IMPACT OF IMAGE PREPROCESSING ON RECOGNITION OF LETTERS OF SIGN LANGUAGE

Paweł Abramowicz, Magdalena Topczewska

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: The article concerns the problem of the selected sign language letters in the form of images classification. The impact of the image preprocessing methods as adaptive thresholding or edge detection is tested. In addition, the influence of the found shapes filling is checked, as well as centering the hands on the images. The following classification methods were chosen: SVM classifier with linear kernel function, Naive Bayes and Random Forests. The accuracy, F-measure, the AUC, MAE and Kappa coefficient were reported as measures of classification quality.

Keywords: image preprocessing, sign language, classification

1. Introduction

In Poland according to the Universal Newborn Hearing Screening Program 3 children out of 1000 are born with hearing impairment [12], while the report of the Central Statistical Office shouts that 14% of people ranging in age from 15 to 70 have hearing defect [18,5]. Thus the problem may be present from birth or may also occur at a later age. Some deaf or hearing-impaired people use a sign language as a form of communication and expressing emotions.

There are several hundred sign languages around the world. Each sign language consists of ideographic and dactylographic signs. Ideographic signs can be considered as equivalents of short phrases in spoken language, while among dactylographic signs finger alphabet, characters assigned to punctuation or numerals may be mentioned.

To recognize hand postures many different techniques can be applied not only to sign language applications, but also to games or human-computer interaction sys-

Advances in Computer Science Research, vol. 14, pp. 5-16, 2018. DOI: 10.24427/acsr-2018-vol14-0001

tems. Some approaches concern the preprocessing of images, some – feature extraction. The differences also apply to the classification methods. For instance, in [16] a cascade classifier applying AdaBoost method was used to separate 21 letters in Thai finger-spelling. The main feature of hand postures was an object detection approach based on Histogram of Orientation Gradient (HOG). In case of [17], detection of the finger contour using hidden Markov models in the American Sign Language gestures was described, while in [13] Canny edge detection and boundary tracing algorithm were applied to detect fingers location. Additionally, in many applications contour detection techniques were applied to extract features, for instance, in [14] the Moore–Neighbor algorithm was used to obtain the external shape of every image. Next, the number of pixels to form the shape was reduced, and finally neural networks were implemented to classify objects. Finger detection could also be achieved by a color segmentation and a contour extraction [9,7]. The SVM method and HOG descriptors were used to recognize Arabic Sign Language alphabet [2].

In this paper selected methods of image preprocessing were used to verify the ability to improve the classification quality of the chosen finger alphabet letters collected as photographs. The comparison between original images and images after preprocessing (adaptive thresholding or edge detection) was performed. Additionally, the results of two more experiments were obtained and confronted – shapes filling and centering. All mentioned transformations were compared using three well-known classification algorithms (SVM, Naive Bayes, and Random Forests) and detailed performance measures as the classification quality, F-measure, the AUC, Kappa coefficient and MAE were reported.

The data set used in the experiments was a subset of the Hand Posture and Gesture Datasets [11] and contained four letters of the sign alphabet: A, B, C, and V. The set consisted of 191 elements. Each character was photographed on a dark and light background – 24 photographs were available for each character on a given background. Only the letter V was photographed 23 times on a dark background. A size of each photo was 128×128 pixels. The colors of the images were represented in the grayscale.

2. Selected methods of image preprocessing

In this paper two main approaches have been selected to convert original photographs into images that should enable methods of classification improvement of the allocation of the letters to classes: adaptive thresholding and edge detection. Next using obtained results shape filling and objects centering have been additionally applied to investigate the total impact on the results.

2.1 Thresholding

Thresholding is an image segmentation method that based on a colour or a grayscale image creates a binary image as a result [4,6]. The algorithm in its simplest form adapts the threshold value on each pixel comparing it to the intensity of a pixel. Pixels with intensity lower than the threshold are replaced by pixels with one colour (ex. black), while these with intensity greater that the threshold are replaced by pixels with the second colour (*maxValue*, ex. white):

$$destination(x,y) = \begin{cases} maxValue & \text{if } source(x,y) > threshold, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $source(x,y)$ is an intensity of a pixel, $threshold$ is set by the user.

Due to the fact that on the considered images hands showing sign alphabet letters are illuminated unevenly, a simple thresholding may not give the expected effect. Therefore, the adaptive thresholding is worth using.

