Detection and Recognition of Sign Language Protocol using Motion Sensing Device

Rita Tse
Computing Programme
Macao Polytechnic Institute
Macao, China
ritatse@ipm.edu.mo

Zachary Chui
MPI-QMUL Information Systems Research Centre
Macao Polytechnic Institute
Macao, China
zacharychui@gmail.com

AoXuan Li
Computing Programme
Macao Polytechnic Institute
Macao, China
P1308512@ipm.edu.mo

Marcus Im
Computing Programme
Macao Polytechnic Institute
Macao, China
marcusim@ipm.edu.mo

***Abstract*—This paper explores the possibility of implementing a gesture/motion detection and recognition system to recognize the American Sign Language (ASL) protocol for communications and control. Gestures are captured using a Leap Motion sensing device with recognition based on a Support Vector Regression algorithm. There is a high correlation between the measured and predicted values in those samples; it is able to recognize those sample data with almost 100% accuracy. Development of network connectivity and establishment of communication protocol enables "smart" objects to collect and exchange data. With encouraging results given the limitations of the hardware, this work can be used to collect and exchange data with devices, sensors and information nodes and provide a new solution for Internet of Things in the future.**

Keywords—Leap motion sensing device; American Sign Language detection and recognition; support vector regression; Internet of Things

I. INTRODUCTION

Smart City and Internet of Things (IoT) research places emphasis on the development of communication to connect devices, sensors and information nodes. Establishment of network connectivity and protocol enables objects to collect and exchange data. In instances where human nonverbal interaction is required, a keyboard, or a touch device is most common for entering information. Recent research on voice recognition technology has made voice-controlled devices obtainable in the very near future. However, there are instances that make voice-control less attractive: "noisy" environments may create detection and recognition errors that make voice-controlled devices impractical.

An alternative to using voice-control is to implement a gesture or motion-control interface. Once an "action" is detected and recognized, other device actions may follow. As previously mentioned, there may be advantages using a gesture/motion control interface: 1) the environment is "noisy" or impractical to have voice-control; 2) the physical position of the user has to be in a specific area (such as user's extremities) for safety reasons.

Many established gesture/motion communication protocols come to mind. Sign language is one such protocol. Sign language, like any language, has a long history and many forms. Modern sign languages developed as a protocol to help the hearing-disabled and other non-verbal speaking persons to communicate. They are generally visual-based languages, with gestures and motions to represent alphabet letters and numerical digits. This paper investigates the possibility to implement a gesture/motion detection and recognition system to recognize the American Sign Language (ASL) protocol for communications and control.

II. RELATED WORK

Sign language recognition is composed mainly of two parts: the first part is the method to capture sign gestures while the second part is using a reliable and accurate recognition algorithm for the captured gestures. The technologies used to capture motion-based information or data, such as sign language, in general can be classified in two major groups: 1) sensor-based; and 2) image-based.

For sensor-based technology, many researchers have used sensor gloves in their studies. Bukhari et al. [1] assembled a system for American Sign Language translation with a sensor glove, then using Principal Component Analysis to "train" a computer to recognize various gestures and classifying into an alphabet system in real time. The glove was found to have an accuracy of 92%. Gaikwad and Bairagi [2] also used sensor gloves to build a recognition system for Indian Sign Language. The glove was equipped with flex sensors, which produced analog output values converted into digital values using an ADC converter, along with accelerometer sensors to measure the motion of hand. Hand gestures were processed using a LPC2148 microcontroller, with the digital signal sent to a cell phone via a Bluetooth module. The phone would announce the corresponding alphabet of the recognized gesture. Lokhande, Prajapati and Pensar [3] developed an embedded system for

sign language recognition using a glove with flex sensors and a 3-axis accelerometer. The motion/gesture was converted to digital signals using comparator circuits and analog-to-digital converters (ADC). The results were displayed using a liquid crystal display (LCD) and a speaker. Pati et al. [4] built an American Sign Language detection system in a similar approach. They monitored hand movements and fed data into a microcontroller for comparison and display.

