UDC 519.7
DOI: 10.15587/1729-4061.2021.239253

SIGN LANGUAGE DACTYL RECOGNITION BASED ON MACHINE LEARNING ALGORITHMS

Chingiz Kenshimov
PhD*
Zholdas Buribayev
Corresponding author
Master**
E-mail: zholdas_87@mail.ru
Yedilkhan Amirgaliyev
Doctor of Technical Sciences, Professor*
Aisulyu Ataniyazova
Masters Student**
Askhat Aitimov
PhD Student
Department of Computer Science
Suleyman Demirel University
Abylai khan str., 1/1, Kaskelen,
Republic of Kazakhstan, 040900
*Department of Robotics and Artificial Intelligence
Institute of Information and Computational Technologies
Pushkina str., 125, Almaty,
Republic of Kazakhstan, 050010
**Department of Computer Science
Al-Farabi Kazakh National University
Al-Farabi ave., 71, Almaty, Republic of Kazakhstan, 050040

In the course of our research work, the American, Russian and Turkish sign languages were analyzed. The program of recognition of the Kazakh dactylic sign language with the use of machine learning methods is implemented. A dataset of 5000 images was formed for each gesture, gesture recognition algorithms were applied, such as Random Forest, Support Vector Machine, Extreme Gradient Boosting, while two data types were combined into one database, which caused a change in the architecture of the system as a whole. The quality of the algorithms was also evaluated.

The research work was carried out due to the fact that scientific work in the field of developing a system for recognizing the Kazakh language of sign dactyls is currently insufficient for a complete representation of the language. There are specific letters in the Kazakh language, because of the peculiarities of the spelling of the language, problems arise when developing recognition systems for the Kazakh sign language.

The results of the work showed that the Support Vector Machine and Extreme Gradient Boosting algorithms are superior in real-time performance, but the Random Forest algorithm has high recognition accuracy. As a result, the accuracy of the classification algorithms was 98.86 % for Random Forest, 98.68 % for Support Vector Machine and 98.54 % for Extreme Gradient Boosting. Also, the evaluation of the quality of the work of classical algorithms has high indicators.

The practical significance of this work lies in the fact that scientific research in the field of gesture recognition with the updated alphabet of the Kazakh language has not yet been conducted and the results of this work can be used by other researchers to conduct further research related to the recognition of the Kazakh dactyl sign language, as well as by researchers, engaged in the development of the international sign language

Keywords: Gesture recognition, sign language, feature extraction, hand tracking, algorithm evaluation

Received date 05.07.2021
Accepted date 16.08.2021
Published date 31.08.2021

How to Cite: *Kenshimov, C., Buribayev, Z., Amirgaliyev, Y., Ataniyazova, A., Aitimov, A. (2021). Sign language dactyl recognition based on machine learning algorithms. Eastern-European Journal of Enterprise Technologies, 4 (2 (112)), 58–72. doi: https://doi.org/10.15587/1729-4061.2021.239253*

1. Introduction

A major advance in the field of information technology over the past ten years can be called the digitalization of human-computer interaction at the visual level. This achievement primarily solves communication problems of people with hearing disabilities and allows for rapid human-computer interaction. In this regard, the gesture is one of the main forms of visual communication of people. The actions and relative positions of body parts and their changes over time correspond to certain messages, and recently became also promising in the interaction of technical systems and humans. Thanks to the detection capabilities of visual communication primitives, gesture recognition has become one of the most widely researched topics in recent years [1, 2]. The results of automatic gesture recognition and classification are used to train people with hearing impairments and help them communicate with strangers using sign language. They can also be used as a quick message for digital smart devices. This is the social significance of sign language recognition. As video data has become ubiquitous in practical applications, the research and development of gesture recognition automation is finding application in many human-machine communication systems.

The phonological structure of a sign language is usually divided into five elements: articulation point, hands configuration, movements type, hands orientation, and facial expressions [1]. Each gesture is perceived through a combination of these elements. These blocks represent valuable sign language elements and can be used by automated intelligent sign language recognition (SLR) systems. It should bear in mind, that in a sign language, one gesture means one whole word. In contrast, dactylology is a peculiar form of speech where the dactylic alphabet is used. Each hand gesture illustrates a specific letter of this language. Each natural language, like the Kazakh language, has its own dactylic language, which is also different from the dactylic language of other languages.

Research on the development of a Kazakh sign dactyl language recognition system is currently insufficient for a complete representation of this language. When developing methods and systems for recognizing the Kazakh dactyl sign language, a number of difficulties arise, mainly associated with spelling, sign language and other features of the language [3]. The alphabet of the Kazakh language has 42 letters, of which 33 are borrowed from the Russian alphabet, the remaining 9 are specific to this language. This condition is also relevant for the Kazakh dactyl language. Since the Kazakh language belongs to the family of Turkic-speaking languages and most words, letters and sounds are similar, these tasks are relevant for the majority of the population of Turkic-speaking peoples, which now numbers more than 200 million people. But it should also be borne in mind, that the Kazakh language, unlike its relatives, is just beginning to move from the Cyrillic to the Latin alphabet.

Also, one of the problems is the division of the types of gesture into static, when there is no need to make any movement of the hands, the position of the hand and fingers is stationary in space during the considered time, as well as dynamic when gestures are reproduced by moving the hand. In most cases, systems that provide real-time gesture reading have only one of the proposed forms. That is, they rely in comparison on static or dynamic data in their database.

2. Literature review and problem statement

The paper [1] provides an overview to create a consistent taxonomy to describe recent research, divided into four main categories: development, structure, recognition of other hand gestures, and reviews. An analysis of glove systems for the characteristics of SLR devices was carried out, a technology development plan was developed, existing limitations were considered, and valuable information about technological environments was provided to help explore opportunities and challenges in this area. This paper uses the low-level features of the human hand, used for machine learning algorithms. It then applies this data in the recognition and classification process. The main disadvantage of this approach was that special gloves for recognizing the position of the hands cannot always be at your fingertips, and not everyone has them.

The article [2] aims to recognise sign language characters, trained using images of American sign language letters. The use of capsule networks for learning processes was proposed. The test results were compared with the results of the LeNet architecture. As a result of the study, it was noticed that for effective character recognition in sign language, capsule networks are useful and give a successful result than LeNet.

In [4], a technique for visual gesture recognition is proposed by combining several spatial and spectral representations of gesture images manually using a convolutional neural network. The technique, proposed in this paper, allows us to calculate the Gabor spectral representations of spatial images of hand gestures and uses an optimized neural network to classify gestures into appropriate classes. The authors of this paper have considered various ways to combine both types of modalities to determine a model that increases the reliability and accuracy of recognition. It should be noted, that the material of this work emphasizes the development of sign language recognition, gradually moving from a variety of auxiliary tools to more everyday ones, such as a smartphone or tablet, which a person can carry with them in everyday life, without additional cargo, to provide convenience and save the budget of the average consumer.

In the paper [5], a classification of the Turkish sign language was implemented using finite automata based on pose marking, which uses depth values in location-based functions. A grid-based signature space clustering scheme has been developed, and cluster numbers are used as objects for a set of connections. A pose marking algorithm for recognizing a predefined set of gestures in TSL is proposed. The labels, assigned to poses, are used to classify gestures concerning known vocabulary using the FSA. A set of complex gestures is selected to evaluate the technique; however, their scheme is also expanded for a new gesture, simply providing an appropriate FSA based on its poses. The general classification scheme deals only with position labels and not low-level and spatiotemporal features, reducing the space and time requirements.

The authors of [6] described the results of using the long-term, short-term memory (LSTM) model, which improved the machine translation of Google Translate.

One of the closest works on the study of the Kazakh sign language is [3], where the Kinect sensor is used for gesture recognition, the coordinates of the skeleton of the hand and key characteristics are processed through XML files using tools and calculations in MATLAB. It is easy to understand that this approach was implemented for the old Kazakh alphabet.

In the article [7], several real-time gesture recognition systems were compared using convolutional neural networks. The system, proposed in this paper, can recognize words from a natural language with gestures, using signs for each letter. The approach of this work was evaluated in the American and Russian sign languages. For the American sign language, a data set, prepared by Massey University and the Institute of Information and Mathematical Sciences, was used. Russian sign language recognition quality lagged behind the high result due to the complexity of the real data set for the Russian sign language. According to the results of the study, the accuracy of typing the American Sign Language showed a high result, which we took into account for the experience in the design of architecture.

