

ELM based two-handed dynamic Turkish Sign Language (TSL) word recognition

Zekeriya Katılmış^{a,*}, Cihan Karakuzu^{b,*}

^a Institute of Graduate, Department of Electronics and Computer Engineering, Bilecik Seyh Edebali University, Bilecik 11210, Turkey

^b Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik 11210, Turkey

ARTICLE INFO

Keywords:

Sign language recognition
Hand gesture recognition
Dynamic word gestures
Leap Motion Controller (LMC)
Extreme Learning Machine (ELM)

ABSTRACT

Hearing impaired individuals can easily overcome the barriers in communicating with other members of the society via computer technology. In this study, the recognition of dynamic words in Turkish Sign Language (TSL) with two hands was studied using the Leap Motion Controller (LMC) device. 50 dynamic words were determined considering the similarities and differences among themselves, and a dataset was created using 4 signers. For the system proposed in this paper, a comprehensive feature extraction process is executed. Applying feature selection algorithms and PCA, LDA and PCA+LDA dimension reduction methods to this dataset, new datasets with less dimension were obtained. For the first time, ELM architectures were operated as a classifier in a sign language recognition system. Recognition performance was tested with 5 different ELM networks and 2 classical classifiers and the results were compared. In addition, comparisons of classical and ELM based classifiers were presented. The 10-fold cross validation method was used to test the validity of the proposed system and the accuracy of the results obtained. Based on the results obtained by a comprehensive analysis, it was observed that the ML-KELM classifier maintains its performance rate and gives the highest performance rate. At the same time, it has been observed that ML-KELM classifier has a stable structure, which offers less user intervention.

1. Introduction

Sign language is an important communication tool used by hearing-impaired people to communicate with each other and with other people. Sign language is defined as the mother tongue of the hearing-impaired and they provide a great part of their communication with this language. According to the statistics of the World Health Organization, the number of hearing-impaired people is approaching 466 million in the world and 3 million in Turkey. This number is increasing day by day, and it is known that 90% of these people do not have the necessary devices. For this reason, the recognition and interpretation of sign language by computer systems is both socially and technologically important.

Many recognition systems have been designed and developed with machine learning methods in sign language recognition. So far these systems can be categorized as electronic glove-based (Oz & Leu, 2011), image-based (AL-Rousan, Assaleh, & Tala'a, 2009), Kinect-based (Tiwari, Anand, Keskar, & Satpute, 2015), EMG-based (Kosmidou, Hadjileontiadis, & Panas, 2006), Leap Motion-based (Mohandes, Aliyu, & Deriche, 2014) and their combined use with multimodal-based

(Kumar, Gauba, Roy, & Dogra, 2017a, 2017b) systems. Although the systems other than Leap Motion have their own advantages, their usability in daily life is very low. For this reason, Leap Motion Controller (LMC) has been preferred both in every part of daily life and for developing a portable system.

In this study, it has been studied to recognize the dynamic (gesture) words performed with two hands in Turkish Sign Language (TSL) by using the Leap Motion device, which is a motion-based platform based on a digital sensor system. The reason why Leap Motion device is preferred for this study is that it is cheaper, lighter and easily portable compared to similar sensors and cameras, and can be easily embedded or directly integrated into any technological device used today. These features of this device are important for hearing impaired people to easily communicate with other people anytime anywhere. There is a few number of Leap Motion based dynamic word recognition studies in the literature. In these studies, the words of languages such as ASL, ArSLR, ISL and SIBI were studied. However, TSL dynamic word recognition study is not available.

The dataset created for this study has been reduced gradually by applying the feature selection algorithm first. In the second stage, the

* Corresponding authors.

E-mail addresses: zekeriya.katilmis@bilecik.edu.tr (Z. Katılmış), cihan.karakuzu@bilecik.edu.tr (C. Karakuzu).

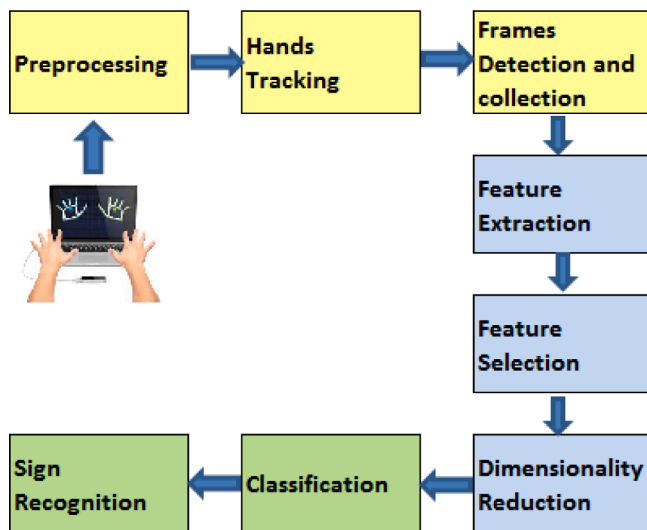


Fig. 1. Workflow of the proposed recognition model.

dimension has been reduced by using dimension reduction methods. Using Extreme Learning Machine (ELM) learning methods, the classification success rates of the original dataset and the reduced dataset dimension has been compared. The 10-fold cross validation method was used to test the validity of this proposed system and the accuracy of the results obtained.

The general workflow of the proposed recognition model consists of 3 stages and 8 steps as shown in Fig. 1. Leap Motion stage (yellow blocks) consists of preprocessing, tracking and image frame collection steps. The dataset stage (blue blocks) consists of feature extraction, feature selection and dimension reduction steps. The application stage (green blocks) consists of training and testing steps. The first step is the preprocessing step that starts the leap motion service. The second step is the tracking step, in this step the tracking layer matches the data by extracting tracking information such as hand and fingers. The third step is to collect image frames from data that is matched using LMC API features. In the fourth step, the dataset is created by extracting the determined features from the image frames collected from LMC. In the fifth step, feature selection is performed by applying the feature selection algorithm on these features. In the sixth step, the dimensions of the selected features are reduced by reduction method. The seventh step is the stage in which these features are presented to the classifier as a vector and training is done to define words. In the last step, the trained classifier is used to recognize the gestures of users. The contributions of this study to the literature can be summarized as follows.

- LMC-based dynamic word recognition for TSL
- Using signal processing technique for feature selection that will enable us to distinguish between different and similar various gestures in sign language recognition
- Using PCA, LDA, and PCA+LDA dimension reduction methods in sign language recognition
- Using the ELM method in sign language recognition.

2. Related works

There are many studies on sign language in the literature. Among these studies, in the context of the subject and scope of this study, LMC-based studies have been examined under two headings as static letter and dynamic word recognition. The literature summary of static letter recognition studies has been prepared to present comparison with dynamic word recognition studies. The summary of static letter and dynamic word recognition studies is presented in Table 1 and Table 2, respectively. In the tables, different colors are used for differences

languages in order to provide distinction.

2.1. Static letter recognition works

According to the literature review given in Table 1, there are very few studies for static Turkish Sign Language (TSL) recognition based on LMC device. Demircioğlu et al. conducted a study with LMC to recognize the selected basic hand movements of TSL. They used the data consists of 2700 records with 139 features, 5 or 10 samples from each of 18 hand signals, 30 frames for each sample for static and single-handed signs. They used this data set to test various machine learning methods and comparing (Demircioğlu, Bülbül, & Köse, 2016a, 2016b). On the other hand, Karacı et al. created a dataset by taking 5 single-handed signs and 5 two-handed signs in TSL from 2 different signers using LMC. For one-handed and two-handed signs, they used the 120 and 390 featured datasets, respectively. They stated that the decision tree and deep learning models were more successful for a data set containing a very limited number of signs, only 5, (Karacı, Akyol, & Gültepe, 2018). We proposed two-handed static TSL alphabet recognition system using a dataset consisting of 3672 samples with 50 repetitions from 2 different signers for 17 letters. In addition, by applying PCA, LDA and PCA+LDA methods, 3 dimensionally-reduced datasets were created. The results are compared by testing with traditional machine learning methods. The highest performance rate was obtained with RF with 98.5% (Katılmış & Karakuzu, 2020).

