Hand Gesture Recognition Based on Skeletal Image Properties

J. Ma and V. Yu. Tsviatkou and A. A. Boriskevich
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: majun1313@hotmail.com, vtsvet@bsuir.by, anbor@bsuir.by

***Abstract*—Hand gesture Recognition is an important task and can be used in a lot of applications. In intelligent systems, hand gesture recognition can be used to access information through a video interface. In recent years, skeleton-based hand gesture recognition become a popular research topic. The existing methods have the low discriminative power due to sensitivity of features to image noise. We have proposed new methods to decrease the influence of the noise to extract hand image features. The objective of the research is to improve the hand gesture classification accuracy. A hand gesture recognition method based on skeleton image properties is developed. For 5 classes recognition, this approach allows us to increase the classification accuracy on test set from 0.4% till 20.4% as compared with existing well-known methods.For 10 classes recognition, this approach allows us to increase the classification accuracy from 5% till 18% as compared with existing well-known methods.**

***Keywords*—Color images, Skeleton images, hand gesture feature, Machine Learning, Classification Accuracy**

I. Introduction

Hand gesture recognition is widely used in many applications such as sign language recognition [1], clinical and health [2], and robot control [3]. Open semantic technologies provide the ability to access the knowledge base of an intelligent system using a video interface. The inclusion of a hand gesture recognition system in the video interface makes it easier to enter commands and data into an intelligent system.

There are two existing practical approaches to recognizing hand gestures [4]. The first approach is based on data gloves (wearable or direct contact) [5], [6], and the second is based on computer vision, which does not require special sensors except cameras [7]. Moreover, vision-based methods can provide contactless communication between humans and computers. Therefore they are considered ordinary, suitable approaches.

As one of the types of camera vision-based approaches [4], skeleton-based approaches are attracting much attention in recent years since the skeleton feature describes geometric attributes and constraints and easily translates features and data correlations [8]. The skeleton-based approaches can further be classified as RGB-based [9] and RGB-D-based [10]- [12] approaches according to the different ways of obtaining the skeletal images. The RGB-D-based approaches adopt the depth sensor of the Kinect camera to obtain the skeletal image. One of the merits of these methods is that the lighting, shade, and color did not affect the obtained skeletal image. However, the depth camera's cost, size, and availability will limit their use. On the contrary, RGB-based approaches only require standard cameras. However, it must first convert the RGB images into grayscale images and then follow binarization and skeletonization to extract the skeletons. The skeletons extracted by this kind of method may include many useless skeletal branches or skeletal rings that are caused by the noise. The noise problem will be evident when the contrast of the input image is low. Since the existence of the noise, the accuracy of hand gesture recognition using RGB skeleton-based approaches are not satisfying. However, it would be a promising method if the effect of the noise could be reduced.

In the past several decades, many denoise methods have been proposed to alleviate the effects of noise on the skeletonization algorithm and produce stable skeletons as much as possible. These methods can be concluded into three different types, which are skeletonization-based denoising approaches [13]- [16], pruning-based denoising approaches [17]- [19], and scale-space-based denoising approaches [20], [21].

In this paper, a hand gesture recognition system based on skeleton image properties is developed, in which skeleton images are extracted by using different combinations of the skeletonization and denoising method. The objective of the research is to improve hand gesture classification accuracy.

II. Related Methods

In this section, some skeletonization and denoising methods used in our hand gesture recognition system are introduced. There are five skeletonization methods implemented, two of which are classical skeletonization methods and others proposed by us. In addition, a post-pruning method and a space-based denoising approach are also deployed in this system.

A. Image Skeletonization Method

OPTA algorithm [22] is a classical parallel skeletonization method proposed by Roland T. Chin et al. This

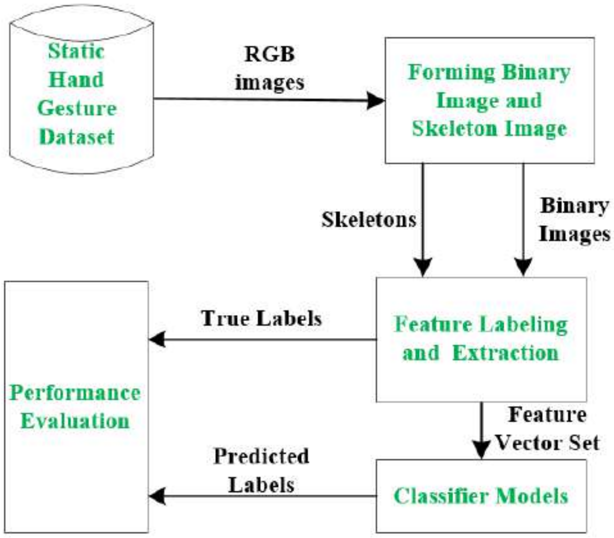


Figure 1. General block-scheme of the hand gesture reognition.

algorithm uses eight 3×3 thinning templates to remove the pixels. In addition, two other restoring templates are applied to address the breakage and disappearance of horizontal and vertical limbs with double-pixel widths. The drawback of this algorithm is that it is susceptible to noise.

ZS algorithm [23] is another classical parallel skeletonization method, which is one of the most popular methods since it can offset the influence of the noise to some extent by breaking one iteration in OPCA into two sub-iterations. Therefore, the computation speed of the ZS method is slower than OPTA. In addition, the ZS algorithm has some potential problems, such as sometimes it may suffer the problem of excessive erosion, and it fails to maintain one-pixel width, which may increase the difficulty when applied to recognition tasks.

