Sign Language Recognition with CW Radar and Machine Learning

Yilong Lu
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
EYLu@ntu.edu.sg

Yue Lang
School of Electrical and Information Engineering
Tianjin University
Tianjin, China
LangYue@tju.edu.cn

***Abstract*—Sign language is the primary communication medium for the deaf-mute community. However, the literacy of understanding and using sign language is hard to gain without professional training. In this paper, we explore the use of low power frequency modulated continuous wave radar for automatic sign language recognition. The proposed system is composed of a radar, a sound cluster and a computer for transforming signals to spectrograms. Furthermore, as the time-frequency spectrograms are high-dimensional data with redundant information, we then perform dimensionality reduction by extracting the histogram of oriented gradients features from these spectrograms. The features are finally classified by the k-Nearest Neighbour algorithm and a classification result of 95.8% is achieved on the five testing signs. The impact of the k value in the k-Nearest Neighbour is also investigated.**

***Index Terms*—Radar micro-Doppler, machine learning, sign language recognition.**

I. Introduction

Sign language, as a special means of communication, is of vital importance to the hearing-impaired community. It was reported in 2008 that there were about 500 million deaf-mute people all over the world [1], and the number kept increasing in the last decade. However, the users of sign language are quite limited, especially among the hearing community. In this case, the hearing-impaired people are hard to adapt to the social life. To provide a automatic and reliable translation in the form of text/speech, In this case, sign language recognition (SLR) has attracted considerable interest in recent years.

A majority of SLR methods are based on vision, such as video camera, stereo camera, and Kinect sensor. For example, the authors of [2] extract the trajectories features and the hand-shape features from the sign videos and utilize a Hidden Markov Model (HMM) for SLR. The authors of [3] designed a recurrent convolutional neural network (RCNN) to address the mapping of video segments to glosses and achieve promising result in continuous sign language recognition. Yin et al. [4] focused on the inter-signer variation problem and proposed a Reference Driven Metric Learning (RDML) strategy. For stereo-based SLR, Grzeszcuk et al. [5] proposed a hybrid gesture representation that could model the user's arm as a 3D line and reported a 96% recognition rate on a set of six gestures. Compared with the former two sensors, Kinect is more popular these days and has been applied to many sign languages [6]–[8]. Though widely studied, these sensors mainly suffers from the self-occlusion problem and are susceptible to the illumination when tracking the movement of the hands.

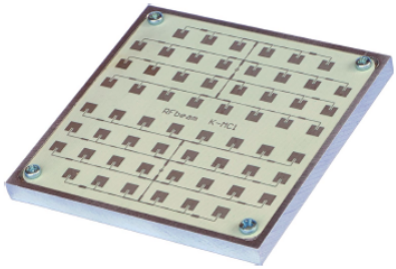
Radar, due to its robustness to the environmental changes, is considered as an alternative to the SLR problem. When a signer performing a sentence, there will be spatial relative translation between the human hands and the radar, causing the Doppler effect. Meanwhile, the movement of the signer's fingers will induce micro-Doppler signatures in addition to the central Doppler frequency. Some studies have employed radar micro-Doppler signatures for hand gesture classification [9]–[11], whereas the referred gestures in these works are mainly short and simple movements with no semantic meanings. To the best of our knowledge, few works have been done for sign sentence recognition using radar sensors.

In this paper, we propose a SLR method using a low-cost Continuous Wave (CW) radar. As a preliminary study, we investigate five simple sign language words and sentences, that is "Yes", "No", "Hello", "Are you deaf ?", and "I am learning sign". The sign signals are first captured by a radar sensor and is then digitized by a ADC and fed into a notebook PC for data processing. The digitized signals are first transformed to spectrograms, and machine learning algorithms are applied and compared for classification.