2.2. Dynamic word recognition works

Aliyu, Mohandes, and Deriche (2017) proposed a dual LMC-based ArSLR recognition system to meet the challenges associated with finger clogs and missing data. For feature extraction, 17 geometric features were selected from both devices, while Bayesian approach Gaussian Mixture Model (GMM) and simple LDA approach were used for classification. Data were collected from one adult signer for isolated double-handed 100 Arabic dynamic signal words, recognition accuracy of 94% was achieved. In the second study of the same authors (Deriche, Aliyu, & Mohandes, 2019), 92% success was obtained using 2 adult signers. Pramananto, Sumpeno, and Legowo (2017) identified 19 features by examining 5 static and 5 dynamic one-hand gestures in Indonesian sign language (SIBI). In the study, the dataset was created by one signer by making 25 repetitions for each sign. In the classification using the Naive Bayes (NB) method, 80% accuracy was obtained. In the study of Avola, Bernardi, and Cinque (2019), dynamic hand gestures were represented by feature vector sets that change over time. Recurrent Neural Networks (RNN) were trained by taking the angles formed by the finger bones of the human hand as a feature. The dataset contains 12 dynamic and 12 static American Sign Language (ASL) words. The dataset consists of 1200 hand gesture sequences taken from 20 different signers. Each sign was made by 15 men and 5 women. A performance above 96% was achieved in this study. In Mittal, Kumar, Roy, Balasubramanian, and Chaudhuri (2019), Mittal et al. proposed the LSTM model for continuous sequences of gestures and a sign language recognition system that recognizes sequence of connected gestures. The system is based on splitting continuous signs into sub-units and modeling them with neural networks. The dataset was created by 6 signers repeating each sign 15 times. 36 dynamic features were extracted for the dataset. The proposed system has been tested using 35 double-handed isolated different sign words and 942 signed sentences from the Indian Sign Language (ISL). Average accuracy of 72% and 89%, respectively, was obtained. In the study of Kumar, Gauba, Roy, and Dogra (2017a), a multi-sensor fusion framework for sign language recognition using the Coupled Hidden Markov Model (CHMM) is proposed. This framework is used to recognize dynamic isolated sign gestures performed by hearing impaired persons. The best recognition accuracy was achieved with CHMM as high as 90%. For 25 dynamic single-handed sign words belonging to ISL, the dataset was created by repeating 8 times by 10

Table 1
Static sign recognition literature summary. (See below-mentioned references for further information.)

Ref.	Approach (Feature extraction, Feature number and Language)	Hand	Dataset, Classifier, Accuracy
(Mohandes, Aliyu, & Deriche, 2014)	3D finger length,width,tip position, palm position,hand pitch,roll and yaw. 12 ArSLR	single hand	28 alphabet, 1 signer, 10 repetitions,10 frames MLP NBC 99 98
(Mohandes, Aliyu, & Deriche, 2015)	3D finger length,width,tip position, palm position,hand pitch,roll and yaw. 12 ArSLR	single hand	28 alphabet, 1 signer, 10 repetitions,10 frames LDA 97
(Mapari & Kharat, 2015)	Positional values,distance values,angle values 48 ASL ISL CSL	single hand single hand both hand	32 alphabet, 146 signers, 1 repetitions 34 alphabet, 100 signers, 1 repetitions 33 alphabet, 10 signers, 10 repetitions MLP SVM GFF 90 93 96
(Mapari & Kharat, 2017)	Positional values,distance values,angle values 48 ISL	single hand	31 alphabet, 10 signers, 1 repetitions ED CS 88 90
(Naglot & Kulkarni, 2016)	Consecutive finger tip position to palm position,finger tip position of each consecutive finger 9 ASL	single hand	26 alphabet, 4 signers, 5 repetitions MLP 96
(Chuan, Regina, & Guardino, 2014)	Average distance, average spread, aveage tri-spread. 35 ASL	single hand	26 alphabet, 2 signers, 4 data set KNN SVM 72 79
(Chophuk, Pattanaworapan, & Chamnongthai, 2018)	3D fingertip position,3D palm position 18 ASL	single hand	7 alphabet, 4 signers, 30 samples DT 96
(Eqab & Shanableh, 2017)	3D Hand directions,3D Fingertip positions,hand grab strength 19 ArSLR	single hand	15 words, 1 signer, 5 repetitions MDC 98
(Wibowo, Nurtanio, & Ilham, 2017)	ISL	single hand	24 alphabet, 3 signers, 10 repetitions NBC 95
(Anwar, Basuki, Sigit, Rahagiyanto, & Zikky, 2017)	3D fingertip position,3D palm position 11 ISL	single hand	26 alphabet, 5 signers, 10 repetitions,10 frame KNN SVM 95 93
(Insani, Nurtanio, & Ilham, 2019)	Positional values,distance values 17 ISL	single hand	24 alphabet, 1 signer, 40 repetitions SVM 96
(Khotimah, Saputra, Suciati, & Hariadi, 2017)	Positional values,distance values 34 ISL	single hand	26 alphabet, 1 signer, 5 repetitions,10 frames BPGA -NN 93
(Naidui & Ghotkar, 2016)	3D fingertip position,3D palm position 8 16 ISL	single hand double hand	45 signs, 10 signers, 1 repetitions ED 90
(Demircioğlu, Bülbül, & Köse, 2016 a)	Finger extended,palm and finger direction, finger distance,finger tip position,finger direction, bones angles 139 TSL	single hand	18 signs, 1 signer, 5 samples,30 frame MLP RF 96 93
(Demircioğlu, Bülbül, & Köse, 2016 b)	Finger extended,palm and finger direction,finger distance,finger tip position,finger direction and bones angles 139 TSL	single hand	18 signs, 1 signer, 10 samples,30 frame RF 93
(Karacı, Akyol, & Gültepe, 2018)	Bone directions,palm position,arm direction, wrist position, elbow position 120 390 TSL	single hand double hand	5 alphabet, 2 signers, 5 repetitions 5 alphabet, 2 signers, 5 repetitions DNN ANN DT 100 93 98
(Katılmış & Karakuzu, 2020)	Fingers extended,position distance, direction angles,3D Palm, hand, arm wrist and elbow position 119 TSL	double hand	17 alphabet, 2 signers, 50 repetitions SVM RF LDA 95 98 93

different signers. In this study, LMC and Kinect devices were used in parallel to capture the motion of the sign gesture. 66 features were specified for the dataset. In a similar study by Kumar, Gauba, Roy, and Dogra (2017b); they proposed a multimodal framework for isolated sign language recognition using sensor devices. Kinect and LMC devices are used in a multimodal framework created to capture finger and palm positions from two different views during the gesture. A set of features

were extracted from the raw data captured with the both sensors. Recognition was carried out separately by the Hidden Markov Model (HMM) and Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) based sequential classifiers. The framework was tested on a dataset of 7500 ISL gestures containing 50 different sign words. Accuracy is improved if data from both sensors are combined compared to single sensor-based recognition. The success rates for HMM and BLSTM-

Table 2
Dynamic sign recognition literature summary. (See below-mentioned references for further information.)

Ref.	Device	Approach (Feature extraction, Feature number and Language)		Hand	Dataset, Classifier, Accuracy			
(Aliyu, Mohandes, & Deriche, 2017)	Dual LMC	3D Fingertips position, 3D palm Position, finger distance, hand Pitch, Roll, Yaw, hand sphere radius	16	ArSLR	double hand	100 words, 1 signer, 10 repetitions	LDA GMM	94
(Deriche, Aliyu, & Mohandes, 2019)	Dual LMC	3D Fingertips position, 3D palm Position, finger distance, hand Pitch, Roll, Yaw, hand sphere radius	16	ArSLR	double hand	100 words, 2 signers, 10 repetitions	LDA GMM	92
(Pramunanto, Sumpeno, & Legowo, 2017)	LMC	Finger and palm angle, palm distance, palm elevations	19	SIBI	single hand	5 words, 1 signer, 25 repetitions	NBC	80
(Avola, Bernardi, & Cinque, 2019)	LMC	3D Fingertip positions, 3D palm position, internal angles, intra-finger angles	23	ASL	single hand double hand	12 words, 20 signers, 2 repetitions	RNN LSTM	96
(Mittal, Kumar, Roy, Balasubramanian, & Chaudhuri, 2019)	LMC	3D Fingertip positions, 3D palm position	36	ISL	double hand	35 words, 6 signers, 15 repetitions	LSTM	89
(Kumar, Gauba, Roy, & Dogra, 2017a)	LMC, Kinect	3D Fingertip position, 3D fingertip direction	66	ISL	single hand	25 words, 10 signers, 8 repetitions	CHMM	90
(Kumar, Gauba, Roy, & Dogra, 2017b)	LMC, Kinect	3D Fingertip position, 3D fingertip direction	132	ISL	single hand double hand	50 words, 10 signers, 15 repetitions	HMM BLSTM -NN	97 94
(Hisham & Hamouda, 2017)	LMC	3D Directional and positional features, finger angles, subtraction of each phalanx's end position and the palm's position vectors	170	ArSLR	single hand double hand	16 words, 3 signers, 25 repetitions	DTW SVM KNN ANN	97
		3D Directional and positional features, finger angles, subtracting the phalanx's start position from phalanx's end position vectors	140					96
This work	LMC	Fingers extended, position distance, direction angles, 3D Palm, hand, arm wrist and elbow position	119	TSL	double hand	50 words, 4 signers, 40 repetitions	ELM ML-KELM	96 99

