Recognition of Sign Language Hand Shape Primitives With Leap Motion

Burçak Demircioğlu, Güllü Bülbül, Hatice Köse
Computer Engineering Department, Istanbul Technical University
Istanbul, Turkey
{demircioglubu, bulbulg, hatice.kose}@itu.edu.tr

Abstract

In this study, a rule based heuristic method is proposed to recognize the primitive hand shapes of Turkish Sign Language (TID) which are sensed by a *Leap Motion* device. The hand shape data set was also tested with selected machine learning method (*Random Forest*), and the results of two approaches were compared. The proposed system required less data than the machine learning method, and its success rate was higher.

Keywords: Sign Language, Sign Recognition, Leap Motion

1. Introduction

Sign recognition is vital for an efficient and robust human-machine interaction for hearing impaired people. To be able to use such a system in as many contexts as possible, the technology should be as small and as adaptable as possible. To achieve this goal, the hardware chosen for this study is the *Leap Motion* sensor[1], which is smaller than similar sensors, and can be integrated to any device with USB port, such as robots, computers or smart phones. This study is a part of an ongoing project on the recognition and generation of Turkish Sign Language (TID) gestural vocabulary by computer aided methods. For this purpose, depth sensors such as *Kinect* and *Leap Motion* are used to recognize a selected corpus from TID and this corpus is also generated via humanoid robots and virtual avatars, therefore a two way nonverbal communication on TID is achieved. Due to the modular structure of the project, it can be extended to any similarly structured sign language with minimum effort. The main purpose is to use this multi-modal platform to teach children sign language using interactive games (Köse et al., 2014), (Köse et al., 2015a), (Köse et al., 2015b).

This project consists of two phases: A real-time solution, and an offline solution for hand sign recognition. The real-time solution is based on the heuristic models we developed for every sign on *Leap Motion* based system after analyzing every hand sign for Turkish Sign Language. The data gathered to test this system is then used to train and test selected offline machine learning technique via the *Weka*[2] system to verify the success of the system. The second stage is offline. After all these stages, the same data is given to *Weka*, and offline working code, then the two results of these processes are compared.

There are some previous studies on sign recognition for Turkish Sign Language as well as for other sign languages. However, they differ from this project in their approaches and the technology used. For example, a system aiming at improving learning speed of children learning Turkish Sign language which uses machine learning techniques is presented in (Haberdar and Albayrak, 2005). Keskin et al. (2013) studied recognition of real-time data from selected TID signs (digits) by using *Kinect* technology. The key advantage of our approach is that it requires fewer sampling and therefore less preparation time. So, algorithm coding is chosen for this project which makes the project more efficient than others. To the best of our knowledge, this is the first study using a *Leap Motion* sensor for hand shape inventory recognition in Turkish Sign Language. This alphabet consists of 32 static hand shapes which are used to produce all signs in Turkish Sign Language (Kubuş, 2008), and different from the finger alphabet (for finger spelling). Finger alphabet greatly differs from this alphabet structurally, therefore the recognition systems for finger alphabet can not be employed for this study.

2. Theoretical Information

2.1. Sample Sign Data Set

The main purpose of the project is the recognition of the static hand shape inventory (not the alphabet) of the Turkish Sign Language; because static hand shapes are distinctively and categorically highest phonological features of sign languages as stated in (Kubuş, 2008). The static hand shape inventory of TID which is used in this paper is based on the same work which is the main linguistic source on this language feature. This reference work indicates that TID is a rich language in terms of morphology, phonology and classification. TID has significantly different linguistic properties than other sign languages, as in the case of spoken languages. Although several handshapes in TID are similar to other sign languages, most of them are different. For example; some ASL (8,E,K,M,N,T) and Taiwanese Sign Language (middle finger, ring finger) handshapes are absent in Turkish Sign Language. Therefore Turkish native signers can not differentiate some similar signs of ASL which indicates that TID has a unique handshape inventory (Kubuş, 2008).

Some of the handshapes in the sample data set are not included in this study as the *Leap Motion* sensor is inadequate to sense overlapping fingers. Also, the sensor is not precise enough to differentiate similar signs such as *ASL C-handshape* and *Hooked Flat Extended*. 18 signs from the sample data set are included in this project: 5 Handshape, V&2 Handshape, L Handshape, O Handshape, C

[1]URL: https://developer.leapmotion.com/. Last access date:16/01/16.

