Real-Time Turkish Sign Language Recognition Using Cascade Voting Approach with Handcrafted Features

Abdulkadir Karacı^{1*}, Kemal Akyol², Mehmet Ugur Turut³

¹⁻³ *Kastamonu University, Faculty of Engineering and Architecture, Computer Engineering, Kastamonu 37200, Turkey*

Abstract – In this study, a machine learning-based system, which recognises the Turkish sign language person-independent in real-time, was developed. A leap motion sensor was used to obtain raw data from individuals. Then, handcraft features were extracted by using Euclidean distance on the raw data. Handcraft features include finger-to-finger, finger-to-palm, finger-to-wrist bone, palm-to-palm and wrist-to-wrist distances. LR, k -NN, RF, DNN, ANN single classifiers were trained using the handcraft features. Cascade voting approach was applied with two-step voting. The first voting was applied for each classifier's final prediction. Then, the second voting, which voted the prediction of all classifiers at the final decision stage, was applied to improve the performance of the proposed system. The proposed system was tested in real-time by an individual whose hand data were not involved in the training dataset. According to the results, the proposed system presents 100 % value of accuracy in the classification of one hand letters. Besides, the recognition accuracy ratio of the system is 100 % on the two hands letters, except "J" and "H" letters. The recognition accuracy rates were 80 % and 90 %, respectively for "J" and "H" letters. Overall, the cascade voting approach presented a high average classification performance with 98.97 % value of accuracy. The proposed system enables Turkish sign language recognition with high accuracy rates in real time.

Keywords – Cascade voting, handcrafted features, machine learning, sign language recognition, Turkish sign language.

I. INTRODUCTION

Individuals with speaking and hearing impairments use body and hand gestures to communicate with other individuals. These signals using gestures and facial expressions are named Sign Language (SL) [1]. SL is the primary means of communication for hearing-impaired people [2]. Most of the countries have their own SL different from the other sign languages. In Turkish Sign Language (TSL), gestures are used for common words and the letters in the Turkish alphabet. Whereas SL is affected by spoken language, it has a distinct grammatical structure. A relation between TSL and Turkish cannot be taken for granted [3].

Automatic hand posture recognition is an important research topic in computer science and there are a large number of

applications in this area. Some of them include driving support via manually-controlled cockpit instruments, gesture-based consumer electronics and house automation, gaming industry applications, interaction with virtual objects and supporting technologies for disabled individuals [4].

For SL recognition, researchers mostly employed different sensors such as Kinect, Leap Motion (LM), data gloves, accelerometers and depth camera. LM sensor uses two infra-red IR cameras and three infra-red LEDs to observe a hemispheric area approximately with one meter distance. It was especially designed for recognising hand and finger movements. The device enables direct access to different parts, such as points of fingertips, the position of the palm and the direction of the palm [5], [6]. This device has a narrower field of vision than Kinect. Therefore, its function is limited to obtaining hand and finger movements [5]. Kinect was designed to track various movements performed by individuals. A coloured camera attached to the device transmits the obtained scene, depth presentation and the skeletal structure perceived by it to the computer [1]. Data gloves involve attaching sensors such as flex sensors, gyro sensor and accelerometer on a wearable glove and enable sign language recognition [7].

In this study, Turkish sign language recognition system was developed in real time. A core benefit of the system is person-independent running. LM sensor presents raw data obtained from individuals. Handcraft features were extracted from the raw data by using Euclidean distance. Logistic Regression (LR), K -Nearest Neighbourhood (k -NN), Random Forest (RF), Deep Neural Network (DNN), Artificial Neural Network (ANN) single classifiers were used in machine learning phase, and also cascade voting approach was used to improve the system performance.

The main contributions of this paper are as follows:

1. A hybrid machine learning mechanism (including two voting methods) has been proposed for Turkish sign language alphabet recognition.
2. Handcrafted features, which are person-independent and also hand position-independent, were extracted.

* Corresponding author's e-mail: akaraci@kastamonu.edu.tr

3. Experimental validation was performed in real time by a person whose sign language data was not included in the training dataset.
4. Higher classification performance was obtained with the cascade voting approach.
5. The proposed system recognises all of the one-hand letters and especially many of the two-hands letters in Turkish sign language alphabet with a high accuracy value in real time.

In the second section, related studies are discussed. In the third section of the article, data collection, handcrafted feature extraction method, cascade voting approach are presented. In the fourth section, the experimental study and its results are explained in detail. The results are summarised in the fifth section.

II. RELATED STUDIES

The number of studies conducted on TSL is rather limited. In addition, these studies do not involve real-time systems, i.e., they are offline. The sign language alphabets of many countries contain letters mostly presented by one-handed. In the Turkish Sign Language alphabet, most of the letters (22) consist of two-handed letters. This, in turn, significantly affects the performance of machine learning models. There is one notable study, which employed an LM sensor. Demircioglu et al. [8] managed the offline recognition of the Turkish sign language alphabet using Heuristic, RF and multi-layer perceptron (MLP) methods and LM sensor. They obtained high classification accuracy of 99.03 % using the Heuristic method. There are also TSL recognition studies using a Kinect sensor. Oktekin and Cavus [6] used this device to develop a system that would recognise in real time 33 sign language expressions performed using one hand. The accuracy rate of the system developed in this study was 82 %. Kakayev and Albayrak [9] recorded the skeletal structure using Kinect and based on these data developed another system that could recognise 15 and 18 TSL words offline. The study presented maximum accuracy for 15-word recognition by 98 %. Memis and Albayrak [10] developed a system that recognized 111 TSL words in 3 different classes with the k -NN machine learning method employing scenes provided by the Kinect sensor. They tested the recognition system using the 3-fold method and were able to achieve 91.52 % recognition accuracy rate. Unutmaz et al. [1] designed a system that recognised through a Convolution Neural Network (CNN) 9 different sign language expressions performed by 12 different individuals and collected via Kinect. The results were evaluated offline by setting the test data rate at 5 %, 10 % and 20 %. The highest recognition rate was obtained using 20 % test data rate by 99.25 %. Haberdar and Albayrak [3] recognised TSL expressions in real time using Hidden Markov Model (HMM) from video camera images. For 22 one-handed and 28 two-handed expressions, the accuracy rate was found to be 95.7 %. Yalcinkaya et al. [11] used the Motion History Images method for representing the expressions on the training data that comprised 8 different sign language videos. They performed the classification process at 95 % accuracy rate with k -NN. Ezel and Baykan [12] used CNN for the recognition

