IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 December 2022, accepted 29 December 2022, date of publication 5 January 2023, date of current version 11 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3234526

RESEARCH ARTICLE

Fingertip Detection Algorithm Based on Maximum Discrimination HOG Feature in Complex Background

PENG JI, JINXIANG FENG, FENGYING MA, XIANJIAN WANG, AND CHENGLONG LI
School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

Corresponding author: Fengying Ma (mfy@qlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61903207, in part by the Youth Innovation Science and Technology Support Plan of Colleges in Shandong Province under Grant 2021KJ025, in part by the Industry-University-Research Collaborative Innovation Fund under Project 2020-CXY32, in part by the Key Research and Development Plan of Shandong Province under Grant 2019JZZY010731, and in part by the Jinan Scientific Research Leader Studio under Grant 2019GXRC062.

ABSTRACT Gesture fingertip detection plays a vital role in human-computer interaction applications such as VR and robot control. To improve the accuracy of fingertip detection in complex background, this paper proposes a fingertip detection algorithm based on a novel maximum discrimination HOG feature. Firstly, the Holistically-nested edge detection algorithm is used to detect the edge of hand images and non-hand images, and the HOG features of the contour edge are extracted to obtain the training set of positive and negative samples to reduce the influence of illumination, color, and texture on fingertip detection. Secondly, the maximum discrimination features are filtered from the positive sample set by a custom maximum discrimination feature filter and stored as a feature dictionary. The filtered maximum discrimination features and negative sample set are input into the XGBoost classifier for training and the voting rights classifier is obtained. Filtering maximum discrimination features can reduce the interference of irrelevant features and improve the performance of the classifier. Then, the KNN algorithm is used to find the best match in the dictionary, and the final fingertip position is obtained by the Meanshift algorithm. Finally, the algorithm in this paper is tested. The test results show that the accuracy of fingertip detection in this paper can reach 99 %, and the detected RMSE is kept in the range of 5 pixels, which is higher than that based on YCbCr skin color segmentation, YOLO target detection, and YOLO-YCbCr.

INDEX TERMS Fingertip detection, HED edge detection, HOG features, human-computer interaction, XGBoost.

I. INTRODUCTION

With the development of human-computer interaction (HCI) technology, gesture interaction, as the most natural interactive interaction, has been widely used in virtual reality (VR) and augmented reality (AR), such as aerial writing [1], game control [2], interactive projector [3] and robot control. For example, for the reconnaissance robots control, if soldiers can use head-mounted displays and visual gesture interaction technology (similar to air mouse technology), it will greatly facilitate the operation of soldiers. However, visual gesture interaction depends on gesture detection or fingertip detection. There are many challenges in fingertip detection. The environment where soldiers and reconnaissance robots are located may be complex, and the light changes dramatically. Soldiers may also wear gloves or hold other devices, and need to use infrared cameras to take images at night. Therefore, it is essential to study how to realize the visual detection of gestures in the all-day and complex background, even in the presence of attachment interference (such as gloves).

At present, fingertip detection methods mainly include IMU sensor-based contact detection methods and vision

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

based non-contact detection methods. Among them, fingertip detection algorithms based on contact detection methods mostly use data gloves, depth cameras, marker finger sets, and other equipment to obtain hand information, and then obtain fingertip position through image processing technology and the geometric relationship of the hand. For example, A.Z.Shukor et al. [4] designed a data glove, which achieves sign language recognition by equipping each finger with a finger-bending sensor to obtain hand information. Data gloves can accurately obtain hand information, but expensive, and not easy to promote. T.Dinh-Son et al. [5] extracted the hand region and palm center of interest through the joint information of the Kinect depth camera and then used the tracking algorithm to extract the contour of the hand. Finally, based on the hand contour coordinates, the k-cosine algorithm is used to detect the fingertip position. Through the depth camera, the hand area can be easily obtained. However, the existing depth cameras are limited by distance, illumination and other factors when shooting depth images, which is difficult to meet the needs of multi-scene applications. In addition, the hand position can also be obtained by giving different color markers to different fingertips [6]. However, this method requires re-marking before changing the experimental background each time, which will cause a lot of inconvenience to the operation.

Compared with the above methods, vision based non-contact detection methods can be applied to a variety of scenarios. It only needs an ordinary camera, which is cheap and suitable for promotion. Vision-based fingertip detection methods generally include two steps: first, obtaining the hand from the image, and then locating the fingertip through artificial features or machine learning methods. The detection accuracy of these methods is often affected by complex backgrounds, dramatic changes in illumination, hand occlusion and other factors in the image. Therefore, it is very important to design a reasonable algorithm to accurately extract the hand in the image. Therefore, L.Lae-Kyoung et al. [7] proposed to combine YCbCr model with a genetic algorithm to optimize the segmentation of the hand. In [8], the method of dense optical flow combined with the YCbCr model is used to obtain the hand area by setting the skin color threshold, and then the hand center of gravity is obtained by dividing the hand area into blocks. Finally, the maximum center of gravity distance method is used to detect the fingertip. The method of hand segmentation based on YCbCr skin color model is simple, but it is difficult to deal with the problem of illumination change and skin color background. Due to the limitations of skin color segmentation, researchers began to try to obtain hand regions through deep learning methods. M.P. et al. [9] proposed using the target detection model based on deep neural network (DNN) to segment gestures from the scene, and then use the MobileNetv2 architecture to estimate the fingertip position. H.Y. et al. [10] proposed to detect the hand area by the Faster R-CNN algorithm, and then use the CNN network to detect the fingertip position. In [11], the YOLO algorithm was used to obtain the hand area, and then the fingertip detection was regressed by the VGG16 full convolutional neural network. Y.-H.Chen et al. [12] proposed an improved masked region convolutional neural network (Mask R-CNN), which uses a region-based CNN network to detect fingers and achieves fingertip detection through a three-layer CNN network. Deep learning-based algorithms (such as YOLO, etc.) usually use color images. For gesture images captured by infrared cameras at night, gesture detection or fingertip detection based on deep learning is not good. Moreover, when performing operations in complex environments, the hand images captured by the camera often suffer from partial occlusion. The method based on deep learning is also difficult to deal with complex occlusion problems.

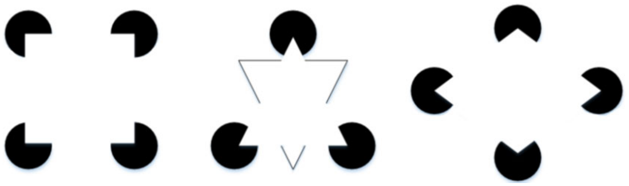


FIGURE 1. Illusion contour diagram.

The characteristic of human vision, illusory contours [13] (or subjective contours), is a visual illusion in which an observer perceives the presence of contours in the absence of changes in brightness or color. As shown in Fig.1, the square, triangle, and diamond are not complete, but they are subjectively given complete contours. This shows that human vision can find and infer the location of familiar objects based on incomplete contours. Considering the problems existing in the current fingertip detection algorithm, this paper proposes a fingertip detection algorithm based on subjective contour characteristics, which can detect the familiar target according to the partial contour. The HOG feature used in this algorithm is first used in pedestrian detection. Through the cooperation of HOG feature extraction and SVM classifier [14], it has achieved great success in the field of pedestrian detection. HOG feature has excellent characteristics such as geometric and optical invariance, and describes the shape of local regions by statistical gradient information. S. R. et al. [15] applied the HOG feature and edge HOG feature to gesture recognition, which provides us with a new idea. Edge has the characteristics of high tolerance to illumination, color and texture, which is very suitable for complex gesture detection and recognition environment. And combined with the idea of voting based on contour components, gesture detection in complex backgrounds with occlusion can be solved. Common edge detection methods, such as algorithms based on differential operators such as Canny, are difficult to effectively detect opponents in complex backgrounds. Therefore, this paper uses a Holistically-nested edge detection (HED) method [16]. This method achieves end-to-end image detection by using a deep learning model of a fully convolutional neural network and a deeply supervised network, and has a good effect on distinguishing the hand contour from the background contour. In addition, because the SVM classifier uses quadratic programming to solve the support vector, in the face of large sample data, the classification will consume a lot of machine memory and operation time, and in the case of linear inseparable, the algorithm is prone to overfitting. Therefore, this paper chooses XGBoost [17] as the voting rights classifier. XGBoost adds a regularization term to the objective function, which can effectively avoid overfitting, and uses parallel processing to greatly improve the calculation speed.

