A Hand-Washing Support System Based on Center of Gravity in Hand Correction

Katsumi Nagata

September　2021

Department of Information Science and Intelligent Systems
Graduate School of Advanced Technology and Science
Tokushima University, Japan

CONTENTS

SUMMARY

Aiming at solving the problem of recognition accuracy decrease during hand-washing in existing systems due to noise produced by hand-shaking, lighting changes, and so on, we develop a novel system. The system focuses on proper hand-washing patterns to examine whether hands are being washed correctly. By comparing videos of hand-washing captured at sinks and learning model videos, the system informs users in real time whether they are using proper hand-washing patterns. Optical flows and skin-color areas are used as data features to recognize proper hand-washing patterns. Support vector machine is used as the recognition model in this system. The hand-washing pattern recognition accuracy is improved by removing noise sources and capturing pattern characteristics by labeling and using a correction that considers the center of gravity in hand.

Keywords: Health system; Image processing; Machine learning; Support vector machine.

1. Introduction

Every year, the Ministry of Health, Labor, and Welfare announces the infection status, vaccines, and treatment methods due to the emergence of new strains of influenza, noroviruses, and coronaviruses. These diseases are highly contagious and their symptoms are often severe. According to the 2012 Vital Statistics[1], of approximately 1.26 million deaths, approximately 30,000 were due to infectious and parasitic diseases and approximately 200,000 were due to respiratory diseases caused by infectious diseases.

Vaccination is one of the most promising methods for preventing infection, but it is not a panacea. Reasons for this include "high cost," "need for periodic vaccination," and "inability to cope with unknown pathogens." Even though the vaccines is not yet complete, the new coronavirus has killed about 4 million people in the worldwide.[2] In addition, since infectious diseases have different causative pathogens and different routes of infection, their preventive methods also differ. For these reasons, it is difficult to say that vaccination is a perfect preventive method.

On the other hand, there are some common preventive methods, such as gargling and hand-washing. In particular, hand-washing is a common method for preventing infection in various places.

Generally, hand washing removes physical dirt from our daily lives (particularly, transient bacteria; transient germs are germs that are temporarily attached to the skin). There are different types of hand-washing, such as routine hand-washing, hygienic hand-washing, and surgical hand-washing. Daily handwashing is handwashing with liquid soap and running water to wash away and sterilize transient bacteria. (Scrubbing method) [3]. The latter hand-washing method is commonly found in medical settings and is intended to prevent contact infections. Hygienic hand-washing refers to hand-washing with disinfectant and running water for sterilization and elimination of transient bacteria. (Rubbing method)[3] Operative handwashing is handwashing to disinfect and eliminate transient bacteria and eliminating indigenous bacteria. Correct hand-washing is a washing method that essentially consists of hygienic hand-washing[4].

Hand-washing is an important preventive measure, but few people are aware of proper hand-washing. Some reasons for this are: "I don't know how to wash my hands properly," "Hand-washing is tedious," and "It is difficult to know whether my hands are clean or not.”

The washing method using the hygienic hand-washing movement is used as the correct hand-washing method in this study.

Several studies on hand-washing have demonstrated that proper hand-washing is effective in terms of hygiene[5][6][7]. To prove this, it is necessary to verify that dirt is properly removed via chemicals or visual inspection by experts[4]. However, while it is a good idea to use chemicals to check if dirt has been washed properly during experiments, it is not convenient to use chemicals or instruments for washing in daily life because it is expensive and difficult to check.

In this study, we focused on the correct behavior of hygienic hand-washing based on the premise of alcohol disinfection and developed a system to verify whether hands are being washed correctly. By comparing hand-washing videos taken at a washbasin with those of a learning model, the system can notify a user in real time whether he or she is following an appropriate hand-washing pattern. To determine the appropriate hand-washing pattern, we developed a system for extracting features related to "correct hand-washing" by applying methods using motion and shape features as features related to hand movements in research on sign language recognition[8]and gesture recognition[9][10]. We have developed a system for extracting features related to "correct hand washing.”

The structure of this paper is as follows. First, Chapter 2 describes related research, and Chapter 3 describes the hand-washing support system based on the existing method. In Chapter 4, we describe the hand-washing support system using the proposed method. An experiment to show the effectiveness of the proposed method is conducted in

Chapter 5, and a discussion is presented in Chapter 6. Finally, in Chapter 7, we summarize and discuss future issues.

An overview of the system is shown in Figure 1. A user washes his or her hands in a real environment, and the scene is recorded by a fixed camera. Recorded hand-washing images are uploaded to the system in real time. The system analyzes the uploaded video and aims to notify the user of inappropriate hand-washing patterns.

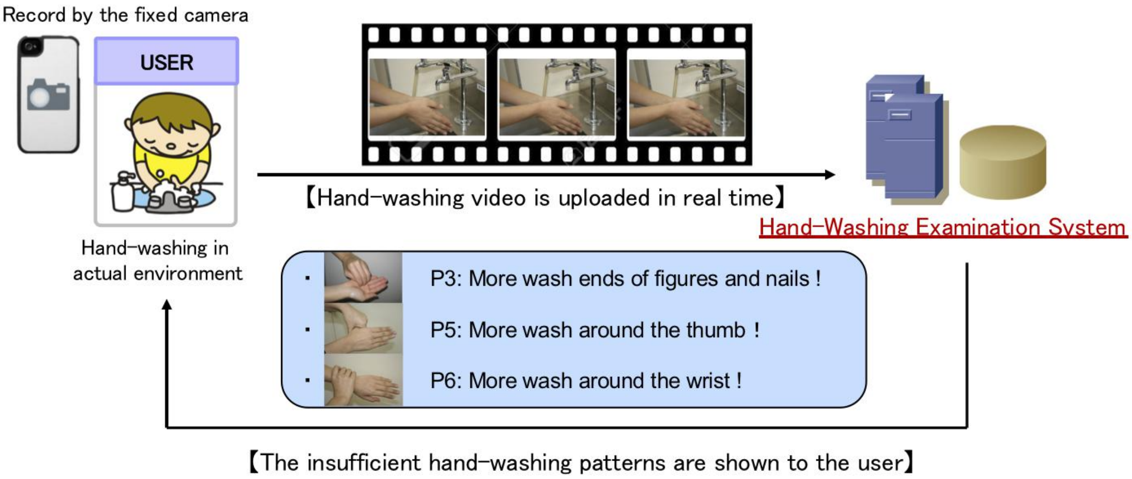

Record by the fixed camera
USER
Hand–washing in actual environment
【Hand–washing video is uploaded in real time】
Hand–Washing Examination System
・ P3: More wash ends of figures and nails !
・ P5: More wash around the thumb !
・ P6: More wash around the wrist !
【The insufficient hand–washing patterns are shown to the user】

Fig. 1: Hand-washing support system.

2. Related Research

In existing research, inspection systems for proper hand-washing have been developed using features of motion and shape associated with hand gestures found in sign language recognition [11] and gesture recognition [8]. However, recognition accuracy is degraded by noise caused by hand tremors and changes in lighting during hand-washing.

Motion features are often used in videos, which are composed of multiple images. A video consists of multiple images, and motion features can be determined from the motion of objects in the images. To identify a hand, optical flow is used to extract motion features from the hand motion.

In this section, we discuss related previous studies.

2.1 motor characteristics

Motion features are often used in videos, which are composed of multiple images. A video consists of multiple images, and motion features can be identified from the motion of objects in the images. To identify a hand, we use optical flow to extract motion features from the motion of the hand.

More individual "three-dimensional(3D) motions" than the two-dimensional (2D) model of parametric motion can be seen in many real-world videos. There is a limit to motion estimation by applying only 2D motion models such as parallel motion and rotation to videos. In addition, even in a video in which an object (or camera) moves horizontally and the motion can be expressed almost only by parametric motion, there are individual pixel-by-pixel motions in other parts of the video. There is a limit to the number of pixels that can be estimated. Therefore, motion estimation using "optical flow," a motion model that assigns independent movement parameters to each pixel, was proposed around 1980 and has been widely used since then.

