2014 13th International Conference on Machine Learning and Applications

American Sign Language Recognition Using Leap Motion Sensor

Ching-Hua Chuan*, Eric Regina†, Caroline Guardino‡
*School of Computing, †Department of Mathematics and Statistics, ‡Exceptional, Deaf and Interpreter Education
University of North Florida
Jacksonville, FL, USA
{c.chuan, n00868514, caroline.guardino}@unf.edu

***Abstract*—In this paper, we present an American Sign Language recognition system using a compact and affordable 3D motion sensor. The palm-sized Leap Motion sensor provides a much more portable and economical solution than Cyblerglove or Microsoft kinect used in existing studies. We apply *k*-nearest neighbor and support vector machine to classify the 26 letters of the English alphabet in American Sign Language using the derived features from the sensory data. The experiment result shows that the highest average classification rate of 72.78% and 79.83% was achieved by *k*-nearest neighbor and support vector machine respectively. We also provide detailed discussions on the parameter setting in machine learning methods and accuracy of specific alphabet letters in this paper.**

Keywords—American Sign Language; 3D Leap Motion sensor; k-nearest neighbor; support vector machine; deaf education

I. INTRODUCTION

In this paper, we apply machine learning methods to American Sign Language (ASL) recognition using a 3D motion sensor. Parents of children who are born deaf have three communication options: (1) equip their child with a technological hearing device (i.e., digital hearing aid or cochlear implant) and enroll them in listening and spoken language education, (2) simultaneously learn a manual sign system (i.e., American Sign Language, Sign Exact English), with their child, or (3) equip their child with hearing technology *and* simultaneously learn a manual sign system. Because listening and spoken language education takes months or years to develop, parents wishing to immediately communicate with their child often choose options 2 or 3; learn a manual communication system, typically ASL. However, learning a manual sign system such as ASL often requires a synchronous language role model. It is our goal to develop an affordable and reliable ASL recognition system to provide instantaneous feedback to individuals learning ASL, such as parents and family members of children who are deaf, as well as the child who is deaf. Such a system has the potential to bypass the need for a language role model if it can provide sign accuracy feedback instantaneously.

Comparing with the existing ASL recognition systems that use Cyberglove [1] or Microsoft Kinect [2], we focused on a much more affordable and compact sensor from Leap Motion in this paper. We first examined the sensory data provided by the sensor's application programming interfaces (APIs). We then described the derived features relevant to ASL signs. Two machine learning methods, *k*-nearest neighbor and support vector machine, were applied to classify the 26 letters of the English alphabets in ASL signed by two faculty members one deaf and one hearing. Results presented in this paper included classification correct rates with varying parameter settings and accuracy of specific letters.

ASL is the fourth most studied foreign language in the nation [3]. Instructors and students of ASL may also benefit from using the affordable recognition system. Instructors of ASL typically have students videotape themselves for viewing and grading at a later date. Students typically have to learn ASL from a live or videotaped instruction. The ASL recognition system could potentially give students instant feedback on their signing accuracy and skills, thus minimizing the wait time for instruction. ASL instructors will also benefit by reducing the time spent viewing and grading student videos.

II. LEAP MOTION CONTROLLER AND ATTRIBUTES

A. Leap Motion Controller And Its APIs

Leap Motion controller [4] is a compact and affordable commercialized sensor for hand and finger movements in 3D space of approximately 8 cubit feet above the device. As shown in Figure 1, the sensor reports data such as position and speead of palm and fingers based on the sensor's corrdinate system. Data are transmitted to a computer via a USB connection. The frame rate of data transmission is set at 15 frames per second in this study.

The controller comes with APIs supported by the maker. Via the API, the hand and finger data can be sent to user-designed programs to use the sensor as an alternative computer-human interface. Many apps have been created using the controller, and most of them are gaming apps and apps for music-making.

Table I lists the hand and finger features obtained from the API in this study. Palm related features include normal (a unit direction vector), position (the center position of the palm), and velocity (in millimeter per second). The API also reports a float number representing the confidence of the data's accuracy. Grab strength is also a float number between zero and one, with zero indicating open hand and one for closed hand. Similar to grab strength, pinch strength shows the openness

978-1-4799-7415-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ICMLA.2014.110

between thumb and any other finger of the same hand. Sphere center and radius are calculated based on the estimated sphere placed as if the hand were holding a ball.