For imaged-based technology, some researchers used the Leap Motion sensing device in their studies. Mapari and Kharat [5] proposed a method using the Leap Motion sensing device to convert a hand gesture to digital data. The data contained 48 features including position, distance and angle values. A multilayer perceptron (MLP), a type of feed forward artificial neural network, was used to classify the data. This technique classified 32 signs with near 90% accuracy in cross validation. Simos and Nikolaids [6] built a similar system for Greek Sign Language alphabet recognition. They used support vector machine (SVM) classification with radial basis function (RBF) kernel to achieve over 99% classification accuracy in cross validation. However, they found that the Leap Motion sensing device has limitations. Potter, Araullo and Carter [7] analyzed the Leap Motion sensing device and concluded that it can be a potential tool for recognizing Australian Sign Language, but further development of the Leap Motion application program interface (API) is required.

III. Experimentation

This work aims to build a system to use the American Sign Language (ASL) as a protocol to communicate to devices, sensors and information nodes.

A. System Architecture

This system includes a Leap Motion sensing device and a Raspberry Pi computer, with a Machine Learning algorithm to learn and predict data. Machine learning is a popular algorithm for object classification and data prediction.

The Leap Motion sensing device is a motion detector. Compared with other sensors, it employs proprietary algorithms to calculate near 3-dimensional output of detected motion. For example, if a hand is detected, it can display the position of each bone of the hand and return its corresponding coordinates in three dimensions. It also provides an API for use with many popular programming languages. However, it has limitations. In some occasions if a finger is bent, it cannot be distinguished from the hand. The thumb often is not recognized even if the hand is extended. Also, if fingers are positioned close to each other, they may not be distinguishable. Positioning the palm perpendicular to the camera can make gestures undetectable.

These limitations are realized through the experimentation with the Leap Motion sensing device and using its API. These observations affected this design work in the following ways:

1) Recognizing a static gesture
2) Using Leap Motion coordinate data
3) Calculation of the angle of the joints in the hand

The Raspberry Pi 3 model B+ was used as the component responsible for result prediction and display in this work. The Raspberry Pi computer is a low cost, credit card sized computer that plugs into a computer's monitor or TV, and can use a standard keyboard and mouse. As a microprocessor development board, the Raspberry Pi is powerful enough to handle complex calculations. Furthermore, Raspberry Pi has a built-in Wi-Fi module to connect with other devices. For example, it can send signals to communicate with smart home devices.

B. Detection

The first step for machine learning is to build a dataset by pre-processing data then sampling, as shown in Fig. 1.

Gestures are visual movements for humans. However, it needs to convert to digital data for computing programs to process, which means gestures must be sampled. It is easy to get locations of a hand's joints by using the API provided by the Leap Motion sensing device. Each hand has 19 bones. The coordinates of the bone tips can locate each bone specifically, which means 28 three-dimensional coordinates can represent a gesture fully. That is, gesture captured by the Leap Motion sensor can be directly converted to data. However, the data obtained using this method is sensitive to the shape of the user's hand and location. If users, who have different hand shapes, place their hand in different locations relative to the sensing device, data received from the Leap Motion sensing device may be significantly different for the same gesture. To build a workable dataset, the tester needs to sample gestures many times from various users with different hand locations. Each sample will have almost 90 features. As a result, this method needs to generate significant amounts of data. To use a machine learning algorithm, it needs to reduce the number of random variables (features) under consideration, and to obtain a set of principal variables.

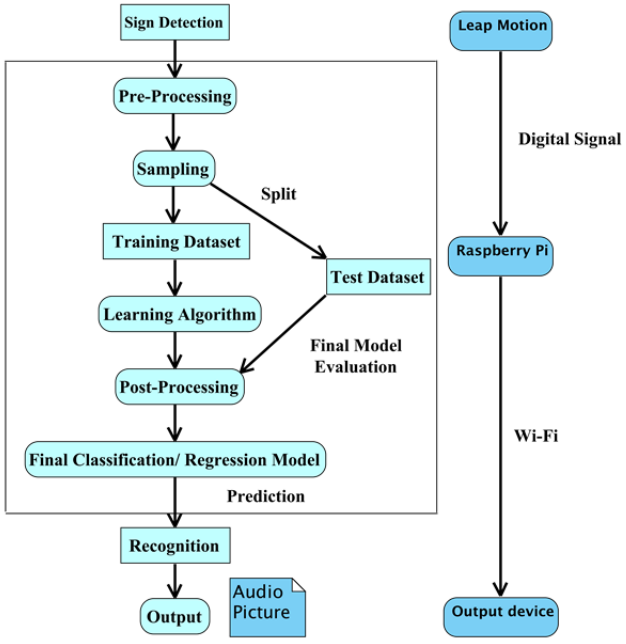

Fig. 1. Overall system structure.