of 28 images of sign language alphabet. For the performance test, 3-fold cross validation was used and the accuracy was 84.48 %. Aktas et al. [13] managed to recognise 5 different sign language images that involved facial expressions and head movements with CNN. They evaluated the performance with 5-fold cross validation and obtained 78.49 % accuracy. Overall, a review of the literature reveals that the majority of the Turkish sign language recognition studies involved mostly offline systems and focused on the Kinect and video images.

There are also a number of studies on the sign languages of countries other than Turkey. These studies mostly involve offline classification and the use of Kinect. Kumar et al. [5] proposed a study that used Kinect and LM sensors for Isolated Sign Language Recognition. In this study, they extracted a feature set using the data they obtained with the two different images received from the positions of fingers and palm. They employed the Hidden Markov Model (HMM) and the Bidirectional Long Short-Term Memory Neural Network-based (BLTSM-NN) classifiers for sign recognition and evaluated the results they obtained jointly for increasing the recognition performance. The study was tested offline with 50 different sign words from the Indian Sign Language (ISL) and 97.85 % accuracy was obtained. Kumar et al. [2] used LM and Kinect sensors to present a new Sign Language Recognition System, which combined signal gestures and facial expressions. In their study, the authors recorded the hand signal gestures with LM and captured the signer's facial data with Kinect to perform recognition on a training dataset comprising 51 dynamic sign words using HMM. Afterwards, they aimed at improving the recognition performance by combining the separate recognition results of different methods with the Independent Bayesian Classifier Combination. The recognition rates were 96.05 % and 94.27 % for one-handed and two-handed gestures, respectively. Almeida, Guimarães and Ramírez [14] investigated the phonological structure of the Brazilian Sign Language and obtained features of intensity, position and depth from the RGB-D (Kinect) sensor. They analysed the relation between structural elements such as shape, movement and position and the extracted features. They also managed to classify offline the signs according to these features and linguistic elements with an average 80 % accuracy rate using support vector machine (SVM). To recognise the 24 signs from the American Sign Language, Tao et al. [15] presented a model of the CNN using the multiview augmentation of the images captured by the Kinect and inference fusion. The model they proposed enabled successful training that decreased the overfitting. Different tests revealed the accuracy rates of 100 % and 93 %. Sidiq et al. [16] normalized the features extracted from the sign images using Fourier, Hartley and Log-Gabor transformations and investigated the performance of the KNN, SVM, and MLP classifier algorithms. The data obtained for the 23 signs in the Arabic Sign Language were divided to test and training data. The system performance was tested, which indicated that the highest accuracy rate was 99 %. Almasre and Al-Nuaim [17] proposed a Kinect-based prototype model for recognition of specific Arabic Sign Language hand gesture words. This model obtained 83 % offline classification

accuracy using SVM, RF and KNN algorithms. Mohandes et al. [18] performed recognition of 28 Arabic Sign Language expressions using Naive-Bayes (NB) and MLP neural networks. The dataset was obtained with Leap Motion. The training and testing were performed on the offline data by 5-fold cross validation. The highest accuracy rate was 99.1 %.

III. THE PROPOSED SYSTEM

This section presents a detailed information about the proposed model developed for obtaining the sign language data and the real-time recognition of sign language alphabet. Using the raw data captured by the LM sensor does not allow for sufficient recognition performance. Therefore, a new mathematical model was proposed in this study for extracting handcraft features. The details of this mathematical model are presented in this section. In addition, machine learning algorithms used for real-time recognition of Turkish Sign Language alphabet are explained in detail.

Figure 1 shows an overview of the proposed model. The first phase of the development of the sign language alphabet recognition system involved a set of experimental studies to render the raw data obtained by the LM sensor meaningful.

Fig. 1. Overview of the proposed model: (a) training of the models; (b) real-time prediction.

As a result of these experimental studies, the handcraft feature extraction, the mathematical model of which is explained in Section 3.2, was performed and a new dataset was produced that contained these features. Then, LR, k -NN, RF, DNN and ANN machine learning algorithms were trained using the handcrafted feature vector. For classifier algorithms to be able to classify real-time sign data, the training phase should be conducted beforehand. With this aim, firstly a dataset

containing approximately 3000 samples for each sign was obtained. The entire dataset was used for training. The handcrafted features extracted from the data obtained real time with the LM sensor were fed into the trained machine learning models as input for conducting the test phase. In this approach, the first voting process (Voting-1) was applied to obtain the final predictions of the individual classifiers, while the second voting process (Voting-2), which used predictions coming from Voting-1, was applied to obtain the prediction of the proposed model. Detailed information is given in the sub-sections.

