Tables

Chapter 1
Introduction

As smart homes and augmented reality (AR) become popular, the need for human-computer interaction (HCI) is also increasing. Among many HCI methods, gesture recognition can be a good solution because it is simple and intuitive for humans. For example, to increase volume of a TV or a radio, users only need to raise their hands without holding any devices.

Previous works on gesture recognition typically utilized camera-based[1-4] and sensor-based[5-8] approaches. Camera-based gesture recognition generally shows excellent performance. However, the personal information of the user may be captured as an image when using the camera at home. It is also well known that they are sensitive to changes in light that can degrade performance. On the other hand, sensor-based gesture recognition is relatively free from the above problems. However, since the user must wear a special device, it may cause inconvenience to the user. Recently, WiFi-based gesture recognition is receiving more attention in the research field because it does not have aforementioned problems. As shown in Figure 1, when a person moves, the WiFi signal is reflected by the movement. Thus, movement-related information can be found in the received signal at the receiver. When analyzing the WiFi signal, both received signal strength indicator (RSSI) and channel state information (CSI) can be used. However, RSSI is unstable due to the multipath effect, so it shows poor performance in indoor environment. CSI, On the other hand, contains amplitude and phase of multiple subcarriers. Thus, it has more fine-grained information than RSSI.

Therefore, most of recent WiFi-based gesture recognition utilizes CSI for gesture recognition.

Many recent researches leveraged deep learning (DL) models to predict what gesture occurred from CSI data. DL is a powerful tool that can show high accuracy without the human-engineered complex features. However, learning common patterns from data requires large amounts of training data. It also requires large memory and high computational complexity due to many parameters. Therefore, it is difficult to deploy a model on an AP because a typical AP is not so powerful to handle heavy DL model. Reducing model complexity is necessary for such situation. Therefore, we propose a gesture categorization method that can lighten the classifier. The main idea is that gestures have their own stopping points. For example, when changing direction, the user must stop at some points. We can categorize the gesture by detecting these stopping points which will be called gesture segments in this paper. The variance of amplitude of the CSI is the good indication to detect gesture segments. As shown in Figure 3, the variance of amplitude increases when the hand of user moves. Also, when the hand of user stops, the variance is almost zero. We can categorize the gesture by counting the number of gesture segments. For example, draw M and draw Rectangle fall into the same category because they have four gesture segments. After categorizing the gesture, each category contains only two or three gesture targets. As the result, ML classifiers such as support vector machines (SVMs) are sufficient to achieve high accuracy. This is the key enabler for the low-complexity system because ML classifier consumes little memory and requires low latency compared to the deep learning model. Furthermore, it also requires small training data which can reduce the

data collection effort. The main contributions of this paper are as follow:

- We propose the simple categorization method to enable lightweight WiFi-based gesture recognition even in the APs or IoT devices that have limited capability. Because our method divides ten gestures into four categories, each classifier only needs to classify two or three gestures. By making the problem simpler, machine learning (ML) models such as SVMs can show good performance for the gesture recognition. In general, the ML models require relatively less computation and memory than the DL models. Therefore, our method can reduce the complexity caused by the model.
- The proposed method shows superior performance compared to the traditional DL models and ML models. Although the traditional ML models and even DL models show 60~70% accuracy, our method shows nearly 90% accuracy for the data of user who is unseen during the training. Furthermore, If the hardware has sufficient memory ans computational capability, the DL models also can leverage our methods to improve the performance.

In chapter 2, we show our system model. Then, we explain the detail of our proposed methods such as preprocessing, categorization, and feature extraction in chapter 3. In chapter 4, we show the performance of our methods and draw our conclusions in chapter 5.

Chapter 2
System Model

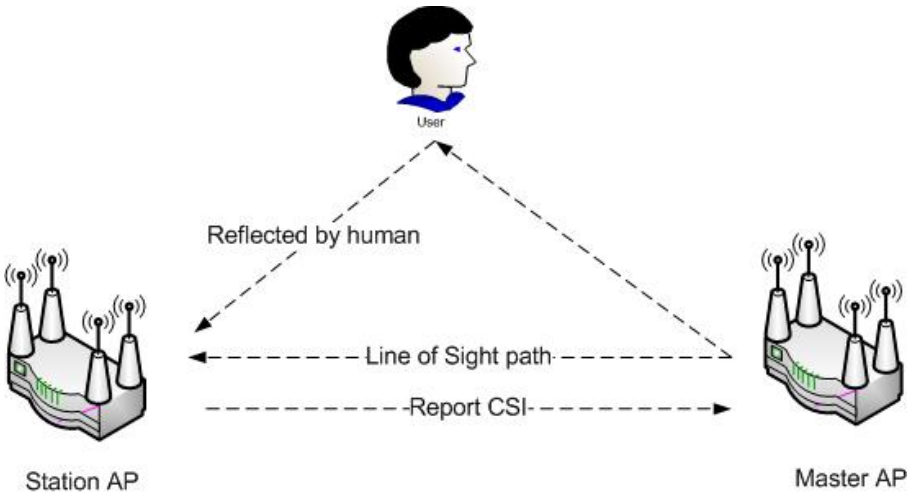


Figure 1. System model

As shown in Figure 1, the system consists of one master AP and one station AP. The master AP sends a null frame to the station AP every 25ms. After the station AP receives the null frame, it measures the CSI from the signal. As there are multiple OFDM subcarriers, the CSI contains several different channel frequency response (CFR) values. The CFR values can be represented as [9]:

$$H_{nm}(t) = |H_{nm}(t)|e^{j\angle H_{nm}(t)} \quad n \in [1,N] \quad m \in [1,M] \qquad (Eq.\ 1)$$

where N is the number of subcarriers, M is the number of chains, $H_{nm}(t)$ is the CFR value of subcarrier index n and chain index m measured at time t, $|H_{nm}(t)|$ is the amplitude of the CFR value, and $\angle H_{nm}(t)$ is the phase of the CFR value. Data collected from

qualcomm chipset contains the CSI for each antenna, so we can get MxN CFR values every 25ms. When there are L packets, the data dimension is LxMxN.

When the master AP sends a signal, the transmitted signal is affected by multipath effect. As shown in Figure 1, the signal is reflected by a person. As a result, the signal arriving at the station AP shows a change in amplitude and phase. So, by analyzing the patterns in the CFR values, we can tell which gestures occurred. However, the signal is not only affected by human movement but also by noise and human body size. In particular, according to the paper [10], the pattern of the CFR values varies from person to person. This is why previous researches leveraged powerful DL models to achieve good accuracy. However, our proposed method leveraged SVM models without using DL models. Although the SVM model is not so powerful compared to recent DL models, it can achieve similar accuracy with a less memory consumption and computation by using the categorization. We will explain the proposed method in detail at chapter 3.

Chapter 3
Proposed Scheme

3.1 Overview

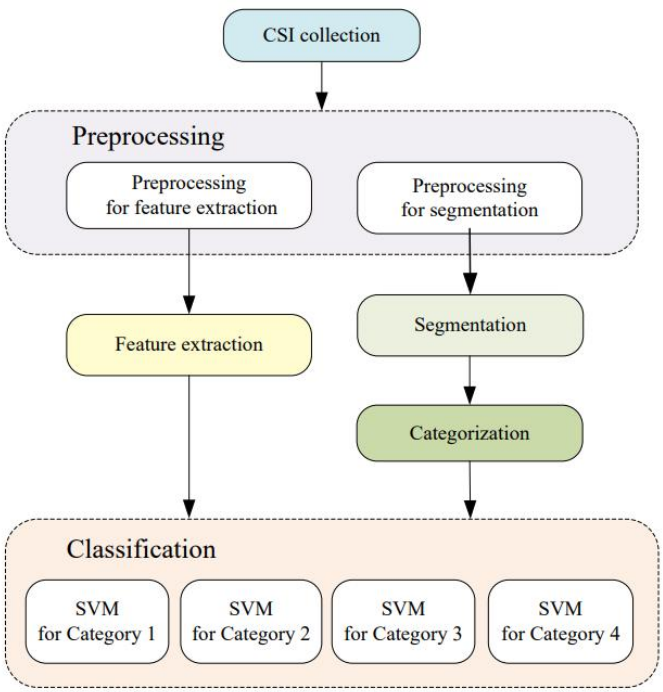


Figure 2. System flow

After CSI collection, we used amplitude and phase of the CFR values for classification. However, amplitude and phase contain noise and static energy that are not related with the human movement. Therefore, amplitude and phase must be preprocessed before classification. Preprocessing part can be divided into two parts. First, we denoise the amplitude using singular vector decomposition (SVD)

and discrete wavelet transform (DWT) denoising for classification. After applying the SVD to the CSI data, the largest eigenvalue is removed and the CSI data is reconstructed. The authors of paper [11] assumed that the eigenvector with the largest eigenvalue is the static energy caused by static object. After removing this static energy, the CFR values mostly contain the dynamic part related with human movement. Also, high frequency components must be eliminated since they are not likely from the human movement. However, some high frequency components may contain gesture-related information. So we utilized DWT denoising, which preserves some sudden changes while removing high frequency components. Second, we applied more preprocessing techniques for categorization. Variance in amplitude is a key enabler for categorization. However, even a small number of outliers and noise can have a significant impact on variance. Therefore, removing these outliers is important for categorization, even at the expense of some gesture-related information. Therefore, amplitude are preprocessed using primary component analysis (PCA), hampel filter, high pass filter (HPF), DWT denoising, and normalization. Details are described in the preprocessing section. After preprocessing, the denoised amplitude is fed to segmentation module. This module divides a gesture into units called gesture segments. As shown in Figure 3, each gesture shows a certain number of peaks. For example, since the gesture contains two gestures segments (push and pull), there are two peak in Figure 3(a). Therefore, we can find the start and end of each gesture segment using threshold-based segmentation. When the variance exceeds threshold, the gesture started and vice versa. Next, the start and end of gesture segments are fed to feature extraction module and categorization module. Simply, the number of gesture segment is a category of the gesture