[2]URL: http://www.cs.waikato.ac.nz/ml/weka/index.html. Last access date:27/03/16.

Handshape, ASL A Bar, ASL 8 Handshape, ASL I&H Handshape, ASL I Handshape, ASL Y Handshape, ASL 3 Handshape, 4 Claw Handshape, 8 Handshape, 9 Handshape, ASL A Handshape, Baby O Handshape, Open 8 Handshape, and ASL Q Handshape (figure 1).

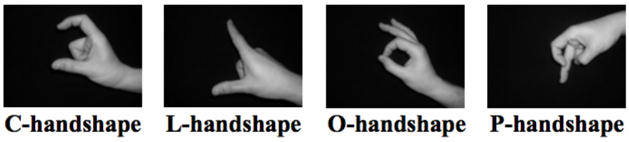

Figure 1: Sample data set from (Kubuş, 2008)

2.2. Related Works

There are several studies on the recognition of Sign Languages using depth sensors such as *Leap Motion* and *Kinect* which are summarized as follows:
The first study (Chuan et al., 2014) is based on the recognition of the 26 static single-hand finger alphabet signs of the ASL using *Leap Motion*. This study uses two machine learning methods which are *k-nearest Neighbor(NN)* and *Support Vector Machine(SVM)* based on the samples from human subjects with *Leap Motion*. The average success rate of this study is measured as k-NN=72.78% for four-fold cross validation with k=7 and as SVM=79.83% for four-fold cross validation using Gaussian radial basis function (RBF) kernel.
The second study (Mohandes et al., 2014) presents a project for the recognition of the 28 static single-hand finger alphabet signs in Arabic Sign Language using *Leap Motion*. Two machine learning methods, *Naïve Based Classifier(NBC)* and *Multilayer Perceptron (MLP)*, are used for the hand shapes classification. The success rate of these methods are NBC=98.3% with 76/2800 mis-classified samples and MLP=99.1% with 26/2800 mis-classified samples.
The last project (Keskin et al., 2013) is based on the hand gesture recognition using *Kinect*. Recognition of the ASL signs representing the 10 digits is presented in the project. The average success rates of real-time test achieved by the cross-validation tests is ANN (Artificial Neural Networks)=98.81% and SVM(Support Vector Machine)=99.90%. The average success rates for a synthetic data set test achieved by the cross-validation tests is ANN=99.89% and SVM=99.96%.

3. Analysis And Modeling

Leap Motion has two infrared sensors which are both directed towards the y-axis from the *Leap Motion* (axes of leap motion can be seen in Figure 2). These sensors detect the world as a funnel which extends as sensing reaches outward from the leap motion by maintaining end-to-end 150 degrees gap in three dimensions. The height of the sensing extends from 25 to 600 millimeters (Leap Motion, 2016). Because the sensing is done by the infrared sensors, when a finger is in front of another finger in the y axis, the finger positioned behind cannot be seen by the sensor and this situation lowers the precision of the sensing process. Also for precise sensing, the view must be clear in the sensing funnel and the light must be controlled to keep the high contrast stable, which is vital for infrared sensing.
139 different attributes, which are received from the sensor at a speed of approximately 127fps, are used to recognize the signs. By using these attributes, necessary angles and the distances between joints and bones (when angles are not sufficient enough) are calculated and the signs' limit values are defined. This solution is explained in details in the section 4.2.
In this study, the distances between the data points are not preferably used; because the distances may vary from person to person and from hand to hand. For example, distance between palm center to a finger tip may be considerably smaller for a child's hand compared to an adult's hand. For this reason, the angles rather than the distances are used for the calculations (when angles are sufficient enough).