Optical flow is the apparent motion in an image caused by the relative motion between the camera and the external world and corresponds to the point correspondence between the image at time t and the image at t+dt, a few seconds later. The term optical flow (or simply flow) may refer to a velocity vector at a single point in the image, or it may refer to a dense velocity vector field in the image. The computation of optical flow plays an important role as a preprocessing step for various tasks such as video analysis, shape restoration, region extraction, and object tracking. A similar term, motion the field, refers to a velocity field obtained by projecting a 3D velocity vector on the surface of an object in 3D space onto the image plane and is different from the optical flow. On the other hand, optical flow is an apparent motion that can be interpreted from an image and does not necessarily correspond to the physical motion of the object. In addition, optical flow cannot be uniquely determined from an image.

For example, if the size of a video is 640 × 480 pixels, the "flow vector" for each 640 × 480 pixel is predicted in the second image. The "field of 2D mobility vectors for each pixel between frames" is the optical flow.

Optical flow is a technique for predicting the flow vectors of the optical flow between neighboring frames "Image 1" and "Image 2" in a video, taking into account the pixel values around each pixel in the entire video and the number of flow vectors in the surrounding areas.

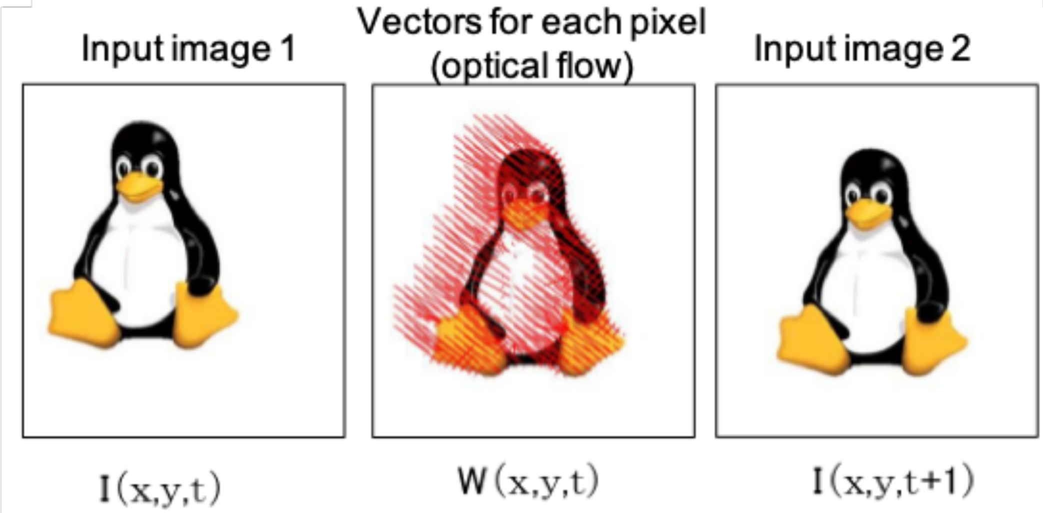


Fig. 2.1: Optical flow image

From the example shown in Fig. 2.1, we can see the movement vectors for each pixel between frames (input image 1 and input image 2). Next, an overview of two types of optical flow methods and the optical flow method used in this study [12] are described below.

2.1.1. Feature point-based method: Lucas-Kanade method

First, we assume that each subregion N (approximately 15 × 15 pixels) is moving in an independent parametric motion (affine transformation or translation). Then, the optical flow of each subregion is estimated independently by searching for a suitable location as the destination subregion, which is a method typical of the LK method (hereafter referred to as the LK method[13]). Since the correspondence search is performed for each subregion, the method appears to be tracking feature points. In this section, we will use the LK method as an example to provide an overview.

The basic mechanism of feature point-based optical flow estimation, including the LK method, is to calculate the correspondence score of the gray value of the entire surrounding region N for each pixel and to estimate the pixel with the highest correspondence as the destination. However, if we simply search for the destination using only the similarity of each small region N, there will be a high degree of ambiguity as to whether each candidate is the correct destination in images in which there are many regions with high similarity, depending on the size of the window and the search order.

Therefore, the LK method assumes that each pixel in image 1 is moved independently by a separate "movement vector (flow vector)" and that each pixel value in image 2 after the movement remains (almost) the same as the luminance value in image 1 before the movement. This equation is called the "Luca-Kanade equation."

The Lucas-Kanade equation is prepared for each pixel, and a total of 5 × 5 = 25 equations can be prepared as constraint equations before and after the movement of a small area N (e.g., 5 × 5 pixels). The optical flow of the entire region N is finally estimated using the least-squares method to find parameters when the sum of the errors in these 25 pixels is minimized. Since the approximation is based on the Taylor expansion, the solution that can be calculated here is limited to the mobility in a small area, so a sparse search using pyramid images is used to estimate larger motions.

Fig. 2.2 shows an optical flow image using the LK method.

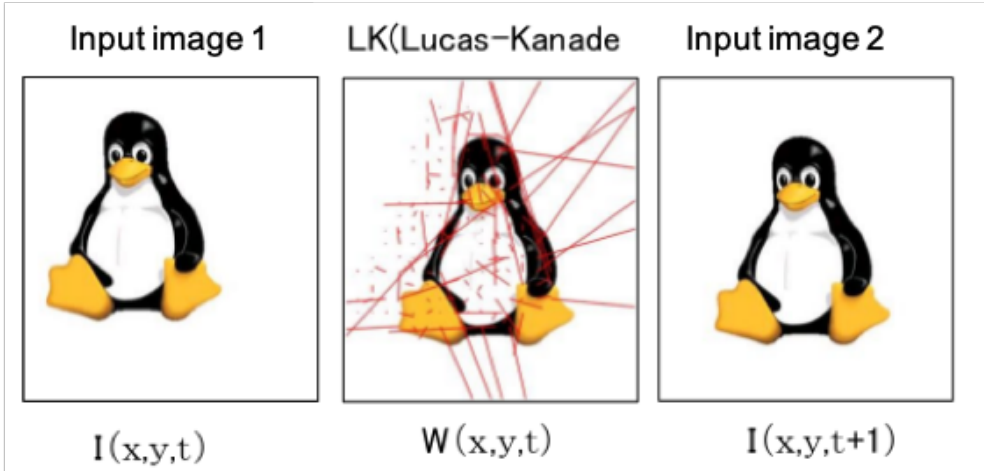


Fig. 2.2 : LK method

However, in image regions, where the degree of texture correspondence between regions N is ambiguous (e.g., pure white texture), it becomes difficult to determine where the destination will be, and even if the error function is least-squares for such pixels, a good optical flow cannot be estimated. For this reason, a feature point tracking mechanism called the "KLT tracker" [14] is provided.

The KLT tracker is an optical flow gradient method that can detect only feature points with the least ambiguity and compute the flow vector only at those points. Since feature points suitable for tracking are extracted beforehand when calculating the flow, fast and accurate computation is possible. However, there is a problem when extracting human motion: the number of flows that can be obtained is limited when the feature points are concentrated in a static area. Therefore, when feature points are concentrated in some image regions, it is difficult to capture the motion in other regions.

2.1.2. Variational-based method: Horn-Schunk method

There is a variational method, such as the Horn-Schunk method, that estimates the optical flow of all pixels at once by minimizing the error function of the entire image while considering the smoothness between adjacent flow vectors.

The variational method is a general framework for finding the optimal parameter values by iteratively minimizing an error function with many parameters. The method in (1) estimates the value of flow vectors by searching for the optimal destination for each pixel independently, but the difference is that in (2), the variational method can introduce smoothness between adjacent flow vectors. In (2), the smoothness of values between neighboring flow vectors is introduced by the variational method. In addition, we can use the smoothness transmitted from the surrounding flow vectors to make predictions.

In addition, another point (2) is that the optical flow can be obtained without assuming parametric motion for each small region as in the LK method. The value of each flow vector, which cannot be fully expressed by parametric motion, can be obtained by minimizing the error using the variational method.

Fig. 2.3 shows an optical flow image using the Horn-Schunk method.