In this paper, firstly, the HED algorithm is used to detect the edge of the images. Secondly, the HOG features of each edge point are extracted from the contour edge images, and the maximum discrimination features are selected by a custom maximum discrimination feature filter. A centripetal offset vector is proposed to characterize the offset between the position of the maximum discrimination feature and the position of the fingertip, and the maximum discrimination feature and the centripetal offset vector are saved to the feature dictionary. Then, the XGBoost voting rights classifier is used to classify the maximum discrimination features with voting rights from all contour edge HOG features. Finally, the KNN [18] algorithm is used to match the maximum discrimination features obtained by the classification with the maximum discrimination features in the feature dictionary, and the centripetal offset vector is used to vote on the fingertip position, and then the Meanshift [19] algorithm is used to search the final detected fingertip position.

Aiming at the problem of fingertip detection in human-computer interaction, this paper proposes a fingertip detection algorithm based on maximum discrimination HOG feature in complex backgrounds. The specific contributions of this paper are as follows:

1) In this paper, the edge contour HOG features are used as the basis of image matching to reduce the impact of dramatic changes in light, texture changes, and skin-like backgrounds on fingertip detection.
2) To improve the efficiency and accuracy of fingertip detection, a voting rights classifier is designed to classify the voting features.
3) A maximum discrimination feature filter is proposed to select the most stable and effective features from hand edge contour HOG features, which reduces the influence of irrelevant features on the voting rights classifier.
4) Inspired by the illusory contour, a method of voting with hand contour components is proposed to determine the fingertip position. Depending on this idea, the fingertip position can be detected according to part of the hand information to effectively deal with the partial occlusion problem.

The rest of this article is organized in the following order. The overall overview of the proposed algorithm is introduced in Section II. Section III introduces the extraction of hand contour HOG features and the selection process of maximum discrimination features. Section IV introduces the principle and training process of the voting rights classifier for hand contour components. Section V describes the process of fingertip detection by KNN and the Meanshift algorithm. Section VI is the experimental part, which verifies the effectiveness of the fingertip detection algorithm proposed in this paper. Section VII is the summary part.

II. ALGORITHM OVERVIEW

Fingertip interaction is a very convenient and intuitive human-computer interaction method, which has a very wide range of applications. For example, in some scenarios where the operator and robots perform tasks cooperatively, such as anti-terrorism raids, battlefield reconnaissance, etc., the integrated use of the head-mounted display and fingertip motion trajectory information (similar to the air mouse) will greatly facilitate the operation complexity of operators, especially for fully armed soldiers. However, soldiers and mobile reconnaissance robots usually work in the wild, outdoors, and in other scenes with dramatic changes in light and complex backgrounds. In addition, soldiers often wear gloves or other attachments, which makes the fingertip detection algorithm need to adapt to the partial occlusion of the hand. To improve the effect of fingertip detection, this paper proposes a fingertip detection algorithm based on the maximum discrimination feature. The contour edge HOG feature is used to reduce the influence of illumination, color, and texture on fingertip detection, and the partial contour edge HOG feature is used instead of the whole. Therefore, when the hand is partially occluded, it can also have good robustness.

Through analysis, we find that in many human-computer interaction teleoperation applications based on fingertip information, the degree of freedom of the controlled object is low, such as in military reconnaissance robot systems and various manipulator control systems. Remote operation can be realized by using a single finger just like an air mouse. Inspired by this, we provide a single-finger human-computer interaction method that can use single-finger fingertip detection directly instead of the air mouse.

Target detection methods include holistic and component-based detection methods. The whole-based detection method compares the similarity between the known template image and each region in the image to be detected. If the similarity reaches the set threshold, the matching is considered to be completed. The component-based detection method refers to dividing the target into multiple components, and each component is matched with each region in the image to be detected. Compared with the whole-based detection method, the component-based detection method can effectively improve the detection effect under occlusion, and improve the fault tolerance rate and robustness. According to this idea, this paper designs a fingertip detection algorithm based on maximum discrimination HOG features in complex backgrounds. The algorithm mainly includes three steps: maximum discrimination features extraction of hand

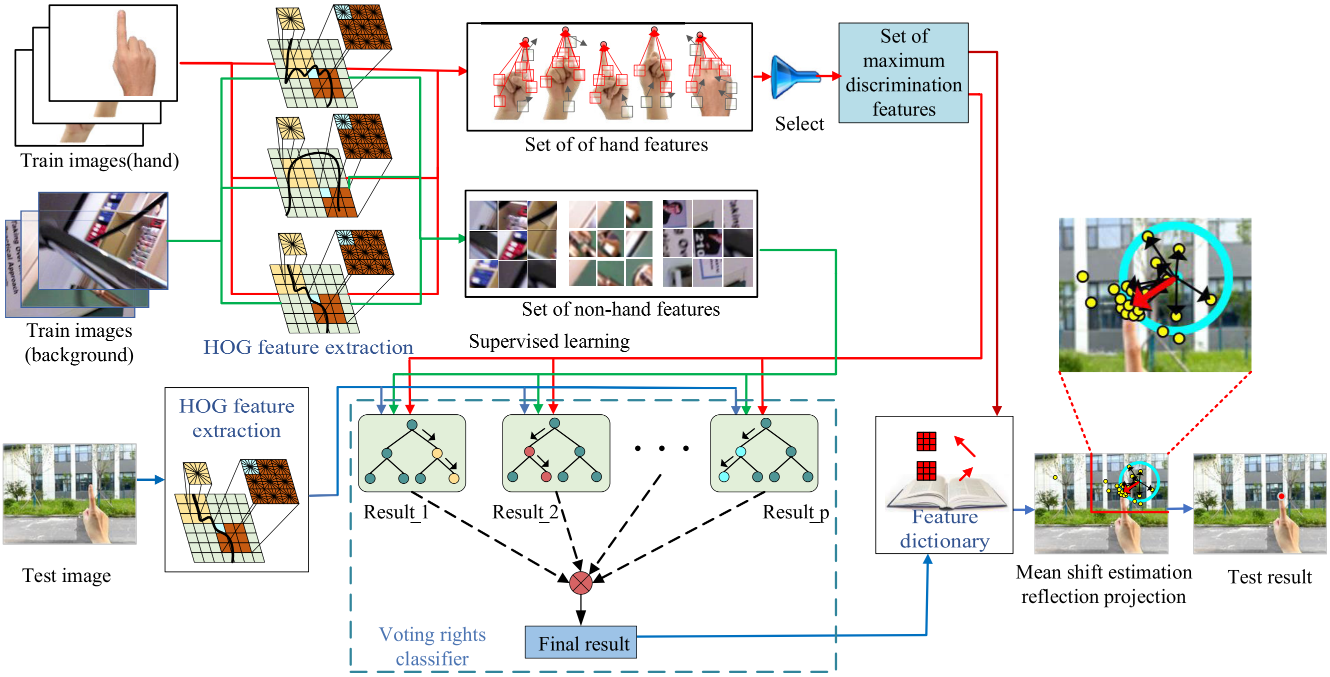


FIGURE 2. Algorithm flow.

contour, training of voting rights classifier of hand contour components, and fingertip detection. The algorithm flow is shown in Fig.2.