The image is divided into separate regions and for each region the threshold is calculated separately. In addition to the parameter corresponding to the size of a region (blockSize), the c parameter as the global threshold is present. It is a constant that is subtracted from the average intensity of pixels in a given region. Hence, the method allows a user to reject background pixels where there is no differential intensity.

By controlling two mentioned parameters different final effects can be achieved.

2.2 Edge detection

The second method of image preprocessing considered in the paper is edge detection [19,10]. The main purpose is to identify points in the digital image in which the light intensity changes rapidly. The Canny's method was used to achieve it [3].

The Canny's algorithm starts by reducing noise in the image. Edge detection is very susceptible to a noise in the raw image and false edges may be created. To reduce a noise a 5x5 Gauss filter is applied to the image resulting in a slightly blurred image that is not affected by interference in a significant way.

In the next step, to detect horizontal and vertical edges, the gradient is searched using the Sobel's operator [10,15]. The operator returns the value of the first derivative for the vertical and for the horizontal direction. The value and the direction of the gradient can then be calculated.

Afterwards, non-maximum suppression pixels are removed, because they are not considered as a part of the edge. Therefore, only thin lines composed of individual pixels remain.

The last step of the algorithm is thresholding using the hysteresis to eliminate irrelevant edges that have a slope below a given threshold. The Canny's method uses two thresholds: lower and upper. If the pixel gradient is greater than the upper threshold, the pixel is considered as the edge. If the pixel gradient is smaller than the lower threshold, the pixel is discarded. Otherwise, if it is between the lower and the upper threshold, the pixel will only be accepted if it is connected to a pixel whose gradient value is above the upper threshold. By controlling these two parameters different final results can be obtained.

2.3 Shape filling and centering

In order to improve image recognition, two additional approaches have been implemented and tested. The first was to fill the shapes of hands obtained using methods described in two previous subsections, and the successive approach was to center the filled hands on the image.

The algorithm of the shape filling is as follows. A given row of the image is checked until the first pixel in appropriate colour occurs. Then the last pixel in such a colour is found. Finally, specified segment of pixels is filled with a chosen colour.

The hand centering algorithm works on a simple principle. To assess whether the photographed hand is in the middle of the image, the last row of pixels on which the wrist of the photographed person is visible is checked. Based on the first and the last pixel of the wrist, the current position of the hand is calculated. If the calculated center of the wrist is not in the center of the image, the hand is moved in the right direction, so that it is exactly in the middle.

3. Experiments

To perform the experiments the Hand Posture and Gesture Dataset [11] was chosen and A, B, C, and V letters of the sign alphabet. Additional impediment is the occurrence of three different types of a background of the photographs: white, black, and mixed. The number of chosen images every type is presented in Table 1.

Each image was processed into a feature vector with 16384 values of attributes. Finally, four experiments were performed using the own implementations and the Java - ML Library [1,8]:

- experiment 1: to examine the classification quality on the feature vectors created on the basis of original images;
- experiment 2: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection;

Table 1. Number of images for each letter and background colour

Letter	Background		
	white	black	mixed
A	24	24	48
B	24	24	48
C	24	24	48
V	24	23	47

- experiment 3: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection, then filled;
- experiment 4: to examine the classification quality on the feature vectors created on the basis of images after adaptive thresholding or edge detection, then filled and centered.

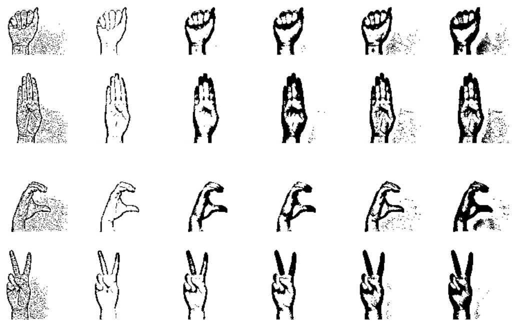

Fig. 1. Selected images after adaptive thresholding with a number of sample parameters

Weka system [20] was used to examine the classification quality with the 10-folds crossvalidation testing. The range of parameters for edge detection was from 10 to 140 every 10, while for adaptive learning it was from 3 to 23 every 2 for the first parameter and from 3 to 24 for the second parameter. The examples of images with a different set of parameters are presented in Figure 1 for adaptive thresholding for white background and in Figure 2 for edge detection and black background. The final dataset contained images as a result of all combinations of parameters to not have biased results by arbitrary choice of best parameters. The goal is to check if these methods work in general, thus the results should be interpreted as average results for methods, but not the best to obtain.