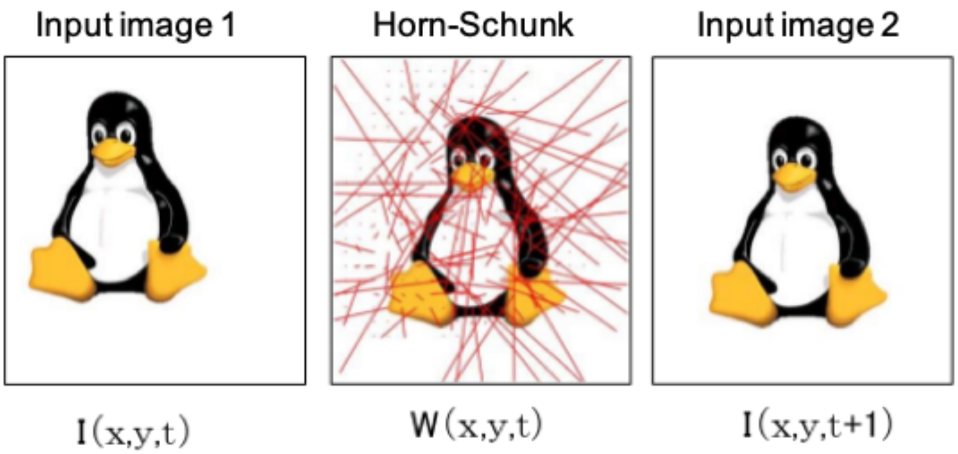


Fig. 2.3: Horn-Schunk method

2.1.3. Pyramid LK method

The pyramid LK method is an iterative implementation of the LK method using a pyramid structure of images.

A pyramid structure is a set of several different resolution images ranging from high resolution to low resolution. Here, we create a low-resolution image of 1/4 size by thinning out the even rows and columns while performing Gaussian smoothing on the input image and then create a low-resolution image from the created image. As an image, we can obtain multiple images such as the pyramid in Fig. 2.4.

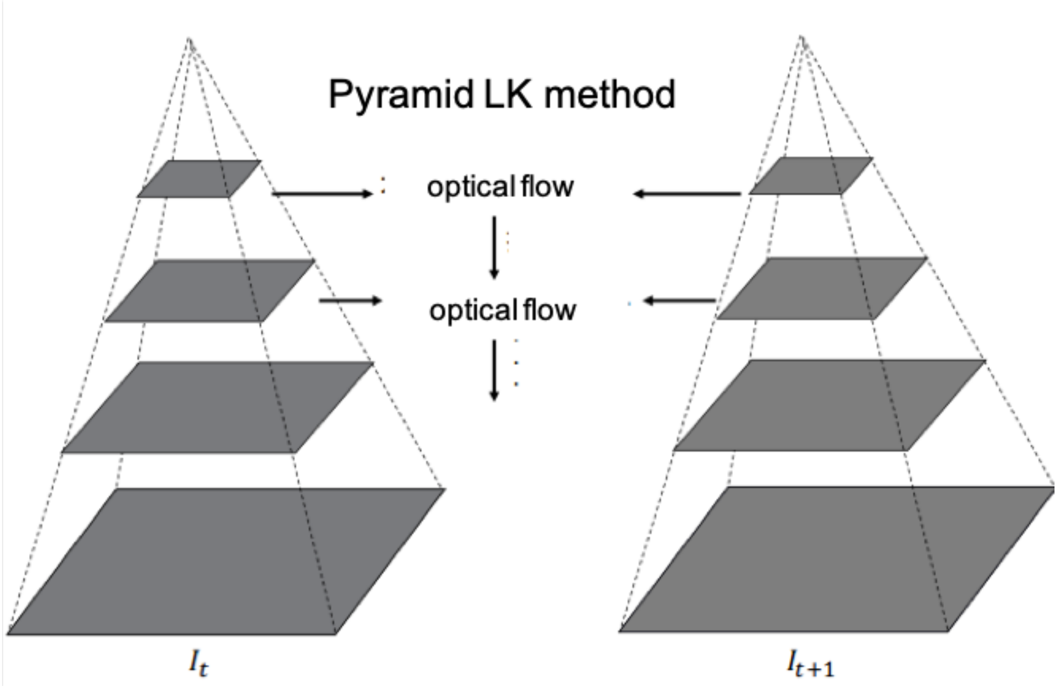


Fig. 2.4: Pyramid LK method: structural drawing

The original image is called hierarchy level 1, the next low-resolution image is called level 2, and so on. In the pyramid LK method, the LK method optical flow is first performed on the lowest resolution image to determine the approximate movement vector of each corner point in the next frame. The obtained result is used as the initial value for the next high-resolution image, and the area around the destination of the movement vector at low resolution is searched. The process is iterated until level 1 is reached, and the final movement vector is obtained. This method can compensate for the shortcomings of the LK method because it can cope with sudden changes in the flow. In addition, since we begin with a search for low-resolution images, unnecessary calculations can be avoided.

2.2 Shape features

Shape features contain shape information about an image, such as geometric features (distance, area, the center of gravity, the center of the target image, etc.) and features of the centerline and closed curve of the region after segmentation. However, when we obtain shape information about an image, we need to obtain shape information about the entire image. We use color information to specify the object for which we want shape information from an image and to obtain its center of gravity and area. To obtain color information from the image, the object is specified using the color information, and the center of gravity and area are obtained. The color specification method for obtaining color information from an image is generally based on the RGB, HSV, or HLS method. The acquisition of color information is described below.

2.2.1. RGB system

The RGB system is the most familiar color representation method for personal computers (PCs). RGB is a color representation method that combines the three primary colors of light, red, blue, and green, in 256 combinations each, to represent a wide range of colors, as shown in the figure below. Displays such as CRTs express colors by changing the brightness of pixels of the three colors R, G, and B. In this sense, it is the color representation method that has the greatest affinity with PCs. The values for R, G, and B range from 0 to 255 (or 00 to FF in hexadecimal), and the colors are specified by adjusting the color values in increments of 1 (Fig. 2.5).

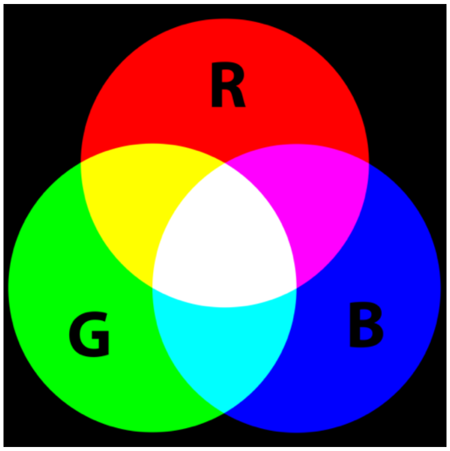


Fig. 2.5: RGB Method: Additive Mixing

In other words, with the RGB method, 256 × 256 × 256 = 16.77 million colors can be specified. However, while this method can maximize the color expressiveness of a PC, it has the drawback of being difficult to understand intuitively. Therefore, the following sections describe the HSV and HSL methods.

2.2.2. HSV method

HSV system (also called HSB system) is a color that is "easy for people to grasp intuitively," and HSV (HSB) color is represented using a color space consisting of three elements: Hue, Saturation, and Value or Brightness. This is a nonlinear transformation of RGB and is sometimes used to convert colors.

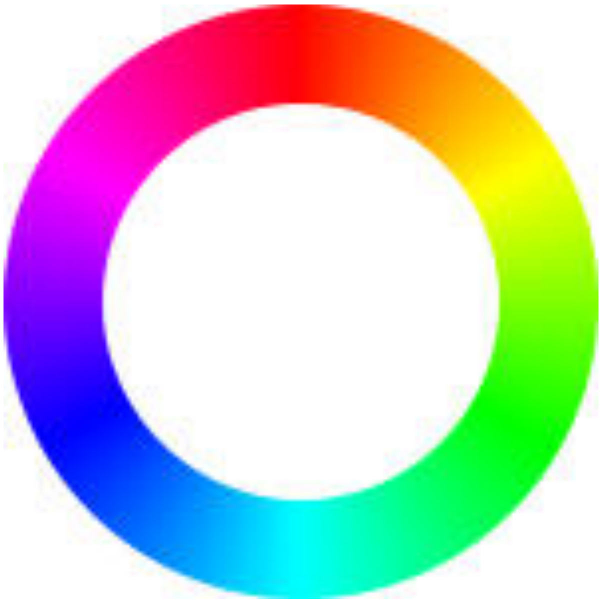


Fig.2.6: HSV color correlation

Hue" refers to the shades of red, yellow, green, blue, and purple, which can be used to classify colors (or more precisely, chromatic colors). If we put red, yellow, green, blue, and purple in order, and finally connect purple and red, a ring is formed, which is called "color correlation," as shown in Fig. 2.7. As can be seen from the fact that it is a ring, the hue can be varied from 0 to 360 degrees with respect to red.