1) In the stage of maximum discrimination feature extraction of hand contour, two groups of images are taken, which are hand images under white background and complex background images without hand. The two groups of images are extracted by the HED method and binarized. Then, HOG features are extracted from the contour edge image of the hand and background to get positive and negative sample sets. To reduce the influence of irrelevant features on subsequent fingertip detection, the most suitable maximum discrimination features for this algorithm are selected from the positive sample set through a custom maximum discrimination feature filter. The offset between the maximum discrimination feature and the fingertip point is defined as the centripetal offset vector, and the maximum discrimination features and the centripetal offset vector are stored in the feature dictionary.
2) The maximum discrimination features and negative sample set are input into the XGBoost classifier for training, and the voting rights classifier of hand contour components is obtained.
3) For the test image, HED edge detection, binarization, HOG feature extraction, and voting rights classifier classification are completed in turn. Then, the KNN algorithm is used to match the maximum discrimination features obtained by the classification with the features in the feature dictionary. The matched features are combined with the concentric offset vector to vote to obtain a sparse fingertip detection space. Finally, the Meanshift algorithm is used to select the final detection result from the fingertip detection space.

III. MAXIMUM DISCRIMINATION FEATURE EXTRACTION OF HAND CONTOUR

A. HOG FEATURE EXTRACTION

The histogram of oriented gradient (HOG) [15] is a feature that describes the local information of the image. This method describes the shape of the object of interest in the image through the gradient or edge direction density distribution. The essence is to count the gradient information. HOG feature has excellent characteristics such as insensitivity to illumination and geometric changes. However, the HOG feature also has the problem of large computation and the inability to deal with occlusion. Therefore, this paper uses HOG feature voting of hand contour edge points to detect fingertips and realizes fingertip detection under partial occlusion.

Since the proposed algorithm aims to solve the problem of fingertip detection in complex background, the edge detection algorithm is expected to distinguish the contours of hand and background as much as possible, and the HED [20] algorithm can deal with the details of the image well. Therefore, this paper uses the HED algorithm to detect the edge of hand images and background images.

The main steps of HOG feature extraction of hand contour edge points used in this algorithm are as follows:

First, the read edge image is normalized and the gradient size $m(x, y)$ and direction $\theta(x, y)$ of each edge point *(x,y)* in

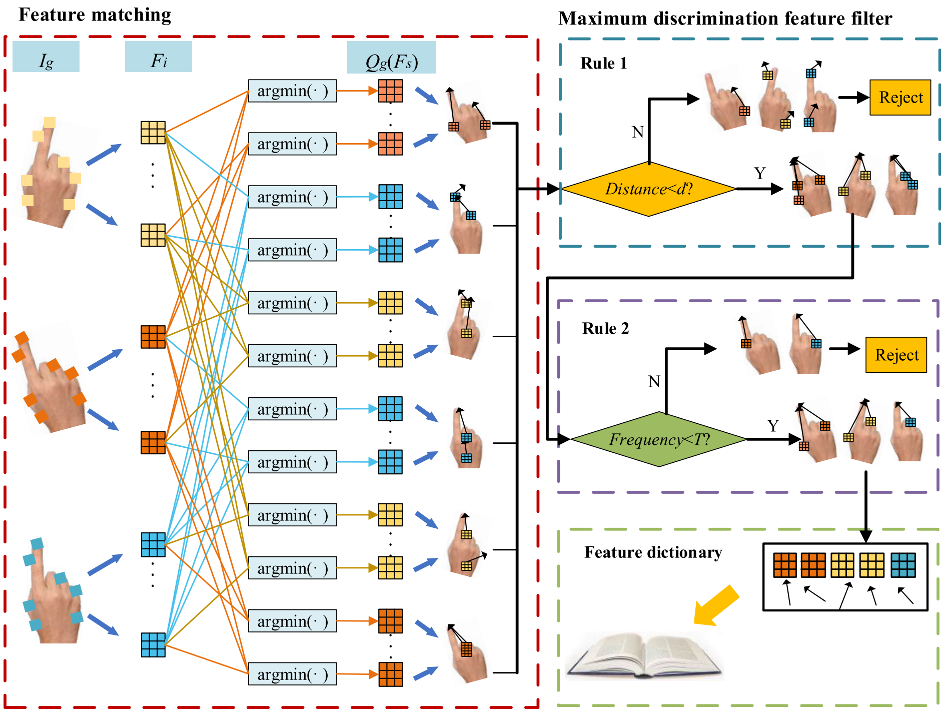


FIGURE 3. Generation of feature dictionary.

the edge image can be obtained by Formulas 3 and 4.

$$H_1 = H(x+1, y) - H(x-1, y) \quad (1)$$

$$H_2 = H(x, y+1) - H(x, y-1) \quad (2)$$

$$\theta(x, y) = arctan(\frac{H_1}{H_2}) \quad (3)$$

$$m\,(x, y) = \sqrt{{H_1}^2 + {H_2}^2} \quad (4)$$

Among them, $H(x, y)$ is the gray value of the normalized image at (x, y).

Then, the gradient size and direction are counted to obtain the gradient histogram. When calculating the gradient histogram, the HOG feature takes the block as the sampling window, a block contains 3 × 3 cells, and a cell contains 9 × 9 pixels. The gradient direction is used to project each pixel in the cell, and the gradient size is used as the weight of the projection.

Finally, $L2$ regularization is performed on the gradient size in the block, which can effectively reduce the influence of illumination on HOG features. The normalization formula is as follows:

$$B_n = \frac{x_n}{\sqrt{\|x_n\|^2 + \xi^2}} \quad (5)$$

where B_n is the normalized result, x_n is the corresponding block vector, and ξ is a very small positive number.

B. FEATURE SELECTION AND FEATURE DICTIONARY GENERATION

After feature extraction, a huge feature set is often generated. At this time, feature selection algorithms are usually used to filter features, remove redundant and unimportant features, and retain the most effective features. This can not only effectively reduce the number of features and avoid dimension disaster, but also reduce the influence of irrelevant features on the classifier and improve the performance of the classifier.

Therefore, to obtain features with good detection effects and stability, this paper uses a custom maximum discrimination feature filter to filter the HOG features of the hand edge and defines the filtered features as maximum discrimination features, which are the features with voting rights. Each Maximum discrimination feature corresponds to a concentric offset vector, which is an offset determined by the maximum discrimination feature and the fingertip position.

The specific process of feature selection and feature dictionary generation is shown in Fig.3. The first part is the process of feature matching, which provides information basis for subsequent filtering. The HOG feature extraction method proposed in Section III-A is used to extract the features of the hand images, where, $I_g(F_i)(g = 1, 2, 3, \ldots, N,\ i = 1, 2, 3, \ldots, n)$ represents the ith HOG feature extracted from the gth hand image. Due to the difference in images, the number of HOG features in each hand image is different. In the second part, a maximum discrimination feature filter is established through (1) and (2) conditions to filter maximum discrimination features.

1) The $L2$ norm between the fingertip position detected by maximum discrimination features and the real fingertip position is very small.

$$Dist_j\,(F_i) = ||P_j(F_i) - Fingertip\,(j)\,||^2 < d \quad (6)$$

$$P_j(F_i) = \varphi(Q_j(F_s)) - Offset(F_i) \tag{7}$$

$$s = \arg\min_k ||I_g(F_i) - I_j(F_k)||^2 \tag{8}$$

where, $P_j(F_i)$ denotes the fingertip coordinates detected by the most similar feature to $I_g(F_i)$ in the jth graph, Fingertip(j) denotes the real fingertip coordinates in the jth graph, $Dist_j(F_i)$ denotes the $L2$ norm of the detected and real fingertip coordinates in the jth graph, d is the threshold, and d is set to 15 pixels in this paper. Less than the threshold d is considered to satisfy very small. $Q_j(F_s)$ represents the most similar feature to $I_g(F_i)$ in the jth graph, s represents the label of the most similar feature to $I_g(F_i)$ in the jth graph, $\varphi(\cdot)$represents the position coordinate function corresponding to a HOG feature, $\varphi(Q_j(F_s))$ represents the coordinate of the most similar feature to $I_g(F_i)$ in the jth graph. Offset (F_i) represents the offset vector between the feature $I_g(F_i)$ and the real fingertip coordinate.