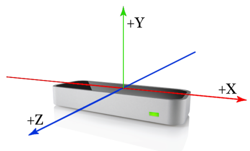

Figure 2: Axes of Leap Motion (Leap Motion, 2016)

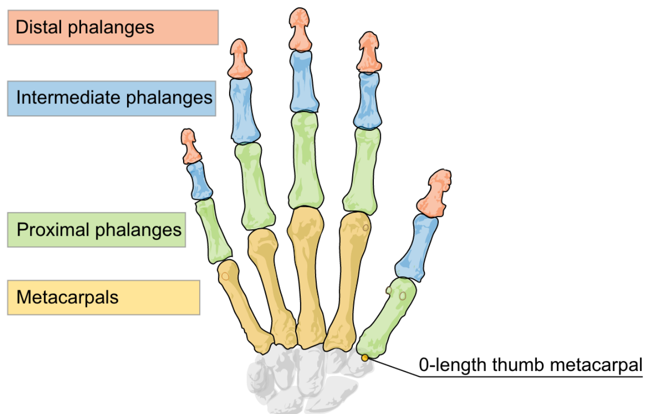

Figure 3: Bone types in hand (Villarreal, 2007)

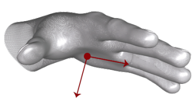

Figure 4: Palm Direction and Normal Vectors (Leap Motion, 2016)

4. Design, Implementation And Test

This project consists of two phases: sign language recognition with real-time data and offline recognition:

4.1. Real Time Recognition

The angles and distances between the defined attributes are calculated to be used in the recognition phase. The

distances are in use when the angles are not sufficient for recognizing the chosen sign. For example, to recognize the ”L” sign; the angle between the thumb’s tip direction and index finger’s tip direction is calculated, which is sufficient to recognize this sign. Here the directions are vectors in three dimensions. The sample pseudo code of the heuristic model of a sign is presented in Figure 5, all of the recognition rules can be seen in the Appendix section.
As a final step of the recognition, a stabilization phase is required for the output to overcome the noise in the real-time data. A *Sliding Window* based approach is used to discard the noise in the data. The window size is chosen as five frames for the method, and the most frequent data item in the window is selected as the output of the window, at each iteration.
In the test phase, real-time data from both hands of the three human subjects (two women and one man) are used as in the Figure 6 and Figure 7.

```
For 8-handshape:
    IF Extended Finger Count is 0
        IF Palm Direction is "Down"
        AND Distance Btw Ring and Pinky Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Thumb and Ring Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Thumb and Middle Finger Tip Positions>DISERRORMARGIN
        AND Angle Btw Index Finger's Proximal Bone's Direction and Hand's Direction Vectors > PI / 2
        AND Angle Btw Index Finger's Proximal and Intermediate Bones' Direction Vectors > PI / 2 - ERRORMARGIN
        AND Angle Btw Middle Finger's Proximal and Intermediate Bones' Direction Vectors > PI / 2 - ERRORMARGIN
            Sign is "8-handshape"
```

Figure 5: A sample from specified rules for a sign’s recognition

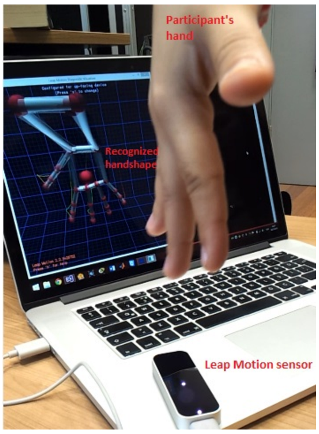

Figure 6: A Screenshot from the experiment 1

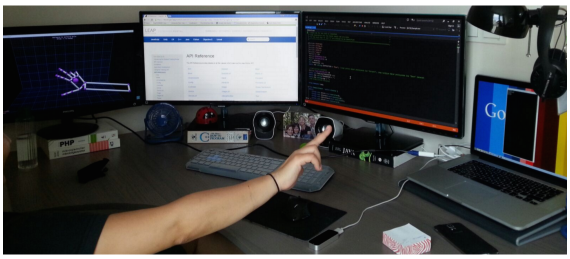

Figure 7: A Screenshot from the experiment 1

4.2. Offline Recognition

For the offline recognition phase, the data obtained from the real time recognition phase is recorded into the text files in the *arff* format. 30 frames are recorded for every trial. For every sign, 10 trials are recorded. The first five of these trials are used as training data and the remaining five are used as test data, and saved as separate *arff* files. In both files there are