A. Dataset

Dataset Preparation:

All coding was carried out with Python language. The raw dataset was obtained by transferring the data captured in real time by the LM sensor into the computer environment with the WebSocket method and in JSON format. LM sensor produces the hand data in a three-dimensional space (x , y , z) and virtualizes as seen in Fig. 2. This virtualization is the expression of how the device perceives the hand. The green ball-shaped figures illustrate the locations, in which it is possible to collect data from the device.

Fig. 2. Virtualization of Leap Motion data.

There are 29 letters in Turkish Sign Language alphabet. Each letter in the raw dataset was represented with data collected for 10 seconds with 5 participants in experimental studies. The frame per second (FPS) value was 60 during data capture. In this respect, 3000 raw data were obtained for each letter. The total dataset obtained for the study consists of 87 000 rows. As the raw data obtained from a single hand contains 195 features, there are a total of 390 features for two hands [19]. The information such as start and finish positions of finger distal, intermediate, proximal and metacarpal bones, directions of bones, palm position, arm direction, wrist position and elbow position were captured in x , y , and z axes. The 3-D position coordinates in the Hand Hierarchy of LM sensor are presented in Fig. 3.

Fig. 3. Positional data obtained from LM [20].

The sign language alphabets of many countries, including USA, Japan and Ukraine, contain letters mostly presented by single hand. On the contrary, Turkish sign language alphabet contains 22 two-handed letters and only 7 one-handed letters, which may significantly affect the performance of machine learning models. Therefore, feature extraction based on the mathematical principle was carried out on the 390 features obtained. The details of the process are explained below.

B. Handcraft Feature Extraction

LM sensor produces the hand data as a value vector in 3-dimensional space. There are two main problems faced during receiving data from this device. The first problem is the re-composition of the data space whenever the LM sensor is turned on. Therefore, even if the position of the hand does not change, the raw data produced by the LM change and become inconsistent. In other words, the x, y, z values produced when the hand is being held straight may differ from the values produced in the next run of the LM. The second problem is about the position of the hand. The experiments showed that people often did not place their hand at the exact middle point of the device in every attempt and did not hold their hand in the position identical to the one used in obtaining the training data, which caused serious problems. These are extremely challenging to be noticed by the human eye. Hence, it is not possible to use the raw data from the LM sensor in the sign language alphabet recognition system. A new feature space that is independent of the hand position on the device, hand size and finger length is necessary. For addressing this need, a new handcraft feature extraction method, which was not used in the previous studies, was developed in this study.

For explaining the feature extraction method, it is necessary to elaborate on the features of the raw data collected by the LM firstly. As explained in the previous section, the number of features that LM collects for single hand is 195. These features contain information about palm position, arm direction, wrist position, elbow position, metacarpal start, metacarpal end, metacarpal direction, proximal start, proximal end, proximal direction, intermediate start, intermediate end, intermediate direction, distal start, distal end and distal direction for the three

axes. The fingers and the distance between fingers are critical parameters for sign language alphabets. Also, the relative position of two hands is an important factor for letters performed by two hands.

The mathematical and algorithmic steps for feature extraction are explained below while also considering the hand positions for letters in the Turkish Sign Language. During feature extraction, finger-to-finger, finger-to-palm, finger-to-wrist bone, palm-to-palm and wrist-to-wrist distances are calculated using the Euclidean distance. This calculation is carried out as in Eq. (1) for x, y and z axes and then this distance information is normalized dividing by the hand size. Thus, the three features of a particular position are converted into one single feature.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (1)$$

A symbolic representation of the feature extraction process in this study can be seen in Fig. 4. Each point in this figure represents the positions in x, y , and z axes. Arrows on the hands show the Euclidean distances calculated for different hand positions. Detailed and progressive information about each calculation is provided below.

Fig. 4. Handcraft feature extraction.

$d(d)$ Calculation: Let x_{id}, y_{id}, z_{id} be the coordinates of the starting point of the distal bone of the i th finger in the x, y , and z axes, respectively, and pw_k be the size of the k th hand ($k = 1, 2$); the distances between tip points of each finger are calculated as in Eq. (2).

$$d(d)_{i,j} = \frac{\sqrt{(x_{id} - x_{jd})^2 + (y_{id} - y_{jd})^2 + (z_{id} - z_{jd})^2}}{pw_k} \quad (2)$$

$d(t)$ Calculation: Let x_{it}, y_{it}, z_{it} be the coordinates of the starting point of the intermediate bone of the i th finger in the x, y , and z axes, respectively; the distances between the intermediate points of fingers are calculated using Eq. (3).

$$d(t)_{i,j} = \frac{\sqrt{(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2 + (z_{it} - z_{jt})^2}}{pw_k} \quad (3)$$

d(r) Calculation: Let x_{ir} , y_{ir} , z_{ir} be the coordinates of the starting point of the proximal bone of the i th finger in the x , y , and z axes, respectively; the distances between the proximal points of fingers are calculated using Eq. (4).

$$d(r)_{i,j} = \frac{\sqrt{(x_{ir}-x_{jr})^2+(y_{ir}-y_{jr})^2+(z_{ir}-z_{jr})^2}}{pw_k}. \quad (4)$$

d(dp), d(tp), d(rp), d(dw), d(tw) and d(rw) Calculations: Let x_{kp} , y_{kp} , and z_{kp} be the palm of the k th hand, and x_{kw} , y_{kw} , and z_{kw} be the wrist of the k th hand, and x_{id} , y_{id} , z_{id} , x_{it} , y_{it} , z_{it} , x_{ir} , y_{ir} , z_{ir} be the coordinates of the starting points of the distal, intermediate and proximal bones of the i th finger in x , y , and z axes; the distances of the tip, intermediate and proximal points of each finger to the palm (p) and the wrist (w) denominated respectively as $d(dp)$, $d(tp)$, $d(rp)$, $d(dw)$, $d(tw)$, and $d(rw)$ are calculated as seen in Eqs. (5)–(10).