The article [8] presents an effective framework for solving the problem of static gesture recognition based on data, obtained from web cameras and the Kinect depth sensor. In this paper, the video sequence is taken as input data. That is, the sequence of frames and the classification is performed separately without any frame information. The accuracy of the method, proposed by the authors, was estimated based on the collected images, consisting of 2700 frames.

In [9], an intelligent system for the Turkish sign language recognition was developed. It is based on 33 basic signs of the Turkish Sign Language. To determine the signals, a Microsoft Kinect v2 sensor was used. The proposed system is designed to help people with hearing and speech impairments and other people and solve communication problems between these people. We can apply this development for our own purposes, however, there is a linguistic difference between the Kazakh and Turkish languages, which is the main barrier.

The scientific works [10] present a review of the scientific literature on sign language recognition systems. In [11, 12] they were identified and analyzed for their direct relevance to sign language recognition systems. In the article [13] the classification is considered based on six dimensions (data collection methods, static/dynamic signs, signature mode, one-handed/two-handed signs, classification technique, and recognition speed).

The research paper [14] analyzes statistics on the use of various data collection methods, used in sign language systems, and 12 sign languages were selected for this purpose. Among these languages, the American sign language is the first to be analyzed. For the review of this language, the literature was used [15, 16].

The article [17] presents a method for recognizing gestures of the American sign language using the method of principal component analysis to minimize the similarity of gesture classes. The scientific work [18] implements a system for recognizing the letters of the American alphabet using surface electromyography to allow people to spell words. The developers of the recognition system, presented in the article [19], used the MAdaline network for image processing and classification.

The article [3] describes the sign symbols, used to record the structure of gestures in writing. The choice of L. S. Dimskis' sign notation in relation to the Kazakh sign language is also justified, the features of the representation of the Kazakh sign language using L. S. Dimskis' sign notation in the course of compiling a dictionary of frequently used gestures are revealed.

As a result of the analysis of the existing methods, proposed in the above works, most are characterized by insufficient accuracy and speed of recognized gestures. Also, many studies often require conditions, such as wearing special gloves and other devices, good lighting, etc.

Many scientific studies related to the recognition of the Kazakh sign language were conducted with the old alphabet, consisting of 42 letters [3, 20]. Also, the recognition accuracy did not exceed 90 %, which requires updating and improving the recognition systems. The current Kazakh alphabet consists of 31 letters, changes were made to the spelling of specific Kazakh letters, digraphs were also introduced. Therefore, the development of an accurate and high-speed algorithm for recognizing the new Kazakh sign language in real time in order to facilitate communication with people with hearing disabilities is an urgent task.

3. The aim and objectives of the study

The aim of this work is to implement a recognition program with the highest accuracy of the Kazakh dactylic sign language with an updated alphabet using machine learning methods. The scientific novelty of this work is the development of a new system that provides a solution to the problems of gesture recognition, both dynamic and static, combined into one base for building practical systems for human-computer interaction.

To achieve this aim, it is necessary to solve the following tasks:

– collect a dataset with images of each gesture for training and testing samples, using the MediaPipe framework, implement hand and finger tracking and identify key points of the hands in three-dimensional space;

– implement a recognition program of the Kazakh dactylic sign language and classify gestures using machine learning algorithms;

– conducting a numerical evaluation of the quality of algorithms in order to determine the best classifier in gesture recognition problems, build a three-dimensional model, containing metrics, such as precision, recall and f1-measure for all gesture classes.

4. Materials and methods

In the course of the work, the American, Russian and Turkish sign languages were analyzed [8, 9, 15], and a program for recognizing the Kazakh sign language was implemented on their basis. In this paper, classical algorithms for gesture recognition are applied, combining two types of data into one database, which is reflected in the architecture of the recognition system.

Therefore, works, devoted to the study of sign dactyl language, mainly take into account only one specific language, determine the type of data (single frame or multiple frames) that we cannot use as proposed solutions for the Kazakh language.

4. 1. Random Forest

The first method, chosen to classify fingerprint language gestures, is the random forest algorithm. Fig. 1 shows how the random forest algorithm works. By representing a set of decision trees, this algorithm combines them to produce a more accurate result. The training sample is divided into subsamples of a certain size, from which the trees are built. To build a split in the tree, the maximum value of the random functions was viewed. Each new partition of the tree is made by determining its random features, the best feature is selected, and the tree structure continues until the choice is exhausted, that is, until only one representative of the class remains. But in the latest implementations of this algorithm in our work, we see that there are parameters that limit the height of trees and the number of objects in the subsample when recognizing gestures.

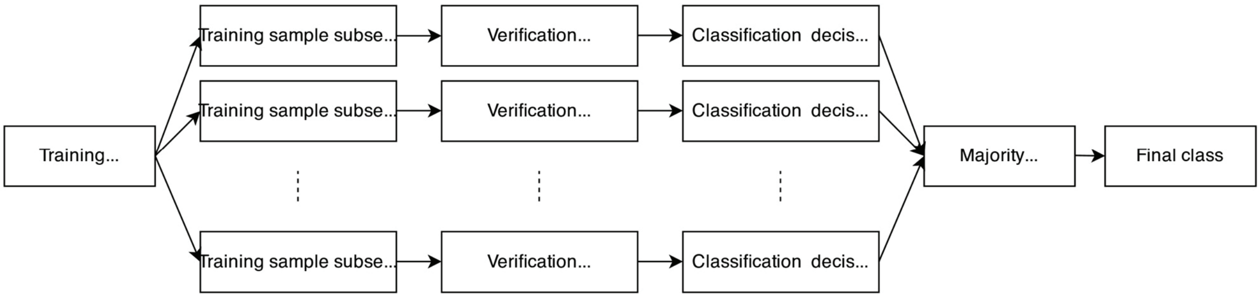


Fig. 1. The process of Random Forest classifier

4. 2. Support Vector Machine

The next research method is the classifier of support vector machines. This algorithm can be divided into two parts: training the classifier and recognizing the characters, supplied to the input.

At the first stage, a software implementation of the mathematical apparatus of the support vector machine is developed to create a classifier model. The SVM model represents different classes on a hyperplane in a multidimensional space. This hyperplane is generated iteratively to minimize the error. The purpose of this classifier is to divide data sets into classes to find the maximum limit hyperplane. One of the important concepts in SVM is support vectors, a collection of data points, located closest to the hyperplane, and using these points to determine the dividing line. A hyperplane is a decision plane, a space, divided between a set of objects with different classes.

In the second stage, the recognition and classification process is implemented. The hyperplane that separates the classes correctly is selected.

The main advantages of the SVM classifier are the ability to show high accuracy and the ability to work well with a large space. SVM classifiers mostly use a subset of training points. Hence very little memory is used as a result.

They have a long learning time, so they are not suitable for large datasets in practice. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

4. 3. Extreme Gradient Boosting

The third algorithm, used in this research paper, is the XGBoost classifier. This algorithm is based on gradient boosting of decision trees. First, we construct an ensemble of weak predictive models, in this case, decision trees. The training of the ensemble is performed sequentially. At each iteration, the deviations of the predictions of the already trained ensemble on the training sample are calculated. By adding the new tree's predictions to the trained ensemble's predictions, the average deviation of the model, which is the goal of the optimisation problem, is reduced. New trees will be added to the ensemble as long as the error is reduced.

The XGBoost algorithm is designed for classification tasks that work with structured and tabular data. Using the gradient descent architecture, the algorithm enhances the performance of weak classifiers. The main parameters of the algorithm are the number of trees, the step size to prevent overfitting, the change in the value of the loss function to divide the leaf into subtrees, the maximum depth of the tree, and the regularisation coefficient. To support the parallelisation of the tree building process, a block structure is used. It is possible to continue training for additional training on new data. The parallelisation of the algorithm is possible due to the interchangeable nature of the loops, used to build the training base: the outer loop lists the leaves of the trees, the inner loop calculates the features. Finding a loop inside another one prevents the algorithm from parallelizing, since the outer loop cannot start its execution if the inner one has not finished its work yet. Therefore, to improve the running time, the order of the loops is changed: initialisation takes place when reading data, then sorting is performed using parallel threads. This replacement improves the performance of the algorithm by distributing the calculations across threads.