2) On the basis of 1), the maximum discrimination feature should also satisfy that they have appeared in most hand images. In other words, the frequency of maximum discrimination features in hand images exceeds the set threshold.

$$Frequency = \frac{\sum_{j=1}^{N} \alpha(Dist_j(F_i) < d)}{N} > T \tag{9}$$

where α represents the indicator function, if $Dist_j(F_i) < d$, the value is 1, otherwise 0. T is the set threshold, which was set to 0.5 in this paper, and N is the number of hand images.

IV. TRAINING OF VOTING RIGHTS CLASSIFIER FOR HAND CONTOUR COMPONENTS

In section III-A, two sample sets are obtained, one is a positive sample set composed of hand contour edge HOG features and the other is a negative sample set composed of background contour edge HOG features. Then, the maximum discrimination feature filter proposed in Section III-B is used to select the maximum discrimination features with good and stable detection effects from the positive sample set, and a voting rights classifier for hand contour components is trained with the maximum discrimination feature set and the negative sample set. In fact, the voting rights classifier of hand contour components is a binary classifier, which can classify the features with voting rights from all the features, namely maximum discrimination features. This method can remove the background features in the figure and the other features of the hand before the voting starts, and only retain the maximum discrimination features with a good fingertip detection effect to improve the detection speed and accuracy.

In this paper, the extreme gradient boosting (XGBoost) [21] algorithm is used as the voting rights classifier for hand contour components. The tree model is used as the base classifier. The number of base classifiers is p, and the number of base classifiers was set to 100. R is the base classifier space. Then the voting rights classifier can be expressed as follows:

$$\hat{C}_\alpha = \sum_{n=1}^{p} r_n(f_\alpha),\ r_n \in R \tag{10}$$

where, f_α represents the extracted αth contour edge HOG feature, $\hat{C}_\alpha$ represents the detected result of the voting rights classifier $\hat{C}$ on the feature f_α, r_n represents the nth base classifier, so $r_n(f_\alpha)$ represents the detected value of the nth base classifier. The objective function of the model training is:

$$\begin{aligned} Aim^{(t)} &= \sum_{\alpha=1}^{m} l\left(C_\alpha, \hat{C}_\alpha^{(t)}\right) + \sum_{n=1}^{t} \Omega(r_n) \\ &= \sum_{\alpha=1}^{m} l[C_\alpha, \hat{C}_\alpha^{(t-1)} + r_t(f_\alpha)] + \Omega(r_t) + constant \end{aligned} \tag{11}$$

$$\Omega(r_n) = \beta T + \frac{1}{2}\gamma \times \sum_{i=1}^{T} \omega_i^2 \tag{12}$$

C_α represents the true label of the feature f_α, and $l\left(C_\alpha, \hat{C}_\alpha\right)$ represents the loss between the detected value and the true label. To prevent over-fitting, the algorithm simplifies the model from two aspects: limiting the number of leaves and weight regularization. In Equation 12, $\Omega(r_n)$ represents the penalty, T represents the number of leaf nodes, β and γ are penalties, and ω_i is the value of leaf nodes. The voting rights classifier uses the Taylor series to approximate the objective function, which can be expressed as:

$$Aim^{(t)} \approx \sum_{\alpha=1}^{m} [G_t r_t(f_\alpha) + \frac{1}{2} H_t w_t^2(f_\alpha)] + \Omega(r_t) \tag{13}$$

where, $G_t = \frac{\partial l\left(\hat{C}_\alpha^{(t-1)}, C_\alpha\right)}{\partial \hat{C}_\alpha^{(t-1)}}, H_t = \frac{\partial^2 l\left(\hat{C}_\alpha^{(t-1)}, C_\alpha\right)}{\partial^2 \hat{C}_\alpha^{(t-1)}}$.

The above algorithm is described by pseudo-code. Algorithm 1 gives the whole training process of the voting rights classifier.

After training the voting classifier, the test image can be input into the voting classifier to obtain the maximum discrimination feature. The classification process of the voting rights classifier is shown in Fig.4. In the figure, the green labels are all the HOG features extracted, and the red labels are the maximum discrimination features.

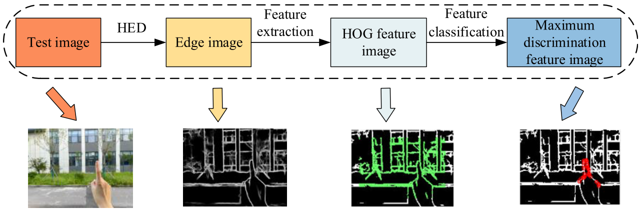


FIGURE 4. The classification process of voting rights classifier.

Algorithm 1 The Training of Voting Rights Classifier

```
Require: $N$ pure hand images with $m \times n$ resolution,$b$ background images with $m \times n$ resolution,Fingertip coordinates,feature selection thresholds $d$ and $T$.
Ensure: Feature dictionary,Voting rights classifier model.
HEDimage($m \times n$) $\leftarrow$ HEDmodel(hand($m \times n$),background($m \times n$))
// HED algorithm is used to detect the edge of two groups of images;
HandBinary($m \times n$),BackBinary($m \times n$) $\leftarrow$ Otsu(HEDimage($m \times n$))
// The Otsu algorithm is used to binarize the edge images;
for $j = 1$ to $N$ do
    edgehog1,postionset1,offset$\leftarrow$getHOG (HandBinary($m \times n$))
    // The HOG feature of binary image is extracted and the position and concentric offset vector of HOG feature are recorded;
end for
for $num = 1$ to num(edgehog1) do
    vector $\leftarrow$ edgehog1(:, $num$)
    for $t = 1$ to $N$ do
        matchNum $\leftarrow$topKneighbors(edgehog1(:, $t$), vector)
        // KNN algorithm is used for feature matching;
        matchX,matchY$\leftarrow$ postionset1(matchNum)
        detectFingertip(1,1) $\leftarrow$ matchX-offset(1,num)
        detectFingertip(2,1) $\leftarrow$ matchY-offset(2,num)
        distance $\leftarrow$getEuclidean(detectFingertip,fingertip)
        // Calculate the Euclidean distance between the detected fingertip position and the real fingertip position;
        if distance<$d$ then
            postivesum $\leftarrow$ postivesum+1
        end if
    end for
    if postivesum/$a > T$ then
        dictionary$\leftarrow$ save(vector, offset(:,num))
    end if
end for
Initialize XGBoostmodel
XGBoostmodel $\leftarrow$XGBoosttrian((HandBinary,HandLabel),(BackBinary, BackLabel))
// Train the XGBoost model as a voting rights classifier;
```

V. FINGERTIP DETECTION

A. KNN FEATURE MATCHING

KNN algorithm [22] is an algorithm that analyzes the similarity of samples by comparing the distance between two samples. In this paper, Euclideandistance is used to measure the similarity of samples. Through the KNN algorithm, each maximum discrimination feature obtained by classification is matched with the maximum discrimination features in the feature dictionary, and the top three features of similarity are taken as the matching features of the feature. Each matching feature is combined with the corresponding concentric offset vector for fingertip position voting to obtain a sparse fingertip detection space.

On the basis of the above voting, the fingertip detection confidence map can be used to represent the fingertip detection space. The calculation formula is as follows:

$$Pr = \sum_{j}^{V} \sum_{t}^{K} \hat{C}_j \times \left(p\left(f_j\right) - offset_t\left(f_j\right)\right) \tag{14}$$

Among them, $\hat{C}_j$ represents the classification result of the Voting rights classifier for feature j, $p(f_j)$ represents the position of the feature in the test image, $offset_t(f_j)$ represents the concentric offset vector corresponding to the feature matching the maximum discrimination feature f_j in the feature dictionary, V represents the number of maximum discrimination features, K represents the number of nearest neighbors, in this paper, set K to 3.