Based on these two classical methods, we have proposed three new skeletonization methods: OPCA, ZSM, and OPTA.

OPCA [24] is a denoising version of the OPTA algorithm by modifying some deletion conditions. It is more robust than the OPTA method and, at the same time, shares a similar computation speed with the OPTA method. In addition, it can achieve a single-pixel width. However, it is more sensitive to the noise than the ZS algorithm.

ZSM [25] is an improved version of the ZS algorithm, in which the drawbacks of excessive erosion are overcomed and it can achieve single pixel width by adopting extra five thinning templates. The denoise ability of this algorithm is similar to the ZS algorithm. However, this method is still a sub-iterative method as the ZS algorithm.

MOPCA [26] is the improved version of the OPCA, in which there is a total of 13 templates are used to enhance the robustness of the algorithm to the noise. This method combines the merits of ZS algorithms and OPTA algorithms. It is insensitive to noise as the ZS and as fast as the OPTA method.

B. Post-pruning and Scale-Space based Denoiseing Method

Using denoise-skeletonization can only partly offset the influence of the noise. Therefore, it is necessary to use other denoising techniques to further improve the noise-against ability. As a result, we also proposed a new post-pruning method and a new scale-space denoising method.

The post-pruning method proposed by us is named DCEM [27], modified from the famous pruning algorithm of DCE [28]. One of the limitations of the DCE is that it requires manual tuning of the parameter of the pruning power's strength. In DCEM, conducting this tedious work is unnecessary, making it more convenient in many applications.

Our proposed scale-space denoising method is ATFM [29], derived from the ATF [30] method. The core idea of the ATF method is first to extract skeletons from different smoothed images that are filtered by using different scale-space filters to the original image. Then, they used their proposed sensitive measure to evaluate these skeletons, from which the skeleton with the lowest score is considered stable. However, their method sometimes may suffered the problem of the deformation of the skeleton. To overcome this problem, we proposed our ATFM method. In our method, the significant modification on the sensitive measure, in which more information is included.

III. Proposed Hand Gesture Recognition Method

The general block scheme of hand gesture recognition includes the following components, which are static hand gesture dataset, binary and skeleton image formation, feature labeling and extraction, classifier models, and performance evaluation. This general blcok-scheme is presented in Fig. 1, all major components are marked with green color.

Static hand gesture images are stored in the images dataset. They are used to train the classifier and evaluate the accuracy of the classification task. These RGB images of hand gestures are first passed into a block that can form the binary and skeleton images from them. All the proposed skeletonization and denoise methods are embedded in this block.

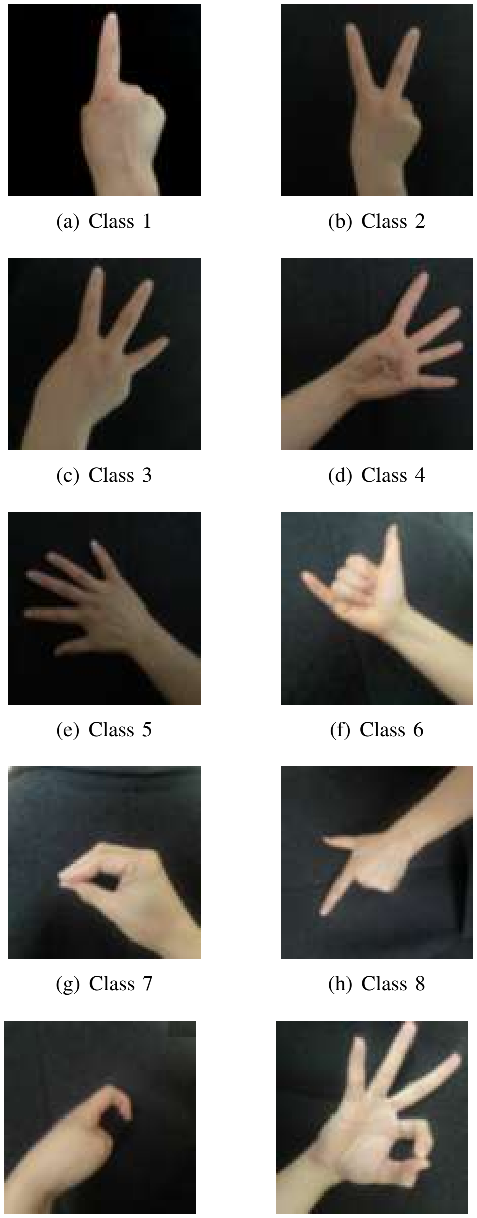
Next, feature extraction is conducted based on these obtained binary and skeleton images. For each pair of binary image and skeletal image, there are nine geometry features should be extracted, and they together compose a feature vector. Next, manual labeling for each pair of binary and skeleton images is also required to get the truth labels that corresponding to each feature vector. These feature vector can passed to the trained classifier for prediction. By comparing the predicted label and the truth label to compute the accuracy. In the classification module, there are six different well-known classifiers for optional, which includes decision tree(DT) [31], k-nearest neighbors (KNN) [32], naïve Bayes (NB) [33], support vector machine (SVM) [34], ensemble learning (EL) [35], multilayer perceptron (MLP) [36].

A. Creation of the Hand Gesture Dataset

All static hand gesture images that in dataset are captured with the iPhone 11. The resolution of images are $3024 \times 3024 \times 3$. Since directly processing these images is time-consuming, resize operation is used to converting these images into $95 \times 95 \times 3$ images. The dataset consists of ten different classes, example pictures are shown in Fig. 2.