5. Results of sign language dactyl recognition based on machine learning algorithms

5. 1. Creating a dataset

To create a dataset, first of all, a real-time image output program was used. After that, the hand was detected, that is, the area of interest for further classification. After detecting and tracking the hand, the frame of the hand is drawn by the key points of the hands. The frame is displayed on an empty frame, which will be saved in the dataset in the corresponding, predefined folder.

The first step of our research is to get an image from a webcam, since the program works in real-time. This is followed by the process of hand detection by the MediaPipe neural network framework, as shown in Fig. 2. The ability to perceive the shape and movement of the hands is used to understand sign language and control hand gestures. Reliable real-time hand perception is a challenge for computer vision, as the hands often close together and do not have high-contrast patterns. The MediaPipe framework, by creating multi-modal machine learning pipelines, returns accurate three-dimensional key points of the hand.

Fig. 3 shows drawing a hand frame and drawing a three-dimensional hand reference using 21 key points from just one frame. The Fig. 4 shows the process of selecting the area of interest, that is, the area of the detected hand (these examples in the pictures were made in the laboratory by our scientists, with their permission).

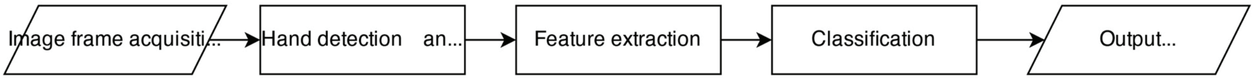


Fig. 2. The process of the classification algorithm

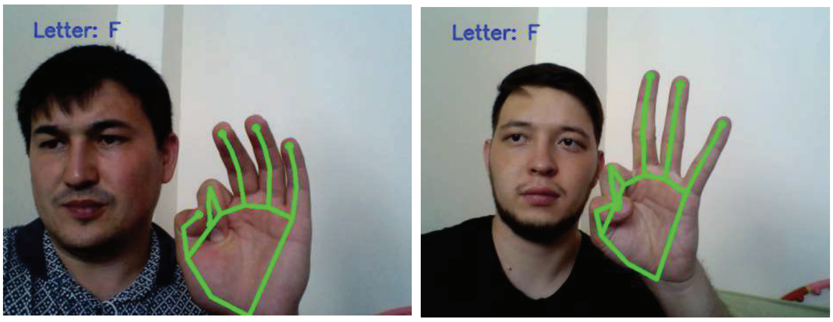


Fig. 3. Drawing a hand frame

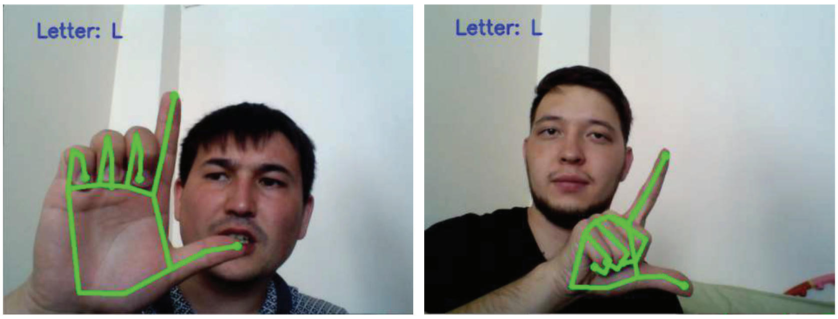


Fig. 4. Selecting a hand area

The frame of the hand (Fig. 5) is moved to another empty window, and the image is saved. The program code for saving the image is shown in Fig. 6.

The first version of the Kazakh alphabet consists of 42 letters. The development of a database for the Kazakh sign language, consisting of a dactylic alphabet of 42 gestures, is the initial step in creating a system for automatic recognition of individual hand gestures. The dactylic alphabet for the first Kazakh sign language is shown in Fig. 7.

In 2017, a decree was signed on the transition of the Kazakh alphabet from Cyrillic to Latin. After the changes, the new Kazakh alphabet includes 31 characters of the Latin alphabet, which completely covers all the sounds of the Kazakh language. This article is relevant because research related to gesture recognition of the updated Kazakh alphabet has not yet been conducted. Fig. 8 shows a dataset of 31 gestures, each gesture corresponds to one letter of the new Kazakh alphabet.

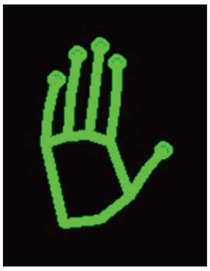
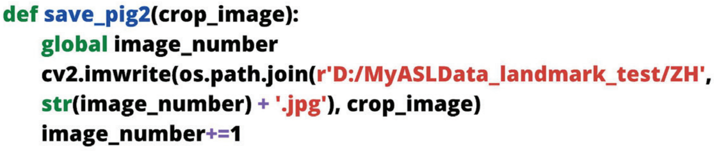


Fig. 5. A hand frame



```
def save_pig2(crop_image):
    global image_number
    cv2.imwrite(os.path.join(r'D:/MyASLData_landmark_test/ZH',
    str(image_number) + '.jpg'), crop_image)
    image_number+=1
```

Fig. 6. Saving an image

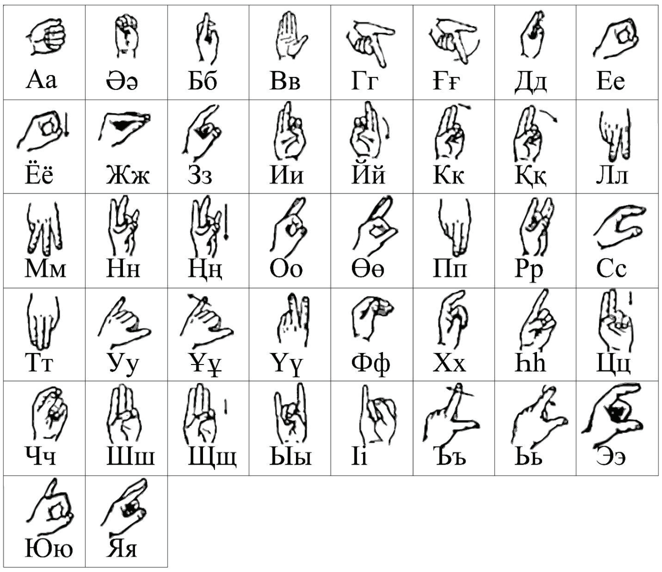


Fig. 7. The first Kazakh dactyl alphabet [28]