$$d(dp)_{i,k} = \frac{\sqrt{(x_{id}-x_{kp})^2+(y_{id}-y_{kp})^2+(z_{id}-z_{kp})^2}}{pw_k}. \quad (5)$$

$$d(tp)_{i,k} = \frac{\sqrt{(x_{it}-x_{kp})^2+(y_{it}-y_{kp})^2+(z_{it}-z_{kp})^2}}{pw_k}. \quad (6)$$

$$d(rp)_{i,k} = \frac{\sqrt{(x_{ir}-x_{kp})^2+(y_{ir}-y_{kp})^2+(z_{ir}-z_{kp})^2}}{pw_k}. \quad (7)$$

$$d(dw)_{i,k} = \frac{\sqrt{(x_{id}-x_{kw})^2+(y_{id}-y_{kw})^2+(z_{id}-z_{kw})^2}}{pw_k}. \quad (8)$$

$$d(tw)_{i,k} = \frac{\sqrt{(x_{it}-x_{kw})^2+(y_{it}-y_{kw})^2+(z_{it}-z_{kw})^2}}{pw_k}. \quad (9)$$

$$d(rw)_{i,k} = \frac{\sqrt{(x_{ir}-x_{kw})^2+(y_{ir}-y_{kw})^2+(z_{ir}-z_{kw})^2}}{pw_k}. \quad (10)$$

All the above features are calculated for both hands. If the sign language letter is performed by one hand, the values for these features are 0 for the other hand.

For the two-handed letters, the relative position of the hands is also important. Thus, features that describe the relative positions of two hands are also calculated as illustrated below.

d(p) and d(w) Calculations: Let x_{kp} , y_{kp} and z_{kp} be the coordinates of the palm and x_{kw} , y_{kw} and z_{kw} be the coordinates of the wrist of the k th hand; the distances between the palms and wrists of the two hands, $d(p)$ and $d(w)$ are calculated using Eqs. (11) and (12).

$$d(p)_{k,j} = \sqrt{(x_{kp}-x_{jp})^2+(y_{kp}-y_{jp})^2+(z_{kp}-z_{jp})^2}. \quad (11)$$

$$d(w)_{k,j} = \sqrt{(x_{kw}-x_{jw})^2+(y_{kw}-y_{jw})^2+(z_{kw}-z_{jw})^2}. \quad (12)$$

d(if) Calculation: Let x_{kd} , y_{kd} and z_{kd} be the coordinates of the starting point of the distal bone of the index finger of the k th hand in x , y , and z axes; the distance between the index fingers of the two hands is calculated using Eq. (13).

$$d(if)_{k,j} = \sqrt{(x_{kd}-x_{jd})^2+(y_{kd}-y_{jd})^2+(z_{kd}-z_{jd})^2}. \quad (13)$$

Since most two-handed letters in the Turkish Sign Language alphabet involve placing the right hand over the left hand, the last feature in the dataset was extracted using an indexing algorithm for the right hand. In this algorithm, a 3-bit binary number was created by coding negative values as 0 and positive values as 1 in the x , y , and z axes. The state of negativity or positivity was determined by the palm position. A new feature was extracted by converting the value obtained in the binary system into the decimal system (i.e., $101 = 5$). In the feature extraction process, 54 and 112 features were extracted for signs made with single and two hands, respectively. In addition, a categorical conversion was carried out for the target variables by transforming [A–Z] to [0–28].

C. Classification

In this section, the LR, k -NN, RF, DNN and ANN classifiers employed for classifying the Turkish Sign Language Alphabet are explained. While the Keras Library was used for DNN, the Ski-Learn Library was employed for the rest of algorithms. Figure 5 shows the cascade voting approach applied to the classification of each sign in the Sign Language alphabet. Within the cascade voting approach, Voting-1 was applied to the predictions obtained by each classifier, and the class with the most votes was considered the output of the relevant classifier. During the test process, data were received from the LM device for 2 seconds. During this time, 120 frames were received. In fact, a single frame was sufficient to describe the sign language letter. However, since the LM device did not give very stable data, 120 frames of information were received. All of these 120 frames were given as input to machine learning algorithms. Machine learning algorithms generated sign language alphabet prediction for each frame. In other words, it produced 120 predicts. Voting-1 was applied to make the final prediction based on these predictions. The sign language alphabet with the most votes was the prediction output of the machine learning algorithm. Then, Voting-2 was used to reach the final decision from the classification results of 5 different classifiers. Thus, final prediction with the most votes was demonstrated as the system prediction. In addition, the test phase of the proposed study was performed in real conditions and test data were not previously recorded input data.

Feature vectors consisting of single-handed and two-handed signs in the feature extraction stage were $7 \times 3000 \times 54$ and $22 \times 3000 \times 112$, respectively. Here, 7 and 22 indicate respectively the number of single-handed and two-handed signs, 3000 indicates the number of sampling for each sign, and finally 54 and 112 indicate the number of handcrafted features for single-handed and two-handed signs, respectively. Here, the features between 55 and 112 to equate the number of features obtained from single-handed signs to the number of features obtained from two-handed signs were filled by a value of 0. The reason is to bring the feature space to the same size for single and two hands. Finally, $29 \times 3000 \times 112$ training data vector was obtained by combining the extracted datasets for single and two-handed signs. The complete this data vector was used for

training the classifiers. As real-time new data were employed for testing, the dataset was not divided into testing and training data. This new testing data comprised 120 rows captured for 2 seconds in real time. Here, each row describes the related sign language letter. Therefore, outputs of class predictions made by the classifiers were obtained for each row during the testing. As a result, 120 class predictions were obtained. These predicts were reduced to one predict by applying Voting-1 for each classifier. Then, the system prediction was obtained by applying Voting-2 to predictions of five classifiers from Voting-1.