In each one class, there are more than 100 different images. As a result, the total amount of our dataset is over 1000 images. These images are randomly divided into train-validation group and testing group. The number of images in testing group is equal to 20% of the initial image set, and the number of images in train-validation group is 80% of the initial image set.

B. Forming Binary Image and Skeleton Image using Hybrid Combining Denoising Techniques and Skeletonization Methods


(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

(e) Class 5 (f) Class 6

(g) Class 7 (h) Class 8

(i) Class 9 (j) Class 10

Figure 2. Example of the Ten Class of Hand Gestures.

The skeleton and pattern images are extracted from the original images by using different combinations of skeletonization method and denoise methods . There are six hybrid methods are used, which including ZS+ATFM, OPTA+ATFM, OPCA+ATFM, ZSM+ATFM, MOPCA+ATFM, and MOPCA+ATFM+DCEM. The time consumption of these methods is listed in Tab. I.

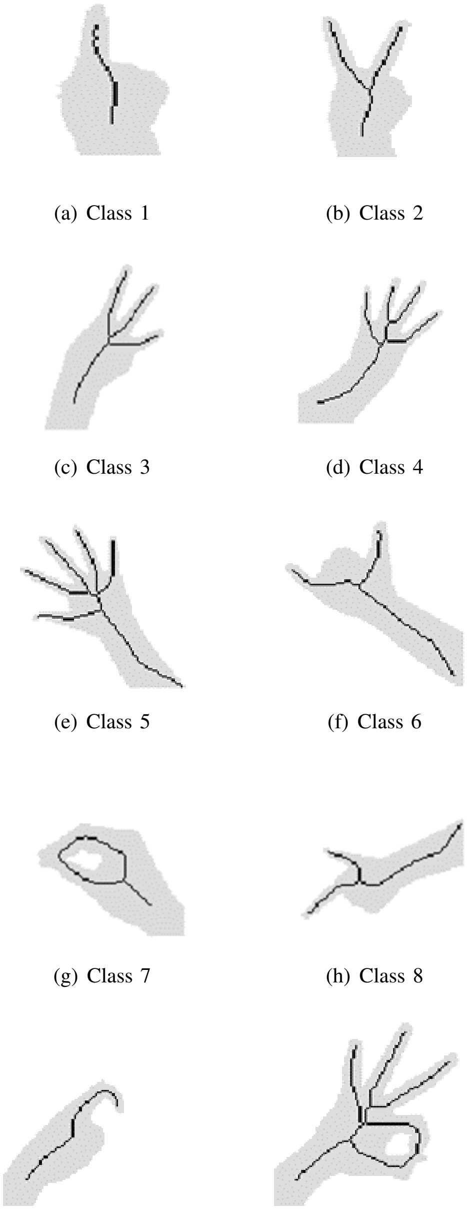
From Tab. I, it is noted that ZS+ATFM, OPTA+ATFM, ZSM+ATFM, MOPCA+ATFM, and MOPCA+ATFM+DCEM respectively spend more 38%, 22%, 33%, 0.4%, and 5% time when compared with the method of OPTA+ATFM. Besides, we can learned the use of DCEM may take extra 0.02 seconds.

In Fig. 3, we listed example skeletons extracted from

Table I
TIME CONSUMPTION OF SIX METHODS

Skeleton Image Extract algorithm	Average Time of skeleton process (s)
ZS+ATFM	0.704
OPTA+ATFM	0.624
OPCA+ATFM	0.510
ZSM+ATFM	0.682
MOPCA+ATFM	0.512
MOPCA+ATFM+DCEM	0.536

the images that are shown in Fig 2 by using the skeletonization method MOPCA with both ATFM and DCEM.



(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

(e) Class 5 (f) Class 6

(g) Class 7 (h) Class 8

(i) Class 9 (j) Class 10

Figure 3. Skeleton examples of ten hand gesture classes.

C. Feature extraction based on Skeleton and Binary Images

After a skeletal image and its pattern image are obtained from an input image, it is necessary to transform the skeletal images along with its pattern image to a 9-dimension feature vector used in the later classification. This 9-dimension vector includes the following significant geometrical features: the number of endpoints (NEP); the number of cross points (NCP); the existence of the inner hole (EIH); the average virtual-real distance rate between each pair of endpoints (AVRD); the number of virtual cross points (NVCP); Rate of the deviation of the thick of the endpoints (RDTE); Average distance between the thickest point in a pattern image and each endpoint in the skeletal image(ADTPE); distance between pattern thickest point and skeletal thickest point (DPSP); average angle of the endpoint (AAEP). Each dimension of this feature vector is manually selected with respect to the topology of these different classes.

The NEP is obtained by summarizing the number of these foreground pixels, which have only one neighbor foreground pixel in its 8-neighborhood window in the skeletal image.

The NCP is obtained by summarizing the number of these foreground pixels, which have more than two neighbor foreground pixels in its 8-neighborhood window in the skeletal image.

The EIH is an important geometry feature with only two values, 0 or 1. The inner hole denotes that the hole should be enclosed by the skeleton. Ideally, only Class 7 and Class 10 have the inner hole. One method to judge the existence of the inner hole for these hand images is to compute the number of closed areas in the skeleton image.

The AVRD describes the similarity of the real connecting line between endpoints to the virtual closet straight line between them. For each pair of endpoints, the real connecting line and its distance can be obtained using breadth-first search (BFS) algorithms, and the distance of the virtual line is calculated using the Euclidean distance formula. Then the average value is easily obtained.