$$30 * \frac{10}{2} = 150 \tag{1}$$

frames for every sign.
For the offline phase, the *Leap Motion API* cannot be used; due to the permission restrictions in the API, the objects’ data is not accessible, therefore it cannot be fed with the offline data. To overcome this problem, the necessary objects are listed, modeled, and created again within header files to be used in the offline phase’s code. The new objects are created with the same names as in the API to make the conversion simpler and error free. After the creation of new object structure, all of the data is read from the previously created *arff* files frame by frame and saved into the object variables in the code.
To adapt the code to the new object structure, some variables, and functions are modified, as well. In the *Leap Motion* API, the output data are obtained from the relevant functions defined in the objects. For example, direction vectors are obtained from the *direction()* function from objects such as *finger*, and the extended finger list is obtained from the hand object with fingers and extended functions such as *hand.fingers().extended()*. With the new object structure which is created for the offline use, all of the variables are directly accessible. Therefore the equivalents of these function calls in the offline code are; *finger.direction* and *hand.extendedFingerList*. In the same way, if just the x direction of direction vector is required, it can be obtained directly by the variable *finger.direction.x*.
Finally, a sample machine learning method (*Random Forest*) is implemeted via the *Weka* system and tested using the offline data. First of all, the method is trained using the *dataTrain.arff* file by 10-fold cross validation method. The trained method is then tested with the offline test data (*dataTest.arff*).

5. Experimental Results

5.1. Heuristic Method Results

The real time recognition phase results are summarized in the Figures 8 to 13. The success rates of the outputs are almost 100%, and the stabilization code avoids the flickers in the data, which makes the output more reliable and efficient in daily use of the program.
To compare the offline phase results with the machine learning method results, the stabilization part of the real time code is omitted and all frames are recognized one by one. Therefore if there are 150 frames per sign, there are exactly 150 recognition results for that particular sign. The sign specific results can be seen at the confusion matrix displayed in the Table 1. As seen in the table, all of the signs are recognized with 100% success rate, except the *Baby O Handshape* with 84% and *Open 8 Handshape* with 98.66%