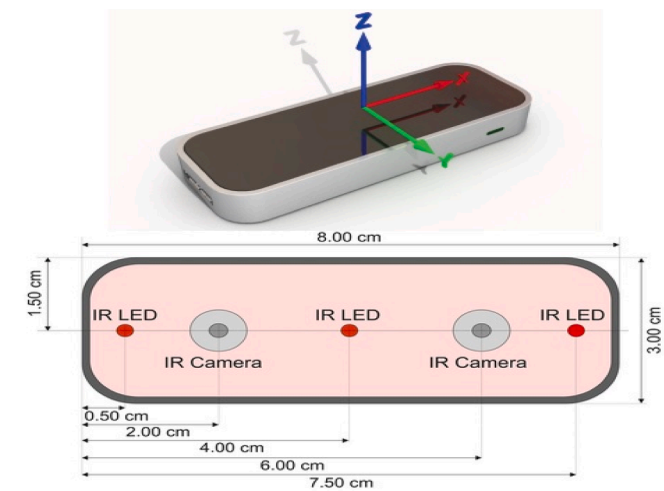


Fig. 2. The LMC device (top) and its schematic view (down) (LMC, 2019).

NN were 97% and 94%, respectively. Hisham and Hamouda (2017) proposed a model to recognize static and dynamic movements in the realization of signs. The proposed model is based on 2 different palm and bone feature sets that have characteristics in common between them. The work includes 38 static gestures and 20 dynamic gestures. 99% and 98% success rate with KNN model for static and dynamic gestures, respectively; it has been reported that 97% and 96% success rates were achieved with the DTW model, respectively.

As can be seen from the works given in Table 1 and Table 2, and briefly summarized above, this study proposes a different perspective and solution method. Briefly, the specificities of this study are:

- The first application of ELM learning as a classification method to sign language recognition
- Being the first study in this contact in dynamic words recognition in Turkish Sign Language (TSL)
- Forming a comprehensive dataset for dynamic word recognition in TSL
- Presentation of recognition performance analysis with 9 different datasets formed by integrating 3 different dimension reduction methods with a different gesture extraction and selection approaches.

3. Leap Motion Controller (LMC)

LMC is a device with the front view given in Fig. 2 that allows interaction with the software using hand gestures. This device uses a pair of cameras and infrared LEDs to obtain a hand image with depth information. LMC can detect and record hand and finger gestures in three-dimensional space at a high speed of 290 fps, up to 1% of a millimeter with zero delay. LMC can track gestures up to 200 frames per second within an effective field of view of 150 degrees from approximately 25 to 600 mm. This device uses the right-hand cartesian coordinate system. The detected values are in millimeters. The LMC application programming interface (API) provides a series of snapshots, called images frames, of motion tracking data. This device (by scanning hands, fingers and similar objects within the field of view) continuously

Table 3
Dynamic words similar to each other in terms of construction.

	Words (Turkish, English)
1	Balta, patlıcan, yarım (Cleaver, eggplant, half)
2	Açık, baklava, yarı final (Open, baklava, semifinal)
3	Farklı, final, sınav, takım (Different, final, exam, team)
4	Fırın, gemi (Oven, ship)
5	Avustralya, Avusturya (Australia, Austria)
6	Bıçak, çabuk (Knife, fast)
7	Damga, köfte (Stamp, meatball)
8	Muhasebe, terzi (Accounting, tailor)
9	Cüzdan, incir (Wallet, fig)
10	Kutu, masa, rekor (Box, table, record)
11	Suç, yumurta (Crime, egg)
12	Felsefe, trafik (Philosophy, traffic)
13	Matkap, pergel, tornavida (Drill, compass, screwdriver)
14	Çatal, fiş (Fork, plug)
15	Kibar, kibrit (Gentle, match)
16	Akraba, dama (Relative, checkerboard)
17	Sosis, tel (Sausage, wire)
18	Çıрак, eş (Apprentice, spouse)
19	Pasta, serseri, silecek, yönetmek (Cake, punk, squeegee, manage)
20	Coğrafya, deprem, kilim, yağmur (Geography, earthquake, rug, rain)

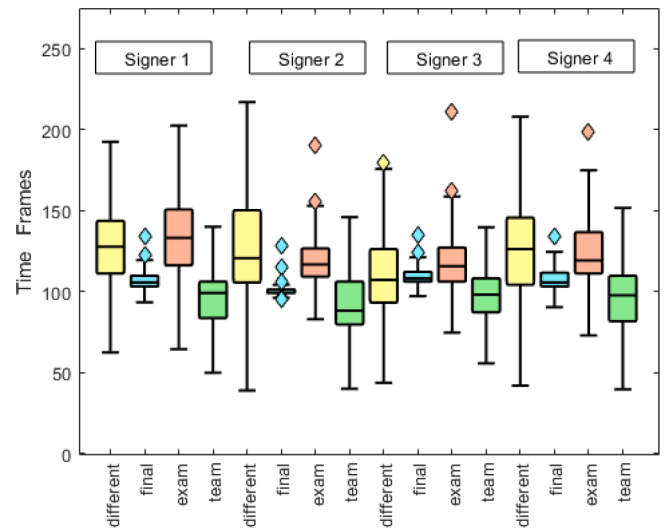


Fig. 4. Box plots for sign gestures of similar words.



Fig. 3. Image sequences for the words “gemi(ship)” (top row) and “fırın(oven)” (bottom row) (TİD, 2019).

provides a data array and image frames. Each image frame contains basic tracking information such as hand, finger, and object, as well as currently recognized gestures and status information. If a hand, finger, object or gesture is detected, the device assigns a unique ID value. This ID value is preserved as long as the object being remains in the field of view of the device.

4. Feature extraction

Within the scope of this study, 50 dynamic words were prepared by considering the similarities and differences among themselves. These are composed of words under 20 different categories that are similar to each other in terms of their construction. The words used in our study are presented in Table 3. The origins Turkish words are given in the first line of the table, and the English equivalents of the words are given in the second line. Words that were created close to each other in terms of their construction time are preferred. With these choices, it is aimed to increase the accuracy and reliability of the classification success rate of words.

In this study, firstly, data were collected by using 4 signers, two adult males and two adult females, for 50 dynamic words performed with two hands in Turkish Sign Language and feature extraction were made. In this context, 119 distinctive features were identified for each of the hand, finger, joint and joint points from features such as direction, length, distance, location, angle, and position. Among these features, there is also angle information, considering that it will be distinctive in

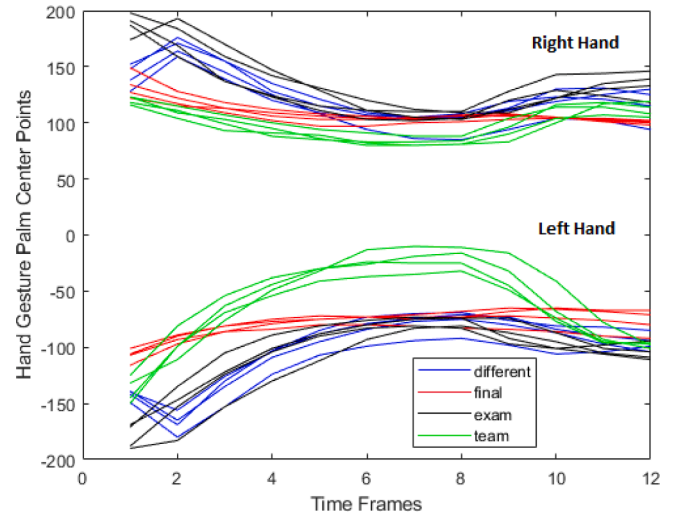


Fig. 5. 2D plot for sign gestures of similar words.

the recognition of similar letters, regardless of hand and finger size alone. A dataset consisting of a total of 8000 records from 4 signers including 40 samples with a single frame for each word was created. The dataset consists of a combination of 480 image frames taken within 12 s for each word. The record created by each signer for each word was saved as a separate file. The image sequence of two dynamic similar words used is presented in Fig. 3.