Fig. 5. The proposed cascade voting approach architecture for sign language prediction.

LR: The LR algorithm aims at learning weights that minimise a cost function, such as a cross-entropy function with ridge regression, and reducing the model complexity as well as preventing overfitting [21]. LR classifier was trained using default parameters. While default parameters in the library were used as the parameters of the algorithm, some values were as follows: solver = 'lbfgs', max iter = 100, multiclass = 'auto', penalty = 'l2', C=1.0, fit intercept = True.

k-NN: *k*-NN is a statistical method, which presents the class label with the highest number of classes of the closest *k* examples for the target variable as the prediction result. *k*-NN classifier was trained using parameters such as number of neighbors = 5, weight function = uniform, leaf size = 30 and algorithm = auto.

RF: The acceptance towards RF, a combination of decision tree classifiers, is on constant increase due to its classification performance. In addition, parameter determination is easier in RF when compared to other methods that require rigorous adjustments [22], [23]. The RF classifier was trained using the parameter n estimators = 100, criterion = 'gini', max depth = None, min samples split = 2.

ANN: The structure of ANN resembles that of the human brain with interconnected artificial neurons in the place of

highly cross-linked axons and dendrites. The model does not involve specific equations [24]. The ANN classifier was trained using the parameter hidden layer number = 1, hidden layer neuron number = 100, activation = 'relu', solver = 'adam', learning rate = 'constant', learning rate init = 0.001, max iter = 200.

DNN: Deep learning is a sub-domain of machine learning described as artificial intelligence. Using deep learning, it is possible to establish Deep Neural Networks (DNN) with deep architectures [25]. DNN is a multilayer perceptron with several hidden layers and fully connected weights. DNN is almost always initialized using an unsupervised or a supervised pre-training technique [26]. The DNN classifier used in this study had 3 hidden layers with 184, 368 and 552 neurons. This classifier was trained with input and hidden layer activation = 'relu', output layer activation = 'softmax', learning rate = 1, loss function = 'Categorical_crossentropy', optimizer = 'Adadelta' for parameters.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the real-time classification performance of the proposed sign language recognition system is explained in detail. The experimental study was carried out on the TSL alphabet presented in Fig. 6. An independent person, whose data were not in the training dataset, conducted the experimental study. Thus, the person-independent classification performance of the recognition system was introduced. In real-time recognition, firstly, the features were extracted using two-second data collection for each sign language alphabet. Then these features were fed into the classifiers, and then the final prediction of the cascade voting approach was presented on the screen. The results obtained in this study and comparison with previous studies are presented below.

Fig. 6. Turkish Sign Language alphabet [27].

Ten experiments were conducted for each letter in the sign language alphabet in the experimental study. The results are presented comparatively in Table I, which illustrates the average accuracy value obtained in 10 experiments using the LR, *k*-NN, RF, DNN and ANN single classifiers and the cascade voting method. The sign language letters predicted inaccurately by classifiers are presented in parentheses. In general evaluation of the letter-wise performance of the single classifiers, the first 5 letters that were most often predicted inaccurately were "Ç, H, J, S and Ğ". Respectively, the LR, *k*-NN, RF, DNN and ANN classifiers were not able to classify

letters “Ç”, “A, Ç, D, E, Ğ, İ and T”, “Ğ and H”, “H” and “Ç”. All of these letters with recognition problem were performed with two hands. The reason is that the LM sensor cannot sense the second hand or that the hands are virtualised inaccurately. On the other hand, using the cascade voting method, all letters in the sign language alphabet were recognised. All letters

excluding “J” and “H” were recognised at the accuracy rate of 100 %. The recognition accuracy rates for these letters are 80 % and 90 %, respectively. Moreover, Fig. 7 graphically summarises the performance of the classifiers for each sign language character.

TABLE I
REAL-TIME RECOGNITION PERFORMANCE OF THE MACHINE LEARNING ALGORITHMS (SINGLE CLASSIFIER AND CASCADE VOTING)