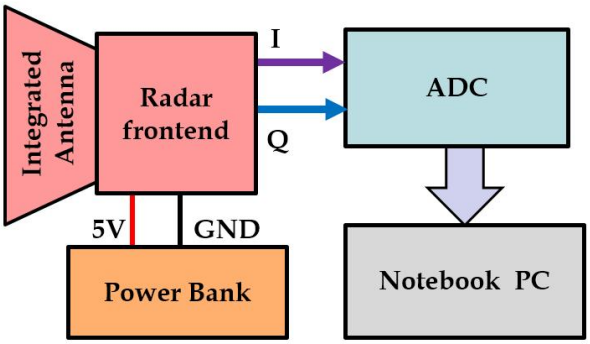
II. Radar Hardware Integration

A low-cost Continuous Wave (CW) radar sensor is integrated based on a K-MC1 radar transceiver from RFbeam [12] and a widely available low cost analog-to-digital converter (ADC) with USB interface for power supply and data recording as shown in Fig. 1. The K-MC1 CW radar transceiver operates at the frequency of 24 GHz with a relatively low cost less than USD 200 and a low transmission power of 100 mW. An high gain antenna with 30 patches for transmission and 30 patches for receiving are integrated on the radar module in a very compact size.

The output Intermediate Frequency (IF) signals are of very small bandwidth and can be digitized by widely available low-cost ADC with a USB interface for power supply and data recording and data processing by a normal notebook PC.



(a)



(b)

Fig. 1. (a) K-MC1 transceiver with antenna; (b) whole radar sensor configuration.

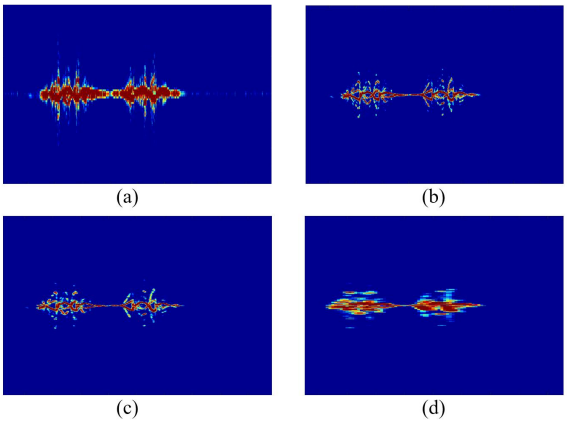


Fig. 2. Spectrogram images of (a) 0.01 sec; (b) 0.05 sec; (c) 0.1 sec; (d) 0.5 sec.

III. Radar Data Processing

Complex radar echo output as I and Q analog signals are digitized by a ADC and delivered to a notebook PC for data processing. The ADC sampling rate and depth can be selected and here we have selected the parameters of 96 kHz and 24 bits for the highest performance.

Fast Fourier Transform (FFT) is then implemented for joint time-frequency analysis. As FFT suffers from the spectral leakage problem, a Taylor window, which is prominently used in radar signal processing, is adopted. Fig. 2 shows the resulted spectrogram examples with different window sizes.

IV. Machine Learning for SLR

A. Histogram of Oriented Gradients (HOG) Feature

HOG was first developed as a feature extraction technique for pedestrian detection [13] and has gained great success in many other applications. HOG is a global feature which calculates and presents the statistical result of the gradient direction of the pixels. The essential idea of HOG descriptor is that the directions and magnitudes of the gradients can well characterize a local target's edge and appearance.

The feature extraction process is done with the following steps:

Step 1: The image is first filtered with the operators $[-1, 0, 1]$ and $[-1, 0, 1]^T$ to calculate the oriented gradients. Given the pixel value at (x, y) as $H(x, y)$, the gradients can be calculated by (1) and (2):

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (1)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \quad (2)$$

where $G_x(x, y)$ and $G_x(x, y)$ denotes the horizontal and the vertical gradients, respectively.

Then the gradient magnitude $G(x, y)$ and its angle $\theta(x, y)$ are given by (3) and (4):

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (3)$$

$$\theta(x, y) = \arctan \frac{G_y(x, y)}{G_x(x, y)} \quad (4)$$

Step 2: The image is then divided into multiple spatial regions (termed as cells), whose size is determined according to the resolution of the image and the scale of features of interest. In this work, a cell of 8×8 is selected.