B. MEANSHIFT FINGERTIP DETECTION

In the above sections, the fingertip detection confidence map of the image is obtained. To obtain the final detected fingertip, the Meanshift algorithm [23] is proposed to find the point with the largest detected fingertip density in the sparse fingertip detection space as the final detected fingertip. In addition, considering that the fingertip detection space is a sparse space, there may be isolated points. Therefore, the maximum value in the fingertip detection space is selected as the initial value of the iteration. The formula for detecting the

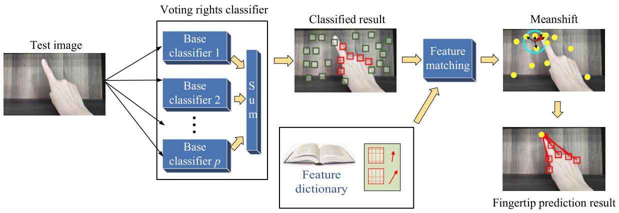


FIGURE 5. **Fingertip detection.**

fingertip iteration update is as follows:

$$Detect = \frac{\sum_{i=1}^{n} Pr(i)\, g\left(\left\| \frac{Detect-Pr(i)}{h} \right\|^{2}\right)}{\sum_{i=1}^{n} g\left(\left\| \frac{Detect-Pr(i)}{h} \right\|^{2}\right)} \tag{15}$$

The fingertip detection process is shown in Fig.5. Firstly, the edge HOG features are extracted from the test image and input into the voting rights classifier to obtain the maximum discrimination features and other features, which are represented by the red and green rectangular boxes in the figure respectively. The two types of features correspond to different centripetal offset vectors. Then, the maximum discrimination features represented by the red rectangular box are matched and voted with the maximum discrimination features in the feature dictionary, and the final detection fingertip position is searched by the Meanshift algorithm. This method of detecting fingertips based on the voting of hand contour components improves the fault tolerance of fingertip detection. Even if the voting rights classifier classification has some inaccurate classification, it will not affect the final detection results. Algorithm 2 gives the whole process of fingertip detection for test images.

VI. EXPERIMENTS

A. DATASET

In this paper, 1000 single-finger hand images under white background and 2000 complex background images without hand were used to train the voting rights classifier. The hand image is a self-made image with different shooting depths and different shapes. The complex background image is a subset C of the NUS-II data set taken on the campus of the National University of Singapore. In this paper, the edge HOG features of the hand image were extracted as the positive sample set, and the edge HOG features of the background image were extracted as the negative sample set. Then the maximum discrimination features were selected from the positive sample set, and the labels of the maximum discrimination feature set and the negative sample set were assigned to 1 and 0 respectively. The training set and the test set were divided according to the ratio of 8:2, and some training data sets are shown in Fig.6.

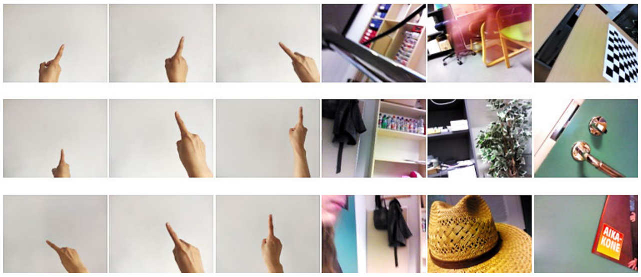


FIGURE 6. **Partial dataset images.**

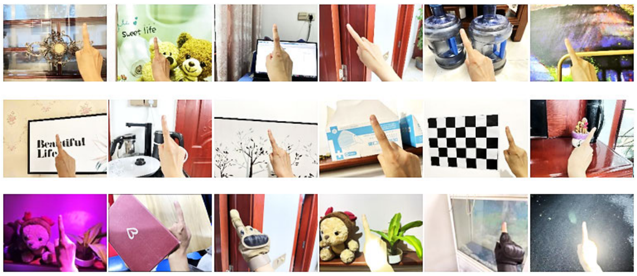


FIGURE 7. **Partial test images.**

When filtering maximum discrimination features, it is necessary to obtain the fingertip position of the hand image and the offset between the position of each HOG feature and the fingertip position. In this paper, the single-finger hand images are under a white background. Therefore, firstly, the maximum center of gravity distance method based on YCbCr skin color segmentation [8] was used to obtain the fingertip position in the image, and then the vector from the position of the HOG feature to the fingertip position was calculated, and the vector was defined as the centripetal offset vector.

In addition, this paper also set up 200 single-finger hand images under complex background to evaluate the effect of fingertip detection. Firstly, hand images under red, yellow, local light, and natural light of different intensities were taken. Secondly, hand images of partial occlusion, wearing all-finger gloves and half-finger gloves were taken. Finally, the images of the left and right hands of different subjects were taken, and some test images are shown in Fig.7. The dataset presented in this paper is available on request from the corresponding author.

B. COMPARING METHODS

1) COMPARISON OF EDGE DETECTION ALGORITHMS

To verify the applicability of the HED edge detection algorithm, the HED algorithm was compared with the Canny algorithm on the test image set. From Fig.8, it can be seen that the HED algorithm focuses on hand contour and ignores unimportant details. On the contrary, the Canny algorithm depicts more irrelevant edges. The edge detection algorithm is required to retain only the edges of the hand as much as possible, which can reduce the difficulty of subsequent HOG feature selection and classification.

These two edge detection algorithms were applied to the proposed algorithm, and XGBoost classifier was used as the voting classifier to detect fingertips. The number of base

Algorithm 2 Fingertip Detection

```
Require: $c$ pure hand images with $m \times n$ resolution,Voting rights classifier model.
Ensure: Detected fingertip coordinates.
HEDimage($m \times n$) $\leftarrow$ HEDmodel(Test($m \times n$))
TestBinary($m \times n$) $\leftarrow$ Otsu(HEDimage($m \times n$))
for $j = 1$ to $c$ do
    edgehog1,postionset1,offset$\leftarrow$getHOG(TestBinary($m \times n$))
end for
edgehog, postionset$\leftarrow$ XGBoostmodel(edgehog1, postionset1)
for $s = 1$ to num(edghog) do
    vector$\leftarrow$ edgehog(:, $s$)
    matchNum$\leftarrow$topKneighbors(dictionary,vector)
    for $k = 1$ to 3 do
        $X(o) \leftarrow$ postionset(1,$s$)-dictionary(1, matchNum($k$))
        $X(o) \leftarrow$ postionset(2,$s$)-dictionary(1, matchNum($k$))
        $o \leftarrow o + 1$
    end for
end for
$(X_c, Y_c) \leftarrow$Meanshift($X, Y$)
// Search the final fingertip position with Meanshift algorithm;
```

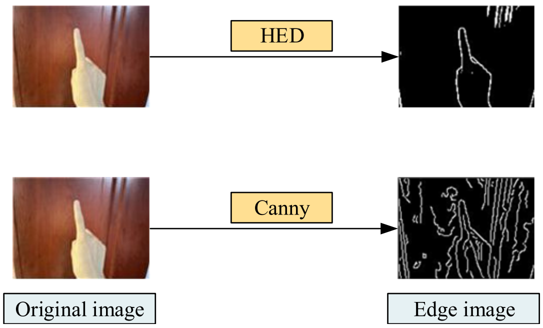


FIGURE 8. Edge detection.

classifiers of the voting rights classifier was set to 100, and the effect of fingertip detection of the two algorithms was evaluated by RMSE(RMSEx, RMSEy), accuracy rate(Acc), and missed detection rate(UnReco), as shown in Table 1. RMSE [1] was used to evaluate the detection accuracy of the detected fingertip points (x_p, y_p). The smaller the RMSE, the higher the detection accuracy.

$$RMSEx = \sqrt{\frac{1}{m}\sum_{1}^{m}(x_p - x_t)^2} \tag{16}$$

$$RMSEy = \sqrt{\frac{1}{m}\sum_{1}^{m}(y_p - y_t)^2} \tag{17}$$

where, (x_t, y_t) is the real fingertip coordinates, and m is the number of test images. The size of the image was 120 × 160, and the Euclidean distance between the detected fingertip and the real fingertip was defined as less than 15 pixels. Accuracy refers to the proportion of detected correct images in all images.

$$Acc = \frac{\sum_{1}^{m} F(\sqrt{(x_p - x_t)^2 + (y_p - y_t)^2} < 15)}{M} \tag{18}$$

where, M is the total number of images, $F(\cdot)$ is the indicator function, if $\sqrt{(x_p - x_t)^2 + (y_p - y_t)^2} < 15$, then $F(\cdot) = 1$, otherwise $F(\cdot) = 0$. In the process of fingertip detection, there is a situation where all the maximum discrimination features in the image are not correctly classified, so the fingertip cannot be detected. To evaluate the frequency of such cases, this paper used UnReco to characterize the proportion of undetected fingertip images in all images.

$$UnReco = \frac{Q}{M} \tag{19}$$

where Q represents the number of images where the fingertip is not detected.