Sign	LR, %	k-NN, %	RF, %	DNN, %	ANN, %	Cascade Voting, %
A	100	0 (Ö and K)	100	100	80 (Ö)	100
B	100	100	100	100	100	100
C	100	80 (Y and L)	100	100	100	100
Ç	0 (Ş)	0 (Z)	100	90 (Ş)	0 (C, Y)	100
D	90 (Ç)	0 (B and N)	50 (B, T and Ç)	100	100	100
E	90 (D)	0 (B, Z and T)	60 (B and F)	100	90 (F)	100
F	100	10 (R and Ö)	100	100	100	100
G	100	80 (J)	90 (B)	100	100	100
Ğ	90 (Ç)	0 (Y)	0 (J)	90 (B)	100	100
I	100	100	100	100	100	100
İ	100	0 (Z and M)	100	100	100	100
J	70 (B, U, Y)	10 (G, Ğ, Y and K)	30 (B)	70 (B)	90 (D)	80 (B)
H	90 (Z)	70 (Ö, A, G)	0 (B, Ü, İ, Z, K)	0 (B, İ, Z, R)	90 (Ö)	90 (Z)
K	100	100	100	100	100	100
L	100	100	90 (C)	100	100	100
M	100	10 (B and N)	100	100	100	100
N	100	40 (B)	100	100	100	100
O	100	100	100	100	100	100
Ö	100	80 (A and D)	60 (İ and Z)	100	100	100
P	90 (U)	100	100	100	90 (U)	100
R	70 (Ö and T)	20 (T, F, Z)	80 (N and E)	100	100	100
S	90 (B)	40 (K, B, Ö and Ü)	10 (E, D and B, Ü, J)	20 (E)	100	100
Ş	100	70 (S and Ü)	40 (Ü, Ö, B)	50 (Ü)	100	100
T	100	0 (A and Ö)	90 (B)	100	100	100
U	100	100	100	100	100	100
Ü	70 (B and F)	40 (B, İ, Ö, K)	90 (B)	90 (Ş)	100	100
V	100	100	90 (N)	100	100	100
Y	100	100	60 (A)	100	100	100
Z	100	50 (F)	100	100	100	100
Mean	91.38	51.72	77.24	90.00	94.48	98.97

Fig. 7. Letter-wise comparison of the accuracy rates of the classifiers.

To evaluate the overall classification performance of the classifiers, the mean accuracy values are important. Figure 8 shows the average accuracy values of the classifiers comparatively. The minimum classification performance belongs to *k*-NN with a mean accuracy value of 51.72. The maximum classification performance was obtained using the cascade voting method at the accuracy rate of 98.97 %. The classification performance of the other methods – RF (77.24 %), DNN (90 %), LR (91.38 %), ANN (94.38 %), respectively.

Fig. 8. Average accuracy presented by classifiers.

To explore the classification details of the cascade voting method that was the most successful in the classification of the proposed system, a confusion matrix was formed (see Fig. 9). As seen in this confusion matrix, this method reaches the accuracy rate of 100 % in recognition performance except for letters “J” and “H”. In other words, the voting method allows for error-free recognition performance for 27 letters. The letter “H” was recognised at the accuracy rate of 90 %. Out of 10 real-time experiments, the letter “H” was accurately recognised 9 times while once being recognised as “Z”. The classification accuracy of the letter “J” was 80 %. Out of 10 real-time

experiments, the letter “J” was accurately recognised 8 times whereas two times being recognised as “B”.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	T	U	Ü	V	Y	Z
A	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	2	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
Ü	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Fig. 9. Voting method confusion matrix.

In the test process performed in real time, the computation time was approximately 5 seconds for each sign language character. In 4 seconds out of this time, raw data were obtained from the LM sensor and then the handcrafted features were extracted. In the rest of this time, the proposed model predicted the sign language character.

A comparison of the present work with the results of previous studies is shown in Table II. Reviewing the prior research does not reveal a study directly related with the present study that uses Leap Motion for real-time classification on TSL. However, there are some studies that conduct real-time sign classification using video images obtained from Kinect device. Oktekin & Cavus [6] classified 33 Turkish sign language expressions in real time with 82 % accuracy using Kinect sensor and HMM model. Haberdar & Albayrak [3] classified 22 one-handed and 28 two-handed sign language gestures in real time with 95.7 % accuracy. As can be seen, the classification performance of these studies in real-time Turkish sign language recognition is lower than that of the proposed approach. Moreover, there are a limited studies performed in real time using LM for other sign languages. For example, the study presented by Naglot and Kulkarni [28], which was performed on American Sign Language alphabet, was quite similar to our study. They classified 26 letters in real time with 96.15 % accuracy using LM sensor and ANN. In another study, Kumar et al. [29] classified 56 sign language gestures in real time with 63.57 % accuracy using SVM classifier and LM sensor data.

In addition, there are three studies that conducted offline classification on the Turkish sign language. The first study conducted by Demircioglu et al. [8] involved offline recognition of 18 sign language alphabets using data captured by LM device at the accuracy rate of 99.03 %. In the second study, Unutmaz et al. [1] classified offline 9 different sign language expressions captured by Kinect using CNN at the accuracy rate of 99.25 %. The last study involved recognition of 28 Arabic sign language alphabets at the accuracy rate of

99.1 % [18]. High performance was expected in offline classification studies as the data was not captured in real time. The success rate is typically lower in real-time classification systems due to the potential changes in real-time data captured by the device, hand position, distance to device and lighting conditions. The proposed system of the current study works in real time; it offers a very close classification performance to

offline studies. In this respect, the design of an expert system that includes the approach we propose in sign language classification can be implemented. This eminence of the classification system is largely due to the new feature extraction method that can extract features independent of environmental conditions and the person performing the sign language expressions.

TABLE II
COMPARISON WITH PREVIOUS STUDIES ON THE TURKISH SIGN LANGUAGE

Reference	Sign Language	Classifier	Method	Dataset	Maximum Recognition Rate
Demircioglu et al. [8]	TSL	Heuristic, RF and MLP	Leap motion Offline	18 sign language alphabets	99.03 %
Oktekin & Cavus [6]	TSL	HMM	Kinect Real-time	33 sign language expressions	82 %
Unutmaz et al. [1]	TSL	CNN	Kinect, Offline	9 different sign language expressions from 12 different people	99.25 %
Ezel & Baykan [12]	TSL	CNN	Image Offline	29 sign language alphabet images	84.48 %
Haberdar & Albayrak [3]	TSL	HMM	Video Images Real-time	22 one-handed and 28 two-handed gestures	95.7 %
Naglot & Kulkarni [28]	American Sign Language	ANN	Leap Motion, Real-time	26 letters	96.15 %
Kumar et al. [29]	Indian Sign Language	SVM	Leap Motion Real-time	56 gestures	63.57 %
Mohandes et al. [18]	ArSL	NB MLP	Leap motion Offline	28 Arabic sign language alphabets	99.1 %
Our Study	TSL	LR, k -NN, RF, DNN, ANN and Voting	Leap Motion, Real-time	29 TSL alphabets	98.97 %