Step 3: The local feature vector can be obtained by collecting and quantifying the gradient information in the cell. The gradients are integrated into nine "bins", which refer to the directions evenly distributed in [0°, 360°]. The histogram is the voting result of magnitude in terms of the angle of each pixel's gradient. It should be noted that any angle between two bins will result in the magnitude of the cell being split and assigned proportionately to the two angles.

Step 4: As contrast between the radar signal and the spectrogram background are varied across different cells, the gradient magnitudes are easily affected by the pixel value and would distribute in a wide range. In this case, a strategy named "block normalization" is adopted to deal with these variances and regularize the the gradient magnitudes. Block normalization is to integrate multiple cells into a larger image patch (termed as "block") and normalize the histograms in each block. It is worth noting that there are overlaps between adjacent blocks so that the histogram in each cell will be normalized for several times in different blocks. Finally, the histograms from the cells are cascaded to form a feature descriptor of the spectrogram.

B. Machine Learning Algorithms

Five popular machine learning algorithms, including Support Vector Machine (SVM) [14], Random Forest (RF) [15], Extreme Learning Machine (ELM) [16], Decision Tree (DT) [17], and k-Nearest Neighbor (k-NN) [18] are selected and compared for SLR based on radar micro-Doppler spectrograms. The algorithms are briefly described below.

1) Support Vector Machine (SVM) is a supervised learning algorithm (i.e. labeled training dataset is required) used for classification and regression problems. The aim of SVM is to find a hyperplane that clearly classifies the given dataset in an N dimensional space and with the maximum margin. The hyperplane is the classification boundary in a classification problem. While there are many possible hyperplanes for a given classification problem, the ideal hyperplane maximizes the margin which is the sum of distances between the hyperplane and the support vectors of each class. Support vectors refer to the data points that are closer to the hyperplane and can influence the position and orientation of the hyperplane. The model with maximum margin distance provides greater confidence for the data points to be classified.

2) Decision Tree (DT) makes decisions based on a tree-like structure and aims at splitting a data set into different categories based on different situations identified from the dataset. Three terms are defined in a decision tree, namely the internal node, the edges and the leaf. The internal nodes are the conditions for splitting into the different branches. When each branch ends, and there are no more branches from that node, it is called a "leaf". A leaf is a decision that has been made after all the branches being gone through. In a decision tree, there are many branches and conditions, allowing accurate classification of complicated algorithms. The root node is to make the split decision leading to the least loss in accuracy. The decision tree ends based on the longest path from the root node to the last leaf.

3) Random forest (RF) consists of multiple relatively uncorrelated decision trees. Each individual tree in the random forest votes for a class prediction and the class with the most votes would be chosen as the final prediction. The aforementioned DTs are the basic structural elements of the RF model. A node in a tree acts as the point where the training dataset splits into two branches, which depends on whether the data meets the criteria of the node. The ideal node is one that splits data in a way that the resulting two groups are as discriminative as possible. In a RF model, the trees should be relatively uncorrelated so that the ensemble prediction could avoid individual error of each tree. Thus, the result would be more accurate than the prediction of any individual tree. The possibility of getting a correct classification result increases with the number of uncorrelated trees in the model.

4) Extreme Learning Machine (ELM) is a neural network consisting of multiple layers that contribute to the final output. ELMs are based on single-layer feed forward neural networks with random weights assigned to them. There are typically two types of hidden nodes in an ELM, namely the additive hidden nodes and the radial basis hidden nodes. The ELM will take the input as an array of nodes and will match the array to its hidden node set. During this progress, no adjustments from the user is required. The hidden neurons will transform the input data into a different representation by first matching it with the hidden layer neurons through its own specifications and biases. Then the neurons apply a non-linear function onto the hidden neuron layer. The function is then approximated by the weights between the last layer of the hidden nodes and the output node layer.

5) K-Nearest Neighbor (k-NN) is a instance-based machine learning method. At the learning stage, the model stores all of the training samples and calculates the distances between the test sample and each training sample. Then, k nearest training samples are selected, and the label of the current test sample is determined according to the voting result, i.e. the majority category of these nearest neighbors would be assigned to the test sample. In order to obtain better classification performance, the value of k should be chosen carefully. In this work, the k value is set as 5 to achieve the best performance.