Table 1 shows that the HED algorithm is far superior to the Canny algorithm in the detection precision, accuracy, and missed detection rate of fingertip detection. It is proved that the HED algorithm can better deal with the details, ignore the invalid edges, and obtain more effective edges to adapt to the selection and classification of HOG features when dealing with hand images in complex backgrounds, thereby improving the accuracy of fingertip detection.

2) COMPARISON OF FEATURE SELECTION AND NON-FEATURE SELECTION

To verify the maximum discrimination feature filter in the role of the fingertip detection process, this paper used the test images mentioned in section VI-A to do a comparative experiment. First, the unfiltered features were directly used to train the voting rights classifier and complete the fingertip

TABLE 1. Performance of fingertip detection of different edge detection algorithms.

	Evaluating indicators			
Edge detection algorithms	RMSEx	RMSEy	Acc	UnReco
Ours-Canny [24]	8.5710	7.8410	33.50%	53.50%
Ours-HED [20]	2.0097	4.0864	99.00%	0.00%

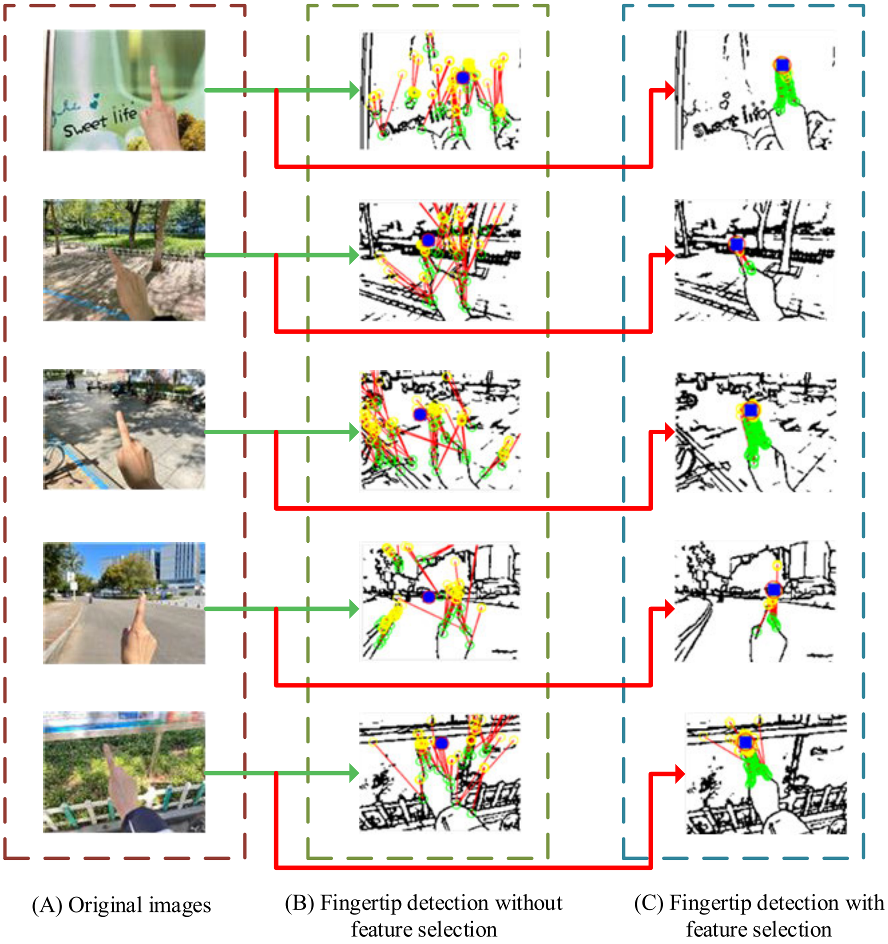

(A) Original images
(B) Fingertip detection without feature selection
(C) Fingertip detection with feature selection

FIGURE 9. Comparison between feature selection and non-feature selection.

detection, some results are shown in Fig.9 (B). Then, the features filtered by the maximum discrimination feature filter were used to train the voting rights classifier and perform fingertip detection, some of the results are shown in Fig.9 (C). In the figure, the green markers are the HOG features, the red lines are the centripetal offset vectors, the yellow markers are the fingertip detection space, and the blue rectangles are the final fingertips.

Fig.9 shows that the fingertip detection effect of (C) is significantly better than that of (B), and compared with (C), there are many background features in (B) that are mistaken for hand features to participate in the voting, so that there are many points in the fingertip detection space that deviate from the real fingertip position. These points affect the search of the final fingertip position by the Meanshift algorithm, which leads to the deviation of the result. The maximum distinguishing feature filter designed in this paper has two filtering conditions: one is that the detection effect of the feature is good, and the other is that the frequency of the feature is high. The features that meet these two conditions were called the maximum distinguishing features and were given voting rights. The selected features are more regular, which is conducive to the training of the voting rights classifier. In addition, the two experiments were evaluated by the evaluation indicators mentioned in Section VI-B1, as shown in Table 2.

Table 2 shows that UnReco can still maintain 0 %, but the Acc reduces to 67 %. This is due to the voting rights classifier using the background features as the hand features for fingertip detection. It is also because of the voting of

TABLE 2. **Performance of fingertip detection of feature selection and non-feature selection.**

	Evaluating indicators			
Algorithms	RMSEx	RMSEy	Acc	UnReco
Non-feature selection	15.2626	9.6569	67.00%	0.00%
Feature selection	2.0097	4.0864	99.00%	0.00%

these features that the RMSE increased significantly and the accuracy of fingertip detection reduced.

3) COMPARISON OF CLASSIFIERS

In this paper, the prepared data set was applied to different classifiers, and the classification effects of different classifiers on maximum discrimination features and other features were compared. The Accuracy,Precision,Recall, and F1-score index [25] were selected to evaluate the classification effect.

Accuracy is used to characterize the correct situation of each classifier in the classification of maximum discrimination features and other features. The calculation formula is as follows:

$$Accuracy = \frac{TP + TN}{Total} \tag{20}$$

Among them, TP is the number of correctly classified maximum discrimination features in the test images, TN is the number of correctly classified other features in the test images, and Total is the total number of features in the test images.

Precision is the proportion of the maximum discrimination features correctly classified in all maximum discrimination features. The calculation formula is:

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

where FP is the number of maximum discrimination features that are misclassified in the test images.

Recall is the ratio of the detected accurate maximum discrimination features to all maximum discrimination features in test images. The calculation formula is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

where FN is the number of other features misclassified in the test images.

F1-score is used to comprehensively evaluate the precision and recall rate. According to the selection of different sizes of β, the algorithm 's emphasis on recall and precision is determined. In this paper, β was set to 1, and the calculation formula is as follows:

$$F_{\beta} = \left(1 + \beta^2\right) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{23}$$

The evaluation results are shown in Table 3.