Large variation in data is measured using a statistical analysis technique known as box-plot. It uses median, approximate quartiles, and lowest / highest data points to convey the level, spread, and symmetry of distribution of the data. A sample box-plots representation for the hand gestures of 4 similar words of 4 signers is presented in Fig. 4. These gestures were captured using LMC. In addition, since a sign gesture sequence is performed in air, thus, can be visualized using 2D plot that shows the variation within the hand gestures of the similar words when performed by 4 signers. Fig. 5 has been obtained by plotting the palm center points during the whole sequence. Each color in the plot is used to discriminate a signer from others which shows the variation in the input sequences. This figure presents a sample 2D plot representation for the hand gestures of 4 similar words of 4 signers.

The proposed system was implemented using Intel(R) Core(TM) i7-3632QM processor @ 2.20 GHZ speed and 6 GB RAM. In order to

Table 4
Features of hands and fingers used in this study.

Hands		Fingers	
Name	Type	Name	Type
Direction	Vector	Distance	mm
PalmNormal	Vector	Angle	Radian
PalmPosition	Vector	IsExtended	Bool
Leftmost	Hand	Direction	Vector
Rightmost	Hand	TipPosition	Vector
Fingers	Fingerlist	Bone(DISTAL).Direction	Vector
Arm.ElbowPosition	Vector	Bone(INTERMEDIATE).Direction	Vector
Arm.WristPosition	Vector	Bone(PROXIMAL).Direction	Vector
Arm.Direction	Vector	Bone(METACARPAL).Direction	Vector
Direction.Pitch	Radian	JointPosition(TIP)	Vector
Direction.Yaw	Radian	JointPosition(DIP)	Vector
PalmNormal.Roll	Radian	JointPosition(PIP)	Vector
		JointPosition(MCP)	Vector

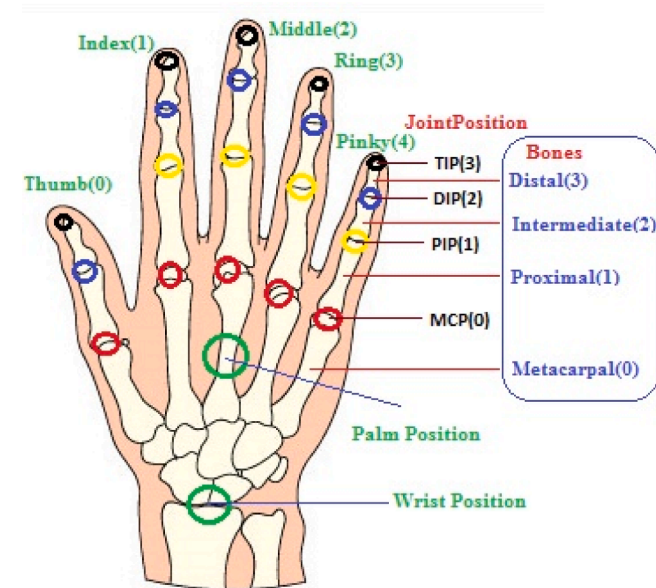


Fig. 6. Feature representation of hand and fingers supported by LMC API (LMC, 2019).

create the dataset, a data collection system was developed in C# language using LMC API version 4.0. Due to the hardware structure of the device, perception difficulties occur in cases such as overlapping and staying in the background, especially in double-handed use. For this reason, the average of 10 image frames taken from the device every 100 ms was recorded in order to reduce the negativity and differences in the data received during the gesture. These gestures were captured using LMC connected to the computer via USB. With this process, it is aimed to prevent such difficulties and to increase the accuracy and reliability of the dataset. In the process of making the gestures by the signers, daylight illumination was used without changing the lighting conditions. Conversion of cvs file into Matlab file format, normalization, feature selection, dimension reduction and classification stages of carried out using Matlab.

Appropriate feature extractions were performed for 50 dynamic words in Turkish Sign Language. The features have been determined by considering the construction state of each word and every point required. Hand and finger features supported by LMC API and used in this study are presented in Table 4.

First of all, the open-closed state feature information of 10 fingers was extracted. $IsExtended_set=\{f1, \dots, f10\}$

Then, distance information is consulted for fingertip points (TipPosition), palm point (PalmPosition) and finger joint points

Table 5
Unit of measure, type, names and number of identified features.

Measure	Type	Features	Piece
IsExtended	bool	Fingers	10
Distance	mm	TipPosition to TipPosition	15
		TipPosition to PalmPosition	8
		TipPosition to JointPosition	13
		JointPosition to JointPosition	7
Angle	radian	Fingers.Direction to Fingers.Direction	6
		Bone.Direction to Bone.Direction	9
		Fingers.Direction to Hands.Direction	4
		Fingers.Direction to PalmNormal	4
		Hands.Direction.Pitch,Yaw,Roll	6
Vector	float	Hands.Direction.PalmPositions	6
		Hands.Direction.PalmNormal	6
		Hands.Direction.Direction	6
		Arm.ElbowPosition	6
		Arm.WristPosition	6
		Arm.Direction	6

(JointPosition). A total of 44 features were determined between aforementioned 3 location points. $Distance_set=\{f11, \dots, f54\}$

Besides these, since finger measures and magnitudes differ, the finger direction (Direction), finger bone direction (Bone) as well as hand and palm direction angle information was needed. In addition, the angle information (pitch, yaw, roll) of the longitudinal, horizontal and vertical axis direction of the hand is also added. A total of 29 features have been extracted for angle information between each direction. $Angle_set=\{f55, \dots, f83\}$

To improve strengthen discrimination, some feature vectors were additionally used. These are vector information of the hand, palm center (PalmPosition) and direction (PalmNormal). In addition, the elbow position, wrist position and direction vector information of the arm are also used. A total of 36 features have been added for the vector direction information of the hand and arm. $Vector_set=\{f84, \dots, f119\}$

Feature representations of hand and fingers supported by the LMC API are presented in Fig. 6. Within the scope of this study, the features determined for the dynamic double-handed 50 words belonging to the TSL were grouped under 4 categories and 16 titles as mentioned above, and totally 119 features were determined to be used in the classification stage. The measurement unit, type, names and numbers of the features grouped under 4 categories and 16 titles are presented in Table 5.

5. Feature selection

Since sign language recognition is motion-based, 5 signal processing techniques are used for feature selection that will enable us to distinguish between different and similar gestures. These were determined as Fast Fourier Transform (FFT), Discrete Wavelet Transform (DWT), Root Squares Means (RMS), Standard Deviation (STD) and Mean (AVG). Our goal here is to select critical features using these signal processing techniques. FFT is used to extract the most common value occurrences in data and to select feature extractions that represent a gesture. FFT only gives frequency information, but we lose all temporal data (Coşkun & İstanbullu, 2012). DWT does not lose temporary data such as SWT. It gives intuition of both frequency and time information of our data. This gives us an idea about the most frequently encountered events together with the information when event occurred (Güler & Übeyli, 2004). RMS provides a single value by measuring the change magnitude of time series data, allowing better separation between data in data analysis. STD and AVG are similar to RMS in that time series data can be compressed into a single value and used as a feature.