Saturation is the vibrancy of a color, and the more vivid the color, the higher the value. Lightness is the brightness of a color and can be varied from 0% to 100%, with black being the lowest value.

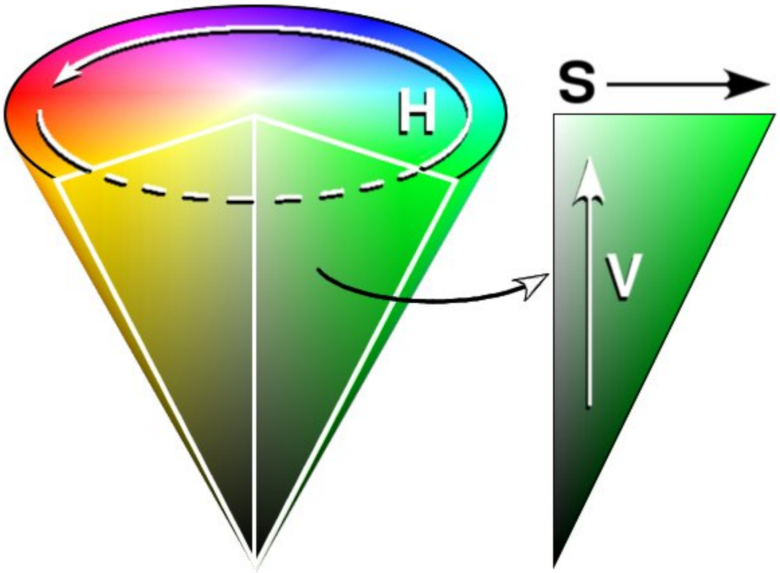


Fig. 2.7: HSV color correlation

Fig. 2.7 shows the HSV model, which is a graphical representation of hue (H), saturation (S), and lightness (v) described above.

RGB to HSV conversion formula

$$V \leftarrow max(R, G, B) \tag{2.1}$$

$$S \leftarrow \begin{cases} \frac{V - min(R,\, G,\, B)}{V} & ifV \neq 0 \\ 0 & otherwise \end{cases} \tag{2.2}$$

$$H \leftarrow \begin{cases} \frac{60(G-B)}{S} + 0 & ifV = R \\ \frac{60(B-R)}{S} + 120 & ifV = G \\ \frac{60(R-G)}{S} + 240 & ifV = B \end{cases} \tag{2.3}$$

If *H<0, H ← H + 360.* The range of output values is 0<=V<=1, 0<=S<=1, and *0<=H<=360.*

2.2.3. HLS method

The HLS method expresses the HLS color in a color space consisting of three elements: "hue," "saturation," and "lightness." It is a color specification method similar to the HSV method, and the basic concept is the same as the HSV method.
Hue" expresses a color as an angle in the range of 0 to 360 degrees.
Unlike HSV, "Saturation" in HLS is different from that in HSV in that saturation decreases from pure color to gray. Unlike HSV, "Lightness" refers to 100% pure color and indicates how much brightness is lost from there. 0% is considered black, 100% is considered white, and the intermediate (50%) is considered a pure color.

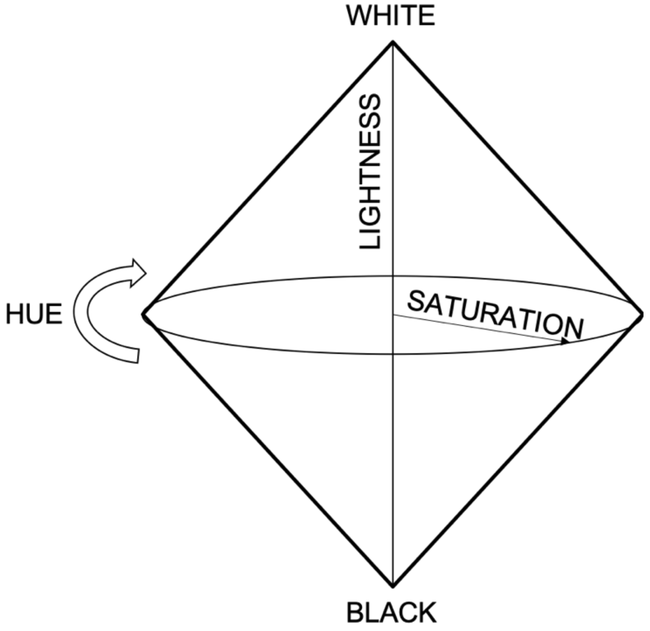


Fig2.8: HLS model

Fig. 2.8 shows the HLS model, which represents hue (H), saturation (S), and lightness (L) described above. However, comparing the figures, it is found the axes of lightness and saturation are taken in different ways.

RGB to HLS conversion formula.

$$V_{max} \leftarrow max(R, G, B) \tag{2.4}$$

$$V_{min} \leftarrow min(R, G, B) \tag{2.5}$$

$$L \leftarrow \frac{V_{max} + V_{min}}{2} \tag{2.6}$$

$$S \leftarrow \begin{cases} \frac{V_{max} - V_{min}}{V_{max} + V_{min}} & ifL < 0.5 \\ \frac{V_{max} - V_{min}}{2 - (V_{max} + V_{min})} & ifL \geq 0.5 \end{cases} \tag{2.7}$$

$$H \leftarrow \begin{cases} \frac{60(G-B)}{S} + 0 & ifV = R \\ \frac{60(B-R)}{S} + 120 & ifV = G \\ \frac{60(R-G)}{S} + 240 & ifV = B \end{cases} \tag{2.8}$$

If H<0, H ← H + 360. The range of output values is 0<=V<=1, 0<=S<=1, and 0<=H<=360.

2.3 Proper hand-washing

Hands are one of the most commonly used tools in our daily lives. The skin that covers our hands is resistant and prevents the entry of harmful microorganisms such as bacteria. However, if we do not wash our hands, viruses can remain on our hands and enter our mouths.

Viruses that cause diseases such as influenza and norovirus can be spread from different sources. For example, doorknobs, handrails, shared computers, and all other objects in our daily lives. A virus can enter the body by rubbing one's nose or eyes with one's hands or by putting food in one's mouth.

2.3.1. Unwashed areas

In routine hand-washing and hygienic hand-washing, if the correct hand-washing is not performed, there will be unwashed areas on the hands. Rinsing with running water is insufficient to reduce the number of germs on the hands. Fig. 2.9 shows a common area of unwashed hands.

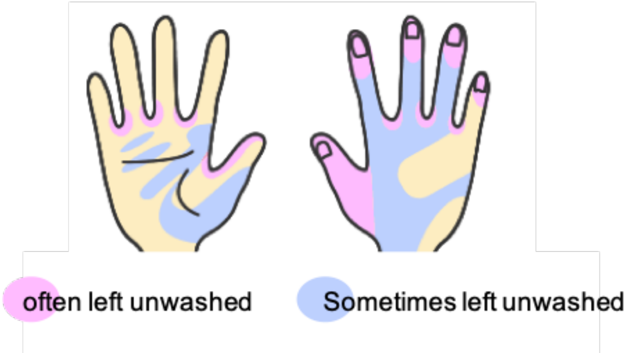


Fig. 2.9: Unwashed areas

Fig.2.9 shows the areas that are often left unwashed, such as the fingertips, thumbs, between the nails, and between the fingers, as shown in Fig. 2.9. If left unwashed, a virus can enter the body and cause illness. Therefore, it is necessary to indicate whether proper hand-washing is being performed.