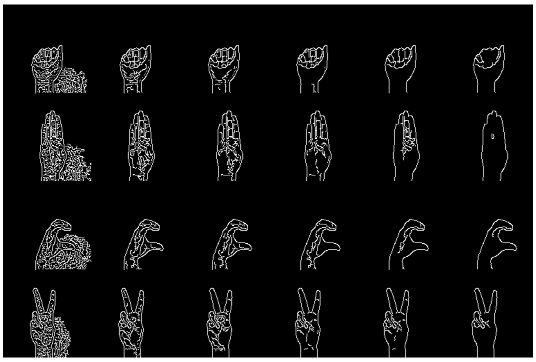

Fig. 2. Selected images after edge detection with a number of sample parameters

Detailed results for all experiments are aggregated in Table 2.

The classification quality (the accuracy) is presented as a percentage of the number of correctly classified objects over the number of all objects in a data set. F-measure considers both precision (the proportion of relevant objects that have been correctly classified over the total amount of objects classified as relevant) and recall (the proportion of relevant objects that have been correctly classified over the total amount of relevant objects) and signifies the harmonic mean of these two measures. The higher and closer to 1, the better the predictive property of a classifier. The Kappa coefficient describes the agreement of prediction with true class, and the value 1 signifies complete agreement. AUC signifies the area under the Receiver Operating Characteristic curve and also quantifies the classifier performance. It determines which of the used models predicts the classes best. It combines true positive rate (recall) and false positive rate (proportion of second class objects classified incorrectly as relevant over the total amount of second class objects). The closer AUC for a model comes to 1, the better it is. The last reported measure is the Mean Absolute Error (MAE). In the classification problem it is the sum over all the objects and their absolute error per object divided by the number of objects in the test set with an actual class label and zero means a perfect classification.

Table 2: Classification results (BC - Background Color; Cl - Classifier; MAE - Mean Absolute Error; F - mean value of F-measure; AUC - mean value of Area under the ROC curve; AT - Adaptive Thresholding; ED - Edge Detection

BC	Cl	Q	Kappa	MAE	F	AUC
Experiment 1						
Mixed	SVM	52.356	0.365	0.320	0.526	0.726
	Naive Bayes	32.984	0.107	0.335	0.338	0.574
	Random Forest	**67.539**	0.567	0.257	0.671	0.866
Black	SVM	**81.052**	0.747	0.275	0.810	0.891
	Naive Bayes	38.947	0.189	0.305	0.389	0.597
	Random Forest	64.211	0.522	0.266	0.628	0.807
White	SVM	**88.542**	0.847	0.264	0.886	0.947
	Naive Bayes	56.25	0.417	0.219	0.543	0.713
	Random Forest	69.792	0.597	0.216	0.695	0.923
AT − experiment 2						
Mixed	SVM	**76.937**	0.692	0.277	0.767	0.890
	Naive Bayes	70.499	0.607	0.148	0.705	0.880
	Random Forest	58.202	0.442	0.283	0.573	0.823
Black	SVM	66.238	0.550	0.296	0.659	0.813
	Naive Bayes	**66.672**	0.556	0.167	0.665	0.855
	Random Forest	48.882	0.317	0.316	0.477	0.732
White	SVM	**71.419**	0.619	0.281	0.712	0.878
	Naive Bayes	69.559	0.594	0.152	0.697	0.876
	Random Forest	56.577	0.421	0.286	0.556	0.822
ED − experiment 2						
Mixed	SVM	**81.136**	0.748	0.273	0.811	0.903
	Naive Bayes	76.589	0.688	0.117	0.766	0.920
	Random Forest	48.948	0.319	0.317	0.472	0.745
Black	SVM	**79.817**	0.730	0.276	0.790	0.881
	Naive Bayes	72.449	0.633	0.137	0.721	0.895
	Random Forest	44.930	0.264	0.331	0.434	0.697
White	SVM	**66.061**	0.547	0.291	0.657	0.833
	Naive Bayes	64.063	0.521	0.180	0.638	0.872
	Random Forest	40.795	0.211	0.339	0.394	0.673
AT, shape filling − experiment 3						
Mixed	SVM	**78.850**	0.718	0.277	0.788	0.891
	Naive Bayes	66.648	0.555	0.167	0.664	0.815
	Random Forest	72.515	0.633	0.204	0.720	0.907
Black	SVM	**67.377**	0.565	0.295	0.673	0.819
	Naive Bayes	62.679	0.503	0.187	0.620	0.789
	Random Forest	60.696	0.475	0.247	0.597	0.825
White	SVM	**82.825**	0.771	0.268	0.828	0.926
	Naive Bayes	67.209	0.563	0.164	0.659	0.844
	Random Forest	75.887	0.678	0.192	0.754	0.936
ED, shape filling − experiment 3						