We also obtained features for fingers from the API. For each of the five fingers, we collected its direction (a unit direction vector) and length (in millimeter). Positions of joints between distal phalanges, intermediate phalanges, proximal phalanges, and metacarpals were also recorded as shown in Figure 1. In addition, we also collected the tip velocity of each finger.

Fig. 1. Leap motion sensor with its corrdinates and data supported by APIs.

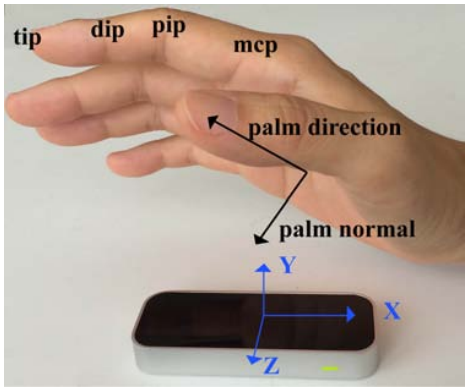


TABLE I. FEATURES OBTAINED FROM THE API.

	Palm		Fingers	
	Name	**Type**	**Name**	**Type**
Hand features[a]	Normal	vector	Direction	vector
	Position	vector	Length	in mm
	Velocity	vector in mm/sec	Tip position	vector
	Confidence	a float in [0, 1]	Tip velocity	mm/sec
	Pinch strength	a float in [0, 1]	Dip position	vector
	Grab strength	a float in [0, 1]	Pip position	vector
	Sphere center	vector	Mcp position	vector
	Sphere radius	in mm		

[a] Features were obtained using API version 2.0.2.

B. *Features Used for Machine Learning*

Some of the features provided by the API are not suitable for sign languages. For example, the absolute position of the palm is irrelevant because a sign can be placed anywhere as long as it is in the detectable range. Therefore, we needed to derive more meaningful features that can be directly obtained from the API.

Table II summarizes the features used as the attributes for machine learning in this paper. In addition to pinch and grab strength for the palm, we derived three more features: average distance, average spread, and aveage tri-spread. Assume that tip_t^n represents the tip position of finger n at frame t, n = {1, 2, 3, 4, 5} for thumb, index finger, middle finger, ring finger, and pinky respectively. The average distance is calculated as the sum of the distance between the finger tip in adjacent frames, averaged across all frames:

$$\text{averageDistance} = \frac{1}{\text{T}-1}\sum_{t=1}^{\text{T}-1}\sum_{n=1}^{5}|\text{tip}_{t+1}^{n} - \text{tip}_t^n|, \quad (1)$$

where T is the total number of frames captured for the sign and the term $|\text{tip}_{t+1}^{n} - \text{tip}_t^n|$ is the distance between figner n's tip in two adjacent frames. The average spread for the palm is estimated based on the distsance between adjacent finger tips:

$$\text{averageSpread} = \frac{1}{\text{T}}\sum_{t=1}^{\text{T}}\sum_{n=1}^{4}|\text{tip}_t^{n+1} - \text{tip}_t^n|. \quad (2)$$

The tri-spead is the triangular area between two adjacent finger tips and the midpoint of the two finger's metacarpal positions (mcp as shown in Figure 1). Assume that tip_n, tip_{n+1}, and $\text{mcp}_{n,n+1}$ are the 3D corrdinates of fingers n and n+1 and the midpoint of their metacarpal positions, the area of the triangle defined by these three points are calculated as half of the cross product of the two vectors $\overrightarrow{\text{tip}_n\text{mcp}_{n,n+1}}$ and $\overrightarrow{\text{tip}_{n+1}\text{mcp}_{n,n+1}}$:

$$\text{triArea}_{n,n+1} = \frac{1}{2}\overrightarrow{\text{tip}_n\text{mcp}_{n,n+1}} \times \overrightarrow{\text{tip}_{n+1}\text{mcp}_{n,n+1}}$$
$$= \frac{1}{2}|\overrightarrow{\text{tip}_n\text{mcp}_{n,n+1}}||\overrightarrow{\text{tip}_{n+1}\text{mcp}_{n,n+1}}|\sin(\theta), \quad (3)$$

where θ is the angle between the two vectors.