2.3.2. Correct handwashing pattern

The MHLW recommends the implementation of the six steps of correct hand-washing to avoid unwashed hands, as described above. (Fig. 2.10) If the number of times you rub your hands and the movements you make during each step are correct, you can be sure that you are washing your hands correctly.

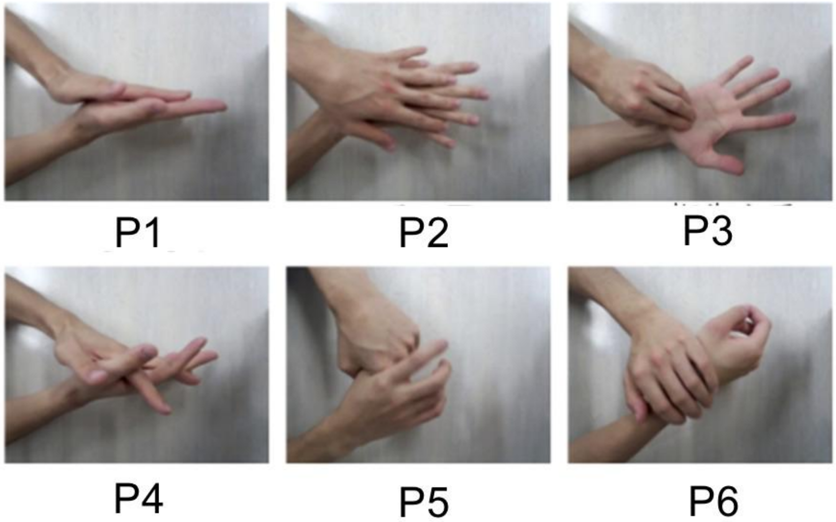


Fig.2.10: Patterns of correct hand washing

In this study, we defined these six patterns as the correct hand-washing patterns in hygienic hand-washing.

As shown in Fig. 2.10, the six patterns are "P1: rubbing the palm"; "P2: rubbing the back of the hand"; "P3: rubbing the fingertips and nails"; "P4: rubbing between the fingers"; "P5: twisting the thumb"; "P6: turning the wrist." By carefully washing each of these patterns for at least five seconds, we can ensure that we are washing our hands correctly.

This has been shown by experiments using chemicals. Therefore, by identifying the actions of the six patterns of correct hand-washing, we can determine whether we are washing our hands correctly.

3. Hand-washing Support System

It has been proven that correct handw-ashing with no unwashed areas is effective in preventing infectious diseases. [7]. If it is possible to automatically recognize whether correct hand-washing is being performed, it can be used not only in medical institutions but also in our daily lives. Although many studies have been conducted on correct hand-washing, it is difficult to accurately recognize correct hand-washing with existing inspection methods for correct hand-washing.

Prior studies include kits using chemicals and formulations that require observation by experts. Tsuchida[15] developed a "hand-washing learning system." With this system, users can learn how to move their hands for proper hand-washing, but they need to wear a 3-axis accelerometer on their wrist. In addition, this system cannot identify hand-washing patterns. Fujitsu Laboratories has also developed a new artificial intelligence-based detection method that automatically recognizes complex hand movements during hand-washing from images based on the overall shape of both hands and a series of hand-washing movements[16]. This enables highly accurate detection but requires a large amount of training data, and it is difficult to use specific features for hand motion recognition. This makes it difficult to compensate for unsuccessful recognition, which can lead to unexpected results. For these reasons, it is difficult to use the system correctly in general.

First, based on the "Study on Camera-Based Handwashing Inspection Method," we developed a handwashing support system using image processing technology. In the following, we describe the hand-washing inspection system and the problems of that research.

3.1 System Overview

Fig. 3.1 shows the outline of the hand-washing support system. As an input image, a real-time hand-washing image is used. An appropriate hand-washing video was used as the training video. Support vector machine (SVM) [17] was used for recognition.

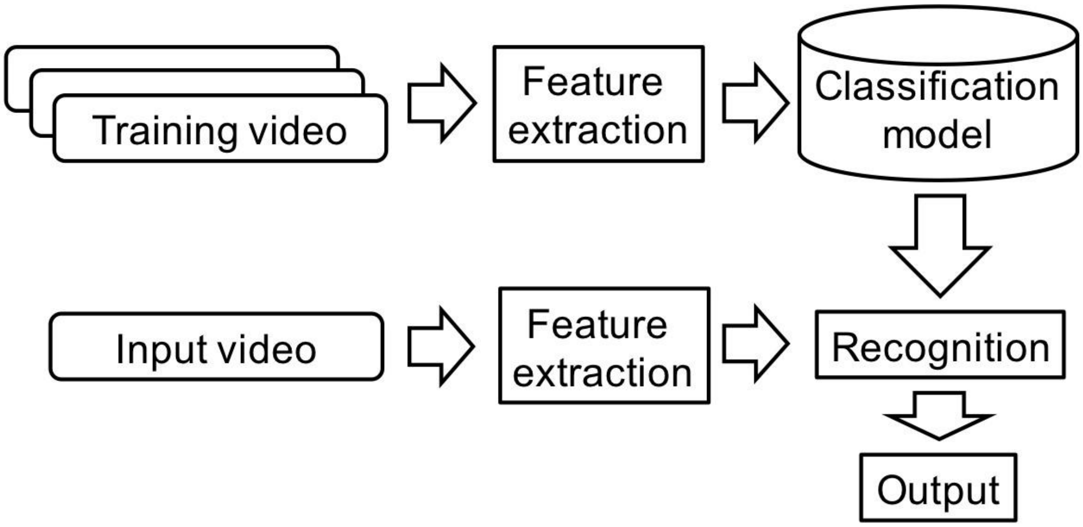


Fig. 3.1: Overview of hand-washing support system using image processing technology

Machine learning models based on deep learning models such as recurrent neural networks [18] and long short-term memory [19] are extremely effective in solving the problem of pattern matching of time-series data. However, these models require huge amounts of big data. On the other hand, SVM performs well, even with small amounts of data, so we introduced SVM as the learning model in our system.

The following sections describe in detail the input frames, features, and discrimination process (SVN and other methods).

3.2 Input Frames

A video can be converted into multiple images (frames). Features are obtained from these frames. However, it is extremely time-consuming to obtain features from each frame and identify them. Therefore, frames can be grouped as input frames, and feature values can be obtained to reduce the identification time.

We proposed two ideas for changing the interval. Ideas 1 and 2 are described below.

3.2.1. Constant frame interval (idea 1)

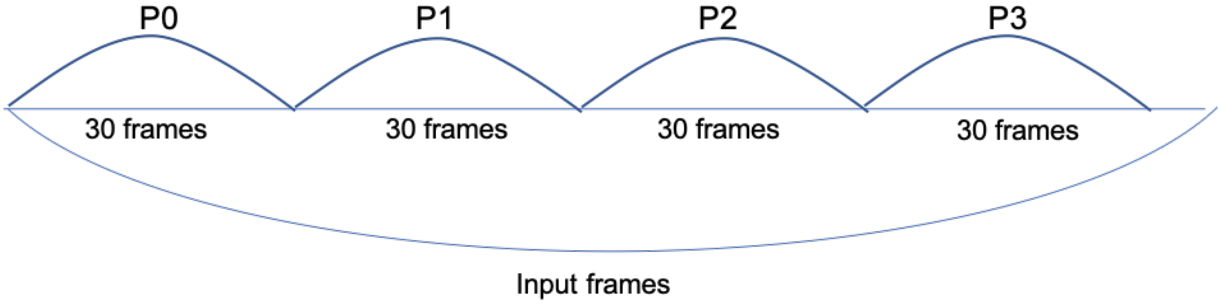


Fig. 3.2: onstant frame interval

In idea 1, we simply set the interval to 30 frames (approximately 1 s) and take them as input frames while shifting them. However, since we do not wash at regular intervals during actual washing, there is a possibility of false detection regarding the switching operation of each pattern.