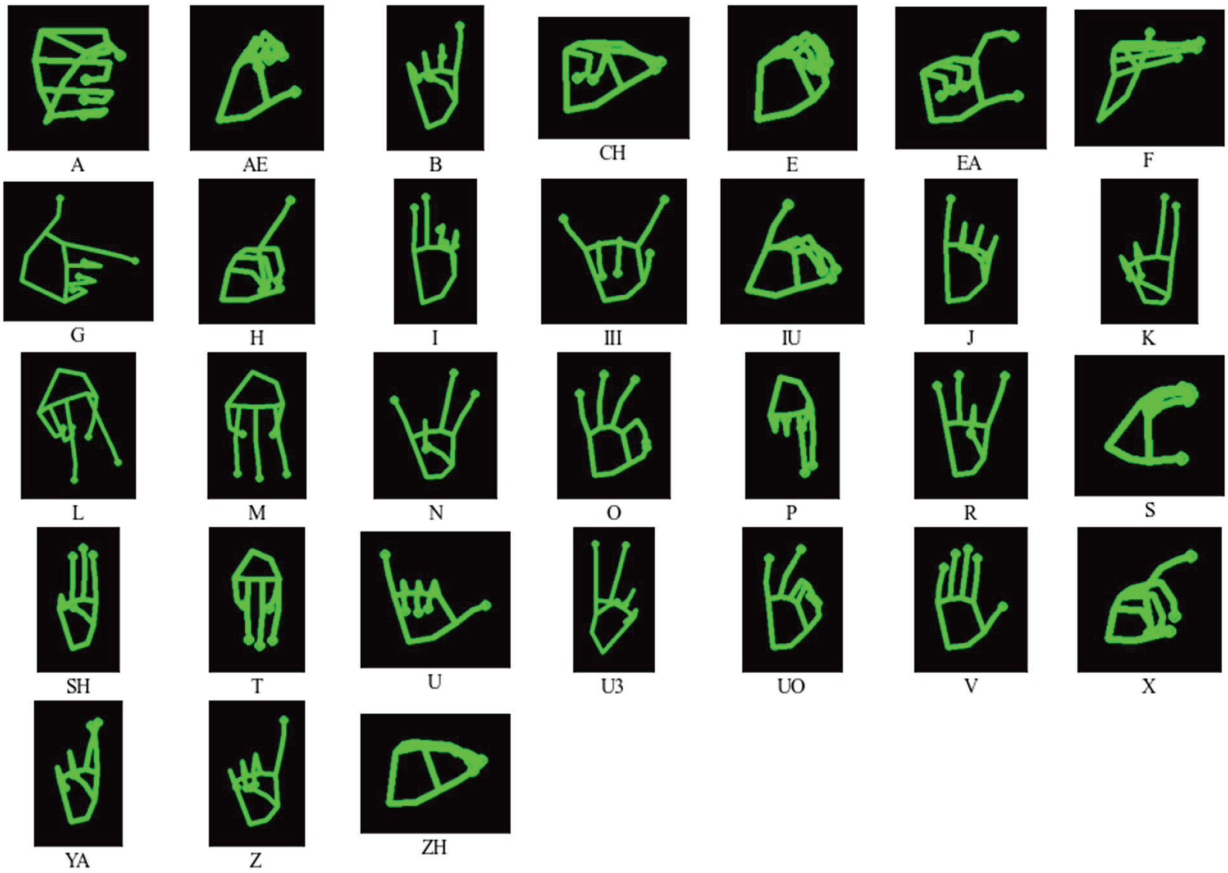


Fig. 8. Creating a dataset

After forming the dataset, we proceed to the recognition process. The collected data is divided into training and test data. For the correct recognition and classification of the Kazakh sign language, the machine learning algorithms, specified in section 4, were applied.

5. 2. Development of a program for recognizing the Kazakh sign language

After the dataset is collected from the image for each gesture, a program for recognizing the Kazakh sign language using machine learning methods will be implemented. As shown in the pseudocode of the algorithm (Fig. 9), real-time streaming video is accepted as input. Then the process of reading the capture from the camera comes. If the hand area is in the frame, the coordinate calculation function is performed. The coordinates are calculated based on the key points found. After the coordinates are determined, the function of drawing the frame of the hand is performed. The resulting image is converted to an array of data and goes to the classification function. As a result of the classification, you will get a text with a label about the gesture. Otherwise, when the hand is out of the frame, the label "None" is displayed since no gesture will be detected.

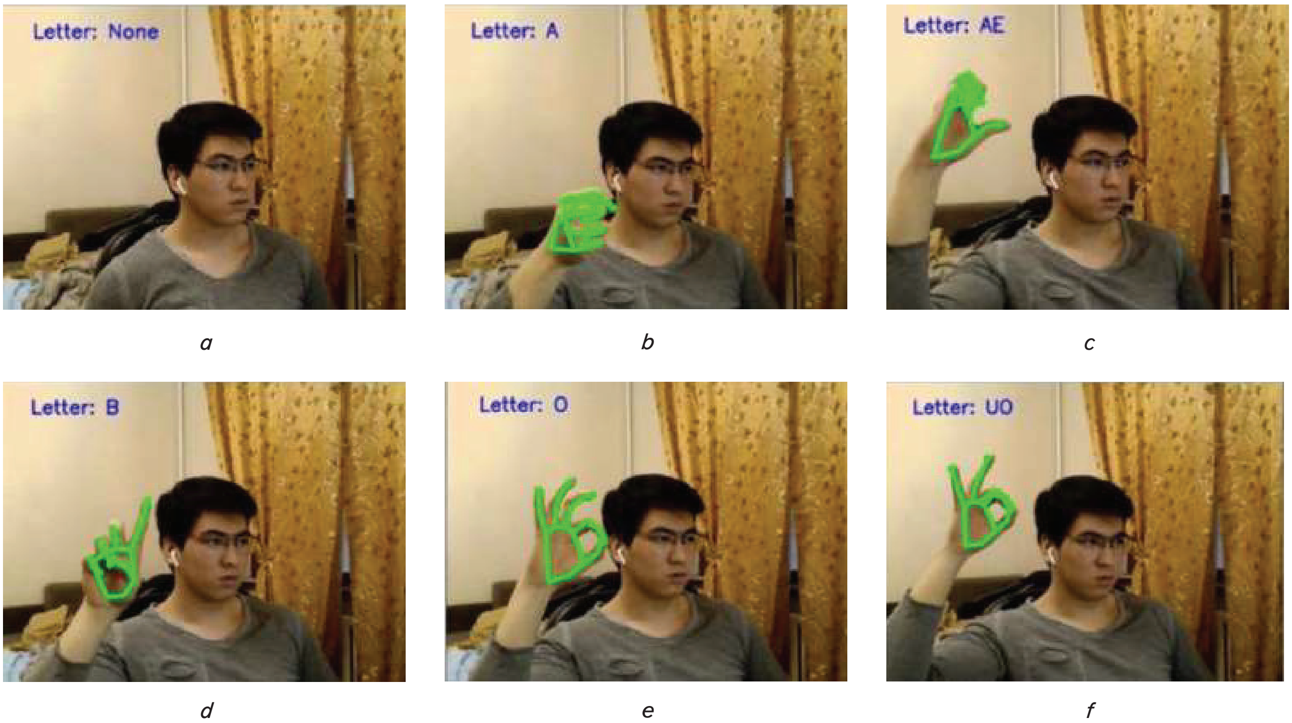
```
Input: Live-streaming video
Output: Live-streaming video with name of gesture
While True:
    Read capture from camera
    If hand in the frame:
        Function coordinate calculating
        Function drawing a hand frame
        Function image conversion
        Function classification
        Text(result of classification)
    Else:
        Text('Letter: None')

    If press Esc:
        Break (close)
```

Fig. 9. Pseudocode of the algorithm process

Fig. 10 shows the output results of the labels corresponding to each class of hand frames shown.

Fig. 11 shows the result of detecting each gesture class in the dataset.

The dataset consists of 31 classes. Each class contains more than 5000 drawings of the hand frame for a single gesture. As shown in the figure, the recognition of gestures, corresponding to the letters of the Kazakh alphabet, occurs in real time.



a *b* *c*

d *e* *f*

Fig. 10. Gesture recognition result: *a* — no gesture detected in the process; *b* — recognition of the letter “a”; *c* — recognition of the letter “ae”; *d* — recognition of the letter “b”; *e* — recognition of the letter “o”; *f* — recognition of the letter “uo”