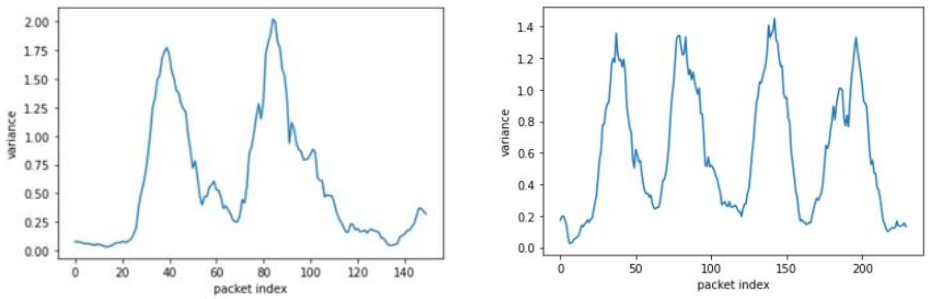
because each gesture has a certain number of gesture segments. Therefore, categorization module returns the number of gesture segments. Feature extraction module extracts statistical features from each gesture segment. Finally, the features and category number are fed to classification module. Classification module first selects the SVM corresponding to the category number. Then, the selected SVM the uses the given features to predict the gesture label. As the SVM only consider gestures corresponding to the category, prediction becomes easier.

3.2 Preprocessing

Naturally, the CSI of the WiFi signal is noisy due to multipath effect. Therefore, we need to preprocess the signal for better classification performance. In this paper, only amplitude is preprocessed. The raw phase is difficult to completely eliminate noise because there is a lot of noise caused by hardware defects such as CFO and SFO [12]. However, the raw phase still helps to improve the model accuracy. Therefore, we utilized both preprocessed amplitude and raw phase for gesture recognition. There are two preprocessing ways when denoising amplitude. The first is for the classification and the second is for the categorization. In the first preprocessing, we apply the SVD to each M amplitude matrix in LxN dimension and remove the first eigenvalue. Since the static energy due to the static environments is likely the largest component of the signal, removing the first eigenvalue and reconstructing the amplitude matrix will help remove this static energy. Then we also remove the high frequency components of signal. In general, the high frequency signals are noise. However, simply removing these high frequency signals can result in loss of meaningful information from high

frequency signals related to human movement. Therefore, DWT denoising is applied to preserve meaningful high frequency components while removing random noises. Then, the second preprocessing for categorization involves more preprocessing techniques. We used the variance of amplitude to categorize the gesture. Even a small noise can have a large impact on the variance. Therefore, we remove noise at the expense of some gesture-related information. First, we apply PCA and take only the first 20 primary components. PCA helps reduce the data dimensions while retaining important information. Then, we utilize a hampel filter to remove outliers. HPF is also applied to remove low frequency components that are not likely related with gesture. In addition, DWT denoising is applied to remove high frequency noise. Finally, we normalize the amplitude scale of each subcarrier. As a result, the amplitude values range from 0 to 1. This is good for solving the different transmission power problem mentioned in the paper[13].

3.3 Gesture Segmentation and Categorization



(a) push & pull (b) draw M

Figure 3. Variance when gesture occurred

The fluctuations in amplitude are affected by the movement of the person and vice versa. For example, variance of amplitude becomes smaller when the person stops moving. Therefore, we used variance to detect gesture segments. To analyze variance changes in the time-domain, the variance is calculated using a sliding window as below:

$$Mean_{nm}(t) = \frac{1}{w+1} \sum_{i=t-w/2}^{t+w/2} |H_{nm}(i)| \qquad (Eq.\ 2)$$

$$Var_{nm}(t) = \frac{1}{w+1} \sum_{i=t-w/2}^{t+w/2} (|H_{nm}(i)| - Mean_{nm}(t))^2 \qquad (Eq.\ 3)$$

$$Var_{m}(t) = \frac{1}{N} \sum_{i=1}^{N} Var_{im}(t) \qquad (Eq.\ 4)$$

$$Var(t) = \frac{1}{M} \sum_{i=1}^{M} Var_{i}(t) \qquad (Eq.\ 5)$$

Where w is the size of moving window. To obtain a variance having only one dimension, equation 4 and equation 5 are used to average all variances. As shown in the Figure 3(a), push & pull can be separated because there is a certain variance pattern when human moves and

stops. For example, we can find two peaks representing push and pull respectively. To find the start and end points of these gesture segments, we used the segmentation method in the paper [9]. The detail is described in the Figure 5. After finding the gesture segments, we only need to count the number of segments for categorization. As mentioned earlier, the gesture has the certain number of gesture segments. So the number of gesture segments is equal to the category of the gesture. In this paper, ten gestures are categorized into four categories. As shown in Figure 4, the first three gestures are category 1, the next three gestures are category 2, the next two gestures are category 3, and the last two gestures are category 4. Since there are only two or three gestures in each category, the SVM can easily find the decision boundaries of classes compared to the traditional methods. Finally, the module outputs a list of start and end points of each segment for feature extraction.