V. CONCLUSION

In this study, a real-time Turkish sign language alphabet recognition system was developed using LM sensor. There are many factors that may influence the system performance in real-time sign language recognition. Some of these factors are: (i) the x, y and z coordinates of the hand on the LM device, (ii) the hand sizes, (iii) the distance between two hands, (iv) the palm direction, (v) failure to detect one of the hands in letters performed using both hands, (vi) the lighting conditions of the environment. The study explained the new handcraft feature extraction method, which was based on mathematical model and allowed for recognition independent of these factors or the person performing the sign language. The extracted features were used for training LR, k -NN, RF, DNN, ANN single classifiers and also cascade voting approach, and their classification performance was evaluated in detail. The best classification performance was obtained using the cascade voting method as it integrated the classification results of the other five machine learning methods. The classification accuracy rate of the cascade voting method was 98.97 %. In comparison to the previous studies on Turkish sign language recognition, the present system was tested by an individual out of the training dataset, employed real-time classification and achieved person-independent high classification accuracy. Furthermore, the proposed feature extraction method can be

adapted to other sign language recognition systems. Our future research will focus on expanding the sign database and presenting the system to disabled individuals by the means of integration to expert systems.

VI. ACKNOWLEDGMENT

This study has been supported by Kastamonu University within the scope of Scientific Research Projects under grant no. KÜ-BAP01/2018-30.

REFERENCES

- [1] B. Unutmaz, A. C. Karaca, and M. K. Gullu, "Turkish Sign Language Recognition Using Kinect Skeleton and Convolutional Neural Network," in *2019 27th Signal Processing and Comm. Appl. Conference (SIU)*, Apr. 2019, pp. 1–4. <https://doi.org/10.1109/siu.2019.8806380>
- [2] P. Kumar, P. P. Roy, and D. P. Dogra, "Independent Bayesian Classifier Combination Based Sign Language Recognition Using Facial Expression," *Information Sciences*, vol. 428, pp. 30–48, Feb. 2018. <https://doi.org/10.1016/j.ins.2017.10.046>
- [3] H. Haberdar and S. Albayrak, "Real Time Isolated Turkish Sign Language Recognition from Video Using Hidden Markov Models with Global Features," *Lecture Notes in Computer Science*, pp. 677–687, 2005. https://doi.org/10.1007/11569596_70
- [4] T. Kapuściński and D. Warchol, "Hand Posture Recognition Using Skeletal Data and Distance Descriptor," *Applied Sciences*, vol. 10, no. 6, p. 2132, Mar. 2020. <https://doi.org/10.3390/app10062132>
- [5] P. Kumar, H. Gauba, P. Pratim Roy, and D. Prosad Dogra, "A Multimodal Framework for Sensor Based Sign Language Recognition," *Neurocomputing*, vol. 259, pp. 21–38, Oct. 2017. <https://doi.org/10.1016/j.neucom.2016.08.132>