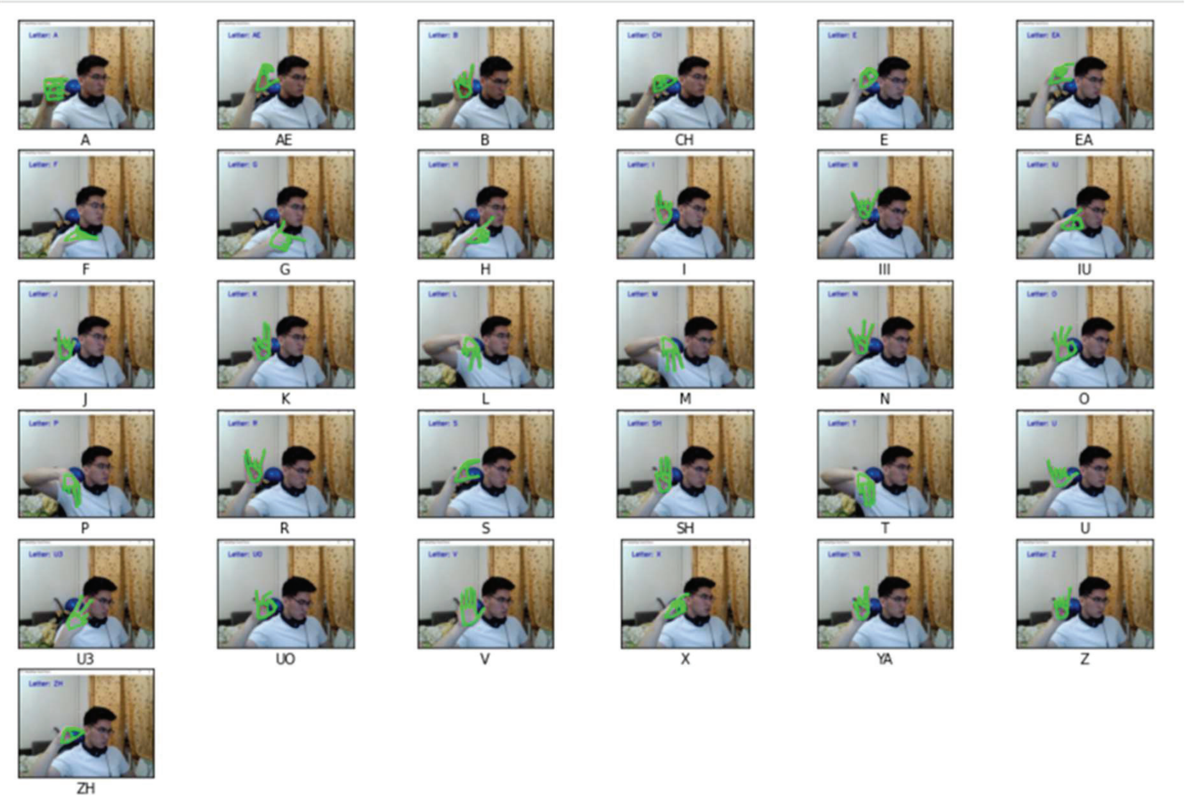


Fig. 11. Result of detecting all classes

5. 3. Estimating the quality of the algorithms

The next stage evaluates the quality of classifiers using the precision, recall, f1-measure, and accuracy metrics. To evaluate the accuracy of the classifier, a confusion matrix is used, which consists of combinations of predicted and target values. The values, predicted by the classifier, are described as positive and negative values, and the actual values are defined as true and false values. The first combination of the confusion matrix represents a true-positive solution (TP): the case of the prediction, matching the actual value, and the false-positive solution (FP) has the error of predicting a positive result, when in fact, it is negative. And the combination of a false-negative solution (FN) is responsible for a negative result that coincides with the prediction of the machine learning model.

Precision represents the percentage of objects that are called positive by the classifier and are actually positive.

$$\text{Precision} = \frac{TP}{TP+FP}. \tag{1}$$

That is, it is responsible for the ability to distinguish a given class from other classes. When the model makes many incorrect positive classifications, the value of this metric decreases.

Recall measures the model's ability to detect samples that belong to the positive class. It is responsible for the ability to detect a particular class. Recall takes into account the correctness of the prediction of all positive samples.

$$\text{Recall} = \frac{TP}{TP+FN}. \tag{2}$$

However, it ignores the erroneous classification of representatives of negative ones, predicted as positive. And the f1-measure contains information about these two metrics, defined as their average harmonic value.

$$F1 = \frac{2*TP}{2*TP+FN+FP}. \tag{3}$$

Accuracy is a metric that describes the overall accuracy of the model classification across all classes.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}. \tag{4}$$

Table 1 shows the value of these metrics for each class, and it shows that the overall accuracy of each class is at least 98–99 %.

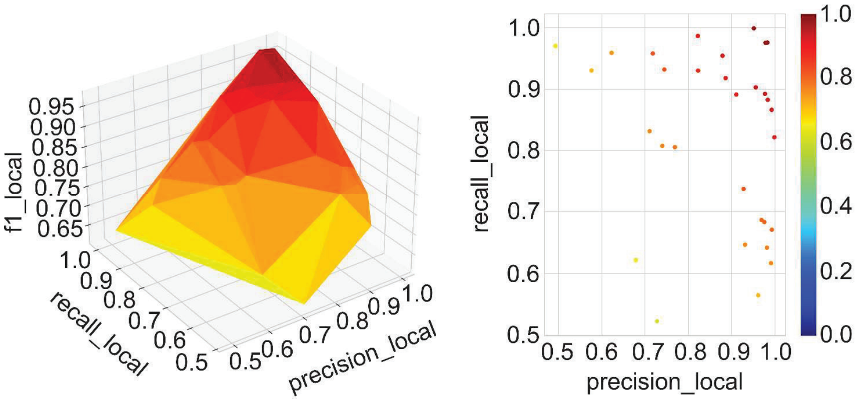
Fig. 12 shows the accuracy and completeness diagram (x and y axes, respectively) and their corresponding $F1$ score (z-axis) for the Random Forest classifier. When the precision value reaches one, and the completeness is zero, the $F1$ measure remains 0, ignoring the precision. If one parameter is small, then the second parameter will not matter, since the $F1$ measure emphasises the smallest value. Using the colour indicator, shown on the right side of the picture, you can see the ratio of accuracy and completeness for each class.

Fig. 13 shows the total confusion matrix of the Random Forest classifier and the number of values of each combination of this matrix. The FN and FP values are 5651, which shows classification errors. The number of correctly predicted positive solutions is 26538.

Table 1

Metrics for evaluating the quality of the Random Forest classifier

Class	Precision	Recall	*F*1-measure	Accuracy
0	0.6226	0.9591	0.7551	0.980
1	0.6786	0.6214	0.6487	0.978
2	0.5762	0.9304	0.7117	0.976
3	0.9909	0.6165	0.7601	0.987
4	0.9988	0.8217	0.9016	0.994
5	0.9108	0.8912	0.9009	0.993
6	0.7105	0.8316	0.7663	0.983
7	0.9513	0.9990	0.9746	0.998
8	0.7445	0.9319	0.8277	0.987
9	0.8225	0.9302	0.8730	0.991
10	0.8861	0.9180	0.9018	0.993
11	0.9310	0.6464	0.7630	0.986
12	0.9275	0.7376	0.8217	0.989
13	0.9925	0.8664	0.9251	0.995
14	0.4927	0.9703	0.6535	0.965
15	0.9816	0.6416	0.7760	0.988
16	0.7398	0.8077	0.7723	0.984
17	0.9692	0.6864	0.8036	0.989
18	0.7276	0.5225	0.6082	0.977
19	0.9929	0.6708	0.8007	0.989
20	0.8219	0.9866	0.8968	0.992
21	0.9834	0.8826	0.9303	0.995
22	0.9755	0.6832	0.8036	0.989
23	0.9783	0.9755	0.9769	0.998
24	0.9613	0.5645	0.7113	0.984
25	0.8789	0.9543	0.9151	0.994
26	0.9771	0.8924	0.9328	0.995
27	0.9825	0.9758	0.9791	0.998
28	0.9556	0.9030	0.9285	0.995
29	0.7689	0.8057	0.7869	0.985
30	0.7179	0.9580	0.8208	0.986
Average	0.8596	0.8252	0.8420	0.988



a

b

Fig. 12. The ratio of quality metrics of the RF algorithm: *a* – the ratio of precision, recall and f1 metrics; *b* – the ratio of precision and recall metrics

Another metric for assessing the classification quality is the ROC curve, which represents a graph of the relationship between true-positive and false-positive indicators.

The quantitative interpretation of this curve gives an indicator of the area of the AUC (Fig. 14), bounded by the ROC curve and the axis of the proportion of false-positive classifications. The higher the AUC result, the better the classifier works. Table 2 shows the numerical AUC values of the Random Forest classifier for each class.

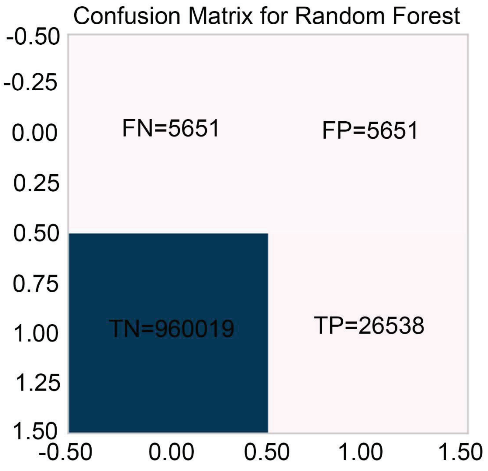


Fig. 13. Confusion matrix of the RF classifier

Table 3 shows the quality metrics of the Support Vector Machine classifier. According to the data, you can see that this algorithm was slightly mistaken in the classification of objects of the 1st and 7th class. In other cases, it showed good results.