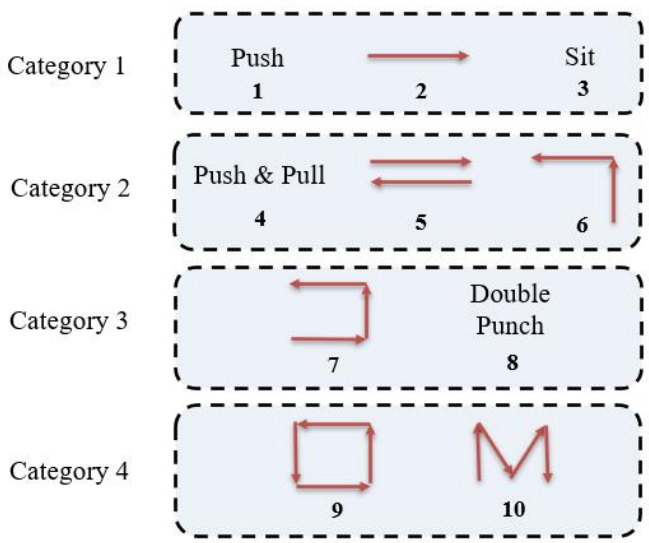


Figure 4. Gesture list

3.4 Feature Extraction

Table 1. Feature list

Domain	Features
Time	Mean, Variance, STD, Max, Min, Median, First quantile, Third quantile, skewness, kurtosis
Frequency	Max Frequency

DL models generally do not rely on feature engineering because they automatically find useful features in large amounts of data. However, DL models require many parameters and computations. Therefore, we utilized the SVM with less memory and computation load compared to DL models. However, ML techniques such as SVM perform worse than powerful deep learning models. Feature extraction is very important to improve the performance of the SVM. In this paper, we utilize both time-domain and frequency-domain features similarly in paper [14]. The features are shown in table 1. Features are calculated in every segment of the gesture. As a result, each category has different number of features. Since we have four categories, we used a total of four SVMs with different input sizes.

3.5 Classification

Algorithm 1 Proposed Scheme

Input:
CSI data ($L \times M \times N$), SVM for each category, threshold
Output: Gesture_label

$phase \leftarrow$ phase of CSI data
$amp \leftarrow$ amplitude of CSI data
$amp_{\text{feature}} \leftarrow \text{preprocess}_{\text{feature}}(amp)$
$amp_{\text{seg}} \leftarrow \text{preprocess}_{\text{seg}}(amp)$
$var \leftarrow$ get_variance(amp_{seg})

gesture_seg $\leftarrow \emptyset$; start $\leftarrow false$
start_idx $\leftarrow 0$; end_idx $\leftarrow 0$
seg_count $\leftarrow 0$
for $l \leftarrow 0$ to $L-1$ **do** /* l is the index of var */
  if $\text{var}_l \geq threshold$ **then**
    if $start = false$ **then**
      start $\leftarrow true$
      start_idx $\leftarrow l$
      end_idx $\leftarrow l$
    end if
  else
    if $start = true$ **then**
      start $\leftarrow false$
      end_idx $\leftarrow l$
      $\text{gesture\_seg}_{\text{seg\_count},start} \leftarrow$ start_idx
      $\text{gesture\_seg}_{\text{seg\_count},end} \leftarrow$ end_idx
      seg_count $\leftarrow$ seg_count + 1
    end if
  end if
end for

features_list $\leftarrow \emptyset$
for $g \leftarrow 0$ to seg_count **do** /* g is the segment index */
  start_idx $\leftarrow \text{gesture\_seg}_{g,start}$
  end_idx $\leftarrow \text{gesture\_seg}_{g,end}$
  $amp_{\text{seg}} \leftarrow amp_{\text{feature}}$[start_idx, end_idx]
  features $\leftarrow$ feature_extraction(amp_{seg})
  $\text{features\_list}_g \leftarrow$ feature_list $\cup$ features
end for
category_idx $\leftarrow$ seg_count
Gesture_label = $\text{SVM}_{\text{category\_idx}}$(features_list).

Figure 5. Algorithm of proposed method

To classify gestures, we utilize the SVM. The whole process of the algorithm is shown in Figure 5. First, we get the amplitude and phase from CSI data. Then, the amplitude is preprocessed for classification and categorization. After calculating variance, the start and end points of each gesture segment is found by applying threshold to variance. Then, features are calculated from every segment. After getting the features and the number of segments, the SVM corresponding to the category is selected. Using the selected SVM, we predict the gesture label from given features.