The average tri-spread is calculated by adding the triangle area of all pairs of fingers and divided by the total number of frames:

$$\text{averageTrispread} = \frac{1}{\text{T}}\sum_{t=1}^{\text{T}}\sum_{n=1}^{4}\text{triArea}_{n,n+1}. \quad (4)$$

For each finger, we derived four features including extended distance, dip-tip projection, orderX and angle. The extended distance is the maximum distance of all points of the finger (tip, dip, pip, and mcp as shown in Figure 1) from the palm center. Dip-tip projection is the projection of the dip-to-tip vector onto the palm normal vector. OrderX is the order of the figner alone the *x*-*z* plane with respect to other fingers. The feature of angle is the angel between the finger's direction vector and the *x*-*z* plane.

TABLE II. FEATURES USED FOR MACHINE LEARNING.

	Palm		Fingers	
	Name	**Type**	**Name**	**Type**
Features used	Pinch strength	a float in [0, 1]	Extended distance	in mm
	Grab strength	a float in [0, 1]	Dip-tip projection	vector
	Average distance	in mm	OrderX	vector
	Average spread	in mm	Angle	degree
	Average tri-spread	in mm		

III. DATA COLLECTION

For this pilot study, we focused on the recognition of 26 English alphabets in American Sign Language. The data were collected from two faculty members including one deaf person in deaf education. When collecting data, the signer was provided with visual feedback via Leap motion's API so that the signer can see his or her sign as being read by the sensor. When the signer was satisfied with the reading, the signer was asked to hold for five seconds while the program recorded data frames. Overall, four data sets were collected from the two signers with two sets from each individual.

IV. Machine Learning Experiments And Results

A. Experiments

The four data sets collected from the two signers naturally formed the base for four-fold cross validation in the supervised classification. Each of the four data sets was selected as the test set while the machine was trained on the other three. The experiment result was reported in terms of classification correct rate for each fold as well as the average over the four folds.

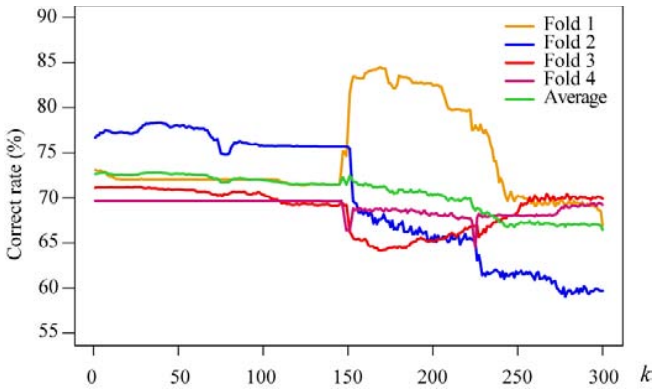
The five-second data frames were processed in two ways for the experiment. In the first setting, the data captured in each frame were considered an instance or observation. The four datasets consist of over 7900 observations. In the second setting, we applied overlapped sliding windows to generate smoothed data from the frames. The window size was measured in number of frames, ranging from 1 to 30 (roughly 2 seconds) in the experiment. As a result, the smoothed data in each window was considered an instance.

We applied *k*-nearest neighbor (*k*-NN) and support vector machine (SVM) for the alphabet recognition. We reported the classification result using *k*-NN with varying values of *k*. We also experimented with different kernel functions used in SVM.