The NVCP is obtained by summarizing the total number of points at the intersection of the virtual line and the real line.

Before presenting the definition of RDTE, ADTPE, and DPSP, the concept of thickness is first introduced. The thickness of a pixel is defined by the distance between this pixel and its closest pixel located on the boundary in the pattern image. Boundary pixels comprise the foreground pixel, whose four neighbors have at least one background pixel.

For a given skeleton with n endpoints, all endpoints can form a set S_{EP}, in which the i-th endpoint is denoted as S_{EP_i}. The thickness of S_{EP_i} can be denoted as T_{EP_i}. The set formed by all T_{EP_i} is denoted as $T_{S_{EP}}$. Then,

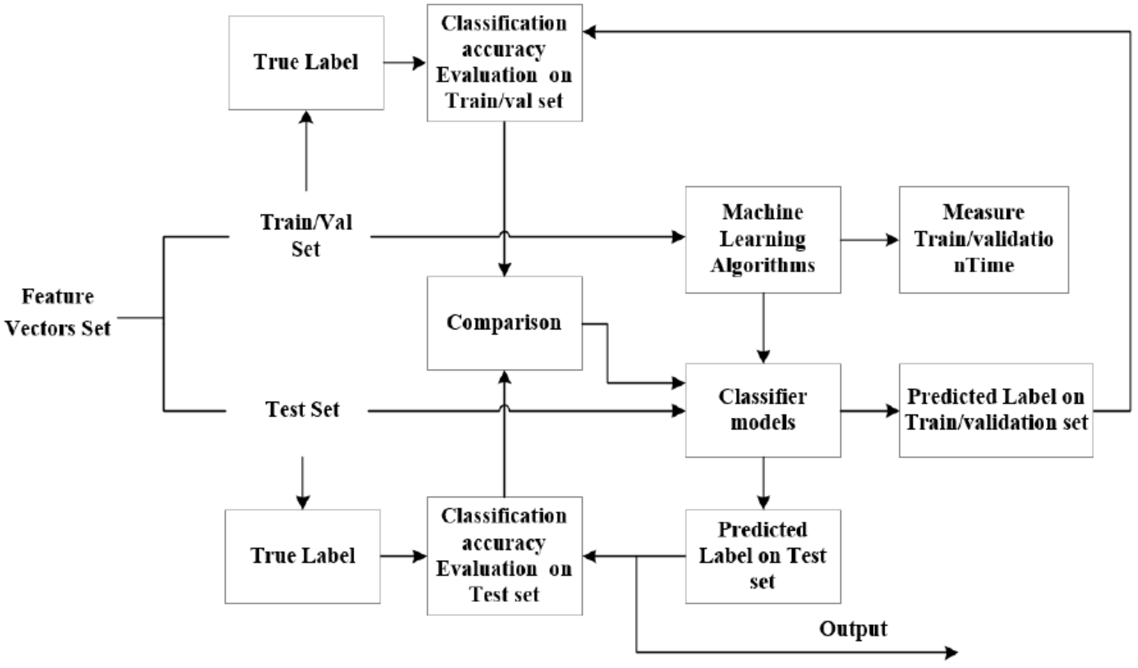

True Label
Classification accuracy Evaluation on Train/val set
Train/Val Set
Machine Learning Algorithms
Measure Train/validatio nTime
Feature Vectors Set
Comparison
Classifier models
Predicted Label on Train/validation set
Test Set
True Label
Classification accuracy Evaluation on Test set
Predicted Label on Test set
Output

Figure 4. Classification Models and Performance Evaluation.

the RDTE for this skeleton can be computed by using the following formula:

$$RDTE = \begin{cases} 0 & n \leq 1 \\ \sum_{i=1}^{n} \frac{\sqrt{(T_{EP_i} - \frac{1}{n}\sum_{i=1}^{n} T_{EP_i})^2}}{max(T_{S_{EP}}) - min(T_{S_{EP}})} & n > 1 \end{cases} \tag{1}$$

We suppose the coordinates of the thickest pixel in the pattern image are P_x and P_y, and its thickness is T_p. We suppose that in a skeletal image, there are n endpoints. The coordinates of the i-th endpoint are denoted as EP_{i_x} and EP_{i_y}. Then, the ADTPE can be calculated by using the following formula:

$$ADTPE = \begin{cases} 0 & n = 0 \\ \frac{\sum_{i=1}^{n} \sqrt{(P_x - EP_{i_x})^2 + (P_y - EP_{i_y})^2}}{nT_p} & n > 0 \end{cases} \tag{2}$$

Supposing the coordinate of the thickest pixel in the pattern image is P_x and P_y, and the coordinate of the thickest pixel in the skeletal image is S_x and S_y, the DPSP can be calculated according to the following formula:

$$DPSP = \sqrt{(P_x - S_x)^2 + (P_y - S_y)^2} \tag{3}$$

Before obtaining the value of the AAEP, the main axis is defined by the thickest point in the pattern image and the farthest endpoint in the skeletal image from that point. Based on that, it is easy to calculate the relative angle of the remaining endpoint to these axes, and the AAEP is the mean of these angles. If the number of endpoints is less than 2, the AAEP is set as 0.

D. Classifier Models and Performance Evaluation

The obtained feature vectors of the images from the training set of the dataset and their labels are passed to classifiers metioned, then conduct the learning process. The hyperparameter of these classifiers is listed in Tab. II.