Regardless of the differences of users' hand, if they display the same gesture, the angle between two bones, called 'joint', should be similar. Every bone can be represented by its direction or orientation, regardless of size or shape. Given two vectors, using (1), the degree of each joint can be calculated:

$$\frac{\overrightarrow{ab} \times \overrightarrow{ab}}{|\overrightarrow{ab} \times \overrightarrow{ab}|} = \angle abc \quad (1)$$

However, the complexity of ASL limits this method's workability. For example, the thumb can move in a very limited range. For two varying gestures representing '4' or '5', this method cannot distinguish them successfully. The reason is the processed data "losses" the information of spatial relationship between the two fingers. In this work, six additional appended variables (features) are proposed to solve this problem: five features to calculate the angle between each distal phalanx and the hand, and one feature to indicate the number of pointed fingers on the hand. The direction of each distal phalanx and the hand, and the number of fingers pointed, can be calculated from the Leap Motion sensing device data. These six additional features increase the recognition of different gestures.

For this work, each gesture is converted into a 20-dimensional vector: 14 angles of joints, 5 angles between distal phalanxes and the hand, and the number of pointed fingers. This method requires a smaller dataset, and is a more general approach compared to other designs using measurement of distance and directions of the fingers in their research.

C. Regression Analysis

As illustrated in Fig. 1, the second step of the machine learning process is to train the dataset and to generate an output to the regression model for prediction in the recognition module. The dataset built for this work has the following characteristics. For ease of use, the dataset has no feature name. Column A is the label of the dataset. Each value represents a character mentioned in Table 1. Data in column B to column U are angles of each joint: B to E for the thumb, F to I for the index finger, J to M for the middle finger, N to Q for the ring finger and R to U for the pinky. Column V shows the number of extended finger. This dataset contains 20,000 samples for 10 gestures: 'a', 'b', 'c', 'd', 'e', 'l', 'v', 'y', '3', '5'.

Considering the many gestures of ASL, regression analysis using Redial basis function (RBF) kernel was chosen. Using the correct algorithm is critical for machine learning. RBF is a popular Gaussian kernel function used in various learning algorithms. A support vector machine (SVM) employing RBF is commonplace. It is the same Gaussian kernel used in the video process. If there are two samples, x and x', represented as feature vectors, the RBF kernel of x and x' is defined as in (2).

$$K(x, x') = \exp\left(-\frac{||x - x'||_2^2}{2\sigma^2}\right) \quad (2)$$

TABLE I. ASSIGNED VALUE TABLE FOR CHARACTER TABLE II TABLE TYPE STYLES

Character	Assigned Value	Character	Assigned Value
a	1	s	19
b	2	t	20
c	3	u	21
d	4	v	22
e	5	w	23
f	6	x	24
g	7	y	25
h	8	z	26
i	9	0	27
j	10	1	28
k	11	2	29
l	12	3	30
m	13	4	31
n	14	5	32
o	15	6	33
p	16	7	34
q	17	8	35
r	18	9	36

Because the alphabet of ASL is well labelled with each alphabet being a category, regression analysis with supervised learning process is a suitable algorithm for this work [8]. In order to use this regression algorithm, each alphabet is assigned a number, e.g. 'a' is assigned to 1, 'b' is assigned to 2, etc. This converts the process of predicting a category to predicting a number. Since machine learning creates a heavy loading of the CPU during classification, this process helps the CPU to calculate more efficiently. Table 1 shows the assigned value for every character.

Following regression analysis, the size of the dataset needs to be determined. It is reasonable to estimate that less than 100 thousand samples will be enough for machine learning in this work. Another question is: which are the most important features for the machine learning algorithm. Consider that each ASL alphabet has unique gestures, and all features are of equal importance, SVR classification with radial basis function kernel was employed for this work. This algorithm provides an effective and accurate method for gesture detection.