V. Experimental Results

With a total of 1000 training data for the five simple sign language words and sentences "Yes", "No", "Hello", "Are you deaf ?", and "I am learning sign" (200 for each sign language word or sentence), the five classifiers are trained respectively. Additional 100 testing data (20 for each sign language word or sentence) was used to test and compare the performance of the five classifiers. Sample spectrogram images of these five sign languages are shown in Fig. 3.

The test results are shown in Table 1. Based on the preliminary study, it seems that k-NN can achieve an superior accuracy of 95.8%.

We then investigated the impact of the k value. Fig. 4 shows the classification accuracies and the training time when k value varying from 1 to 30. It can be observed that the accuracy of k-NN is unstable when k is small (i.e. k = 1 or 2). This is because a testing sample's nearest neighbour may not have the same class with it. A more reasonable way is to extend the reference range to three or more neighbours, and the classifier gains the best performance at k = 5. On the other hand, the training time declines when k changes from 1 to 6, and increases dramatically when k = 7. Overall, to seek for the balance between these two key factors, k = 5 is chosen for this work and the processing time is about 26 seconds on a normal notebook PC with a 2.7 GHz CPU and 8 GB Memory.

TABLE I
Classification results comparison between different algorithms.

Algorithm	Accuracy
Support Vector Machine (SVM)	94.5%
Random Forest (RF)	93.5%
Decision Tree (DT)	65.0%
Extreme Learning Machine (ELM)	83.3%
K-Nearest Neighbor (k-NN)	**95.8%**

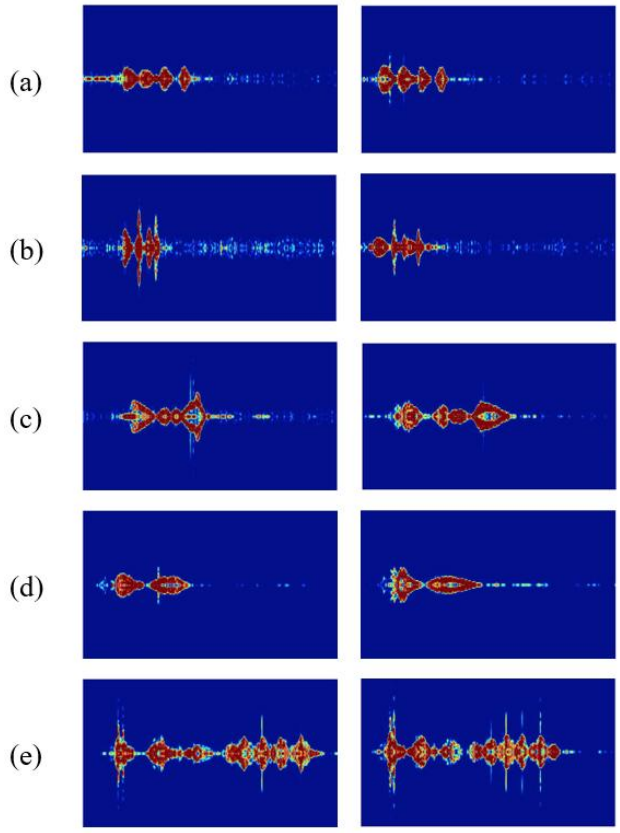


Fig. 3. Sample spectrograms of sign language words/sentences (a) “Ye”, (b) “No”, (c) “Hell”, (d) “Are you deaf?”, and (e) “I am learning sign”.