3.2.2. Variation Frame Interval (idea 2)

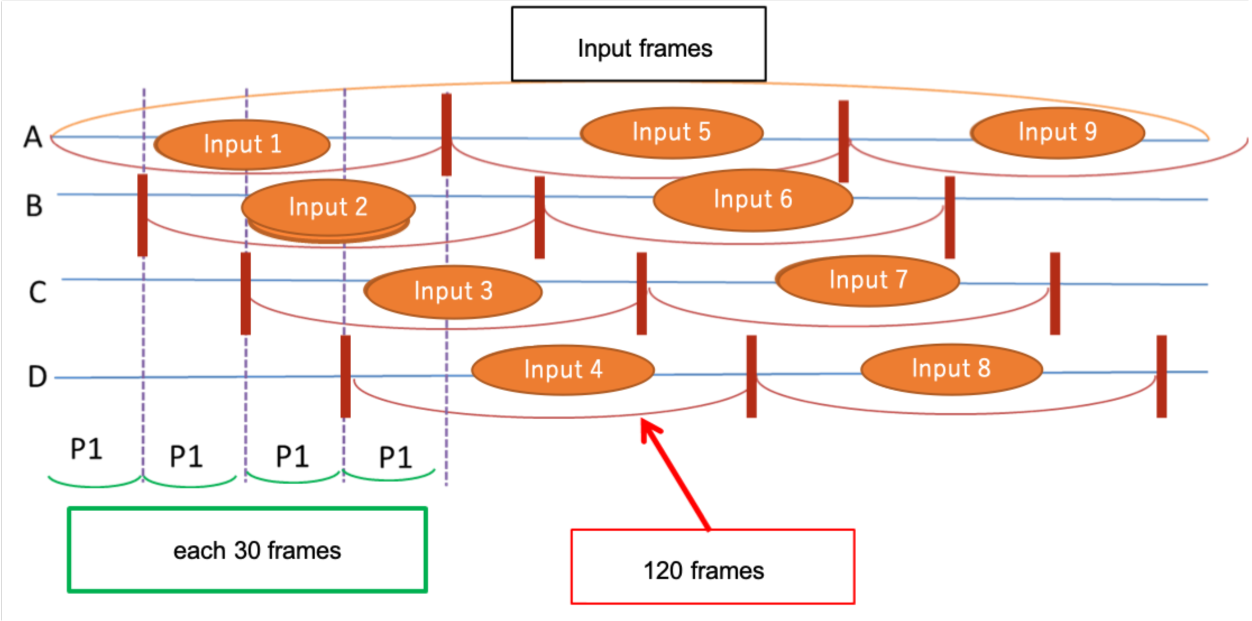


Fig. 3.3: Changing frame interval

In Fig. 3.3, a frame interval of 120 frames (approximately 4 s) is used as input, and frames are shifted by 30 frames. This allows us to determine whether there is a change in the pattern or not.

In this study, we used the change frame interval in Method 2 and experimented with the conventional and proposed methods.

3.3 Features

To determine the area of a hand in an image, there are methods to obtain the optical flow and area using skin color information from motion and shape features.

Fig. 3.4 shows a system diagram of features.

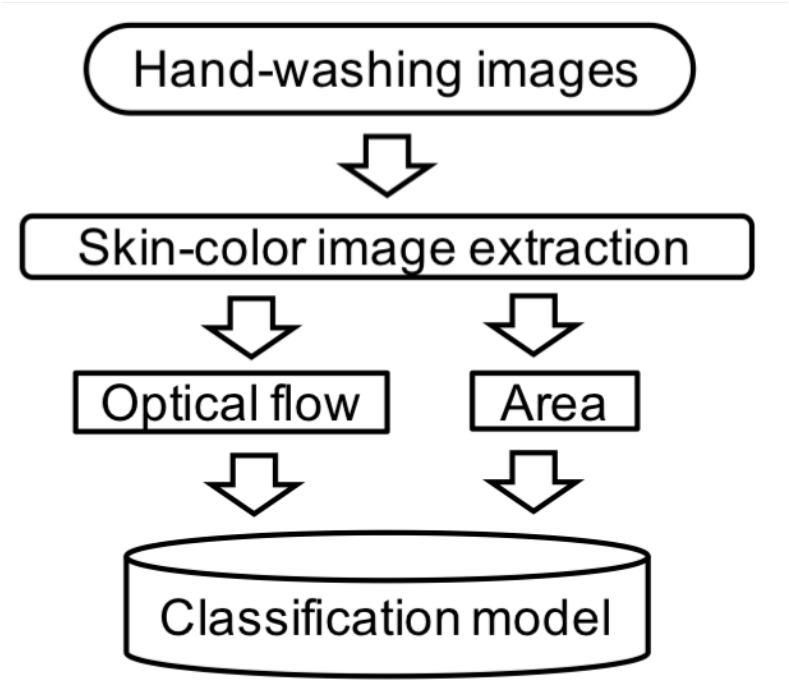


Fig. 3.4: System diagram of features

A video is converted into an image (frame), and the HLS method is used to remove fine noise by contraction and expansion, leaving only skin tone features, i.e., the hand parts. Optical flow and area features are obtained from the remaining skin tone area.

We explain how to obtain the optical flow and area in 3.3.1 and 3.3.2. We also discuss the number of dimensions.

3.3.1 Optical flow division method

The optical flow shows pixel movement vectors from the previous to the current frame. The pyramid LK method, which iteratively implements the original LK method, was used as the optical flow detection method. This method is extremely robust to noise and is computationally inexpensive, enabling stable and swift detection. Therefore, it is suitable for a real-time examination system.

Feature extraction was performed by calculating vector angle T acquired in the optical flow (Table 3.1) and categorizing angles into the following eight ranges.

Table 3.1: ranges of vector angle for categorize

1	0° –45°
2	45° –90°
3	90° –135°
4	135° –180°
5	180° –225°
6	225° –270°
7	270° –315°
8	315° –360°

The frequency of occurrence and size of vectors in the categorized ranges were obtained every few frames. The vectors in each range are defined as the frequency of occurrence, and the mean size and variance as feature values.

Therefore, the total number of dimensions is 24: the frequency of vectors: 8 dimensions and the size of vectors (mean and variance values): 8× =16 dimensions.

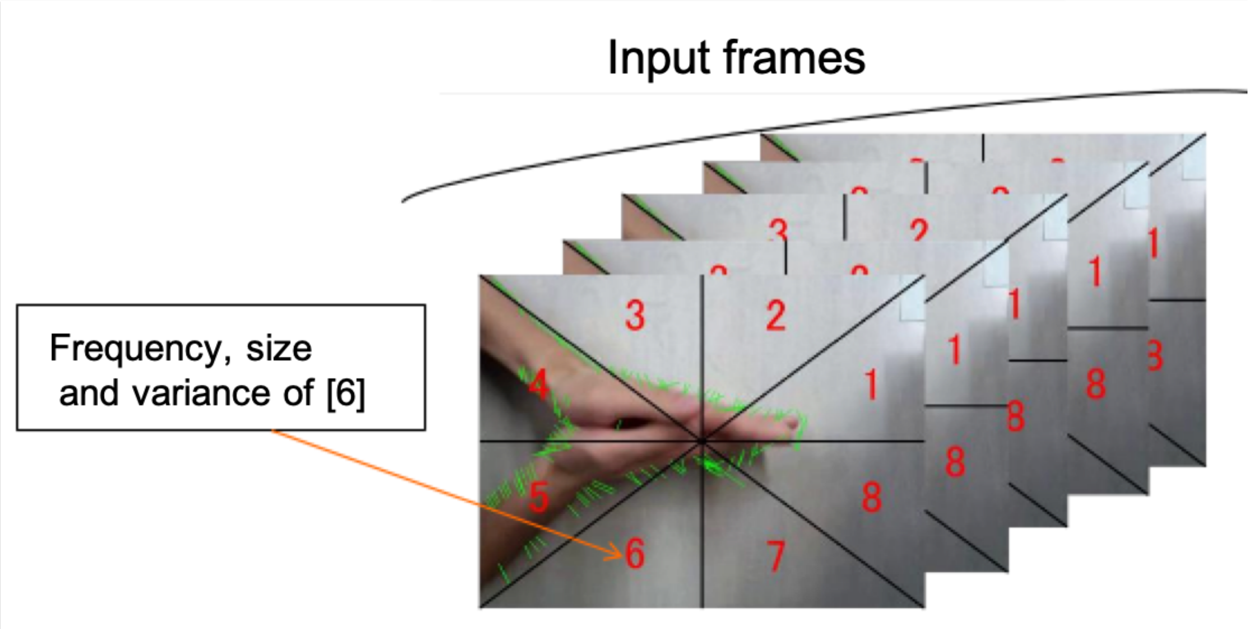


Fig. 3.5: Optical flow extraction (classified into 8 categories)

3.3.2. Skin-Color Area Extraction

By using HSL (hue, saturation, luminance) color spaces, the skin-color region is cut out as the hand region. By performing contraction and expansion on this skin-color region, noise is removed. As shown in FIg. 3.6, feature extraction is performed by identifying the center of gravity of the hand and dividing the skin-color area into the following four ranges(Table. 3.2).