Then, the classifier's learning result are evaluated by considering the accuracy of these classifiers on the test set.

Here, our aim is to explore the relationship between the accuracy of the classifiers and the different skeleton extracted by different methods, the relationship between the accuracy of the classifiers, and the difference in the feature selection. In addition, we also study the difference between distinct classifiers and their performance under a different number of classes. The general block diagram is shown in Fig. 4.

There are many criteria to evaluate the classifier's performance, such as accuracy, F1, precision, recall, roc, and so on. Here, we only take accuracy as the evaluation criteria for simplification. The formula of accuracy is described in the following:

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m} I(x_i, y_i) \tag{4}$$

$$I(x_i, y_i) = \begin{cases} 1 & f(x_i) = y_i \\ 0 & f(x_i) \neq y_i \end{cases} \tag{5}$$

Table II
HYPERPARAMETER OF DIFFERENT CLASSIFIERS

Classifier	Hyperparameter	Setting
DT	Maximum number of splits	100
	Split criterion	Gini
KNN	Number of neighbors	1
	Distance metric	Euclidean
	Distance weight	Equal
NB	Kernal type	Gaussian
SVM	Kernal function	Quadratic
	Box constraint level	1
	Kernel scale mode	Auto
EL	Ensemble Method	Bagged Trees
	Learning type	Decision Tree
	Maximum number of splits	99
	Number of learners	30
	Learning rate	0.1
	Subspace dimension	1
MLP	Number of fully connected layers	1
	Layer size	25
	Activation	Relu
	Iteration limit	1000

Where m is the number of the pair of feature vectors x and its corresponding true label y. $f(x)$ is predicted label for feature vector x.

IV. EXPERIMENTAL RESULTS

Two separate experiments are conducted for the purpose of evaluating the performance of the proposed recognition method. In the first experiment, the recognition experiment is conducted on five hand gesture classes, which include Class 1 to Class 5. Whereas in the second experiment, the recognition experiment is conducted on all ten hand gesture classes.

A. Performance evaluation of static hand gesture Classification for five classes

In order to explore the influence of the features on the classification task for five classes, an experiment of the feature selection has been conducted based on the classifiers of KNN and SVM, in which we removed one feature from the feature vector and conducted the classification task by using remaining features. Experimental results are shown in Tab. III.

From Table III, it is clear that the deletion of the NEP or the deletion of NCP may decrease the accuracy of the classification result in both KNN and SVM. Deleting the NVCP, ADTPE, and AAEP may improve the accuracy of the KNN classification, whereas deleting them may not change the performance of the SVM classification. The reason for that is the KNN classifier used in our experiment only considers the closest neighbors.

Next, the influence of the different skeleton extractions on the accuracy of all six classifiers is also studied. The results are shown in Tab. IV and Tab. V.

From Tab. IV and Tab. V, it is obvious that skeletonization methods can affect the accuracy of the classification. Among the methods with ATFM denoise operation, the proposed three skeletonization methods: ZSM, OPCA, and MOPCA, have higher accuracy of classification over ZS and OPTA in all six classifiers, in which the MOPCA has the highest average accuracy, which is 96.15% and 97.17% on the validation set and testing set respectively.

Table III
CLASSIFICATION ACCURACY COMPARISON FOR 8 AND 9 FEATURES AND 5 CLASSES

Feature Deleted	Classification Accuracy			
	Train/Validation Set		Test Set	
	KNN	SVM	KNN	SVM
NEP	92.30%	97.00%	88.80%	97.80%
NCP	93.80%	97.90%	90.30%	98.50%
EIH	94.80%	97.60%	94.00%	99.30%
AVRD	94.40%	97.20%	94.00%	99.30%
RDTE	95.30%	97.90%	92.50%	99.30%
NVCP	96.40%	97.80%	96.30%	99.30%
ADTPE	95.90%	97.40%	97.00%	99.30%
DPSP	94.40%	97.40%	97.00%	99.30%
AAEP	94.60%	97.40%	94.80%	99.30%
Full Features	94.40%	97.80%	94.00%	99.30%

In addition, the denoising methods influence the performance of classification. For example, we can see that the average accuracy of MOPCA with ATFM+DCEM is 96.67% and 97.55% on the validation and testing sets, respectively, which is 2% higher than that of MOPCA with ATFM.

From the perspective of the classifiers, the decision tree and ensemble learning are the top two best classifiers in the task of five classes classification on all skeletons extracted by different methods, which have up to 98.5% and up to 98.3% accuracy on the validation set, respectively. For the testing set, both have up to 100% accuracy of classification. In contrast, the naïve Bayes has the worst performance in terms of accuracy, which has only

Table IV
CLASSIFICATION ACCURACY EVALUATION ON TRAIN/VALIDATION SET THAT HAS 5 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	88.4%	82.4%	93.0%	92.9%	97.4%	98.5%
NB	73.6%	73.8%	87.3%	87.5%	96.6%	95.7%
SVM	86.2%	78.5%	93.1%	87.3%	96.8%	97.8%
KNN	81.5%	71.8%	87.9%	83.6%	93.2%	94.4%
EL	90.7%	83.7%	95.5%	94.6%	97.3%	98.3%
MLP	85.4%	74.0%	88.4%	83.0%	95.6%	96.3%
Mean	84.3%	77.3%	90.8%	86.8%	96.1%	96.6%