Chapter 4
Performance Evaluation

4.1 Experimental Setup

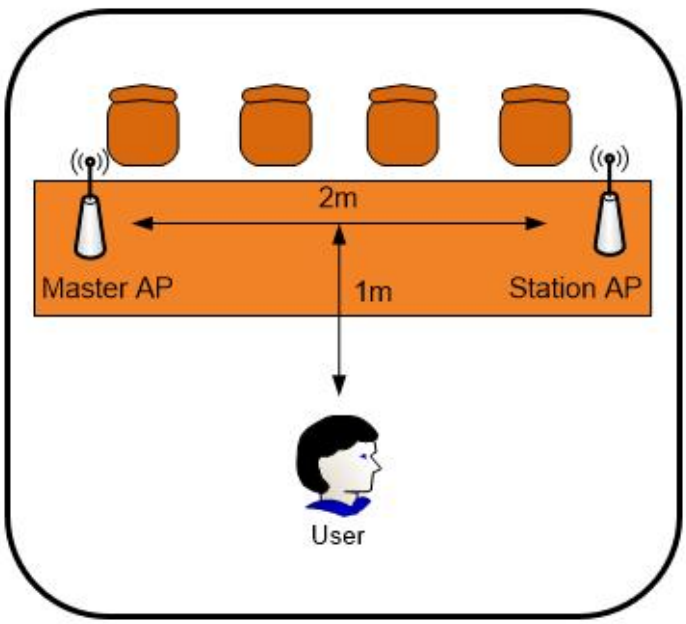


Figure 6. Experimental environment

The CSI data was collected using a qualcomm chipset. For stable acquisition, we set the data rate to 40 packets/sec. To avoid and to get more information, experiments are performed on the channel of 5GHz frequency band with 80MHz bandwidth. As shown in Figure 6, each user performed gestures in front of APs. We collected the CSI data from three people. Each person performed ten gestures 20 times. As a result, a total of 600 (3*10*20) CSI data were collected. Then, the system categorize the recorded data and only correctly categorized data is utilized as training dataset and test

dataset. After categorization, the training dataset is fed to the SVM and test dataset is used to validate the performance of SVM. In order to show the latency of CPU and GPU, training and testing are conducted in Google colab, which provides high-performance CPU and GPU for free. Google colab offers Ubuntu 18.04.5 LTS OS, Intel(R) Xeon(R) CPU(2.20GHz), and Tesla P100-PCIE-16GB(GPU).

4.2 Categorization Performance

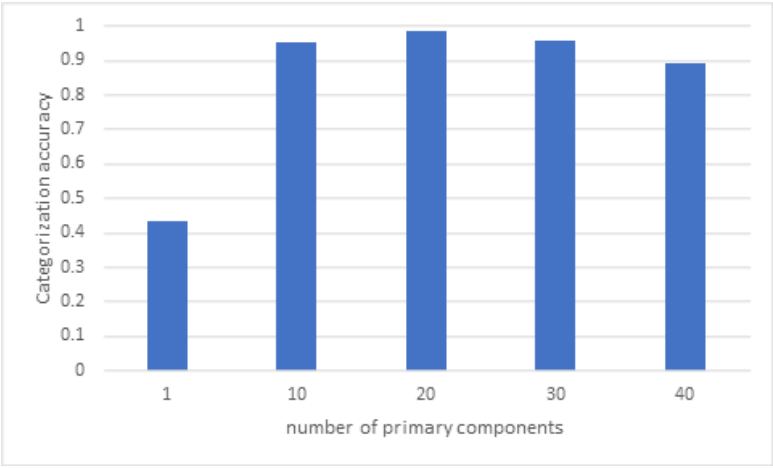


Figure 7. Relationship between categorization accuracy and number of primary components

After one gesture segment, the participant pauses the gesture for approximately 0.5 seconds. This is realistic because person usually stops before performing the next gesture segment. Categorization parameters such as thresholds were set empirically. In addition, since the number of primary components can affect the categorization accuracy, we analyze the relationship between categorization accuracy and number of primary components. When the number of primary components is too small, the information is not enough for

categorization. On the other hand, when the number of primary components is too big, useless information can interfere the categorization. Therefore, we chose the number of primary components as 20 which showed best accuracy.

Table 2. Confusion matrix of categorization

	1	2	3	4
1	0.99	0.01	0	0
2	0.03	0.97	0	0
3	0	0.02	0.98	0
4	0	0	0	1

Table 2 shows the confusion matrix of the categorization when using 20 primary components. The rows in the table represent the actual category, and the columns in the table represent the predicted category. As it shows, 98.5% of the CSI data is correctly categorized. Some gestures are not categorized correctly, but the prediction is still close to the right answer.