Prior to the feature subset selection and feature matrix creation stages, the dataset is pre-processed. Transposition of each one word sample (12 × 119) is taken. Each transposed one word sample is combined and transformed into a (4760 × 12)dimensional matrix. It is transformed into a (19040 × 12)dimensional dataset for 4 signers and

Table 6
Number of peak value and feature matrix for 15 feature subset selections.

Technique	Number of row		Number of column		Feature matrix
	Number of signer	Number of sample	Number of peak value	Number of feature	
FFT	4	40	9	15	160 × 135
DWT	4	40	9	15	160 × 135
RMS	4	40	1	15	160 × 15
STD	4	40	1	15	160 × 15
AVG	4	40	1	15	160 × 15
Total number of feature selections and matrix				75	160 × 315

ready to use. To select the feature subset, the first algorithm whose pseudo code is given in Algorithm 1 is used. The outputs from the algorithm are a subset of the features for each technique. After calculating the content of each feature using each one technique (160 × 12), it is averaged (1 × 12) and combined (50 × 12). In the algorithm, the corre-

lation coefficient for FFT and DWT is used to determine which feature set distinguishes the gestures better. For STD, RMS and AVG techniques, a scalar value is returned instead of the vector output. All combinations of the feature vector (50 × 12) created for each feature are summed up by calculating the similarity criteria. The first 15, 10 and 5 feature subsets were selected from among the results for 119 features and used in the feature matrix.

In the second stage, firstly, the second algorithm whose pseudo code is given in Algorithm 2 is applied to the selected feature subset set and the set of peak values is determined. In the algorithm, nine peak values are determined at the end of sorting made from each of the feature subsets selected for FFT and DWT techniques, and one value for other techniques. The output is a feature matrix with rows specifying multiple actions of the same gesture for each gesture, and columns specifying different feature extracted by the techniques used. The number of peak values and feature matrix for 15 feature subset selections are presented in Table 6. The created feature matrix datasets are used in dimension reduction and classification stages.

Algorithm 1. Feature subset selection algorithm pseudo code

```

fname='f1,f2,...,f119'
numoffeature=119
For each technique (t=1:5)
    For each feature (f=1:119)
        For each gesture word (g=1:50)
            For each sample (s=1:160) //4x40
                r = numoffeature * s + f
                case 'FFT' : x1 = fft(content(r:end)) // 1x12
                case 'DWT' : x1 = dwt(content(r:end))
                case 'RMS' : x1 = rms(content(r:end))
                case 'STD' : x1 = std(content(r:end))
                case 'AVG' : x1 = avg(content(r:end))
            end
            x2=cat(1,x2,x1) // 160x12
        end
        x3=mean(x2) // 1x12
        x4 =cat(1,x4,x3) // 50x12 // MeanedValuePerFeature
    end
    [row,col] = size(x4)

    for each row (i=1 : x4-1)
        for each row+1 (j=i+1 : x4)
            if FFT or DWT
                coff=corrcoef(x4(i:end),x4(j:end))
                sum=sum+coff(1,2)
            elseif RMS or STD or AVG
                mnmx=min(x4(i:end),x4(j:end)) / max(x4(i:end),x4(j:end))
                sum=sum+mnmx(1,2)
            end,end,end

            corrCoffSum(f)=sum
        end
    end
sortedCoffSum = sort(corrCoffSum,DESC)
for n=1:15
    csvwrite (fname(find(corrCoffSum == sortedCoffSum(n))))
end
end

```

Algorithm 2. Pseudo-code of the peak set of values and feature matrix algorithm

```

numberOfPeakValues=9
numoffeature=119
For each gesture word      (g=1:50)
  For each technique      (t=1:5)
    For each select feature (f=1:15)
      For each sample      (s=1:160)
        r = numoffeature * s + f
        case FFT
          y1=fft (content(r:end))
          y1=sort(y1, DESC)
          y1=y1(:1,numberOfPeakValues)
        case DWT
          y1=dwt (content(r:end))
          y1=sort(y1, DESC)
          y1=y1(:1,numberOfPeakValues)
        case RMS :y1=rms(content(r:end))
        case STD  :y1=std(content(r:end))
        case AVG  :y1=avg(content(r:end))
        end
        y2=cat(1,y2,y1)
      end
      y3=cat(2,y3,y2)
    end
  end
end
csvwrite(y3)
end

```

6. Dimension reduction methods

Dimension reduction methods are used to remove unnecessary and dependent features to obtain a dataset with lower dimensions from a high dimensional dataset with a large number of features. In this study, PCA, LDA and PCA+LDA methods were used for dimension reduction in preprocessing stage for ELM machine learning applications.

6.1. Principal Component Analysis(PCA)

Principal Component Analysis (PCA) (Martinez & Kak, 2001) is a linear dimension reduction method that is frequently used in many fields such as statistics, machine learning, and data mining. PCA is a statistical method that uses orthogonal transformation to convert possible dependent variables of the dataset into linear independent variables called principal components. In the PCA method, for a d dimensional n sample dataset $X = \{x_1, x_2, \dots, x_n\}$, the covariance matrix is expressed in (1).

$$cov_{ij} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (1)$$

In this calculation, firstly, the average value of the data is subtracted from each one element and centering process is performed. In the next step, the covariance matrix is calculated to provide a measure of the correlation power between two or more sets of random variables. Then, the eigenvalues and eigenvectors of the covariance matrix are found by

the eigenvalue–eigenvector decomposition method. Next, a feature vector is created by selecting the components. Here, the eigenvector with the highest eigenvalue is determined as the fundamental component of the dataset. In the last stage, the new dataset whose dimension has been changed is obtained (Smith, 2002).

6.2. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a linear dimension reduction method that aims to find the vectors in the space where the classes are best decomposed by using the class information of data objects (Martinez & Kak, 2001). It is among the most commonly used methods with the PCA method. This method makes dimension reduction by maximizing the linear discrimination of groups belonging to different classes in the data. For the LDA method, inter-class scattering matrix S_B and intra-class scattering matrix S_W in dimension reduction process are defined as follows:

$$S_B = \sum_{i=1}^c \left(\bar{x}_i - \bar{x} \right) \left(\bar{x}_i - \bar{x} \right)^T \quad (2)$$

$$S_W = \sum_{i=1}^c \sum_{j=1}^{N_i} (x_{i,j} - \bar{x})(x_{i,j} - \bar{x})^T \quad (3)$$

First, the distance between the mean of the different classes, called the inter-class scattering matrix S_B , is calculated. In the next step, the

distance between the mean and sample of each class is calculated, called the intra-class scattering matrix S_w . Next, the eigenvectors for the scattering matrices and the corresponding eigenvalues are calculated. Then the eigenvalues are ordered and a new matrix is created by selecting the eigenvectors with the largest eigenvalues determined. In the last stage, the new dataset is obtained by taking the inner product of the matrix created with the dataset.

7. Extreme Learning Machines (ELM)

Recently, ELM has become the learning method of interest in the field of machine learning. Huang, Zhu, and Siew (2006) proposed the learning algorithm that they called the Extreme Learning Machine (ELM) in their study. ELM is a customized version of a Single-hidden Layer Feed forward Neural network (SLFN) whose input weights are randomly assigned but output weights are determined analytically. For this reason, the working principle of extreme learning machines is, to a certain extent, the same as the working principles of ANN. However, in extreme learning machines, the weights in the hidden layer are randomly assigned and these values are kept fixed during raining. On the other hand, the weights between the hidden layer and the output layer are determined analytically and quickly with the help of a linear model in one go (Tang, Deng, Huang, & Zhao, 2015). The most important feature of ELM is that the learning process takes place very quickly. In ELM, activation functions such as sigmoid, sine and Guassian are used in the hidden layer, while a linear function is used in the output layer (Huang, 2014). Experimental results have shown that this algorithm offers a good generalization performance. It has been stated that they can learn thousands of times faster than known algorithms for feed forward neural networks.