Table V
CLASSIFICATION ACCURACY EVALUATION ON TEST SET THAT HAS 5 DIFFERENT CLASSES

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	85.8%	82.1%	93.3%	94.0%	96.3%	100.0%
NB	79.9%	75.4%	86.6%	82.8%	95.8%	94.0%
SVM	90.3%	78.4%	91.8%	86.6%	97.0%	99.3%
KNN	85.1%	75.4%	86.6%	82.8%	97.8%	94.0%
EL	90.3%	82.8%	93.3%	94.0%	98.3%	100.0%
MLP	85.1%	68.7%	82.8%	88.1%	97.8%	98.0%
Mean	86.0%	77.1%	89.0%	88.0%	97.1%	97.5%

95.7% in the validation set and 94.00% in the testing set. Regarding training time, when skeletonization is set as MOPCA+ATFM+DCEM, the average time consumed by the decision tree is about 0.6s, which is faster than ensemble learning, which consumes about 4.2s. In Fig. 5, training Time consumed by different classifiers in 5 classes is presented.

For five classes classification task, the best combination method is using MOPCA skeletonization to extract the skeleton, using ATFM and DCEM to offset the noise's influence, and selecting decision tree to predict the class of the static hand gesture. The overall accuracy can reach 98.5%, and the train time is 0.6435s.

B. Performance evaluation of static hand gesture Classification for 10 classes

Similar to the previous section, the experiment of the feature selection has been conducted based on the classifiers of KNN and SVM once more. The only difference is that the current experiment considered more classes, which increased from 5 to 10. Experimental results of the feature selection are shown in Tab. VI.

By comparing Tab. VI and Tab. III, it is notable that the overall accuracy of classification is significantly reduced with the increasing number of classes since there are more complicated hand gestures are considered. In addition, the importance of each feature is also altered. For example, in Tab. III, we knew that the deletion of the NEP and NCP might significantly worsen the accuracy; however, in Tab. VI, the degree of the influence caused by them is much slightly when compared with the feature of AAEP. On the other hand, removing the NVCP and RDTE may increase the classification accuracy on both KNN and SVM.

Table VI
CLASSIFICATION ACCURACY COMPARISON FOR 8 AND 9 FEATURES AND 10 CLASSES

Feature Deleted	Classification Accuracy			
	Train/Validation Set		Test Set	
	KNN	SVM	KNN	SVM
NEP	81.30%	79.70%	84.00%	80.60%
NCP	81.80%	81.10%	84.80%	81.00%
EIH	82.60%	81.50%	84.40%	81.60%
AVRD	82.50%	80.50%	84.40%	79.70%
RDTE	84.50%	80.90%	85.70%	82.70%
NVCP	84.40%	80.90%	88.20%	82.70%
ADTPE	82.50%	78.70%	83.10%	79.30%
DPSP	83.80%	81.20%	85.20%	81.90%
AAEP	73.20%	74.40%	73.40%	78.10%
Full Features	82.70%	81.60%	84.40%	81.00%

Similar to the previous section, the influence of the different skeletonization methods and the different classifiers on the accuracy are explored in static hand gesture classification. The accuracy of 10 classes on the validation and testing sets are presented in Tab. VII and Tab. VIII, respectively. Training time consumed by different classifiers on ten classes is shown in Fig. 6.

From Tab. VII and Tab. VIII, we can see that the average accuracy of classification based on skeletons extracted by distinct methods are all decreased to some extent when comparing with the results in Tab. IV and Tab. V. However, the MOPCA+DCEM+ATFM method still outperforms other methods. For example, for classifying ten types of static hand gesture tasks, the average accuracy of classification of the MOPCA+DCEM+ATFM

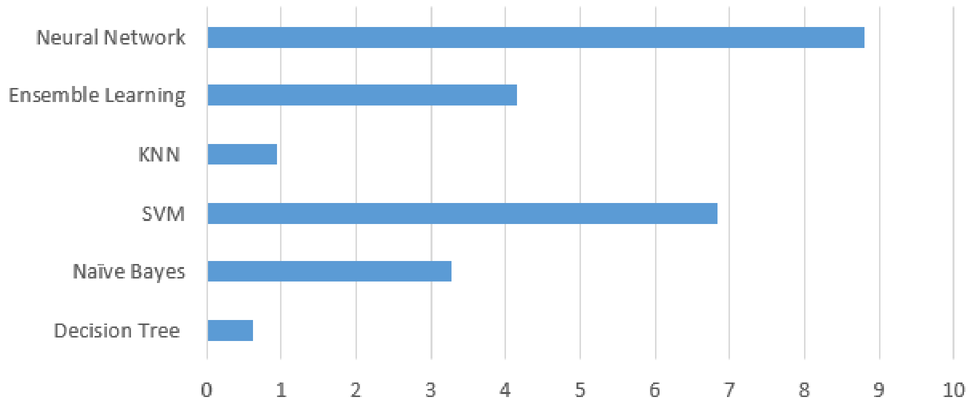


Figure 5. Training Time Consumed by 6 Classifiers on 5 classes dataset.