A general problem for machine learning is that there is always noise in the data. The cause may be due to: spurious readings, measurement errors, and background noise. In this work, measurement error occurs frequently due to the limitations of the Leap Motion sensing device. The data classifier in the regression model needs to be tested to indicate if it can predict unseen data successfully. A popular approach is to use cross validation. Some variables are tested using this algorithm. Cross validation in this work are shown in Fig. 2, with results close to 100%.

```
Train the dataset

Result of cross-validation

[0.991453, 0.991883, 0.997761, 0.998717, 0.999844, 0.997424, 0.997199, 0.997086,
 0.996484, 0.993035]

Test data is [0 0.767866671     0.84500128      -0.682040095    0.947621703     0
.996863008      0.997722745     -0.991320074    0.976764143     0.997374356     0
.998098254      -0.990855098    0.990595639     0.997041106     0.997957349     -
0.986448288     0.973952711     0.995342672     0.996807754     -0.993659198    4
]

The result of test data is B
```

Fig. 2. Result from cross validation overall system structure.

D. Recognition

The third step of the machine learning algorithm is prediction. In reality, there will be a difference between predicted value and expected value, e.g. the gesture of 'a' will not return "l", but a real number around "1". If the regression model is workable, the predicted value will not be too far from the expected value. A reasonable design is to round the result into an integer, e.g. if the result is "0.7", it will be round to "1" which means the gesture is an 'a'.

Fig. 3 depicts the system architecture. Machine learning program loads the dataset sampled from the user's gesture and predicts the dataset. After getting the results, this recognition module will send a socket to a web server. This socket contains the result of the prediction and some message such as 'waiting', 'storing to dataset' or 'predicting'.

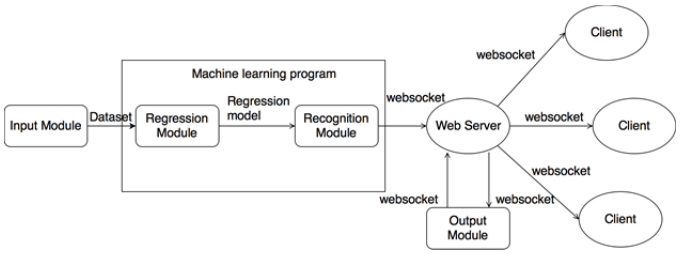

Fig. 3. System architecture.

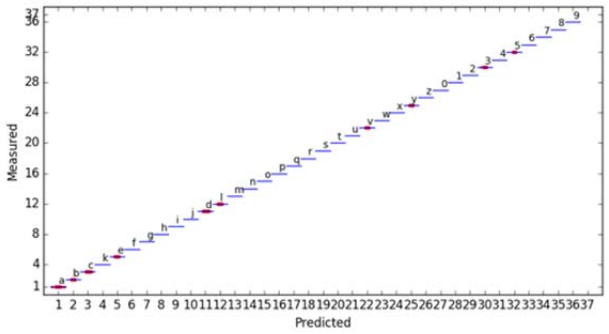

Fig. 4. Comparison between predicted value and measured value.

After the output module receives a response message from the web server, it closes the connection immediately. The web server will send the web socket to all clients (user browsers) visiting the GUI website. The browser then executes a JavaScript program to receive this web socket and display the relative information on the web page. This approach allows the machine learning program and the browser to communicate asynchronously with each other. The browser connects with the server and sets up a handshake connection initially. Since the web socket is a full-duplex communication channel over a single TCP connection, the browser and server can communicate with each other until the browser closes the connection.

This design separates the recognition and the output module, but connects them together through a socket. As a result, changes made to one module will not affect other modules, which assures that this system is portable for other devices. In the future, the output module can communicate directly with other data modules such as smart home devices.

E. Results

The results of this work are shown below. Some important scores to indicate the quality of predictions are summarized in Table 2. According to both the regression score functions, the accuracy rate is 99.99% for the 10 sample categories.