4.3 Overall Performance

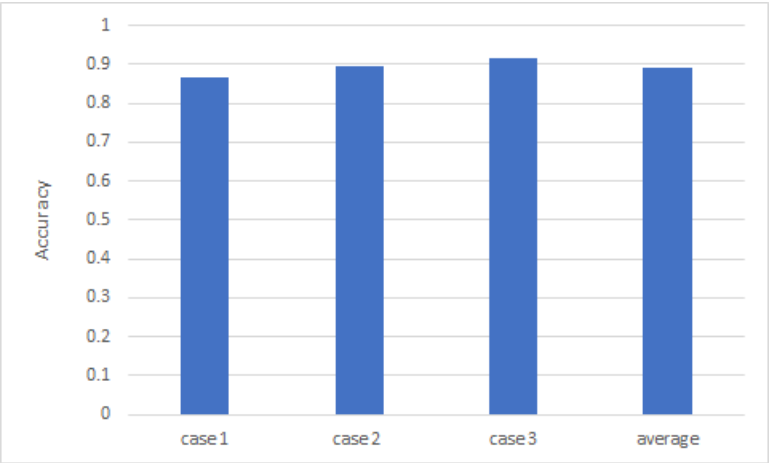


Figure 8. Overall performance of C+SVM

Figure 8 illustrates the overall performance of the SVM using categorization (C+SVM). For the case 1, we train the SVM using data of user 2 and user 3. Then, we test the SVM using data of user 1. Similarly, data of user 2 is test data in case 2 and data of user 3 is test data in case 3. The SVM should work well for the unknown users as it does not know the user's information in advance. Therefore, we evaluate the performance using data of the unknown user. As shown in Figure 8, the average accuracy for test data is approximately 89%. We focused on the case 1 which shows worst performance. In case 1, the heights of user 2 and user 3 is similar, but the height of user 1 is different. Thus, the SVM could not generalize to user 1 and was less accurate. Collecting data of various users may be helpful improving the model. Confusion matrix for the case 3 is shown in Table 3. As we only used correctly categorized data, prediction only occurred within the category.

Table 3. Confusion matrix for case 3

	1	2	3	4	5	6	7	8	9	10
1	0.9	0	0.1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0.1	0	0.9	0	0	0	0	0	0	0
4	0	0	0	0.95	0	0.05	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0.15	0.05	0.8	0	0	0	0
7	0	0	0	0	0	0	0.9	0.1	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0	0.95	0.05
10	0	0	0	0	0	0	0	0	0.1	0.9

4.4 Performance comparison with baseline

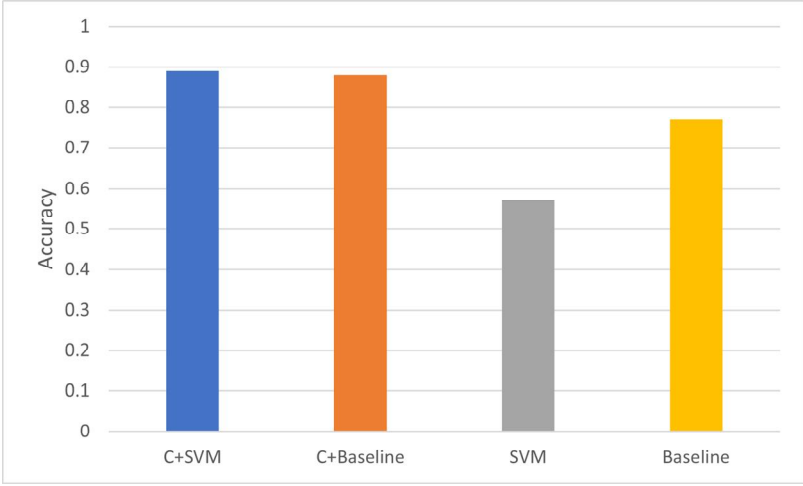


Figure 9. Accuracy comparison

We also compare the performance with the baseline [9] that used DL models. The baseline leveraged pre-trained CNN to extract spatial features from the CSI data. By doing so, it overcomes the need of large training data that is the shortcoming of the DL models. As shown in Figure 9, the baseline performs better than the SVM without categorization. However, with categorization, C+SVM shows higher accuracy than the baseline. Unexpectedly, C+Baseline shows rather lower accuracy than C+SVM because the SVM can be more effective than DL model when there are few targets and few data. Although DL models can perform better than C+SVM when the amount of data is large, categorization still can be helpful improving the DL model. In addition, we compare the memory and latency of the proposed system and the baseline. Since APs and IoT devices have limited hardware performance, memory and latency are significant issues. Memory and latency comparisons are shown in Table 4. As shown in Table 4, the baseline requires 417.74MB to store the parameters of DL model. On the other hand, the proposed method only requires 0.28MB (sum of the four SVMs' memory). That is, the memory consumption of the C+SVM can be significantly reduced. Beyond that, the latency of the proposed method (mean of the four SVMs' latencies) is also lower than the baseline. For CPU, proposed method shows about 3000 times lower latency. When using the GPU, the difference is not so significant because the DL model can leverage parallelism to speed up. However, in real deployments, GPUs are often not available in typical APs and IoT devices. Therefore, our proposed method can be better than DL models in these situations.