B. Results Using k-NN

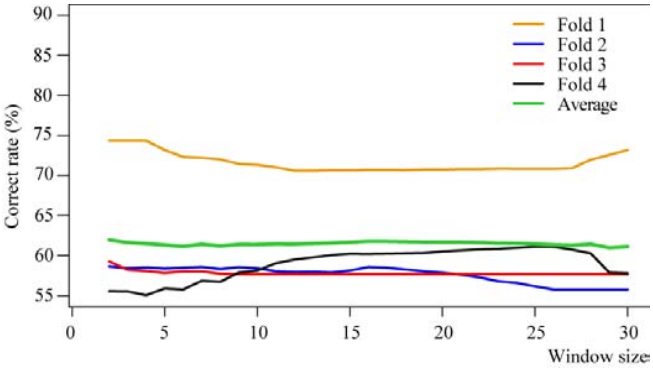
Figure 2 shows the classification results using *k*-NN for the alphabet recognition. Using four-fold cross validation, the highest average correct rate of 72.78% was achieved with $k = 7$. The overall highest performance was 84.50% on test set fold 1 with $k = 169$ while the lowest performance between $k = 1$ and $k = 149$ was 66.36% on fold 4. Generally, the correct rate remains consistent for values of *k* between 1 and 150 but with a significant decline after $k = 150$.

Fig. 2. Alphabet classification results using *k*-NN.



The classification result using *k*-NN with varying size of sliding window smoothing is shown in Figure 3. The result was generated using $k = 7$, the value that produced the higest performance in Figure 2. Comparing with the result in Figure 2, the correct rate using sliding window smooth decreases roughly 10%. The higest average correct rate of 61.95% was achived with a window size of two. However, the performance on fold 1 with sliding window smooth is slightly better than non-smoothed data.

Fig. 3. *K*-NN classification results with sliding window smoothing.



C. Results Using SVM

The classification results using SVM with eight different kernel functions are shown in Figure 4. Using four-fold cross validaiton, the best average result of 79.83% was achieved using the Gaussian radial basis function (RBF) kernel. The overall higest correct rate was 83.39% on fold 2 and the lowest was 74.06% on fold 4 using the Gaussian RBF kernel.

Figure 5 shows the variations in the result using SVM/RBF with varying sliding window sizes. Comparing with the result shown in Figure 4, the performance of using sliding window smooth also decreases around 10% in SVM. However, the average correct rate slightly increases as the window size grows.

Fig. 4. Classification results using SVM with different kernel functions.

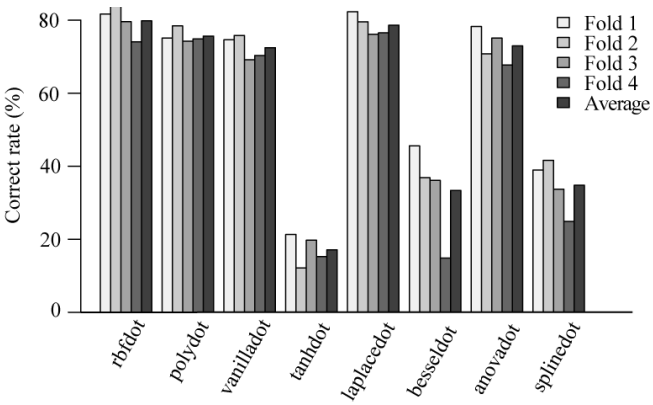
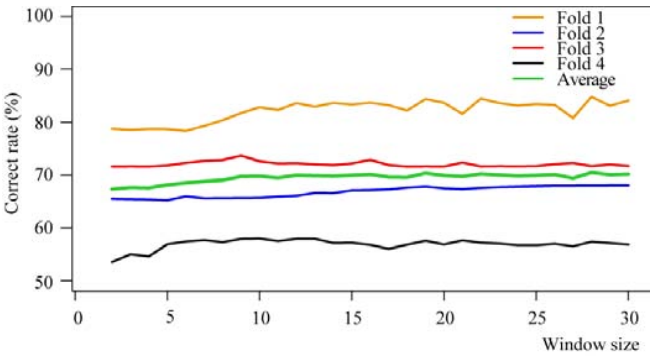


Fig. 5. SVM classification results with sliding window smoothing.