Cao, Lin, Huang, and Nan (2012) and Cambria et al. (2013) introduced the Kernel-based Extreme Learning Machine (KELM) by combining the kernel method and the extreme learning machine. The outputs of the hidden ELM layer can be regarded as nonlinear mapping of samples. The most popular kernel functions of KELM are Gauss and RBF. The Multi-Layer Extreme Learning Machine (ML-ELM) is a multi-layer ELM architecture based on the Extreme Learning Machine (ELM) for Single-hidden-Layer feeding Forward Neural networks (SLFNs) proposed by Cambria et al. (2013). At the same time, ML-ELM is a kind of deep neural network that not only performs a complex function but also does not require iteration in the training process. In this architecture, firstly the parameters in each layer are obtained by unsupervised training, then the network is fine-tuned with supervised learning. ML-ELM was recommended by Kasun et al. (2013). Another ML-ELM uses unsupervised learning to train the parameters in each layer. Besides, unlike deep networks, ML-ELM does not require fine tuning. Therefore, compared to other deep learning algorithms, ML-ELM does not need to spend long time on network training. ML-ELM hidden layer activation functions can be linear or any non-linear function.

Inspired by kernel learning, a kernel model of ML-ELM was developed and a Multi-Layer Kernel-based ELM (ML-KELM) was proposed in Wong, Vong, Wong, and Cao (2017). ML-KELM does not require tuning parameters for all layers like other ELM models. ML-KELM learns the optimal model at one time under fixed parameters. In summary, the proposed ML-KELM is a kernel version of the ML-ELM by stacking

Table 7
Number of function, layer and cell used in classifiers.

Classifier	Number of layer	Number of cell	Function
ELM	1	100	sigmoid
MLP-ELM	3	100	sigmoid
ML-ELM	3	100	sigmoid
KELM	1	-	RBF_kernel
ML-KELM	3	-	RBF_kernel

multiple KELM-AEs with less user intervention and eliminating many drawbacks compared to other ELM models. ML-KELM and KELM-AE methods are explained in detail with the algorithm given pseudo code in Algorithm 3.

Algorithm 3. Pseudo code of ML-KELM and KELM-AE algorithms

Input: Input matrix $X^{(i)}$,
 Regularization C_i ,
 Kernel parameter σ_i ,
 Activation function g_i ,
 Output matrix T ,
 Number of layers N_{Layer}
Output: Output weight β ,
 Kernel matrix Ω ,
 New data representation $X^{(i+1)}$,
 Transformation matrix $T^{(i)}$,
 Representation matrix X^{final} ,
 Two transformation matrix $T^{(1)}$ and Γ
Algorithm KELM-AE for the i th Layer
 Step 1: Calculate $\Omega_{k,j}^{(i)} \leftarrow K(x_k, x_j, \sigma_i)$
 Step 2: Calculate $T^{(i)} \leftarrow (I / C_i + \Omega^{(i)})^{-1} X^{(i)}$
 Step 3: Calculate $X^{(i+1)} \leftarrow g_i(X^{(i)} T^{(i)T})$
return $X^{(i+1)}$, and $T^{(i)}$
Algorithm ML-KELM
 Step 1: Initialize $\Gamma \leftarrow \Gamma^{(1)}$
for $i = 1$ to $N_{Layer} - 1$ **do**
 Step 2: Calculate $X^{(i+1)}, T^{(i)} \leftarrow \text{KELMAE}(X^{(i)}, C_i, \sigma_i, g_i)$
 Step 3: Update $\Gamma \leftarrow T^{(i)} \Gamma$
 Step 4: $i \leftarrow N_{Layer}$, $X^{final} \leftarrow X^{(i)}$
 Step 5: Calculate $\Omega_{k,j}^{(i)} \leftarrow K(x_k^{(i)}, x_j^{(i)}, \sigma_i)$
 Step 6: Calculate $\beta \leftarrow (I / C_i + \Omega^{(i)})^{-1} T$
return $X^{final}, T^{(1)}, \Gamma$, and β

In this study; basic ELM, MLP-ELM (MLP-ELM, 2019), ML-ELM, KELM and ML-KELM are used as classifier in the proposed sign language recognition model. In basic ELM, MLP-ELM, and ML-ELM classifiers, the cell number was 100 and the sigmoid activation function was used. In multi-layer classifiers, the number of layers is 3. In kernel-based KELM and ML-KELM classifiers, the **RBF kernel** function was preferred. There is no need to use cell number in KELM and ML-KELM. Function, layer and cell numbers used in classifiers are presented in Table 7.

8. Experimental results

In the evaluation of the performance rates, 3-phase feature selection and dimension reduction process was performed for the original dataset consisting of 8000 samples created by 4 signers. First of all, 3 separate sub-sets for critical features were determined by using the first feature selection algorithm given in algorithm 1. The number of features has been gradually reduced in the selection of critical features. Using the first feature selection algorithm, among of 119 features, 15 features were selected for the first sub-set for each technique mentioned in Section 5, 10 features for the second sub-set and 5 features for the third sub-set, and a total of 75, 50 and 25 features were selected for 5 techniques, respectively. With the second feature selection algorithm given in algorithm 2, 3 datasets belonging to the determined feature subsets were created. 9 new datasets were created by reducing the dimensions of these 3 datasets with PCA, LDA and PCA+LDA methods. A total of 12 datasets, together with the dimensionally-reduced datasets, were used in the classification process. It was aimed to compare the performance rates of datasets obtained by feature selection and dimension reduction methods and to determine the feature subset with the highest performance rate. The formation stages of 12 datasets are presented in Table 8.

The success rates of the dataset containing 75 feature and the datasets created by using 3 dimension reduction method were examined in the first phase by using the ELM-based and classical k-NN and SVM classifiers in machine learning. This phase consists of 4 different applications. The performance rates and time of ELM-based and classical classifiers for these datasets are presented in Table 9. The first dataset

Table 8
The formation stages of 12 dataset.

First stage	Second stage	Third stage
First application	Feature selection application (5 technique × 15 feature) Containing 75 feature dataset	Dimension reduction application PCA (33 feature) LDA (16 feature) PCA+LDA (15 feature)
Original dataset (119 feature)	(5 technique × 10 feature) Containing 50 feature dataset	PCA (29 feature) LDA (16 feature) PCA+LDA (15 feature)
	(5 technique × 5 feature) Containing 25 feature dataset	PCA (17 feature) LDA (12 feature) PCA+LDA (11 feature)

Table 9
Classification success rates and times of datasets containing 75 features.

Classifier	1.PHASE															
	1.Application (75 features)				2.Application PCA (33 features)				3.Application LDA (16 features)				4.Application PCA+LDA (15 features)			
	Training		Testing		Training		Testing		Training		Testing		Training		Testing	
	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)
k-NN	0,069		0,432	98	0,084		0,013	98	0,085		0,013	98	0,083		0,013	97
SVM	22,70		2,188	98	9,620		1,135	98	9,563		1,126	98	9,544		1,124	98
ELM	0,105	97	0,005	96	0,089	96	0,004	96	0,079	96	0,004	95	0,081	95	0,004	95
KELM	4,733	98	0,220	98	4,789	96	0,164	95	4,634	90	0,153	89	4,622	89	0,151	89
MLP-ELM	0,112	94	0,006	93	0,109	95	0,006	95	0,102	94	0,005	94	0,101	94	0,005	94
ML-ELM	0,167	98	0,006	98	0,155	97	0,006	97	0,142	95	0,005	95	0,140	94	0,005	94
ML-KELM	56,88	99	2,375	98	55,20	99	2,442	98	55,28	99	2,462	98	55,57	99	2,355	98

Table 10
Classification success rates and times of datasets containing 50 features.

Classifier	2.PHASE															
	1.Application (50 features)				2.Application PCA (29 features)				3.Application LDA (16 features)				4.Application PCA+LDA (15 features)			
	Training		Testing		Training		Testing		Training		Testing		Training		Testing	
	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)
k-NN	0,069		0,360	98	0,083		0,014	97	0,082		0,013	97	0,082		0,012	97
SVM	20,61		2,16	98	9,540		1,151	98	9,528		1,137	97	9,502		1,133	97
ELM	0,092	97	0,005	96	0,008	97	0,003	96	0,080	97	0,004	96	0,081	96	0,004	96
KELM	4,842	97	0,216	97	4,552	95	0,154	95	4,456	93	0,153	92	4,616	91	0,151	91
MLP-ELM	0,110	96	0,006	95	0,106	96	0,006	96	0,102	96	0,005	96	0,102	96	0,005	95
ML-ELM	0,167	98	0,006	98	0,147	97	0,005	97	0,141	96	0,005	95	0,140	95	0,005	95
ML-KELM	56,84	99	2,410	98	55,47	99	2,387	98	54,97	99	2,399	98	55,02	99	2,369	98

with 75-feature and the other datasets, whose features were reduced to 33, 16 and 15 by PCA, LDA, and PCA+LDA dimension reduction methods respectively. Among all datasets, the highest classification success rate for 50 dynamic words was obtained with ML-KELM classifier. Furthermore, the ML-KELM classifier preserved its performance rate although the number of features reduced sequentially in applications. In addition, the success rate of classical k-NN and SVM classifiers similar to ML-KELM algorithm was obtained. The classifier with the least

time in terms of training and testing times is ELM. Considering the running times of the algorithms, although the number of features was reduced, a slight decrease in the running time was observed.