According to the evaluation results of the above table, compared with other classifiers, XGBoost has the highest accuracy, reaching 99.40%, and the XGBoost classifier also has a better performance on the comprehensive evaluation index F1-score, reaching 0.9970.

That is to say, when classifying maximum discrimination features and other features, the XGBoost classifier can classify maximum discrimination features more comprehensively on the basis of ensuring accuracy. In addition, the XGBoost classifier can avoid overfitting through regularization terms, so it has better generalization ability. The Accuracy and F1-score results of each classifier are shown in Fig.10.

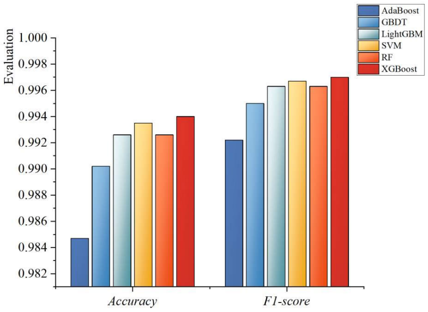


FIGURE 10. **Comparison of classifiers.**

4) TEST IN COMPLEX BACKGROUNDS

The proposed algorithm was compared with skin color segmentation based on YCbCr, object detection based on YOLO, and fingertip detection based on YOLO-YCbCr. The following is a detailed description of various methods.

- Fingertip detection based on YCbCr skin color segmentation [31].
  The algorithm first used the skin color threshold in the YCbCr color space to segment the hand region to obtain the hand contour, and then combined the local maximum value of the cumulative curvature and the detection of convex defects to detect the fingertip.
- Fingertip detection based on YOLO [11].
  The algorithm used the YOLOv5 algorithm to detect the fingertip position directly. During the training experiment, the fingertip positions in the 1000 single-finger hand images mentioned in section VI-A were first marked, and then the YOLOv5 model was trained.
- Fingertip detection based on YOLO-YCbCr [11], [31].
  Firstly, the YOLOv5 algorithm was used to detect the hand region. In the training experiment, 1000 single-finger hand images mentioned in section VI-A were randomly fused with complex background images to

TABLE 3. Performance of feature classification of the comparing classifiers.

Classifiers	Evaluating indicators			
	Accuracy	Precision	Recall	F1-score
AdaBoost [26]	98.47%	99.60%	98.84%	0.9922
GBDT [27]	99.02%	99.45%	99.56%	0.9950
LightGBM [28]	99.26%	99.62%	99.64%	0.9963
SVM [29]	99.35%	99.71%	99.63%	0.9967
RF [30]	99.26%	99.52%	99.74%	0.9963
XGBoost [21]	99.40%	99.69%	99.71%	0.9970

TABLE 4. Performance of fingertip detection of different algorithms.

Algorithms	Evaluating indicators			
	RMSEx	RMSEy	Acc	UnReco
YCbCr [31]	54.8256	28.8637	40.00%	0.00%
YOLO [11]	\	\	57.00%	37.50%
YOLO-YCbCr [11], [31]	26.0165	24.6055	63.50%	12.50%
Ours-AdaBoost [26]	13.6416	9.859	95.50%	2.50%
Ours-GBDT [27]	9.3865	8.5646	80.00%	10.50%
Ours-LightGBM [28]	9.2195	6.5507	88.50%	6.05%
Ours-SVM [29]	10.0898	7.2922	84.50%	8.05%
Ours-RF [30]	3.6417	4.9303	81.50%	13.50%
Ours-XGBoost [21]	2.0097	4.0864	99.00%	0.00%

obtain 1000 fused images. Then, the hand regions in these images were marked, and the YOLOv5 model was trained. Finally, fingertip detection was performed on the detected hand region by the algorithm proposed in [31].

In the test experiment, the proposed algorithm and the above three algorithms were tested using the test image set mentioned in section VI-A. In addition, for the proposed algorithm, an additional set of experiments was set up to verify the influence of different voting rights classifiers on the final fingertip detection effect. Finally, the effect of the above algorithm was evaluated by the evaluation index in VI-B1. Table 4 shows the evaluation results of different algorithms.

Table 4 shows that compared with other classifiers, the RMSE of the XGBoost classifier for fingertip detection is significantly lower than other classifiers, and the Acc and UnReco are also the best in each classifier. This shows that XGBoost can use its own generalization ability to accurately extract maximum discrimination features when classifying complex background features and maximum discrimination features, thereby improving the accuracy of feature matching in the subsequent fingertip detection process. By comparing with the fingertip detection algorithm based on YCbCr skin color segmentation and YOLO target detection, it can be seen that the proposed algorithm has obvious advantages in dealing with complex background challenges. Part of the test results is shown in Fig.11.

From the fingertip detection results in Fig.11, it can be seen that the algorithm can accurately detect the fingertip position in the face of partial occlusion, wearing all-finger or half-finger gloves, dramatic changes in illumination, and different complex backgrounds.

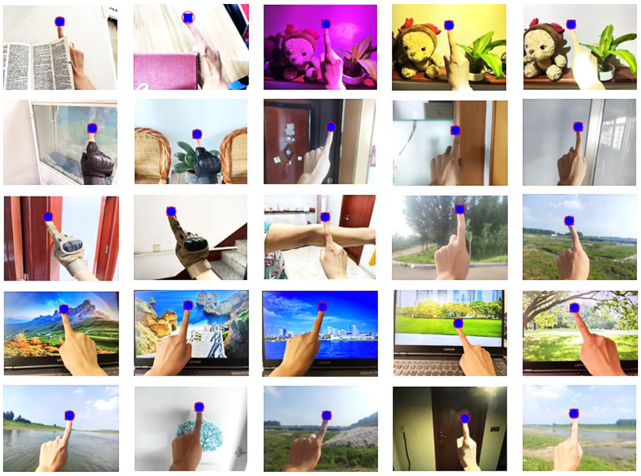


FIGURE 11. Part of fingertip detection results.

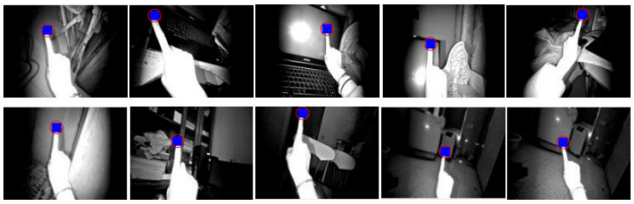


FIGURE 12. Fingertip detection results in infrared images.

To verify the robustness of the proposed algorithm to illumination, a set of hand images under complex background captured by an infrared camera without illumination at night were added to test the proposed algorithm. Some experimental results are shown in Fig.12.

It can be seen from Fig.12 that the proposed algorithm still has a good fingertip detection effect in the night environment, and RMSE maintains within 5 pixels, which shows that the proposed algorithm has good robustness to illumination.

VII. CONCLUSION

To improve the accuracy of fingertip detection in complex environments, this paper designed a fingertip detection algorithm based on the maximum discrimination HOG feature in complex backgrounds. The algorithm can effectively overcome the problems of severe illumination changes and partial occlusion in the process of fingertip detection. The following conclusions can be obtained through the experiments:

1) This paper compared the effects of two edge detection methods on fingertip detection, through experiments, it can be seen that the HED edge detection algorithm can emphasize the contour information between the background and the hand, ignoring irrelevant details, thereby reducing the difficulty of extracting maximum discrimination features, and the contour information is less affected by light and color changes, reducing the impact of environmental factors on fingertip detection.
2) This paper compared the classification effects of different types of classifiers on the maximum discrimination features of the hand and other features. The four evaluation indexes of Accuracy, Precision, Recall, and F1-score prove that the XGBoost classifier has better generalization ability in classifying the hand and background.
3) In this paper, a maximum discrimination feature filter was designed to filter the maximum discrimination features. Experiments show that feature selection can effectively improve the performance of the voting classifier, thereby improving the effect of fingertip detection.
4) In this paper, a fingertip detection method based on hand contour components was adopted, which makes the algorithm still effective when the hand is partially occluded and improves the fault tolerance rate and robustness.
5) The experimental results show that the proposed algorithm can still achieve 99% accuracy and the detected RMSE is kept in the range of 5 pixels in the face of severe illumination changes, partial hand occlusion, and complex background scenes. Compared with the traditional hand gesture detection algorithms based on skin color and deep learning, it can maintain higher accuracy and smaller prediction error in fingertip detection.