BC	Cl	Q	Kappa	MAE	F	AUC
Mixed	SVM	**89.181**	0.856	0.264	0.892	0.948
	Naive Bayes	71.557	0.621	0.142	0.712	0.865
	Random Forest	82.781	0.770	0.155	0.826	0.958
Black	SVM	**83.340**	0.778	0.271	0.833	0.918
	Naive Bayes	66.853	0.558	0.166	0.658	0.828
	Random Forest	70.473	0.606	0.201	0.698	0.900
White	SVM	**84.471**	0.793	0.268	0.845	0.925
	Naive Bayes	66.210	0.549	0.169	0.648	0.835
	Random Forest	74.979	0.666	0.184	0.744	0.933
AT, shape filling, centering − experiment 4						
Mixed	SVM	**80.732**	0.743	0.275	0.807	0.899
	Naive Bayes	68.400	0.579	0.158	0.682	0.828
	Random Forest	74.393	0.658	0.182	0.741	0.914
Black	SVM	**72.318**	0.631	0.289	0.722	0.841
	Naive Bayes	65.411	0.539	0.173	0.650	0.797
	Random Forest	67.342	0.564	0.216	0.665	0.851
White	SVM	**85.580**	0.808	0.265	0.856	0.939
	Naive Bayes	76.438	0.686	0.118	0.763	0.886
	Random Forest	79.468	0.726	0.162	0.793	0.948
ED, shape filling, centering − experiment 4						
Mixed	SVM	**92.403**	0.899	0.259	0.924	0.964
	Naive Bayes	80.142	0.735	0.099	0.800	0.800
	Random Forest	85.543	0.807	0.125	0.854	0.965
Black	SVM	**91.096**	0.881	0.260	0.911	0.961
	Naive Bayes	73.566	0.648	0.132	0.732	0.850
	Random Forest	78.099	0.708	0.164	0.777	0.928
White	SVM	**89.371**	0.858	0.261	0.894	0.954
	Naive Bayes	82.068	0.761	0.090	0.820	0.903
	Random Forest	83.270	0.777	0.139	0.831	0.964

In case of the first experiment and analysis of original images, it can be observed that for all types of classifiers the accuracy for images with white background were higher than for other groups of images. The SVM classifier achieved the highest 88.54% correctness of classification, F-measure (0.886) and the area under the ROC curve (0.947), while Random Forest gave 69.79% of accuracy, F-measure at the level 0.695 and the area under the curve equaled 0.923. The lowest values can be observed for the Naive Bayes: accuracy - 56.25%, F-measure - 0.543 and the area under the curve - 0.713. The Kappa measure estimated the agreement between the original and obtained by the classifier belonging to the class as almost perfect for SVM (0.847), moderate for Random Forest (0.597), and fair for the Naive Bayes (0.417).

The images with black background had similar results as presented for images with white background. The approach with the highest results was SVM - the images

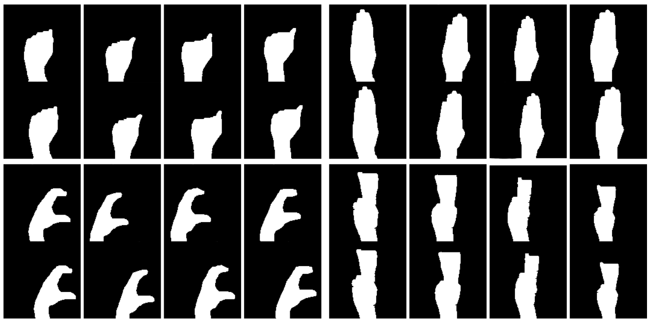

Fig. 3. Selected images after shape filling and edge detection versus centering, shape filling and edge detection

were the most precisely classified with 81.052% accuracy, F-measure 0.81 and the area under the ROC curve 0.891, while the Naive Bayes occurred to have the lowest results: accuracy equaled 38.947, F-measure 0.389 and the area under the ROC curve 0.597. The agreement of real and estimated class belonging was substantial in the case of SVM (0.747), moderate for Random Forest (0.522), while in the case of Naive Bayes it occurred slight.