Table 4. Memory and latency comparison

	SVM 1	SVM 2	SVM 3	SVM 4	Proposed method	Baseline
Memory (MB)	0.08	0.07	0.06	0.07	0.28	417.74
CPU Latency (ms)	0.229	0.211	0.21	0.204	0.2135	678
GPU Latency (ms)	0.209	0.212	0.204	0.204	0.20725	5.11

4.5 Effect of the channel

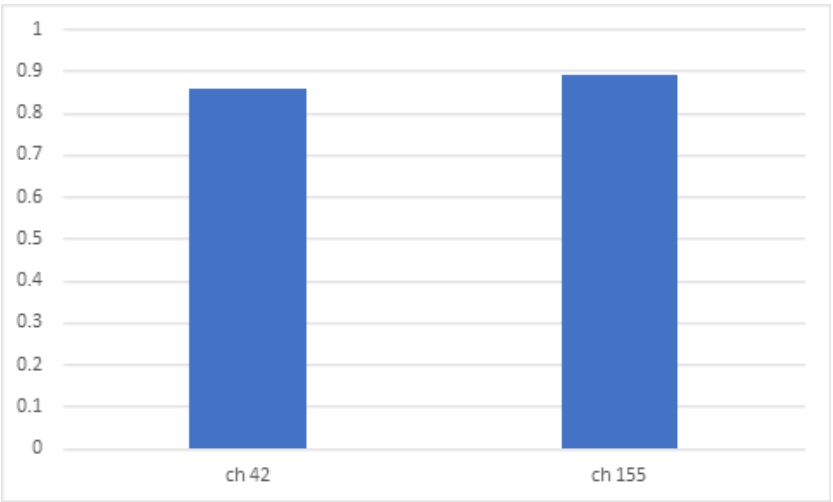


Figure 10. Accuracy of models trained by data at different channels

To see the effect of the channel, we collected data at channel 42 and channel 155. As shown in Figure 10, channel does not significantly affect the performance. However, it is an important issue to select a channel with low interference [15]. Appropriate channel selection method is left for future work.

Chapter 5
Conclusion

As smart home and AR become popular, intuitive HCI is required to control embedded systems. Among many methods, gesture recognition is suitable for HCI because it is very intuitive and natural for human. Therefore, many researchers have focused on gesture recognition using camera and sensor. However, camera-based method can invade user's privacy because it can capture the images of user. In addition, sensor-based method is inconvenient because user has to carry sensor devices. Recently, a new method using WiFi signal is attracting attention. Since WiFi signal do not capture visual information of user, It is free from privacy problem. Furthermore, it is very comfortable because there is no need to carry devices. Although there are many benefits, WiFi signal also has a shortcoming. Due to hardware imperfection and multipath effect, WiFi signal has lots of noises. Thus, recognizing patterns of WiFi signals corresponding to gestures is difficult. Many traditional methods tried to solve the problem using powerful DL models. As the DL models automatically find the good features from data, they showed good performance. However, DL models have too many parameters that consume a large amount of memory and computation. In this paper, we use the SVM with less memory and computation than DL models. In addition, the SVM shows similar performance with DL models using the proposed categorization method. Since each category has only two or three gestures, the SVM can easily find the decision boundary of classes. As a result, the SVM with categorization shows average accuracy of nearly 90% for users who are not seen during

the training. Furthermore, our system requires very little memory and latency compared to the baseline which leveraged a DL model. Our categorization method also can be applied to DL models. Therefore, we expect our method can help improving the performance of gesture recognition.

Bibliography

[1] TRAN, Dinh-Son, et al. Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network. Applied Sciences, 2020, 10.2: 722.
[2] BHAGAT, Neel Kamal; VISHNUSAI, Y.; RATHNA, G. N. Indian sign language gesture recognition using image processing and deep learning. In: 2019 Digital Image Computing: Techniques and Applications (DICTA). IEEE, 2019. p. 1-8.
[3] ARACHCHI, SP Kasthuri, et al. Real-time static and dynamic gesture recognition using mixed space features for 3D virtual world's interactions. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2018. p. 627-632.
[4] LIAO, Bo, et al. Hand gesture recognition with generalized Hough transform and DC-CNN using realsense. In: 2018 Eighth International Conference on Information Science and Technology (ICIST). IEEE, 2018. p. 84-90.
[5] AÑAZCO, Edwin Valarezo, et al. Hand Gesture Recognition Using Single Patchable Six-Axis Inertial Measurement Unit via Recurrent Neural Networks. Sensors, 2021, 21.4: 1404.
[6] COFFEN, Brian; MAHMUD, Md Shaad. TinyDL: Edge Computing and Deep Learning Based Real-time Hand Gesture Recognition Using Wearable Sensor. In: 2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM). IEEE, 2021. p. 1-6.
[7] WU, Jian; JAFARI, Roozbeh. Orientation independent activity/gesture recognition using wearable motion sensors. IEEE Internet of Things Journal, 2018, 6.2: 1427-1437.