Table VII
Classification Accuracy Evaluation on Train/Validation set that has 10 different classes

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	76.5%	70.6%	84.4%	83.3%	86.7%	91.1%
NB	62.1%	61.8%	82.0%	76.1%	77.4%	82.9%
SVM	74.1%	59.5%	79.2%	77.4%	80.4%	81.6%
KNN	74.2%	63.9%	76.7%	75.5%	83.8%	82.7%
EL	80.9%	76.7%	88.7%	88.3%	90.0%	92.9%
MLP	73.4%	59.1%	74.9%	73.0%	82.6%	82.3%
Mean	73.5%	65.2%	80.9%	78.9%	83.4%	85.5%

Table VIII
Classification Accuracy Evaluation on Test set that has 10 different classes

Classifier Models	ZS+ ATFM	OPTA+ ATFM	ZSM+ ATFM	OPCA+ ATFM	MOPCA+ ATFM	MOPCA+ATFM +DCEM
DT	75.9%	73.8%	82.7%	87.3%	86.1%	91.1%
NB	59.5%	62.0%	86.5%	81.4%	80.2%	77.2%
SVM	73.8%	62.4%	77.6%	75.5%	82.3%	81.0%
KNN	69.6%	63.7%	78.9%	75.9%	87.8%	84.4%
EL	81.4%	79.3%	89.0%	91.6%	87.8%	92.8%
MLP	70.5%	60.8%	79.7%	76.8%	83.1%	83.5%
Mean	71.7%	67.0%	82.4%	81.4%	84.5%	85.0%

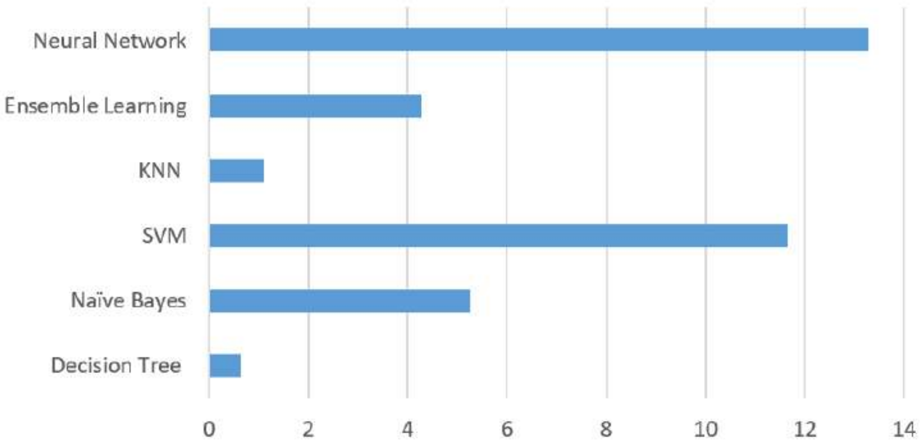


Figure 6. Training Time Consumed by 6 Classifiers on 10 classes dataset.

method can reach 85.58% on the validation set and 85.00% on the testing set. On the other hand, the average accuracy of classification on skeletons extracted by ZSM and OPCA is higher than the average accuracy of classification on skeletons extracted by ZS and OPTA.

In addition, from the perspective of the classifiers, the accuracy of the classification of the decision tree and ensemble learning surpassed all the other classifiers. Based on the skeleton extracted by the MOPCA+DCEM+ATFM method, the decision tree and ensemble learning classifier can obtain 91.10% and 92.80% accuracy on the testing set, respectively. In contrast, the classification accuracy of other classifiers can only achieve about 85%. Although the ensemble learning classifier has a slight bit advantage over the decision tree regarding classification accuracy, the time spent on training the ensemble learning is much more than the decision tree.

In a word, for ten classes classification task, the best combination method is using MOPCA skeletonization to extract the skeleton, using ATFM and DCEM to offset the noise's influence, and using ensemble learning to predict the class of the static hand gesture. The overall accuracy can reach 91.1%, and the train time is 0.6134s.

V. Conclusion

The hand gesture recognition based on the new image skeletonization methods, extracted gesture feature vector and using machine learning technique allow us to increase the classification accuracy. For 5 classes and 10 classes hand gesture classification task, the improvement of accuracy on test set is within the range of 0.4% to 20.4% , and that of 5% to 18% . The MOPCA+ADFM+DCEM method is effective in terms of average classification accuracy on test set. It achieves 97.5% on 5 classes recognition task and 85.00% on 10 classes recognition task. In addition, for 5 classes recognition task and 10 classed recognition task , the training time consumed by six classifiers is within the range of 0.7s to 8.9s and that of the 0.3s to 11s, respectively. It is set that ensemble learning model is the best classifier and it allows us to achieve 100% (5 classes) and 92.8% (10 classes) on test set. Increasing the accuracy of hand gesture classification based on the proposed skeletonization methods improves the technical characteristics of intelligent systems using video interfaces for entering commands and data, and makes a significant contribution to the development of semantic technologies for designing such systems.

References

[1] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Systems with Applications*, vol. 182, p. 115657, 2021.

[2] J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in *2012 Third Global Congress on Intelligent Systems*. IEEE, 2012, pp. 382–385.

[3] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3d hand gesture interaction with a robot for understanding directions from humans," in *2011 Ro-Man*. IEEE, 2011, pp. 357–362.

[4] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[5] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)*, vol. 38, no. 4, pp. 461–482, 2008.

[6] J. J. LaViola Jr, "A survey of hand posture and gesture recognition techniques and technology," 1999.

[7] G. Murthy and R. Jadon, "A review of vision based hand gestures recognition," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 405–410, 2009.

[8] H. Kaur and J. Rani, "A review: Study of various techniques of hand gesture recognition," in *2016 IEEE 1st international conference on power electronics, intelligent control and energy systems (ICPEICES)*. IEEE, 2016, pp. 1–5.

[9] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Sign language recognition based on hand and body skeletal data," in *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2018, pp. 1–4.