In case of mixed background images the best results were obtained for Random Forest classifier (accuracy - 67.539; F-measure - 0.671; the area under the ROC curve - 0.866; Kappa coefficient - 0.567).

Analyzing the use of edge detection or adaptive thresholding in the second experiment, it can be observed that the application of these methods influenced the quality of classification, but not sufficiently. The biggest difference was achieved for edge detection, SVM classifier and mixed background images when the accuracy after edge detection increased by 28.78%, for Naive Bayes by 43.605%. A significant increase can be also mentioned for the black background images and Naive Bayes by about 33.502%. It can be concluded that pictures on a mixed background, which had previously dropped very poorly, now turned out to be the best, while the pictures on the black background from the middle position fell on the last. Differences between the best and the weakest result of the classification decreased.

Trying to achieve better classification results, the obtained hand shapes in the previous two experiments were filled (Fig. 3: the first and third row). Regardless of the edge detection approach final images do not differ practically. Unlike previous experiments, this time the best-classified objects are pictures taken against a white background. Each of the classifiers achieved the highest classification results

- 82.83% SVM, 67.21% Naive Bayes and 75.89% Random Forest. The best classifier for each type of background again turned out to be SVM. In the second place it was Random Forest, whose results were similar to SVM results for mixed and white background images. In case of images with black background, Random Forest managed poorer than the Naive Bayes, which in the remaining groups achieved the lowest results.

The last performed experiment was centering of hands, which underwent adaptive thresholding or edge detection and were filled previously. The received results in the form of images are presented in Figure 3 in the second and fourth row. In case of centering the filled hands after any approach of the edge detection, it can be noticed that it is impossible to define a group of images with the given background, on which the results were the best for each classifier. Whereas comparing classifiers in this paper in case of the SVM the results were the highest - the accuracy, the Kappa coefficient, F-measure or the area under the curve.

Statistical tests were carried out to determine the differences between the methods of image preprocessing. The variables were dependent and could not be described as normally distributed, thus the Friedman test for repeated blocks was applied and then post-hoc tests comparing a single pair of methods. Considering all preprocessing techniques a statistically significant difference in medians was found between this methods (Friedman F=6500; Kendall=0.546; p<0.0001). Similar results were obtained for edge detection (F=3000; Kendall=0.563; p<0.0001) and adaptive thresholding (F=3900; Kendall=0.594; p<0.0001). Comparing the background, the differences in the average results were also obrained: black (F=2000; Kendall=0.519; p<0.0001); mixed F=1200; Kendall=0.3167; p<0.0001); white - (F=2000; Kendall=0.5114; p<0.0001). The results of detailed comparisons are not presented because of the limited space - the only not significant difference was detected for white background images for experiment 2, 3 and 4 and Random Forest classifier.

4. Conclusions

Applying any of the edge detection techniques improved results, but the features obtained by the edge detection combined with the shape fill and centering allowed to achieve the highest classification accuracy regardless of the selected classifier.

SVM among chosen methods of classification turned out to be the best both for the whole set of images and subgroups of images differentiated by the background. The results were obtained for the whole set of parameters' mixture of edge detection techniques. In the next step only the results for arbitrarily chosen the best combination of two parameters for both methods will be checked and presented. The results are much better for any kind of a classifier comparing with the original images.

A proper choice of preprocessing selection method, then a reasonable selection of parameters' values and additional even simple techniques and corrections usage may lead to the binary images on which the selection or extraction methods can give improvements not only concerning the accuracy of classification systems but also the decision-making time, that is one of the most important issue in real-life applications.

References

[1] Abeel T., P.Y.V., Saeys Y.: Java-ML: A Machine Learning Library, Journal of Machine Learning Research, 2009.