In the second phase, the success rates of the datasets created by using 50 feature and 3 dimension reduction methods were examined. This phase consists of again 4 applications. The performance rates and time of ELM-based and classical classifiers for these datasets are presented in Table 10. The first dataset with 50-feature and the other datasets, whose

Table 11
Classification success rates and times of datasets containing 25 features.

3.PHASE																
Classifier	1.Application (25 features)				2.Application PCA (17 features)				3.Application LDA (12 features)				4.Application PCA+LDA (11 features)			
	Training		Testing		Training		Testing		Training		Testing		Training		Testing	
	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)	Time (s)	Accuracy(%)
k-NN	0,068		0,171	96	0,087		0,015	94	0,082		0,012	93	0,081		0,012	91
SVM	15,636		1,691	96	9,660		1,170	95	9,638		1,164	94	9,639		1,165	93
ELM	0,086	96	0,004	96	0,081	95	0,004	95	0,080	94	0,004	94	0,081	94	0,004	94
KELM	4,579	93	0,183	93	4,471	89	0,152	88	4,647	81	0,151	81	4,579	78	0,151	78
MLP-ELM	0,108	95	0,006	95	0,105	94	0,005	94	0,110	94	0,005	94	0,104	94	0,005	94
ML-ELM	0,153	96	0,005	96	0,136	92	0,005	91	0,144	88	0,005	88	0,145	85	0,005	85
ML-KELM	56,42	99	2,550	99	55,01	98	2,369	98	55,38	98	2,437	98	54,44	98	2,348	98

features were reduced by using PCA, LDA, and PCA+LDA dimension reduction methods to 29, 16, and 15, respectively, were subjected to the same classification algorithms. Among all datasets, the highest classification success rate for 50 dynamic words was obtained with ML-KELM classifier. Furthermore, the ML-KELM classifier preserved its performance rate although the number of features reduced sequentially in applications. In addition, the success rate of classical k-NN and SVM classifiers similar to ML-KELM algorithm was obtained. The classifier with the least time in terms of training and testing times is ELM. Considering the running times of the algorithms, although the number of features was reduced, a slight decrease in the running time was observed.

In the third phase, the success rates of the dataset containing 25 feature and the datasets created by using 3 dimension reduction method were examined. This phase consists of 4 applications. The performance rates and time of ELM-based and classical classifiers for these datasets are presented in Table 11. The first dataset with 25-feature and the others, whose features were reduced by using PCA, LDA, and PCA+LDA dimension reduction methods to 17, 12, and 11, respectively, were subjected to the same classification algorithms. Among all datasets, the highest classification success rate for 50 dynamic words was obtained with ML-KELM classifier. Furthermore, the ML-KELM classifier preserved its performance rate although the number of features reduced sequentially in applications. In addition, classical k-NN and SVM classifiers could not maintain their success rates according to ML-KELM algorithm. The classifier with the least time in terms of training and testing times is ELM. Considering the running times of the algorithms, although the number of features was reduced, a slight decrease in the running time was observed.

Considering the applications at all phase, it is observed that, among all classifiers, ML-KELM classifier maintains its performance rates and gives the best performance rates. On the other hand, the ML-KELM classifier, which offers less user intervention, appears to have a more stable success rates. But it has the largest elapsed time for the both training and testing. The reason for this difference in computing and run-time is the difference in the computational style used in this architecture as well as the architecture. That is to say: During the ML-KELM training, it creates a square kernel matrix in the size of the sample number by creating nodes as many as the number of samples used in training for each layer. In the test phase, this process is done once. Therefore, relatively large size matrix operations take a long time. It can be seen from the results given above that this effect is also valid for KELM. However, it is obvious that the briefly mentioned form of calculation brings a more robust and stable generalization capability

compared to other structures. Besides, it was observed that k-NN and SVM classifiers could not maintain their success rates and decreased. In addition, due to the different structure of the algorithms, there were differences in the running times. When we compare PCA, LDA and PCA+LDA applied datasets in terms of performance, running times and number of features, it is seen that PCA+LDA dimension reduction method is successful on classification algorithms. In addition, it can be said that the performance rates of the KELM classifier decrease on dimension-reduced datasets. By performing two-step (feature selection and dimension reduction), reducing the number of features and reducing the dimension, the datasets created offer advantages in terms of calculation, time, cost, storage and performance.

9. Discussion and conclusion

In the study, a new approach style with dynamic word recognition model is presented. We have put forward a dynamic hand gesture recognition approach by utilizing sequential data captured with the LMC. LMC has been preferred over similar devices because of its advantages in many aspects. In this way, the LMC device will make it easier for the hearing impaired to communicate easily and independently with the environment using dynamic sign language recognition. The proposed model is robust as it efficiently captures sign inputs using different and similarly structured words. Similar studies in the literature are mostly on the recognition of static word signs. In this study, an important step was taken in recognizing dynamic word gestures performed with two hands in Turkish Sign Language (TSL) with ELM based LMC. For dynamic gestures, we used different classification methods and compared performance of them. We have collected a sign language dataset consisting of 50 double-handed dynamic sign words of TSL. Using this dataset, we have investigated many aspects of word recognition performance by reducing the number and dimension of the dataset. The recognition has been carried out by different classification types, where the best accuracy of 98% has been recorded using ML-KELM approach. With the strategy developed in this study, high classification success was achieved with feature selection and dimension reduction methods. The dataset forming of signs according to the specified feature extractions and selections can also be easily adapted to other sign languages.

In the future study of ours, it is anticipated that the problem of recognizing dynamic gestures signs outside the field of view of the LMC can be overcome with the Kinect device. At the same time, the method will be extended towards complete sign sentence recognition. In addition to that, the framework can be extended by capturing facial

expressions and lip gestures for further improvement.