[10] C. Xi, J. Chen, C. Zhao, Q. Pei, and L. Liu, "Real-time hand tracking using kinect," in *Proceedings of the 2nd International Conference on Digital Signal Processing*, 2018, pp. 37–42.

[11] G. Devineau, F. Moutarde, W. Xi, and J. Yang, "Deep learning for hand gesture recognition on skeletal data," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 106–113.

[12] F. Jiang, S. Wu, G. Yang, D. Zhao, and S. Kung, "independent hand gesture recognition with kinect," *Signal, Image and Video Processing*, vol. 8, pp. 163–172, 2014.

[13] F. Y. Shih and W.-T. Wong, "Fully parallel thinning with tolerance to boundary noise," *Pattern Recognition*, vol. 27, no. 12, pp. 1677–1695, 1994.

[14] J. Feldman and M. Singh, "Bayesian estimation of the shape skeleton," *Proceedings of the National Academy of Sciences*, vol. 103, no. 47, pp. 18 014–18 019, 2006.

[15] F. Gao, G. Wei, S. Xin, S. Gao, and Y. Zhou, "2d skeleton extraction based on heat equation," *Computers and Graphics*, vol. 74, pp. 99–108, 2018.

[16] C. Yang, B. Indurkhya, J. See, and M. Grzegorzek, "Towards automatic skeleton extraction with skeleton grafting," *IEEE transactions on visualization and computer graphics*, vol. 27, no. 12, pp. 4520–4532, 2020.

[17] W. Shen, X. Bai, R. Hu, H. Wang, and L. J. Latecki, "Skeleton growing and pruning with bending potential ratio," *Pattern Recognition*, vol. 44, no. 2, pp. 196–209, 2011.

[18] A. S. Montero and J. Lang, "Skeleton pruning by contour approximation and the integer medial axis transform," *Computers and Graphics*, vol. 36, no. 5, pp. 477–487, 2012.

[19] L. Serino and G. S. di Baja, "A new strategy for skeleton pruning," *Pattern Recognition Letters*, vol. 76, pp. 41–48, 2016.

[20] M. E. Hoffman and E. K. Wong, "Scale-space approach to image thinning using the most prominent ridge-line in the image pyramid data structure," in *Document Recognition V*, vol. 3305. SPIE, 1998, pp. 242–252.

[21] J. Cai, "Robust filtering-based thinning algorithm for pattern recognition," *The Computer Journal*, vol. 55, no. 7, pp. 887–896, 2012.

[22] R. T. Chin, H.-K. Wan, D. Stover, and R. Iverson, "A one-pass thinning algorithm and its parallel implementation," *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 1, pp. 30–40, 1987.

[23] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[24] J. Ma, T. V. Yurevich, and V. K. Kanapelka, "Image skeletonization based on combination of one- and two-sub-iterations models(in russ.)," *Informatics*, vol. 17, no. 2, pp. 25–35, 2020.

[25] J. Ma, X.-H. Ren, T. V. Yurevich, and V. K. Kanapelka, “A novel sub-iterative parallel skeletonization method,” *Journal of Computers (Taiwan)*, vol. 32, no. 6, pp. 83–97, 2021.

[26] J. Ma, X. Ren, V. Y. Tsviatkou, and V. K. Kanapelka, “A novel fully parallel skeletonization algorithm,” *Pattern Analysis and Applications*, pp. 1–20, 2022.

[27] J. Ma, X. Ren, V. K. Kanapelka, and V. Y. Tsviatkou, “An automatic pruning method for skeleton images,” in *Proceedings of the 15th International Conference,2021*. United Institute of Informatics Problems of the National Academy of Sciences of Belarus, 2021, pp. 232–235.

[28] X. Bai, L. J. Latecki, and W.-Y. Liu, “Skeleton pruning by contour partitioning with discrete curve evolution,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 449–462, 2007.

[29] J. Ma, X. Ren, H. Li, W. Li, V. Y. Tsviatkou, and A. A. Boriskevich, “Noise-against skeleton extraction framework and application on hand gesture recognition,” *IEEE Access*, vol. 11, pp. 9547–9559, 2023.

[30] H. Chatbri and K. Kameyama, “Using scale space filtering to make thinning algorithms robust against noise in sketch images,” *Pattern Recognition Letters*, vol. 42, pp. 1–10, 2014.

[31] O. Z. Maimon and L. Rokach, *Data mining with decision trees: theory and applications*. World scientific, 2014, vol. 81.

[32] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[33] D. J. Hand and K. Yu, “Idiot’s bayes—not so stupid after all?” *International statistical review*, vol. 69, no. 3, pp. 385–398, 2001.

[34] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.

[35] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[36] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

Распознавание жестов рук на основе свойств скелетизированных изображений

Ма Ц., Цветков В. Ю., Борискевич А. А.

Распознавание жестов рук является важной задачей и может использоваться во многих практических приложениях. В интеллектуальных системах распознавание жестов рук может использоваться для ввода информации посредством видеоинтерфейса. В настоящее время распознавание жестов рук на основе скелета стало популярной темой исследований. Существующие методы имеют низкую дискриминационную способность из-за чувствительности признаков к шуму изображения. Мы предложили новые методы уменьшения влияния шума на выделение признаков изображения руки. Разработан новый метод распознавания жестов рук, основанный на свойствах скелетизированных изображений. Цель исследования состоит в повышении точности классификации жестов рук. Данный подход позволяет повысить точность классификации с 5% до 21% по сравнению с существующими известными методами.

Received 12.03.2023