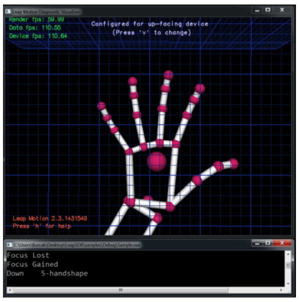

Figure 8:
5 Handshape

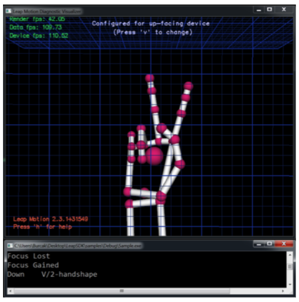

Figure 9:
V & 2 Handshape

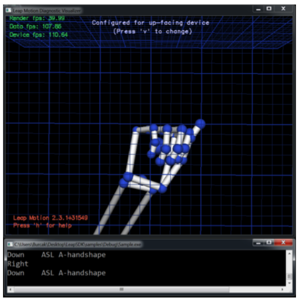

Figure 10:
ASL A Handshape

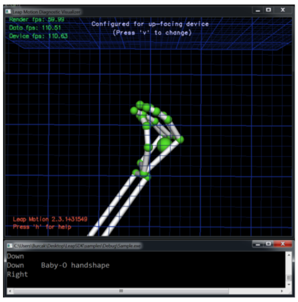

Figure 11:
Baby O Handshape

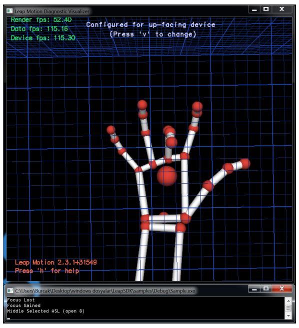

Figure 12:
Open 8 Handshape

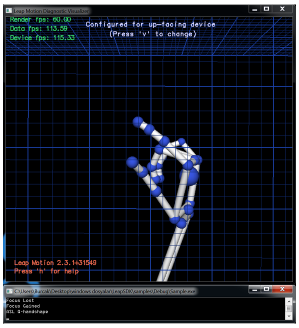

Figure 13:
ASL Q Handshape

success rate. Some of the *Baby O Handshape* sign samples are recognized as *ASL A Handshape* sign because these signs are very similar in the calculation point of view; both signs have no extended finger which is employed in the first rule of their recognition models. As the next step for the *ASL A Handshape* sign's recognition, the distance between the thumb's distal interphalangeal joint, and the index finger's proximal interphalangeal joint is calculated. If this distance is less than a threshold value then the system decides that the sign is *ASL A Handshape*. As the second step for *Baby O Handshape*'s recognition, the distance between the thumb and the middle finger's tip positions is calculated. If the distance is less than a threshold value, it is recognized as *Baby O Handshape*. Therefore when the actual sign is *Baby O Handshape*, sometimes the distance calculated for the recognition of the *ASL A Handshape* can be less than the threshold value because of the noise in the data. The *Open 8* sign is another sign which is occasionally misclassified. As it can be seen from the real time phase results, *Open 8 Handshape* is very similar to *5 Handshape*; just the middle finger's metacarpo-phalangeal joint angle is smaller in the *Open 8 Handshape*. In the recognition model, the middle finger's tip direction is compared to the palm's normal vector's direction. When the result of this comparison is less than a threshold value, the sign is recognized as *Open 8 Handshape*. Because of some slight detection errors of the middle finger's direction vector, a small percent (1.33%) of *Open 8 Handshape* sign samples are misclassified as *5 Handshape*.

5.2. Machine Learning Method Results

A Machine Learning method, namely *Random Forest* is also tested in the study. First, the model is trained using the *dataTrain.arff* file with 10 folds cross-validation. Then to test the trained model, the *dataTest.arff* file is used with the same method as the supplied test set. In the test there were 2700 instances and 139 attributes as explained before. There are no omitted attributes in the test. While using the *Random Forest* method, *Weka* constructed 100 trees by considering eight random features, and model building took 2.74 seconds. The test results of the *Random Forest* method are as follows: The correctly classified instances count is 2527 which is equivalent to 93.5926%, there are 173 misclassified instances which is equivalent to 6.4074%.

The sign-specific results can be seen in the confusion matrix in table 2. As displayed in the Table 2, some signs are recognized with 100% success rate. Those signs are: *V & 2 Handshape*, *L Handshape*, *O Handshape*, *C Handshape*, *ASL A Bar Handshape*, *ASL 8 Handshape*, *ASL I&H Handshape*, *ASL I Handshape*, *ASL Y Handshape*, *ASL 3 Handshape*, *4 Claw Handshape*, *8 Handshape*, *9 Handshape*, and *Open 8 Handshape*. Other signs have lower success rates, as the *5 Handshape* with 65.33%, the *ASL A Handshape* with 80%, the *Baby O Handshape* with 50.66%, and *ASL Q Handshape* signs with a 88.66% success rate.
The signs which had lower recognition rates are *5 Handshape*, *ASL A Handshape*, *Baby O Handshape*, and *ASL Q Handshape* is predicted to be *Open 8 Handshape*, *Baby O Handshape*, *8 Handshape*, *C Handshape* respectively.

5.3. Overall Results

In both methods, the success rate is lower in the same signs, mainly *5 Handshape* vs. *Open 8 Handshape* and *ASL A Handshape* vs. *Baby-O Handshape*. In the *Random Forest* method, *Open 8 Handshape* is also confused with *ASL I Handshape* and *8 Handshape*, and additionally *ASL Q Handshape* is confused with *C Handshape*. The success rate results of every hand shape for both methods can be seen in Table 3. The machine learning method's success rates for the signs are all lower than the heuristic method's success rates except for the *Open 8 Handshape*, which has a very small difference.
The results show that the proposed heuristic model is more robust and precise with 99.03% success rate as average result, while the machine learning method's average success rate remained at 93.59%. Also, the machine learning method requires big sampling sets to create the learning

model while the heuristic method needs none, which makes it easier to integrate more signs. Therefore, for the recognition of the hand shape primitives of Turkish Sign Language, the proposed heuristic method is precise and preferable, especially in real-time recognition systems.