[8]XIE, Baao; LI, Baihua; HARLAND, Andy. Movement and gesture recognition using deep learning and wearable-sensor technology. In: Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition. 2018. p. 26-31.
[9] DING, Jianyang; WANG, Yong. WiFi CSI-Based Human Activity Recognition Using Deep Recurrent Neural Network. IEEE Access,2019, 7: 174257-174269.
[10] LI, Xinyi, et al. CrossGR: Accurate and Low-cost Cross-target Gesture Recognition Using Wi-Fi. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2021, 5.1: 1-23.
[11] BU, Qirong, et al. Deep transfer learning for gesture recognition with WiFi signals. Personal and Ubiquitous Computing, 2020, 1-12.
[12] XIE, Yaxiong; LI, Zhenjiang; LI, Mo. Precise power delay profiling with commodity Wi-Fi. IEEE Transactions on Mobile Computing, 2018, 18.6: 1342-1355.
[13] TAN, Sheng; YANG, Jie. WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition. In: Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing. 2016. p. 201-210.
[14] HUANG, Jinyang, et al. Towards Anti-interference Human Activity Recognition Based on WiFi Subcarrier Correlation Selection. IEEE Transactions on Vehicular Technology, 2020.
[15] ZHENG, Yue, et al. Detecting radio frequency interference for CSI measurements on COTS WiFi devices. In: 2017 IEEE International Conference on Communications (ICC). IEEE, 2017. p. 1-6.

요약

스마트홈과 증강현실(AR)이 보편화되면서 편리한 인간-컴퓨터 상호작용 방식도 주목받고 있다. 그 중 많은 연구자들이 인간에게 간편하고 직관적인 Gesture Recognition에 주목해 왔다. 카메라 기반 및 센서 기반 Gesture Recognition은 매우 성공적이었지만 개인 정보 보호 문제 및 불편함 등의 한계가 있다. 반면, 채널 상태 정보(CSI)를 이용한 WiFi 기반 Gesture Recognition은 이러한 제한이 없다. 그러나 WiFi 신호에 노이즈가 많기 때문에 Gesture Recognition 성능을 향상시키기 위해 딥러닝 모델이 일반적으로 활용되었다. 딥러닝 모델은 대규모 훈련 데이터와 대용량 메모리가 필요하고 높은 계산 복잡도로 인해 실시간 시스템을 방해하는 긴 지연 시간을 초래한다. 이 문제를 해결하기 위해 강력한 딥러닝 모델보다 연산과 메모리가 덜 필요한 SVM을 활용할 수 있다. 하지만 SVM은 대상 클래스가 많을수록 성능이 크게 저하되는 문제가 있었다. 따라서 gesture를 여러 범주로 나눔으로써 대상 클래스를 줄이는 범주화 방법을 제안한다. gesture segment라고 하는 gesture unit을 찾는 것이 범주화 방법의 핵심이다. 각 Gesture는 고유한 gesture segment 개수를 가지므로 gesture를 숫자로 범주화할 수 있다. 예를 들어, 밀기 및 당기기와 같은 연속 gesture에는 두 개의 segment가 있다. 첫 번째 segment는 밀기이고 두 번째 segment는 당기기이다. 사람들이 현재 gesture segment를 중지하고 다음 gesture segment를 수행할 때 segment 사이에 short pause가 발생한다. 우리는 CSI 진폭의 변동을 분석하여 이러한 short pause를 찾을 수 있음을 관찰했다. CSI의 진폭은 사람이 움직일 때 더 커지고 그 반대도 마찬가지이다. 이를 바탕으로 진폭의 변화를 이용하여 short pause를 찾고 gesture segment를 나누는 범주화 방법을 제안한다. 범주화 이후 범주에 해당하는 SVM이 CSI 데이터를 사용하여 발생한 gesture를 결정한다. 제안된 범주화 방법은 98.5%의 정확도를 보였고 최종적으로 10개의 gesture에 대해 SVM의 성능을 약 30% 향상시킬 수 있었다. 또한 우리가 제안한 시스템은 딥러닝 모델을 활용한 비교대상에 비해 훨씬 적은 메모리와 지연 시간을 필요로 한다. 이 결과는 제안된 방법이 준수한 정확도를 가지며 AP 및 IoT 장치와 같은 제한된 하드웨어에도 배포할 수 있음을 보여준다.

주 요 어 : WiFi, Channel State Information, 행동 인식 Categorization, Lightweight

학 번 : 2020-24710