Fig. 15 shows the accuracy and completeness diagram (x and y axes, respectively) and their corresponding $F1$ score (z-axis) for the Support Vector Machine classifier. When the precision value reaches one, and the completeness is zero, the $F1$ measure remains 0, ignoring the precision. If one of the parameters is small, then the second parameter will not matter, since the $F1$ measure emphasises the smallest value. Using the colour indicator, shown on the right side of the picture, you can see the ratio of accuracy and completeness for each class.

Table 2

Quantitative interpretation of the ROC curve of the RF classifier

Class	AUC ROC
0	0.990
1	0.989
2	0.984
3	0.981
4	0.999
5	0.995
6	0.985
7	0.999
8	0.997
9	0.998
10	0.997
10	0.997
11	0.995
12	0.992
13	0.994
14	0.991
15	0.994
16	0.987
17	0.978
18	0.965
19	0.996
20	0.999
21	0.993
22	0.969
23	0.998
24	0.971
25	0.998
26	0.991
27	0.999
28	0.992
29	0.977
30	0.998
Average	**0.990**

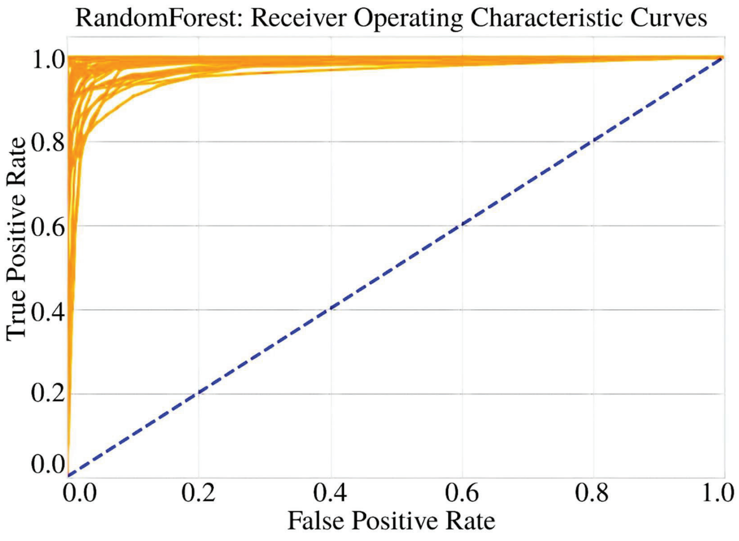


Fig. 14. ROC curves for each class of the RF classifier

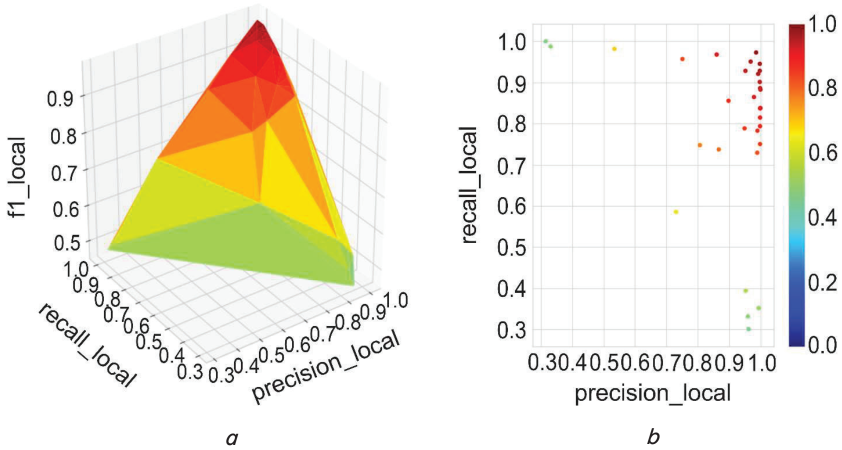


Fig. 15. The ratio of quality metrics of the SVM algorithm: *a* — the ratio of precision, recall and f1 metrics; *b* — the ratio of precision and recall metrics

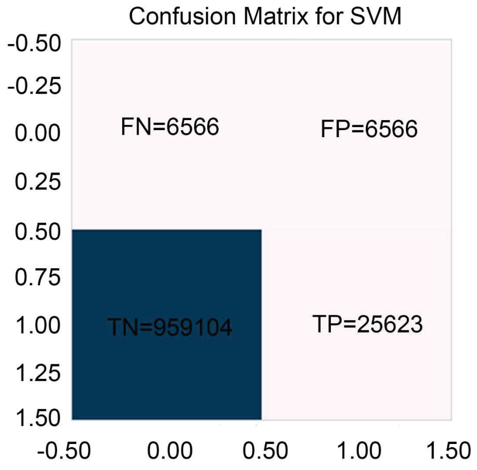


Fig. 16. SVM classifier confusion matrix

Another metric for assessing the classification quality is the ROC curve, which represents a graph of the relationship between true-positive and false-positive indicators.

The quantitative interpretation of this curve is given by the AUC area indicator (Fig. 17), which is bounded by the ROC curve and the axis of the proportion of false-positive classifications. The higher the AUC result, the better the classifier works. Table 4 shows the numerical AUC values of the Support Vector Machine classifier for each class.

Table 5 shows the quality metrics of the XGBoost classifier. The Recall metric reaches the lowest value in detecting Class 1 objects. The ability to distinguish one class from other classes, the Precision metric, showed good results.

Fig. 18 shows the accuracy and completeness diagram (*x* and *y* axes, respectively) and their corresponding *F*1 score (*z*-axis) for the XGBoost classifier. When the precision value reaches one, and the completeness is zero, the *F*1 measure remains 0, ignoring the precision. If one of the parameters is small, then the second parameter will not matter, since the *F*1 measure emphasises the smallest value. Using the colour indicator, shown on the right side of the picture, you can see the ratio of accuracy and completeness for each class.

Fig. 19 shows the total XGBoost confusion matrix and the number of values for each matrix combination. The true-positive solutions (TP) indicators are 24917, and the true-negative solutions (TN) are 958398. The classification errors for false-positive (FP) and false-negative (FN) solutions are 7272 each.

Another metric for assessing the classification quality is the ROC curve, which represents a graph of the ratio between true-positive and false-positive indicators.

The quantitative interpretation of this curve is given by the AUC area (Fig. 20) indicator, which is bounded by the ROC curve and the axis of the proportion of false-positive classifications. The higher the AUC result, the better the classifier works. Table 6 shows the numerical AUC values of the XGBoost classifier for each class.

As shown in the Table 6, the values of the AUC ROC metrics are in the range between 0.92 and 0.99, which proves the good quality of the algorithm. In this section, an assessment of 5 metrics was made to check the quality of algorithms.

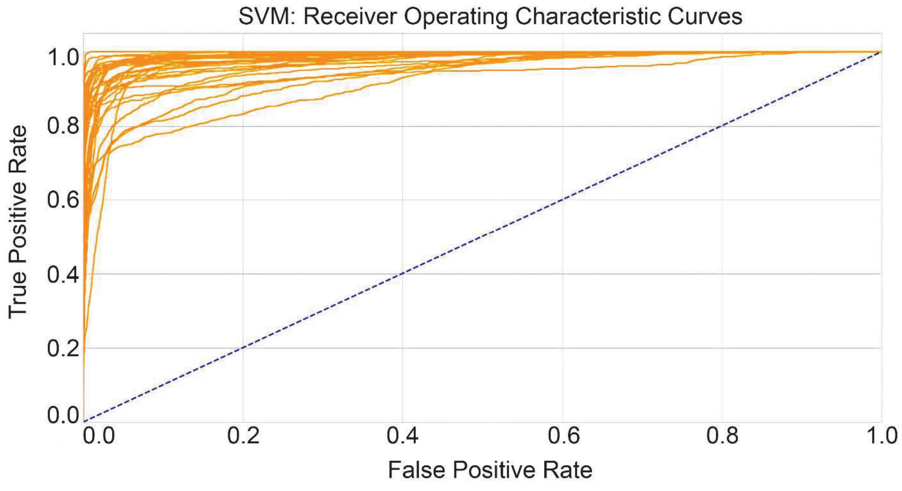
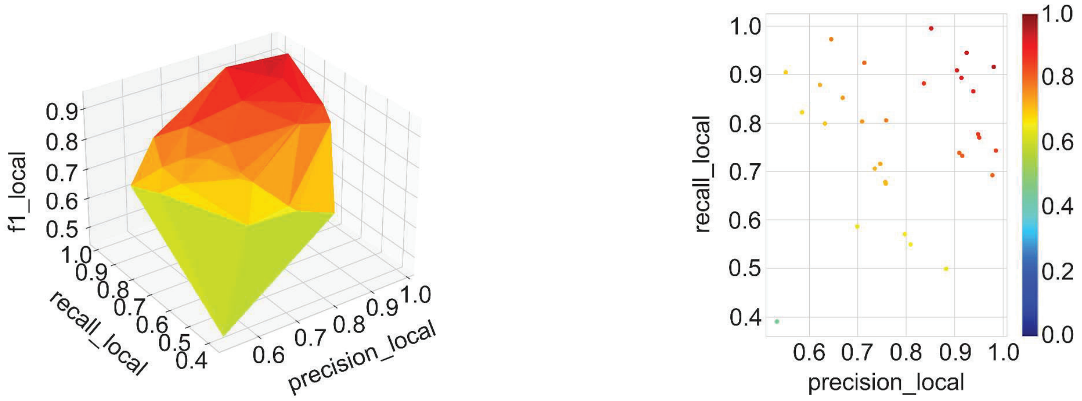