[2] Alzohairi R., Alghonaim R., Alshehri W. and Aloqeely S.: Image based Arabic Sign Language Recognition System. International Journal of Advanced Computer Science and Applications (IJACSA), 9(3), 2018.

[3] Canny J.: A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986.

[4] Davies E. R.: Computer and Machine Vision: Theory, Algorithms and Practicalities, Academic Press, Waltham 1990, pp. 82-94.

[5] Główny Urząd Statystyczny, Departament Badań Społecznych: Stan zdrowia ludności Polski w 2009 r., Zakład Wydawnictw Statystycznych, Warszawa, 2011.

[6] Gonzalez R. C., Woods R. E. Digital Image Processing, Addison-Wesley Publishing Company, New Jersey 1992, pp. 595-602.

[7] Imagawa K., Lu S., and Igi S.: Color-based hands tracking system for sign language recognition. In Proc.: Third IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 462–467, April 14-16, 1998.

[8] Java Machine Learning Library (Java-ML): [http://java-ml.sourceforge.net]

[9] Sung Kwan Kang, Mi Young Nam, Phill Kyu Rhee: Color Based Hand and Finger Detection Technology for User Interaction In Proc.: Intl. Conference on Convergence and Hybrid Information Technology, 2008. ICHIT '08.

[10] Kumar M., Saxena R.: Algorithm and technique on various edge detection: a survey, Signal & Image Processing: An Intl. Journal, vol. 4, no. 3, June 2013.

[11] S. Marcel: Hand Posture and Gesture Datasets, [http://www.idiap.ch/resource/gestures]

[12] Program Powszechnych Przesiewowych Badań Słuchu u Noworodków: [http://www.sluch.ump.edu.pl], [https://www.wosp.org.pl/medycyna/programy/badania-sluchu]

[13] Ravikiran J., Mahesh K., Mahishi S., Dheeraj R., Sudheender S., Pujari N.V.: Finger Detection for Sign Language Recognition. Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, Vol. I, IMECS 2009, Hong Kong, March 18-20, 2009.
[14] Rivera-Acosta M., Ortega-Cisneros S., Rivera J., Sandoval-Ibarra F.: American Sign Language Alphabet Recognition Using a Neuromorphic Sensor and an Artificial Neural Network. Sensors (Basel). 2017 Oct; 17(10): 2176.
[15] Sanduja V., Patial R.: Sobel Edge Detection using Parallel Architecture based on FPGA, International Journal of Applied Information Systems, vol. 3, no. 4, July 2012.
[16] Silanon K.: Thai Finger-Spelling Recognition Using a Cascaded Classifier Based on Histogram of Orientation Gradient Features. Computational Intelligence and Neuroscience. 2017: 9026375 (PMC5611514).
[17] Starner T. and Pentland A.: Real-time American Sign Language recognition from video using hidden Markov models. In Proc.of International Symposium on Computer Vision, pp. 265–270, 21-23 Nov. 1995.
[18] Świdziński M. i in.: Sytuacja osób głuchych w Polsce. Raport zespołu ds. głuchych przy Rzeczniku Praw Obywatelskich. Biuro Rzecznika Praw Obywatelskich, Warszawa, 2014.
[19] Vernon D.: Machine Vision, Prentice-Hall, 1991.
[20] Weka 3: Data Mining Software in Java, `[https://www.cs.waikato.ac.nz/ml/weka/]`

WPŁYW PRZETWARZANIA WSTĘPNEGO OBRAZÓW NA ROZPOZNAWANIE ZNAKÓW ALFABETU MIGOWEGO

Streszczenie Artykuł dotyczy klasyfikacji wybranych liter alfabetu migowego w postaci obrazów. Badany jest wpływ na wyniki kilku metod przetwarzania wstępnego obrazów, w tym progowania adaptacyjnego oraz detekcji krawędzi. Dodatkowo sprawdzane jest wypełnianie znalezionych kształtów, a także centrowanie dłoni na obrazach. Jako metody klasyfikacji wybrane zostały: klasyfikator SVM z liniową funkcją jądrową, klasyfikator Naive Bayes oraz Random Forest. Jako miary jakości klasyfikacji raportowane są jakość klasyfikacji,miara F, pole pod krzywą ROC oraz współczynnik Kappa.

Słowa kluczowe: przetwarzanie wstępne, alfabet migowy, klasyfikacja