CRedit authorship contribution statement

Zekeriya Katılmış: Conceptualization, Methodology, Software, Data curation, Formal analysis, Writing - original draft, Investigation, Visualization. **Cihan Karakuzu:** Conceptualization, Methodology, Supervision, Writing - review & editing, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Aliyu, S., Mohandes, M. & Deriche, M. (2017). Dual LMCs Fusion for Recognition of Isolated Arabic Sign Language Words, 14th International Multi-Conference on Systems, Signals & Devices (SSD). <https://doi.org/10.1109/SSD.2017.8167010>.
- AL-Rousan, M., Assaleh, K., & Tala'a, A. (2009). Video-based signer-independent Arabic sign language recognition using hidden Markov models. *Applied Soft Computing*, 9(3), 990–999. <https://doi.org/10.1016/j.asoc.2009.01.002>
- Anwar, A., Basuki, A., Sigit, R., Rahagiyanto, A. & Zikky, M. (2017). Feature Extraction for Indonesian Sign Language (SIBI) Using Leap Motion Controller, 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand. <https://doi.org/10.1109/ICSEC.2017.8443926>.
- Avola, D., Bernardi, M., & Cinque, L. (2019). Exploiting recurrent neural networks and LMC for the recognition of sign language and semaphoric hand gestures. *IEEE Transactions on multimedia*, 21(1). <https://doi.org/10.1109/TMM.2018.2856094>
- Cambria, E., Huang, G. B., Kasun, L. L. C., et al. (2013). Extreme learning machines. *IEEE Intelligent Systems*, 28(6), 30–59. <https://doi.org/10.1109/MIS.2013.140>
- Cao, J., Lin, Z., Huang, G. B., & Nan, L. (2012). Voting based extreme learning machine. *Information Sciences*, 185(1), 66–77. <https://doi.org/10.1016/j.ins.2011.09.015>
- Chophuk, P., Pattanaworapan, K. & Chamnongthai, K. (2018). Fist American Sign Language Recognition Using Leap Motion Sensor, International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand. <https://doi.org/10.1109/IWAIT.2018.8369790>.
- Chuan, C., Regina, E. & Guardino, C. (2014). American Sign Language Recognition Using Leap Motion Sensor, 13th International Conference on Machine Learning and Applications, Detroit, MI, USA. <https://doi.org/10.1109/ICMLA.2014.110>.
- Coşkun, M., & İstanbullu, A. (2012). EEG İşaretlerinin FFT ve dalgacık dönüşümü ile analizi, *Akademik Bilişim '12 - XIV. Akademik Bilişim Konferansı*, 323–328.
- Demircioğlu, B., Bülbül, G. N. & Köse, H. (2016 a). Recognition of Sign Language Hand Shape Primitives with Leap Motion, 7th workshop on the Representation and Processing of Sign Languages: Corpus Mining, 10th edition of the Language Resources and Evaluation Conference (LREC), Vol.1, No. 1, pp. 47-52, Portoroz, Slovenia. <https://doi.org/10.13140/RG.2.1.1908.6968>.
- Demircioğlu, B., Bülbül, G., & Köse, H. (2016 b). Turkish Sign Language Recognition with Leap Motion, 24th Signal Processing and Communication Application Conference (SIU), Zonguldak. <https://doi.org/10.1109/SIU.2016.7495809>.
- Deriche, M., Aliyu, S. O., & Mohandes, M. (2019). An intelligent Arabic Sign language recognition system using a pair of LMCs with GMM based classification. *IEEE Sensors Journal*, 19(18), 8067–8078. <https://doi.org/10.1109/JSEN.736110.1109/JSEN.2019.2917525>
- Eqab, A. & Shanableh, T. (2017). Android Mobile App for Real-Time Bilateral Arabic Sign Language Translation Using Leap Motion Controller, International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates. <https://doi.org/10.1109/ICECTA.2017.8251936>.
- Güler, İ. & Übeyli, E.D. (2004). Dalgacık Dönüşümünün Kullanımı ile Teşhis Sistemleri için Öznitelik Çıkarma, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu.
- Hisham, B., & Hamouda, A. (2017). Arabic static and dynamic gestures recognition using leap motion. *Journal of Computer Science*, 13(8), 337–354. <https://doi.org/10.3844/jcssp.2017.337.354>
- Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3), 489–501. <https://doi.org/10.1016/j.neucom.2005.12.126>
- Huang, G.-B. (2014). An insight into extreme learning machines: Random neurons, random features and kernels. *Cognitive Computation*, 6(3), 376–390. <https://doi.org/10.1007/s12559-014-9255-2>
- Insani, C. N., Nurtanio, I. & Ilham, A. A. (2019). The effect of light on Leap Motion Controller in the classification of Sign Language Translator System, International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, Indonesia. <https://doi.org/10.1109/ISRITI48646.2019.9034602>.
- Karak, A., Akyol, K. & Gültepe, Y. (2018). Turkish Sign Language Alphabet Recognition with Leap Motion, International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18), May 11–13, Safranbolu, Turkey.
- Kasun, L. L. C., Zhou, H., Huang, G. B., et al. (2013). Representational learning with ELMs for big data. *IEEE Intelligent Systems*. <https://doi.org/10.1109/MIS.2013.140>
- Katılmış, Z. & Karakuzu, C. (2020). Recognition of Two-Handed Posture Finger Turkish Sign Language Alphabet, 5th International Conference on Computer Science and Engineering (UBMK), Diyarbakır, Turkey. <https://doi.org/10.1109/UBMK50275.2020.9219364>.
- Khotimah, W. N., Saputra, R. A., Suciati, N. & Hariadi, R. R. (2017). Alphabet Sign Language Recognition Using Leap Motion Technology and Rule Based Backpropagation-Genetic Algorithm Neural Network (RBBPGANN), JUTI – Volume 15, Number 1, 95 – 103. <https://doi.org/10.12962/J24068535.V15I1.A639>.
- Kosmidou, V. E., Hadjileontiadis, L. J., & Panas, S. M. (2006). Evaluation of surface EMG features for the recognition of American Sign Language gestures, Proceedings of the 28th IEEE EMBS Annual International Conference New York City, USA. <https://doi.org/10.1109/IEMBS.2006.259428>.
- Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P. (2017a). A multimodal framework for sensor based sign language recognition. *Neurocomputing*, 1–18. <https://doi.org/10.1016/j.neucom.2016.08.132>
- Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P. (2017b). Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, 86, 1–8. <https://doi.org/10.1016/j.patrec.2016.12.004>
- LMC. (2019). Leap Motion Controller C# SDK Documentation. Retrieved from <https://developer-archive.leapmotion.com/documentation/csharp/index.html>. Accessed January 11, 2020.
- Mapari, R. B. & Kharat, G. (2015). Real Time Human Pose Recognition Using Leap Motion Sensor, International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India. <https://doi.org/10.1109/ICRCICN.2015.7434258>.
- Mapari, R. B., & Kharat, G. (2017). Analysis of multiple sign language recognition using leap motion sensor. *International Journal of Research in Advent Technology (IJRAT)*.
- Martinez, M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 228–233. <https://doi.org/10.1109/34.908974>
- Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal*, 19(16), 7056–7063. <https://doi.org/10.1109/JSEN.736110.1109/JSEN.2019.2909837>
- MLP-ELM. (2019). Multilayers perceptron based Extreme Learning Machine. Retrieved from <https://www.mathworks.com/matlabcentral/fileexchange/69813-multilayers-perceptron-based-extreme-learning-machine>. Accessed January 13, 2021.
- Mohandes, M., Aliyu, S. & Deriche, M. (2014). Arabic Sign Language Recognition using the Leap Motion Controller, 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey. <https://doi.org/10.1109/ISIE.2014.6864742>.
- Mohandes, M., Aliyu, S. & Deriche, M. (2015). Prototype Arabic Sign Language Recognition using Multi-Sensor Data Fusion of Two Leap Motion Controllers, 12th International Multi-Conference on Systems, Signals & Devices. <https://doi.org/10.1109/SSD.2015.7348113>.
- Naglot, D., & Kulkarni, M. (2016). Real Time Sign Language Recognition using the Leap Motion Controller. In *International Conference on Inventive Computation Technologies (ICICT)*. <https://doi.org/10.1109/INVENTIVE.2016.7830097>
- Naidui, C., & Ghotkar, A. (2016). Hand gesture recognition using leap motion controller. *International Journal of Science and Research (IJSR)*, 5(10). ISSN (Online) 2319-7064.
- Oz, C., & Leu, M. C. (2011). American Sign Language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7), 1204–1213. <https://doi.org/10.1016/j.engappai.2011.06.015>
- Pramunanto, E., Sumpeno, S. & Legowo, R. S. (2017). Classification of Hand Gesture in Indonesian Sign Language System using Naive Bayes, International Seminar on Sensor, Instrumentation, Measurement and Metrology (ISSIMM), Surabaya, Indonesia. <https://doi.org/10.1109/ISSIMM.2017.8124288>.
- Smith, Lindsay I. (2002). A tutorial on principal components analysis, Technical Report OUCS-2002-12, pp. 1–26.
- Tang, J., Deng, C., Huang, G. B., & Zhao, B. (2015). Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3), 1174–1185. <https://doi.org/10.1109/TGRS.2014.2335751>
- TİD Türk İşaret Dili Sözlüğü. <https://www.isaretce.com>. 2019. (Accessed 1 October 2020).
- Tiwari, V., Anand, V., Keskar, A. G. & Satpute, V. R. (2015). Sign Language Recognition Through Kinect Based Depth Images and Neural Network, International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India. <https://doi.org/10.1109/ICACCI.2015.7275608>.
- Wibowo, M. D., Nurtanio, I. & Ilham, A. A. (2017). Indonesian sign language recognition using leap motion controller, 11th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia. <https://doi.org/10.1109/ICTS.2017.8265648>.
- Wong, C. M., Vong, C. M., Wong, P. K., & Cao, J. (2017). Kernel-based Multi-layer extreme learning machine for representation learning. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2016.2636834>