Table 3.2: dividing the skin-color area

1	0° –90°
2	90° –180°
3	180° –270°
4	270° –360°

The skin-color areas in the divided ranges are captured every few frames, and the average area values for each range are used as the number of features.

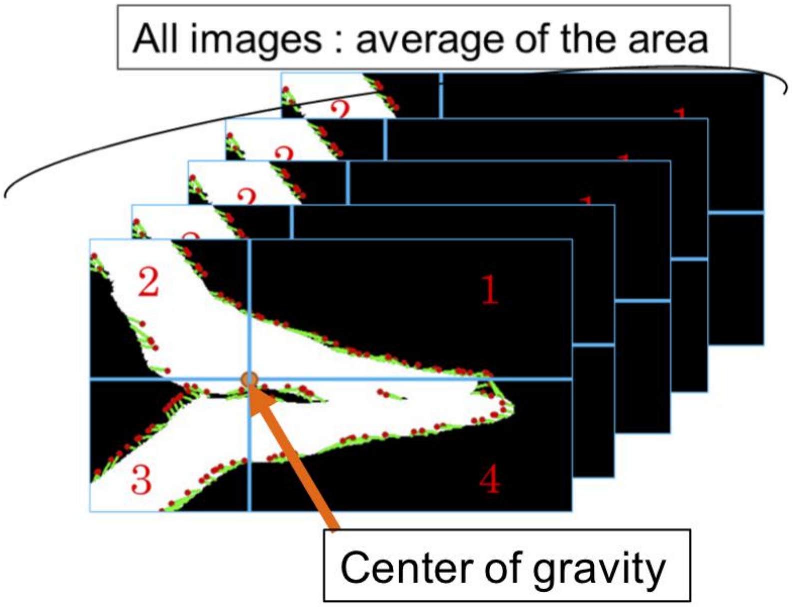


Fig. 3.6: Area acquisition (four divisions based on the center)

As shown in Fig. 3.6, we take the average of the area for each of the classified directions. Therefore, the number of dimensions is the average of the area: four dimensions.

3.4 Discrimination Process

Discrimination is the process of distinguishing between the feature classification model from the training video and input data from the input video. For identification, a classifier is required. The main types of classifiers include the k-Nearest Neighbor algorithm (KNN) [20] and SVM [21].

3.4.1. k-nearest neighbor method (KNN)

In the KNN method, a target object is assigned the largest number of k-nearest objects to it.

For example, the red dot in Fig. 3.7 is assigned the same value as the green dot when k = 3. Similarly, when k = 7, the same value as blue is assigned.

The case where k=1 is specifically called the nearest neighbor method.

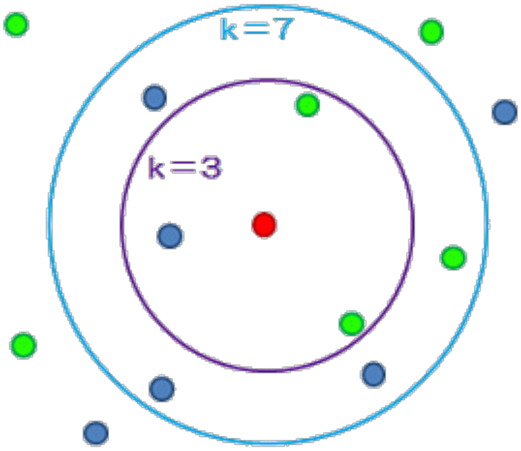


Fig. 3.7: Example of a k-neighborhood

The algorithm using KNN is as follows;

1 ：Determine the value of parameter k. The value of k is 10-NN with k = 10.
2 ：Calculate the similarity between input data and the classification model.

The Euclidean distance is used to calculate the similarity.

To find the Euclidean distance d between any point Q (x, y) and Q' (x', y') on the plane R^2, use

$$d = \sqrt{(x' - x)^2 + (y' - y)^2} \qquad (3.1)$$

3: Sort the data based on the similarity of the calculated data.
4: Select the data similar to the input data based on the similarity for the determined number of k, and vote for the majority of the patterns of the selected data.
5: The pattern of the data with the highest number becomes the inferred pattern of the input data.

Next, the top two results of the majority vote are displayed after 10-NN. However, the results with only one or two votes are excluded. Then, as shown in Fig.3.8, we can judge whether each pattern is correctly washed from the connection between the top 1 and 2 results.

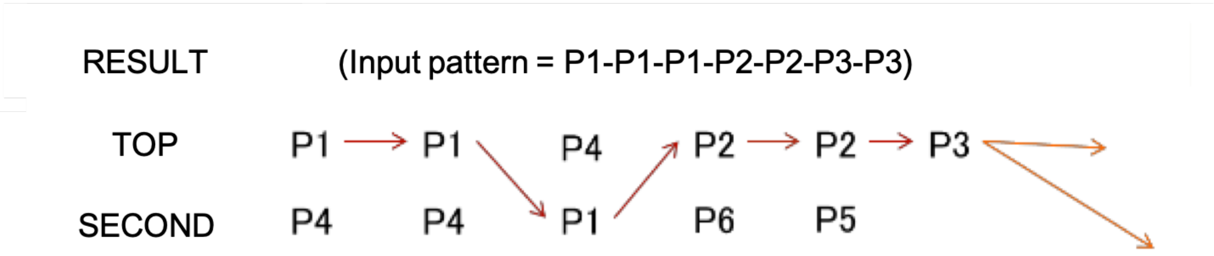


Fig. 3.8: Example of result display

3.4.2. Support Vector Machine (SVM)

SVM is a supervised machine learning method for pattern identification, which has the advantage of not having the problem of local solution convergence. The generalizability of SVM means that it can classify data well, even unknown data that have not been used in training.

The advantage is that the optimal solution is uniquely determined, and there is no need to worry about falling into a local optimal solution. Another advantage is that it does not require a large amount of training data, which makes it easy to control feature points.

3.4.2.1. Overview of SVM

SVM [17] is a pattern recognition model that uses supervised learning proposed by V. Vapnik around 1995. It can be used for classification and regression problems and can achieve high discriminative performance on untrained data using the concept of "margin maximization."

3.4.2.2. Margin Maximization

SVM is a method for classifying classes using boundaries. The boundary line for classifying classes is called the separating hyperplane, and it is obtained on the basis of the concept of margin maximization.

As an example, consider a case, where data are classified into two classes. Each data class is assumed to be a two-dimensional vector with two features, x1 and x2. In such a case, SVM classifies the classes by finding a separating plane, as shown in Fig. 3.9. Here, the vector closest to the separating hyperplane is called the support vector, and the distance between the support vector and the separating hyperplane is called the margin. The larger this margin is, the less likely it is that unknown data will be classified into the wrong class when classified. Therefore, in SVM, the separating hyperplane is determined so that the margin is maximized. This way of thinking is called margin maximization.

As an example, Fig. 3.9 shows two types of data classification.