Act.\Pred.	Handshapes																	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1 (5)	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 (V & 2)	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3 (L)	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 (O)	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5 (C)	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0
6 (asl ABar)	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0
7 (ASL 8)	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0
8 (ASL IH)	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0
9 (ASL I)	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0
10 (ASL Y)	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0
11 (ASL 3)	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
12 (4 Claw)	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0
13 (8)	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0
14 (9)	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
15 (ASL A)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0
16 (Baby O)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	126	0	0
17 (Open 8)	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	148	0
18 (ASL Q)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150

Table 1: Confusion Matrix Table for Heuristic Results (Actual vs. Predicted)

Act.\Pred.	Handshapes																	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1 (5)	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	0
2 (V & 2)	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3 (L)	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4 (O)	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5 (C)	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0	0
6 (asl ABar)	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	0
7 (ASL 8)	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0
8 (ASL IH)	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0
9 (ASL I)	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0
10 (ASL Y)	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0
11 (ASL 3)	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
12 (4 Claw)	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0
13 (8)	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0
14 (9)	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
15 (ASL A)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	30	0	0
16 (Baby O)	0	0	0	0	0	0	0	0	30	0	0	0	39	0	5	76	0	0
17 (Open 8)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0
18 (ASL Q)	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	133

Table 2: Confusion Matrix Table for Machine Learning (Actual vs. Predicted)

6. Acknowledgements

This work is supported by ”Avatar based Machine translation system for Turkish Sign Language” TUBITAK 1003 (FATIH) project with project number 114E263.

We wish to express our sincere gratitude to Researcher Yakup ÖZDEN for his technical assistance about machine learning.

Hand Shape	**Heuristic**	**RF**
5 Handshape	100%	65.33%
V & 2 Handshape	100%	100%
L Handshape	100%	100%
O Handshape	100%	100%
C Handshape	100%	100%
ASL ABar Handshape	100%	100%
ASL 8 Handshape	100%	100%
ASL I & H Handshape	100%	100%
ASL I Handshape	100%	100%
ASL Y Handshape	100%	100%
ASL 3 Handshape	100%	100%
4 Claw Handshape	100%	100%
8 Handshape	100%	100%
9 Handshape	100%	100%
ASL A Handshape	100%	80%
Baby O Handshape	84%	50.66%
Open 8 Handshape	98.66%	100%
ASL Q Handshape	100%	88.66%
Average	**99.03%**	**93.59%**

Table 3: Success Rate of Hand Shapes for Both Heuristic and Machine Learning Methods

7. Bibliographical References

Chuan, C.-H., Regina, E., and Guardino, C. (2014). American sign language recognition using leap motion sensor. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 541–544.

Haberdar, H. and Albayrak, S. (2005). Real time isolated turkish sign language recognition from video using hidden markov models with global features. In Pınar Yolum, et al., editors, *Computer and Information Sciences - ISCIS 2005*, volume 3733 of *Lecture Notes in Computer Science*, pages 677–687. Springer Berlin Heidelberg.

Keskin, C., Kıraç, F., Kara, Y., and Akarun, L. (2013). Real time hand pose estimation using depth sensors. In Andrea Fossati, et al., editors, *Consumer Depth Cameras for Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 119–137. Springer London.

Kubuş, O. (2008). An analysis of turkish sign language (tid) phonology and morphology. Master’s thesis, MIDDLE EAST TECHNICAL UNIVERSITY.

Köse, H., Akalin, N., and Uluer, P. (2014). Socially interactive robotic platforms as sign language tutors. *International Journal of Humanoid Robotics*, 11(01):1–22.

Köse, H., Uluer, P., and Akalin, N. (2015a). Sign language games in child-robot interaction. In *Proceedings of the XVII World Congress of the World Federation of the Deaf (WFD)*, pages 61–62.

Köse, H., Uluer, P., Akalin, N., Yorganci, R., Ozkul, A., and Ince, G. (2015b). The effect of embodiment in sign language tutoring with assistive humanoid robots. *International Journal of Social Robotics*, 7(4):537–548.

Leap Motion, I. (2016). Api overview.

Mohandes, M., Aliyu, S., and Deriche, M. (2014). Arabic sign language recognition using the leap motion controller. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 960–965.

Villarreal, M. R. (2007). File:scheme human hand bones-en.svg - image is retouched by leap motion, inc.

8. Language Resource References

Kubuş, O. (2008). An analysis of turkish sign language (tid) phonology and morphology. Master’s thesis, MIDDLE EAST TECHNICAL UNIVERSITY.