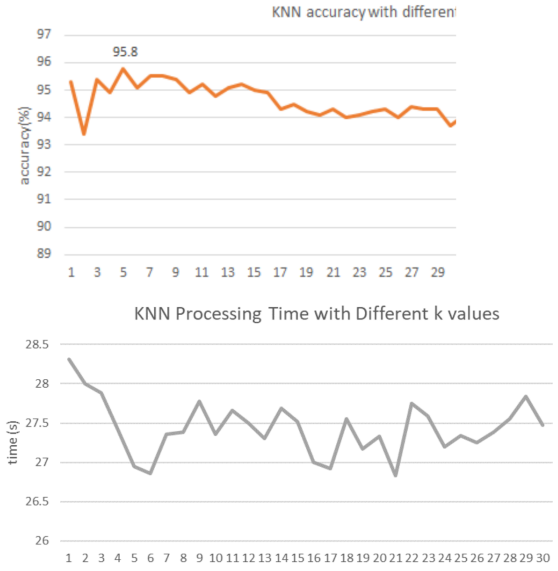


Fig. 4. Classification accuracy and the training time with varying k values.

VI. Conclusion

In this paper, we proposed a radar-based sign language recognition method. The HOG features is extracted from the time-frequency spectrograms. A k-NN classifier is used to classify five common signs and reports satisfying result as 95.8 %. The impact of the k value is further explored and it is worth noting that with the increasing of the k, the classification result increases at the early stage and will then declines, thus the k value should be carefully considered in the classification task. In future work, we will further expand the dataset, and try to realize real time translations of various sign languages into spoken words.

References

[1] I. Yoo and D. Yook, “Automatic sound recognition for the hearing impaired,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 2029–2036, November 2008.

[2] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, “Chinese sign language recognition with adaptive hmm,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, July 2016, pp. 1–6.

[3] R. Cui, H. Liu, and C. Zhang, “Recurrent convolutional neural networks for continuous sign language recognition by staged optimization,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] F. Yin, X. Chai, and X. Chen, “Iterative reference driven metric learning for signer independent isolated sign language recognition,” in *European Conference on Computer Vision (ECCV) 2016 Workshops*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 434–450.

[5] R. Grzeszcuk, G. Bradski, M. H. Chu, and J. . Bouguet, “Stereo based gesture recognition invariant to 3d pose and lighting,” in *IEEE Conference on Computer Vision and Pattern Recognition.*, vol. 1, June 2000, pp. 826–833.

[6] G. C. Lee, F.-H. Yeh, and Y.-H. Hsiao, “Kinect-based taiwanese sign-language recognition system,” *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 261–279, 2016.

[7] J. E. Yauri Vidalón and J. M. De Martino, “Brazilian sign language recognition using kinect,” in *European Conference on Computer Vision (ECCV) 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer, November 2016, pp. 391–402.

[8] C. Chansri and J. Srinonchat, “Reliability and accuracy of thai sign language recognition with kinect sensor,” in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, June 2016, pp. 1–4.

[9] Y. Kim and B. Toomajian, “Hand gesture recognition using micro-doppler signatures with convolutional neural network,” *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[10] G. Li, S. Zhang, F. Fioranelli, and H. Griffiths, “Effect of sparsity-aware time–frequency analysis on dynamic hand gesture classification with radar micro-doppler signatures,” *IET Radar, Sonar & Navigation*, vol. 12, no. 8, pp. 815–820, 2018.

[11] B. Dekker, S. Jacobs, A. S. Kossen, M. C. Kruithof, A. G. Huizing, and M. Geurts, “Gesture recognition with a low power fmcw radar and a deep convolutional neural network,” in *2017 European Radar Conference (EURAD)*, Oct 2017, pp. 163–166.

[12] ”K-MC1 Radar Transceiver”, *RFbeam Microwave GmbH, Switzerland.*

[13] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *International Conference on Computer Vision & Pattern Recognition (CVPR ’05)*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 1. San Diego, United States: IEEE Computer Society, Jun. 2005, pp. 886–893. [Online]. Available: https://hal.inria.fr/inria-00548512

[14] C. Cortes and V. Vapnik, “Support vector machine,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[15] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew *et al.*, “Extreme learning machine: a new learning scheme of feedforward neural networks,” *Neural networks*, vol. 2, pp. 985–990, 2004.

[17] P. H. Swain and H. Hauska, “The decision tree classifier: Design and potential,” *IEEE Transactions on Geoscience Electronics*, vol. 15, no. 3, pp. 142–147, 1977.

[18] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.