- [6] B. Oktekin, "İşitme ve Konuşma Engelli Bireyler için İşaret Tanıma Sistemi Geliştirme," *Uluslararası Kıbrıs Üniversitesi Fen-Edebiyat Fakültesi*, vol. 25, no. 97-1, pp. 593–609, Jan. 2019. <https://doi.org/10.22559/folklor.969>
- [7] Y. Mori and M. Toyonaga, "Data-Glove for Japanese Sign Language Training System with Gyro-Sensor," in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, Dec. 2018, pp. 1354–1357. <https://doi.org/10.1109/scis-isis.2018.00211>
- [8] B. Demircioğlu, G. Bülbül, and H. Köse, "Turkish Sign Language Recognition with Leap Motion [Leap Motion ile Türk İşaret Dili Tanıma]," in *2016 24th Signal Processing and Communication Application Conference (SIU)*, May 2016. <https://doi.org/10.1109/siu.2016.7495809>
- [9] K. Kakayev and S. Albayrak, "Turkish Sign Language Recognition Using Hidden Markov Model," in *Computer Science & Information Technology (CS & IT)*, Jun. 2016, pp. 11–18. <https://doi.org/10.5121/csit.2016.60802>
- [10] A. Memiş and S. Albayrak, "A Kinect Based Sign Language Recognition System Using Spatio-Temporal Features," in *Sixth International Conference on Machine Vision (ICMV 2013)*, Dec. 2013. <https://doi.org/10.1117/1.22051018>
- [11] O. Yalcinkaya, A. Atvar, and P. Duygulu, "Turkish Sign Language Recognition Application Using Motion History Image," in *2016 24th Signal Processing and Communication Application Conference (SIU)*, May 2016, pp. 801–804. <https://doi.org/10.1109/siu.2016.7495861>
- [12] E. Ezel, O. Baykan, "Vision-Based Turkish Sign Language Recognition Using Convolutional Neural Networks," in *International Conference on Theoretical and Applied Computer Science and Engineering (ICTACSE, 2017)*, Nov. 2017, p. 14.
- [13] M. Aktas, B. Gokberk, and L. Akarun, "Recognizing Non-Manual Signs in Turkish Sign Language," in *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov. 2019, pp. 1–6. <https://doi.org/10.1109/ipta.2019.8936081>
- [14] S. G. Moreira Almeida, F. G. Guimarães, and J. Arturo Ramirez, "Feature Extraction in Brazilian Sign Language Recognition Based on Phonological Structure and Using RGB-D Sensors," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7259–7271, Nov. 2014. <https://doi.org/10.1016/j.eswa.2014.05.024>
- [15] W. Tao, M. C. Leu, and Z. Yin, "American Sign Language Alphabet Recognition Using Convolutional Neural Networks with Multiview Augmentation and Inference Fusion," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 202–213, Nov. 2018. <https://doi.org/10.1016/j.engappai.2018.09.006>
- [16] A. addin I. Sidig, H. Luqman, and S. A. Mahmoud, "Transform-Based Arabic Sign Language Recognition," *Procedia Computer Science*, vol. 117, pp. 2–9, 2017. <https://doi.org/10.1016/j.procs.2017.10.087>
- [17] M. A. Almasre and H. Al-Nuaim, "A Comparison of Arabic Sign Language Dynamic Gesture Recognition Models," *Heliyon*, vol. 6, no. 3, p. e03554, Mar. 2020. <https://doi.org/10.1016/j.heliyon.2020.e03554>
- [18] M. Mohandes, S. Aliyu, and M. Deriche, "Arabic Sign Language Recognition Using the Leap Motion Controller," in *2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Jun. 2014, pp. 960–965. <https://doi.org/10.1109/isie.2014.6864742>
- [19] A. Karacı, K. Akyol, and Y. Gültepe, "Turkish Sign Language Alphabet Recognition with Leap Motion," in *Proceedings of the International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18)*, May 2018, pp. 189–192.
- [20] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak, "Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion," *Applied Sciences*, vol. 9, no. 3, pp. 445, Jan. 2019. <https://doi.org/10.3390/app9030445>
- [21] B. K. Dedetürk and B. Akay, "Spam Filtering Using a Logistic Regression Model Trained by an Artificial Bee Colony Algorithm," *Applied Soft Computing*, vol. 91, p. 106229, Jun. 2020. <https://doi.org/10.1016/j.asoc.2020.106229>
- [22] O. Sagi and L. Rokach, "Ensemble Learning: A Survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, Feb. 2018. <https://doi.org/10.1002/widm.1249>
- [23] A. Karacı and A. Kemal, "Classification of Alcohols Obtained From QCM Sensors Using Hybrid Ensemble Classifier," in *2nd International Turkish World Engineering and Science Congress*, Nov. 2019, pp. 159–163.
- [24] A. Karacı, A. Caglar, B. Aydinli, and S. Pekol, "The Pyrolysis Process Verification of Hydrogen Rich Gas (H-rG) Production by Artificial Neural Network (ANN)," *International Journal of Hydrogen Energy*, vol. 41, no. 8, pp. 4570–4578, Mar. 2016. <https://doi.org/10.1016/j.ijhydene.2016.01.094>
- [25] A. Karacı, "Self-Care Problems Classification of Children with Physical and Motor Disability by Deep Neural Networks," *J. Polytech.*, vol. 23, no. 2, pp. 333–341, 2020.
- [26] A. Karacı, "Predicting Breast Cancer with Deep Neural Networks" in *Lecture Notes on Data Engineering and Communications Technologies*, D. Hemanth, U. Kose, Eds. 2020, pp. 996–1003.
- [27] S. Ozbay, and M. Safar, "Real-Time Sign Languages Recognition Based on Hausdorff Distance, Hu Invariants and Neural Network," in *2017 International Conference on Engineering and Technology (ICET)*, Aug. 2017, pp. 1–8. <https://doi.org/10.1109/icengtechnol.2017.8308204>
- [28] D. Naglot and M. Kulkarni, "Real Time Sign Language Recognition Using the Leap Motion Controller," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, Aug. 2016, pp. 1–5. <https://doi.org/10.1109/inventive.2016.7830097>
- [29] P. Kumar, R. Saini, S. K. Behera, D. P. Dogra, and P. P. Roy, "Real-Time Recognition of Sign Language Gestures and Air-Writing Using Leap Motion," in *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, May 2017, pp. 157–160. <https://doi.org/10.23919/mva.2017.7986825>

1372

Abdulkadir Karacı received the Doctoral degree in electronic and computer education (intelligent tutoring system, machine learning and fuzzy logic) from Gazi University, Turkey, in 2013. He is currently an Associate Professor at the Department of Computer Engineering, Kastamonu University. His current research interests include machine learning, convolution neural network, intelligent tutoring system and programming languages.
E-mail: akaraci@kastamonu.edu.tr
ORCID iD: <https://orcid.org/0000-0002-2430-1372>

Kemal Akyol received his B. sc. from Computer Science Department of Gazi University, Ankara/Turkey in 2002. He received his M. sc. degree from Natural and Applied Sciences, Karabuk University, Karabuk/Turkey and PhD degree from the same department. His research interests include machine learning, decision support systems and expert systems. He currently works at the Department of Computer Engineering, Kastamonu University.
E-mail: kakyol@kastamonu.edu.tr
ORCID iD: <https://orcid.org/0000-0002-2272-5243>

Mehmet Ugur Turut received B.sc. in computer engineering from Kastamonu University, Turkey, in 2015. At the moment, he is a Master student at Kultur University, Turkey. His current research interests include machine learning, convolution neural network and programming languages.
E-mail: ugrmhmt@gmail.com
ORCID iD: <https://orcid.org/0000-0001-7841-2660>