REFERENCES

[1] N. Potigutsai and O. Sornil, “Hand and fingertip detection for game-based hand rehabilitation,” in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2021, pp. 36–43.

[2] Y.-H. Chen, P.-C. Su, and F.-T. Chien, “Air-writing for smart glasses by effective fingertip detection,” in *Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE)*, Oct. 2019, pp. 381–382.

[3] J. Cheng, Q. Wang, R. Song, and X. Wu, “Fingertip-based interactive projector-camera system,” *Signal Process.*, vol. 110, pp. 54–66, May 2015.

[4] A. Z. Shukor, M. F. Miskon, M. H. Jamaluddin, F. B. A. Ibrahim, M. F. Asyraf, and M. B. B. Bahar, “A new data glove approach for Malaysian sign language detection,” *Proc. Comput. Sci.*, vol. 76, pp. 60–67, Jan. 2015.

[5] D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, “Real-time virtual mouse system using RGB-D images and fingertip detection,” *Multimedia Tools Appl.*, vol. 80, no. 7, pp. 10473–10490, Mar. 2021.

[6] M. I. H. S. Bahrin, H. M. Yusof, and S. N. Sidek, “Tactile graphics exploration studies using fingertip tracking based on colour markers detection for visually impaired people,” in *Proc. 7th Int. Conf. Mechatronics Eng. (ICOM)*, Oct. 2019, pp. 1–6.

[7] L.-K. Lee, S.-Y. An, and S.-Y. Oh, “Robust fingertip extraction with improved skin color segmentation for finger gesture recognition in human-robot interaction,” in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–7.

[8] G. Wu and W. Kang, “Robust fingertip detection in a complex environment,” *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 978–987, Jun. 2016.

[9] P. Mishra and K. Sarawadekar, “Fingertips detection in egocentric video frames using deep neural networks,” in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Dec. 2019, pp. 1–6.

[10] Y. Huang, X. Liu, X. Zhang, and L. Jin, “A pointing gesture based egocentric interaction system: Dataset, approach and application,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 370–377.

[11] M. M. Alam, M. T. Islam, and S. M. M. Rahman, “Unified learning approach for egocentric hand gesture recognition and fingertip detection,” *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108200.

[12] Y.-H. Chen, C.-H. Huang, S.-W. Syu, T.-Y. Kuo, and P.-C. Su, “Egocentric-view fingertip detection for air writing based on convolutional neural networks,” *Sensors*, vol. 21, no. 13, p. 4382, Jun. 2021.

[13] L. Shen, M. Zhang, and Q. Chen, “The Poggendorff illusion driven by real and illusory contour: Behavioral and neural mechanisms,” *Neuropsychologia*, vol. 85, pp. 24–34, May 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0028393216300677

[14] R. Sun, X. Wang, and X. Ye, “Real-time pedestrian detectionl using OpenCL,” in *Proc. Int. Conf. Audio, Lang. Image Process.*, Jul. 2014, pp. 401–404.

[15] M. Kowdiki and A. Khaparde, “Automatic hand gesture recognition using hybrid meta-heuristic-based feature selection and classification with dynamic time warping,” *Comput. Sci. Rev.*, vol. 39, Feb. 2021, Art. no. 100320.

[16] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[17] A. Deleplace, V. Atamuradov, A. Allali, J. Pellé, R. Plana, and G. Alleaume, “Ensemble learning-based fault detection in nuclear power plant screen cleaners,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 10354–10359, 2020.

[18] L. Cui, Q. Zhang, Y. Shi, L. Yang, Y. Wang, J. Wang, and C. Bai, “A method for satellite time series anomaly detection based on fast-DTW and improved-KNN,” *Chin. J. Aeronaut.*, vol. 36, no. 2, pp. 149–159, Feb. 2023.

[19] W. Min, Y. Zhang, J. Li, and S. Xu, “Recognition of pedestrian activity based on dropped-object detection,” *Signal Process.*, vol. 144, pp. 238–252, Mar. 2018.

[20] K. Heidler, L. Mou, C. Baumhoer, A. Dietz, and X. X. Zhu, “HED-UNet: Combined segmentation and edge detection for monitoring the Antarctic coastline,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 4300514.

[21] M. Rahman, Y. Cao, X. Sun, B. Li, and Y. Hao, “Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray,” *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107252.

[22] X. Wu, “Top 10 algorithms in data mining,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2007.

[23] T. Zhu, X. Wang, J. Zhang, S. Yu, and I. Molotov, “Mean-shift clustering approach to the tracklets association with mean-shift clustering approach to the tracklets association with angular measurements of resident space objects,” *Astron. Comput.*, vol. 2022, Jun. 2022, Art. no. 100588.

[24] X. Hu and Y. Wang, “Monitoring coastline variations in the pearl river estuary from 1978 to 2018 by integrating Canny edge detection and Otsu methods using long time series landsat dataset,” *CATENA*, vol. 209, Feb. 2022, Art. no. 105840.

[25] Y. Guo, H. Wang, Y. Guo, M. Zhong, Q. Li, and C. Gao, "System operational reliability evaluation based on dynamic Bayesian network and XGBoost," *Rel. Eng. Syst. Saf.*, vol. 225, Sep. 2022, Art. no. 108622.

[26] Z. Liu, K. Zhang, C. Wang, and S. Huang, "Research on the identification method for the forest fire based on deep learning," *Optik*, vol. 223, Dec. 2020, Art. no. 165491.

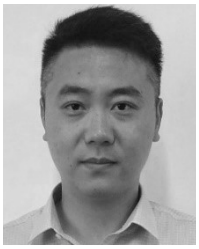
[27] X. W. Liu, Z. L. Long, W. Zhang, and L. M. Yang, "Key feature space for predicting the glass-forming ability of amorphous alloys revealed by gradient boosted decision trees model," *J. Alloys Compounds*, vol. 901, Apr. 2022, Art. no. 163606.

[28] A. Sharma and B. Singh, "AE-LGBM: Sequence-based novel approach to detect interacting protein pairs via ensemble of autoencoder and LightGBM," *Comput. Biol. Med.*, vol. 125, Oct. 2020, Art. no. 103964.

[29] M. Chandrakala and P. D. Devi, "Two-stage classifier for face recognition using HOG features," *Mater. Today, Proc.*, vol. 47, pp. 5771–5775, Jan. 2021.

[30] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J.*, vol. 20, no. 1, pp. 3–29, 2020.

[31] S. Cao and X. Wang, "Real-time dynamic gesture recognition and hand servo tracking using PTZ camera," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27403–27424, Oct. 2019.



PENG JI received the Ph.D. degree in instrument science and technology from Southeast University, Nanjing, China, in 2017. He is currently an Associate Professor with the School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include robotic vision, machine learning, and intelligent robot.

JINXIANG FENG is currently pursuing the master's degree with the School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences). Her research interests include computer vision, machine learning, and intelligent robot.

FENGYING MA received the M.S. degree in control theory and control engineering from the Shandong University of Science and Technology, Jinan, China, in 1999, and the Ph.D. degree in communications and information systems from the China University of Mining and Technology, Beijing, China, in 2009. In 1999, she joined the School of Information and Automation Engineering, Qilu University of Technology (Shandong Academy of Sciences), where she is currently a Professor. Her research interests include machine learning, computer networks, and wireless communications.

XIANJIAN WANG, photograph and biography not available at the time of publication.

CHENGLONG LI, photograph and biography not available at the time of publication.