Fig. 17. ROC curves for each class of the SVM classifier



a *b*

Fig. 18. The ratio of the quality metrics of the XGBoost algorithm: *a* – the ratio of precision, recall and f1 metrics; *b* – the ratio of precision and recall metrics

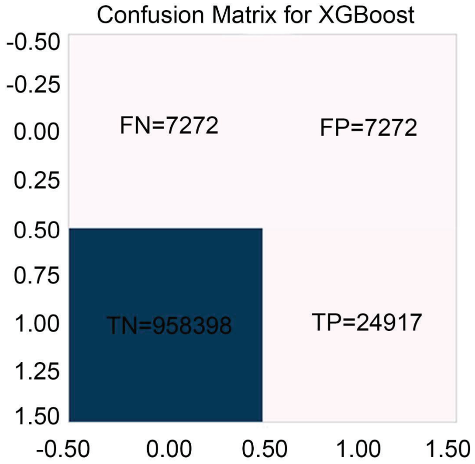


Fig. 19. XGBoost classifier confusion matrix

Table 3

Metrics for evaluating the quality of the SVM classifier

Class	Precision	Recall	*F*1-measure	Accuracy
0	0.9862	0.9727	0.9794	0.998
1	0.3306	0.9873	0.4953	0.988
2	0.9898	0.7293	0.8398	0.936
3	0.9979	0.9012	0.9470	0.990
4	0.9979	0.9012	0.9470	0.996
5	0.9988	0.8147	0.8974	0.994
6	0.7513	0.9569	0.8418	0.988
7	0.3150	1.0000	0.4791	0.930
8	0.9688	0.9506	0.9596	0.997
9	0.8982	0.8555	0.8763	0.992
10	0.8604	0.9678	0.9109	0.993
11	0.9532	0.3941	0.5577	0.978
12	0.7308	0.5855	0.6501	0.980
13	0.9988	0.7505	0.8570	0.991
14	0.5341	0.9814	0.6917	0.970
15	0.9987	0.7938	0.8846	0.993
16	0.8071	0.7478	0.7763	0.986
17	0.9941	0.3520	0.5199	0.980
18	0.9601	0.3315	0.4928	0.976
19	0.9795	0.8645	0.9184	0.994
20	0.9928	0.9208	0.9554	0.997
21	1.0000	0.8378	0.9117	0.994

Ccontinuation of Table 3

22	1.0000	0.8826	0.9377	0.996
23	0.9989	0.8862	0.9392	0.996
24	0.9622	0.3009	0.4584	0.975
25	0.9904	0.7829	0.8745	0.992
26	0.9989	0.8371	0.9109	0.994
27	0.9979	0.9284	0.9619	0.997
28	0.9522	0.9282	0.9400	0.996
29	0.9493	0.7885	0.8615	0.991
30	0.9979	0.9453	0.9709	0.998
Average	**0.8955**	**0.7971**	**0.8435**	**0.986**

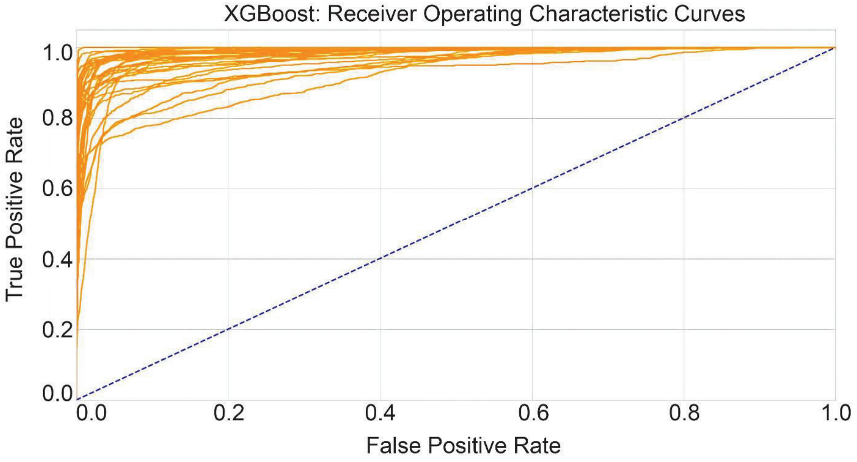


Fig. 20. ROC curves for each class of the XGBoost classifier

Table 6

Quantitative interpretation of the ROC curve of the XGBoost classifier

Class	AUC ROC
0	0.994
1	0.972
2	0.982
3	0.942
4	0.974
5	0.980
6	0.993
7	0.999
8	0.963
9	0.987
10	0.996
11	0.993
12	0.986
13	0.994
14	0.990
15	0.952
16	0.989
17	0.935
18	0.961
19	0.945
20	0.993
21	0.990
22	0.978
23	0.989
24	0.922
25	0.996
26	0.985

Continuation of Table 4

27	0.992
28	0.988
29	0.973
30	0.977
Average	**0.977**

Table 4

Quantitative interpretation of the ROC curve of the SVM classifier

Class	AUC ROC
0	0.994
1	0.972
2	0.982
3	0.942
4	0.974
5	0.980
6	0.993
7	0.999
8	0.963
9	0.987
10	0.996
Class	AUC ROC
10	0.996
11	0.993
12	0.986
13	0.994
14	0.990
15	0.952
16	0.989
17	0.935
18	0.961
19	0.945
20	0.993
Class	AUC ROC
21	0.990
22	0.978
23	0.989
24	0.922
25	0.996
26	0.985
27	0.992
28	0.988
29	0.973
30	0.977
Average	**0.977**

Table 5

Metrics for evaluating the quality of the XGBoost classifier

Class	Precision	Recall	F1-measure	Accuracy
0	0.7135	0.9241	0.8053	0.9857
1	0.5332	0.3905	0.4508	0.9701
2	0.6690	0.8520	0.7495	0.9819
3	0.8821	0.4991	0.6375	0.9812
4	0.9480	0.7771	0.8541	0.9914
5	0.7464	0.7157	0.7307	0.9832
6	0.6220	0.8785	0.7283	0.9787
7	0.8514	0.9952	0.9177	0.9942
8	0.7584	0.8055	0.7812	0.9858
9	0.7089	0.8031	0.7531	0.9831

Ccontinuation of Table 4

10	0.6453	0.9727	0.7759	0.9821
11	0.9093	0.7385	0.8150	0.9886
12	0.7578	0.6750	0.7140	0.9830
13	0.9382	0.8654	0.9003	0.9936
14	0.5515	0.9045	0.6852	0.9721
15	0.7567	0.6787	0.7156	0.9832
16	0.6323	0.7990	0.7059	0.9785
17	0.6989	0.5857	0.6373	0.9800
18	0.7972	0.5702	0.6649	0.9805
19	0.9848	0.7433	0.8472	0.9912
20	0.8362	0.8817	0.8583	0.9905
21	0.9775	0.6925	0.8107	0.9899
22	0.9505	0.7700	0.8508	0.9912
23	0.9137	0.8930	0.9032	0.9939
24	0.8086	0.5491	0.6540	0.9801
25	0.9245	0.9448	0.9345	0.9956
26	0.9155	0.7324	0.8138	0.9890
27	0.9803	0.9158	0.9469	0.9967
28	0.9044	0.9086	0.9065	0.9937
29	0.7351	0.7062	0.7204	0.9821
30	0.5852	0.8221	0.6837	0.9758
Average	0.7947	0.7738	0.7841	0.985

6. Discussion of experimental results of comparative analyzes of algorithms, obtained during the study

In this paper, a system for recognizing the Kazakh dactylic sign language, consisting of the dactylic alphabet of 31 gestures in real time, has been developed.