Appendix: Recognition Rules of the Heuristic Model

Rule for C Handshape

```
For C-handshape:
    IF Extended Finger Count is 1
        IF Sign is !(is 4-Claw)
        AND Sign is !(is ASL A-bar)
        AND Sign is !(is ASL I-handshape)
        AND 0th Extended Finger is "Thumb"
        AND Distance Btw Index and Middle Finger Tip Positions > DISERRORMARGIN
            IF Angle Btw Index Finger's Proximal bone's Direction and Palm Normal Vectors < PI / 2 + ERRORMARGIN
                IF ERRORMARGIN < Angle Btw Index Finger's Proximal and Intermediate Bones' Direction Vectors < PI / 2
                    IF Palm Direction is "Left" or "Right"
                        Sign is "C-handshape"
```

Rule for ASL I&H Handshape

```
For ASL I&H:
    IF Extended Finger Count is 2
        IF 0th Extended Finger is "Index"
        AND 1th Extended Finger is "Pinky"
            IF Palm Direction is "Up"
                Sign is "ASL I&H"
```

Rule for L Handshape

```
For L-handshape:
    IF Extended Finger Count is 2
        IF 0th Extended Finger is "Thumb"
        AND 1th Extended Finger is "Index"
            IF Palm Direction is "Down"
            AND Angle Btw Thumb and Index Finger Direction Vectors > LIMITANGLEFORL
            AND Angle Btw Index Finger's Proximal and Intermediate Bones' Direction Vectors < ERRORMARGIN
                Sign is "L-handshape"
```

Rule for Open 8 Handshape

```
For Open 8 Handshape:
    IF Extended Finger Count is 5
        IF Angle Btw Middle Finger Direction and Palm Normal Direction Vectors <= (PI*1/4)
            Sign is "Middle selected ASL (open 8)"
```

Rule for Q Handshape

```
For ASL Q-handshape:
    IF Extended Finger Count is 2
        IF 0th Extended Finger is "Thumb"
        AND 1th Extended Finger is "Index"
            IF Sign is !(is L-handshape)
            AND Palm Direction is "Left" or "Rigth"
            AND Angle Btw Thumb and Index Finger Direction Vectors < LIMITANGLEFORL
                Sign is "ASL Q-handshape"
```

Rule for ASL I Handshape

```
For ASL I-handshape:
    IF Extended Finger Count is 1
        IF Sign is !(is 4-Claw)
        AND Sign is !(is ASL A-bar)
        AND 0th Extended Finger is "Pinky"
            Sign is "ASL I-handshape"
```

Rule for 4 Claw Handshape

```
For 4-Claw:
    IF Extended Finger Count is 1
        IF 0th Extended Finger is "Thumb"
        AND Palm Direction is "Down"
        AND Angle Btw Index Finger's Proximal bone's Direction and Palm Normal Vectors < PI / 2
            Sign is "4-Claw"
```

Rule for V&2 Handshape

```
For V/2-handshape:
    IF Extended Finger Count is 2
        IF 1th Extended Finger is "Middle"
        AND 0th Extended Finger is "Index"
            IF Palm Direction is "Down"
            AND Angle Btw Index and Middle Finger Direction Vectors > DegreeToRadian(11)
                Sign is "V/2-handshape"
```

Rule for ASL 3 Handshape

```
For ASL 3-handshape:
    IF Extended Finger Count is 3
        IF 0th Extended Finger is "Thumb"
        AND 1st Extended Finger is "Index"
        AND 2th Extended Finger is "Middle"
            IF Palm Direction is "Down"
                Sign is "ASL 3-handshape"
```

Rule for ASL A Bar Handshape

```
For ASL A Bar:
    IF Extended Finger Count is 1
        IF Sign is !(is 4-Claw)
        AND 0th Extended Finger is "Thumb"
        AND Distance Btw Index and Middle Finger Tip Positions<DISERRORMARGIN
        AND Palm Direction is "Down"
            Sign is "ASL A-bar"
```

Rule for ASL 8 Handshape

```
For ASL 8-handshape:
    IF Extended Finger Count is 3
        IF Sign is !(is ASL 3-handshape)
        AND 0th Extended Finger is "Index"
        AND 1st Extended Finger is "Ring"
        AND 2nd Extended Finger == "Pinky"
            IF Distance Btw Thumb and Middle Finger Tip Positions < DISERRORMARGIN
            AND Distance Btw Thumb and Index Finger Tip Positions > DISERRORMARGIN
                Sign is "ASL 8-handshape"
```