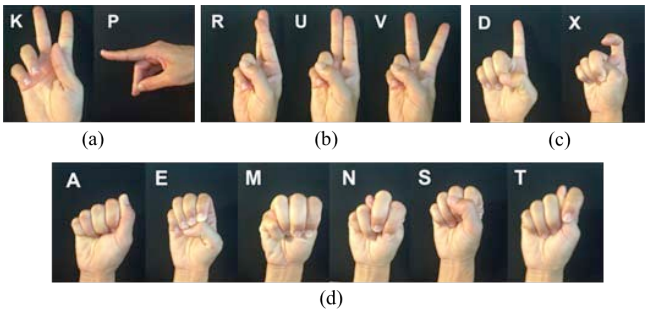
D. Accuracy of Specific Letters

Table III shows the accuracy of classification results for the letters that were not perfectly recognized with classification accuracy of 100% by *k*-NN and SVM. The table lists the letters that were misclassified with a ratio indicating how often such misclassification occurred. Letters excluded from the table were perfectly recognized. Generally, the misclassification can be explained by the similarity between the letters as shown in the groups in Figure 6. For example, the letters K and P in Figure 6 (a) are mere opposites of one another: they share the same handshape but with the palm facing different directions. The letters in Figure 6 (b) all involve the index finger extended with a closed thumb. In Figure 6 (c), the difference between letter D and X is the shape of the index finger. In Figure 6 (d), all letters have the index finger, middle finger, ring finger, and pinky closed but with a closed thumb placed in different places.

TABLE III. ACCURACY OF SPECIFIC LETTERS CLASSIFIED USING *K*-NN AND SVM.

***k*-NN**		**SVM**	
letter	**classified as**	**letter**	**classified as**
E	N (1)	A	A (0.86), S (0.14)
K	P (0.32), R (0.3), V (0.38)	E	N (1)
M	N (1)	K	P (1)
N	M (0.19), N (0.31), T (0.5)	M	M (0.82), N (0.04), S (0.14)
O	N (0.09), O (0.91)	N	M (0.5), N (0.5)
R	R (0.71), U (0.29)	O	M (0.04), N (0.23), O(0.68), T(0.05)
T	M (1)	T	M (0.01), S (0.99)
X	D (0.77), X (0.23)		

Fig. 6. Signs of the letters not perfectly classified by *k*-NN and SVM.



V. DISCUSSIONS AND FUTURE WORK

A major cause of the inaccuracy in the experiment result is the mislabled data from Leap Motion APIs. During the process of data collection, we observed many instances in which the hand and fingers in the visual feedback did not mimic the signer's hand. Since the raw data from the sensor are not publically accessible, we plan to combine the sensor with a web cam, which provides a separate data source for hand gestures. We are currently developing a web application with a back-end database to collect more data from multiple signers.

The Leap Motion sensor in combination with a webcam, has the potential to change the method in which individuals learn and teach ASL. Families with children who are deaf, as well as the children who are deaf themselves, could learn ASL immediately after learning their child has a hearing loss in the hospital. Instructors of ASL and interpreting courses could incorporate the use of the Leap Motion sensor and webcam in their courses to help students achieve signing benchmarks before taking final exams. Students of ASL and interpreting can use the technology to self-assess their signing skills. In order for these potentials to be obtained, the Leap Motion sensor must be able to differentiate between various handshapes, motions, and facial expressions. At times, these variations are subtle and only transparent to the trained eye. While the Leap Motion sensor is still in the initial stages of development, the potential for this technology to impact the field of Deaf Education and Interpreting is great.

ACKNOWLEDGMENT

We would like to thank the signers, Dr. Janice Humphrey and Mr. Jonathan Antal, for their time and effort.

REFERENCES

[1] H. Wang, M. C. Leu, and C. Oz, "American sign language recognition using multi-dimensional hidden Markov models," Journal of Information Science and Engineering, vol. 22, no. 5, pp. 1109–1123, 2006.

[2] S. Lang, M. Block, and R. Rojas, "Sign language recognition using kinect," Artificial Intelligence and Soft Computing, pp. 394–402, Springer Berline Heidelberg, 2012.

[3] L. Tamar, "Colleges see 16% increase in study of sign language," The New York Times, http://www.nytimes.com/2010/12/08/education/08language.html?_r=0 (last access: August 2014)

[4] Leap Motion, http://www.leapmotion.com (last access: August 2014)