The total XGBoost confusion matrix and the number of values for each matrix combination. The true-positive solutions (TP) indicators are 24917, and the true-negative solutions (TN) are 958398. The classification errors for false-positive (FP) and false-negative (FN) solutions are 7272 each.

In other scientific studies [6, 20, 21] of gesture speech recognition, the support vector method was also used, which we also used in our work. But compared to previous works, our recognition accuracy is high. The peculiarity of the method, proposed in our work, is the combination of static and dynamic data types into one database, which makes it possible to interpret gestures in real mode (dynamic gestures), as well as in cases when there is no need to track hands (static gestures).

For this task, there are such limitations as the quality of camera visibility, the quality of illumination of the recognition zone, also the main problem is the moderate use of resources, since it has limitations on computing devices, etc.

The advantage of this work is the high recognition accuracy, which is very important for use in human-machine communication systems. Also, our research work is one of the first works that implemented a gesture recognition system for the updated Kazakh alphabet.

As a lack of research, we can note FPS drawdowns, which affect the speed of recognition of machine learning algorithms. In the future, parallelization is planned in order to improve the performance and increase the speed of the algorithms. To do this, we consider solutions to problems due to the training time that arise when working with a large number of training examples.

7. Conclusion

1. The presented research work is aimed at the correct recognition of the Kazakh sign language. To achieve this goal, a dataset was created that contains more than 5000 images for each 31 gestures. With fewer photographs used, our results were less accurate. Lighting also affects the quality of recognition, and we took this parameter into account in order for our development to give a satisfactory result.

2. The classification of gestures was carried out according to three classification algorithms. The average accuracy of the Random Forest classifier was 98.86 %, the SVM algorithm showed 98.68 % accuracy, and XGBoost has a result of 98.54 % correct recognition. In addition, the classifier's quality is evaluated by the speed of execution and the performance of the algorithm. In terms of training time, Random Forest was faster than the support vector machine and XGBoost. To check the accuracy, cross-validation was performed, where the data was divided into five blocks. As for the speed of predicting in a real-time task, Random Forest, although it won in the learning speed, is inferior in the execution speed, as FPS drawdowns begin. Thus, the prediction accuracy in the three methods is about the same. However, SVM and XGBoost have shown themselves to be better due to execution speed when working in real-time.

3. The conducted research allowed us to draw the following conclusions based on the estimates of the algorithms: the average precision for the RF algorithm was 0.859, for the SVM algorithm was 0.895 and for XGBoost was 0.794. The average recall for the RF was 0.825, for the

SVM algorithm was 0.797, and for XGBoost was 0.773. For most classes, these metrics showed good results, here is the average value of these metrics for each algorithm. In the future, it is planned to improve the performance of these classifiers by parallelizing them using CUDA and OpenCL technologies.

Acknowledgment

This research is funded by grant from the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan within the framework of the Project № AP08053034.

References

1. Ahmed, M. A., Zaidan, B. B., Zaidan, A. A., Salih, M. M., Lakulu, M. M. bin. (2018). A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017. Sensors, 18 (7), 2208. doi: http://doi.org/10.3390/s18072208
2. Bilgin, M., Mutludogan, K. (2019). American Sign Language Character Recognition with Capsule Networks. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). Ankara. doi: http://doi.org/10.1109/ismsit.2019.8932829
3. Kudubayeva, S., Zhussupova, B., Aliyeva, G. (2019). Features of the representation of the Kazakh sign language with the use of gestural notation. Proceedings of the 5th International Conference on Engineering and MIS. doi: http://doi.org/10.1145/3330431.3330440
4. Luqman, H., El-Alfy, E.-S. M., BinMakhashen, G. M. (2020). Joint space representation and recognition of sign language fingerspelling using Gabor filter and convolutional neural network. Multimedia Tools and Applications, 80 (7), 10213–10234. doi: http://doi.org/10.1007/s11042-020-09994-0
5. Saykol, E., Türe, H. T., Sirvanci, A. M., Turan, M. (2016). Posture labeling based gesture classification for Turkish sign language using depth values. Kybernetes, 45 (4), 604–621. doi: http://doi.org/10.1108/k-04-2015-0107
6. Van Houdt, G., Mosquera, C., Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53 (8), 5929–5955. doi: http://doi.org/10.1007/s10462-020-09838-1
7. Makarov, I., Chertkov, M., Veldyaykin, N., Pokoev, A. (2019). American and Russian sign language dactyl recognition. ACM International Conference Proceeding Series, 204–210. doi: http://doi.org/10.1109/tsp.2019.8768868
8. Artyukhin, S. G., Mestetskiy, L. M. (2015). Dactyl alphabet gesture recognition in a video sequence using microsoft kinect. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-5/W6, 83–86. doi: http://doi.org/10.5194/isprsarchives-xl-5-w6-83-2015
9. Oktekin, B. (2018). Development of Turkish sign language recognition application. Nicosia.
10. Wadhawan, A., Kumar, P. (2019). Sign Language Recognition Systems: A Decade Systematic Literature Review. Archives of Computational Methods in Engineering, 28 (3), 785–813. doi: http://doi.org/10.1007/s11831-019-09384-2
11. Munib, Q., Habeeb, M., Takruri, B., Al-Malik, H. A. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. Expert Systems with Applications, 32 (1), 24–37. doi: http://doi.org/10.1016/j.eswa.2005.11.018
12. Oz, C., Leu, M. C. (2007). Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. Neurocomputing, 70 (16-18), 2891–2901. doi: http://doi.org/10.1016/j.neucom.2006.04.016
13. 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. doi: http://doi.org/10.1016/j.engappai.2011.06.015
14. Sun, C., Zhang, T., Bao, B.-K., Xu, C., Mei, T. (2013). Discriminative Exemplar Coding for Sign Language Recognition With Kinect. IEEE Transactions on Cybernetics, 43 (5), 1418–1428. doi: http://doi.org/10.1109/tcyb.2013.2265337
15. Chuan, C.-H., Regina, E., Guardino, C. (2014). American Sign Language Recognition Using Leap Motion Sensor. 2014 13th International Conference on Machine Learning and Applications, 541–544. doi: http://doi.org/10.1109/icmla.2014.110
16. Tangsuksant, W., Adhan, S., Pintavirooj, C. (2014). American Sign Language recognition by using 3D geometric invariant feature and ANN classification. The 7th 2014 Biomedical Engineering International Conference. doi: http://doi.org/10.1109/bmeicon.2014.7017372
17. Zamani, M., Kanan, H. R. (2014). Saliency based alphabet and numbers of American sign language recognition using linear feature extraction. 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), 398–403. doi: http://doi.org/10.1109/iccke.2014.6993442
18. Savur, C., Sahin, F. (2016). American Sign Language Recognition system by using surface EMG signal. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2872–2877. doi: http://doi.org/10.1109/smc.2016.7844675
19. Saha, S., Lahiri, R., Konar, A., Nagar, A. K. (2016). A novel approach to American sign language recognition using Madaline neural network. 2016 IEEE Symposium Series on Computational Intelligence (SSCI). doi: http://doi.org/10.1109/ssci.2016.7850121
20. Kudubayeva, S. A., Ryumin, D. A., Kalzhanov, M. U. (2016). The method of basis vectors for recognition sign languageby using sensor KINECT, KazNU Bulletin. Mathematics, Mechanics, Computer Science Series, 3 (91), 86–96.
21. Gupta, R., Bhatnagar, A. S. (2021). Multi-stage Indian sign language classification with Sensor Modality Assessment. 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS). doi: http://doi.org/10.1109/icaccs51430.2021.9441906