Rule for O Handshape

```
For O-handshape:
    IF Extended Finger Count is 3
        IF Sign is !(is ASL 3-handshape)
        AND Sign is !(is ASL 8-handshape)
        AND 0th Extended Finger is "Middle"
        AND 1th Extended Finger is "Ring"
        AND 2th Extended Finger is "Pinky"
            IF Distance Btw Thumb and Index Finger Tip Positions < DISERRORMARGIN
            AND Distance Btw Thumb and Middle Finger Tip Positions > DISERRORMARGIN
                Sign is "O-handshape"
```


Rule for O Handshape

```
For O-handshape:
    IF Extended Finger Count is 3
        IF Sign is !(is ASL 3-handshape)
        AND Sign is !(is ASL 8-handshape)
        AND 0th Extended Finger is "Middle"
        AND 1th Extended Finger is "Ring"
        AND 2th Extended Finger is "Pinky"
            IF Distance Btw Thumb and Index Finger Tip Positions < DISERRORMARGIN
            AND Distance Btw Thumb and Middle Finger Tip Positions > DISERRORMARGIN
                Sign is "O-handshape"
```


Rule for 5 Handshape

```
For 5-handshape:
    IF Extended Finger Count is 5
        IF Sign is !(is ASL Open 8 handshape)
        AND Palm Direction is "Down"
        AND Distance Btw Index and Middle Finger Tip Positions > DISERRORMARGIN
        AND Distance Btw Ring and Pinky Finger Tip Positions > DISERRORMARGIN
            Sign is "5-handshape"
```

Rule for ASL Y Handshape

```
For ASL Y-handshape:
    IF Extended Finger Count is 2
        IF 0th Extended Finger is "Thumb"
        AND 1th Extended Finger is "Pinky"
            IF Palm Direction is "Down"
                Sign is "ASL Y-handshape"
```

Rule for 8 Handshape

```
For 8-handshape:
    IF Extended Finger Count is 0
        IF Palm Direction is "Down"
        AND Distance Btw Ring and Pinky Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Thumb and Ring Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Thumb and Middle Finger Tip Positions>DISERRORMARGIN
        AND Angle Btw Index Finger's Proximal Bone's Direction and Hand's Direction Vectors > PI / 2
        AND Angle Btw Index Finger's Proximal and Intermediate Bones' Direction Vectors > PI / 2 - ERRORMARGIN
        AND Angle Btw Middle Finger's Proximal and Intermediate Bones' Direction Vectors > PI / 2 - ERRORMARGIN
            Sign is "8-handshape"
```

Rule for 9 Handshape

```
For 9-handshape:
    IF Extended Finger Count is 0
        IF Sign is !(is 8-handshape)
        AND Distance Btw Ring and Pinky Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Thumb and Middle Finger Tip Positions<DISERRORMARGIN
        AND Distance Btw Index and Middle Finger Tip Positions > DISERRORMARGIN
        AND Angle Btw Index Finger's Proximal Bone's Direction and Hand's Direction Vectors >PI / 2
        AND Angle Btw Index Finger's Proximal and Intermediate Bones' Direction Vectors > PI / 2 - ERRORMARGIN
            Sign is "9-handshape"
```

Rule for Baby O Handshape

```
For Baby-O handshape:
    IF Extended Finger Count is 0
        IF Sign is !(is 8-handshape)
        AND Sign is !(is 9-handshape)
        AND Palm Direction is "Down"
        AND Distance Btw Index and Middle Finger Tip Positions < DISERRORMARGIN
        AND Distance Btw Ring and Pinky Finger Tip Positions < 40
        AND Distance Btw Thumb and Middle Finger Tip Positions < DISERRORMARGIN
        AND Angle Btw Thumb Finger's Distal and Intermediate Bones' Direction Vectors < DegreeToRadian(30)
            Sign is "Baby-O handshape"
```

Rule for ASL A Handshape

```
For ASL A-handshape:
    IF Extended Finger Count is 0
        IF Sign is !(is 8-handshape)
        AND Sign is !(is 9-handshape)
        AND Sign is !(is Baby-O handshape)
        AND Palm Direction is "Down" or "Up"
        AND Angle Btw Thumb Finger's Direction and Hand's Direction Vectors  < PI - ERRORMARGIN
            Sign is "ASL A-handshape"
```