TABLE II. STATISTICS SCORE

Function Name	**Score**
Explained variance regression score function	0.999967182448
R^2 (coefficient of determination) regression score function	0.999966798403
Mean absolute error regression loss	0.0479073872591
Mean squared error regression loss	0.0034014307872
Median absolute error regression loss	0.0435534210731

Employing cross validation, the dataset is separated in ten groups, with one set for testing, and another set for training. Using a different testing approach, this work now can predict unseen data. The more accurate the prediction, the closer the error regression loss is to zero. According to Table 2, the prediction is highly accurate. Correspondingly, the higher the accuracy score, the better the program. From a statistical point, this work successfully recognizes eight alphabet and two number gestures. Fig. 4 compares the result between predicted value and measured value for character 'a', 'b', 'c', 'd', 'e', 'l', 'v', 'y', '3', '5'. There is a high correlation between the predicted and the measured values in those samples.

IV. DISCUSSIONS

This work provides a workable and accurate approach for ASL detection and recognition for communicating with information nodes such as smart home devices. The significance of this solution, unlike other works based on image recognition, is this work uses a data based approach. A dataset is built with 20,000 samples for ten characters for training in a machine learning algorithm. It recognizes ten ASL alphabet and number gestures with almost 100% accuracy.

Measurement errors, especially incorrect gestures captured by the Leap Motion sensing device, caused most of the losses. The Leap Motion sensing device is not accurate enough to distinguish between some common but similar ASL gestures, e.g. m and n, p and q, s and t, u and v, d and x. The limitations of the Leap Motion sensing device make it impossible to recognize all ASL alphabet gestures. Those sample characters 'a', 'b', 'c', 'd', 'e', 'l', 'v', 'y', '3', '5', and other recognizable

ASL gestures can be used to communicate with other devices and information nodes.

Selection of the machine learning model is an issue that influences the accuracy of this work. While using a supervised learning algorithm is the correct approach, the selection between regression and classification is still a critical choice. In terms of efficiency, regression is a quicker way to detect gesture with less data. Regression analysis is often used for prediction with continuous data, such as "price". It may be used for small categories of data as in this paper. However, with increasing number of categories, more problems need to be solved, e.g. what value to be assigned to different characters. In such cases, classification is a more suitable approach for category prediction. Multiclass neural network, an example of a classification algorithm, is a more accurate model. However, it needs a longer training period than a regression approach. With simple categories, a regression model is suitable, but classification should be employed if there are many categories.

V. Conclusion

Sign language is a communication protocol to help the hearing-disabled and other non-verbal speaking persons. This paper explores the possibility of implementing a gesture/motion detection and recognition system to recognize American Sign Language (ASL) protocol for communications and control.

The alphabet of ASL is well labelled. A Support Vector Regression, with RBF kernel is used in this work. There was a high correlation between the measured and predicted values using the sample data, with almost 100% accuracy. Based on this work, it is able to create a complex instruction set to communicate with devices, sensors and information nodes in the future.

References

[1] J. Bukhari et al. American sign language translation through sensory glove; signspeak. International Journal of u-and e-Service, Science and Technology 8(1), pp. 131-142. 2015.

[2] P. B. Gaikwad and V. K. Bairagi. Hand gesture recognition for dumb people using indian sign language International Journal of Advanced Research in Computer Science and Software Engineering 4(12), 2014.

[3] P. Lokhande, R. Prajapati and S. Pansare. Data gloves for sign language recognition system. Presented at International Journal of Computer Applications (0975–8887) National Conference on Emerging Trends in Advanced Communication Technologies (NCETACT-2015). 2015, .

[4] K. Patil et al. American sign language detection. International Journal of Scientific and Research Publications 4(11), 2014.

[5] R. B. Mapari and G. Kharat. American static signs recognition using leap motion sensor. Presented at Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies. 2016, .

[6] M. Simos and N. Nikolaidis. Greek sign language alphabet recognition using the leap motion device. Presented at Proceedings of the 9th Hellenic Conference on Artificial Intelligence. 2016, .

[7] L. E. Potter, J. Araullo and L. Carter. The leap motion controller: A view on sign language. Presented at Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration. 2013, .

[8] "Scikit-learn: machine learning in Python — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit-learn.org/stable/. [Accessed: 19- Mar- 2017].