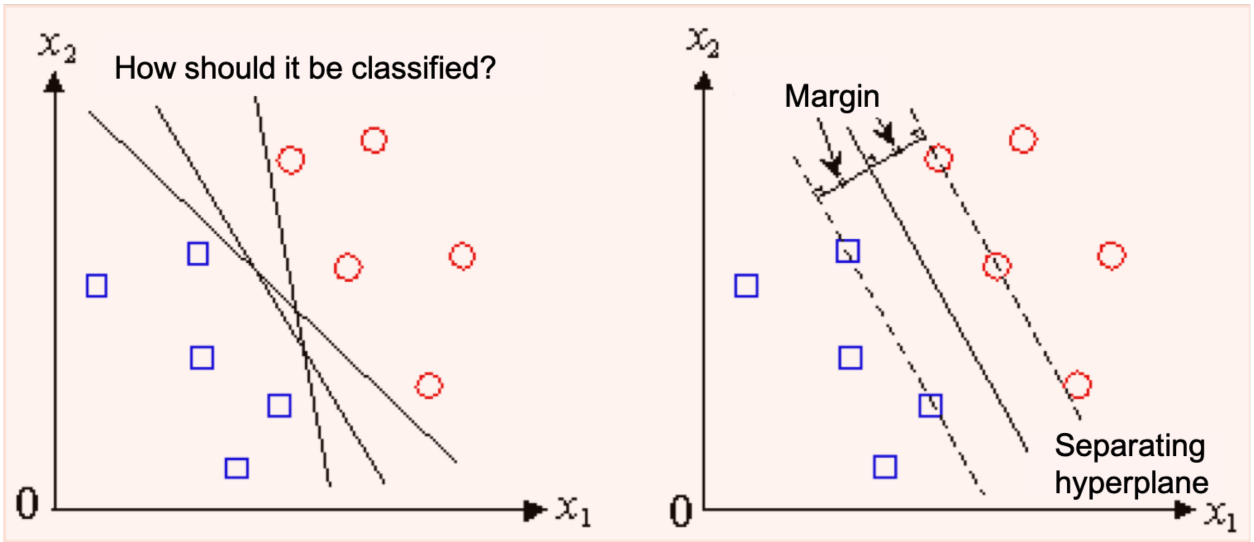


Fig. 3.9: Margin maximization for 2D data classification

We use the fact that a hyperplane with the highest generalization capability is the one that maximizes the distance (called margin) between the separating hyperplane (in this case, a straight line) and the two types of data (two classes). The formulation of the method for maximizing the margin can be attributed to a quadratic programming problem. Therefore, the optimal solution is uniquely determined, and there is no risk of falling into a local optimum.

3.4.2.3. Optimization Problem in Margin Maximization

As described in the previous section, SVM determines the separating hyperplane based on the concept of margin maximization. In this section, we explain how to find the separating hyperplane that maximizes the margin.

As an example, consider the case of classifying data, as shown in Fig. 3.9. Let us assume that n data points are given, and denote x_i(i=1, 2, ...,n). We consider classifying these data points into two classes, K_1 and K_2.

In this case, the separating plane is defined as follows:

$$w^T x + b = 0 \qquad (3.2)$$

In SVM, the n-dimensional real vector w and the scalar variable b, called the bias, are determined so that the margin is maximized. By defining the separating plane, as in equation (3.2), the following condition holds for each class of data.

$$\begin{aligned} &w^T x + b > 0 \quad x_i \in K_1 \\ &w^T x + b < 0 \quad x_i \in K_2 \end{aligned} \qquad (3.3)$$

By introducing the label variable t, the above equation can be expressed as follows:

$$\begin{aligned} &t_i(w^T x_i + b) > 0 \\ &t_i = \begin{cases} 1 & (x_i \in K_1) \\ -1 & (x_i \in K_2) \end{cases} \end{aligned} \qquad (3.4)$$

Next, the support vectors for each class are denoted as x_+ and x_-, respectively. Then, from the formula for the distance between a point and a plane, the margin M can be expressed as

$$\mathrm{M} = \frac{w^T x_+ + b}{\|w\|} = \frac{-(w^T x_+ + b)}{\|w\|} \qquad (3.5)$$

We also assume that the lines passing through the support vectors of each class are $w^T x + b = 1$ and $w^T x + b = -1$, respectively. In this case, on the line passing through the support vector, the following equation holds:

$$\mathrm{M} = \frac{t_i(w^T x_i + b)}{\|w\|} = \frac{1}{\|w\|} \qquad (3.6)$$

Given the above factors, the relationship between the distance between the separating hyperplane and support vector can be shown in Fig. 3.10, and the margin maximization can be replaced by the following optimization problem

$$\max_{w,b}\frac{1}{\|w\|},\quad t_i(w^T x_i + b)\ \geq\ 1\quad (i = 1,2,\dots,n)\qquad (3.7)$$

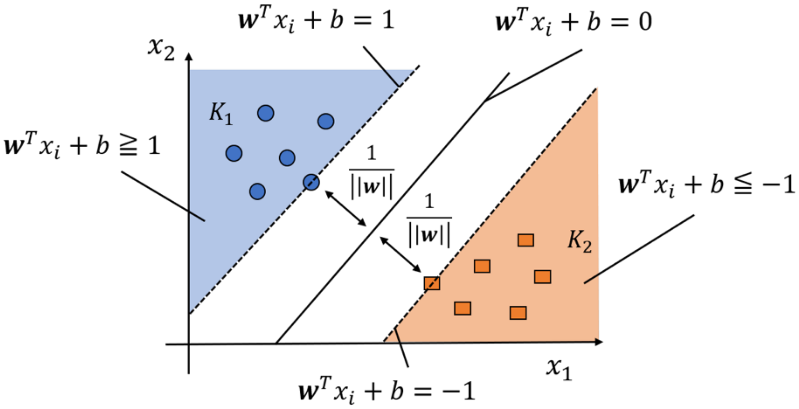

$\boldsymbol{w}^T x_i + b = 1$
$\boldsymbol{w}^T x_i + b = 0$
x_2
K_1
$\boldsymbol{w}^T x_i + b \geqq 1$
$\frac{1}{||\boldsymbol{w}||}$
$\frac{1}{||\boldsymbol{w}||}$
$\boldsymbol{w}^T x_i + b \leqq -1$
K_2
x_1
$\boldsymbol{w}^T x_i + b = -1$

Fig 3.10: Relationship between the distance between separating hyperplane and support vector

For computational convenience, we transform it into the following minimization problem.

$$\min_{w,b}\frac{1}{2}\parallel w\parallel^2,\quad t_i(w^T x_i + b)\ \geq\ 1\quad (i = 1,2,\dots,n)\qquad (3.8)$$

By solving this minimization problem, we can obtain the vector w and bias b that maximize the margin.

However, the constraint in equation (3.8) is only valid when the classes are linearly separable, that is, when the classes can be completely classified by a linear separating hyperplane. This is because this constraint implies that the distance to the separating plane must be greater than or equal to 1 for all data. In other words, it does not allow data to exist inside the margin. In practice; however, it is rarely the case that data classification is linearly separable.

Therefore, by introducing the slug variable ξ, we weaken the constraint in equation (3.8) as follows:

$$t_i(w^T x_i + b) \geq 1 - \xi_i \quad \xi_i = \max\{0, \quad M - \frac{t_i(w^T x_i + b)}{\|w\|^2}\} \qquad (3.9)$$

Equation (3.9) allows the distance between data and separating plane to be less than 1, making it possible to estimate the separating plane even when linear separation is impossible. In addition, by changing the constraint conditions, as in equation (3.9), the minimization problem in equation (3.8) can be replaced by the following

$$\min_{w,b,\xi} \left\{ \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \xi_i \right\}$$

$$t_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (i = 1,2,\dots,n) \qquad (3.10)$$

where C represents the regularization factor, which adjusts the restraining power of the constraint by ξ. The separating plane that maximizes the margins can be obtained by solving the minimization problem in equation (3.9). To solve this problem, we transform it into a dual problem using Lagrange's undecided multiplier method. The reason for transforming the problem into a duality problem is to simplify the computation by reducing the number of variables to be handled. The process of transforming the problem into a dual problem using Lagrange's undecided multiplier method is omitted in this study.

As a result of using Lagrange's undecided multiplier method, the minimization problem in equation (3.10) can be expressed as a dual problem with only α, as shown below

$$\max\left\{\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j t_i t_j x_i^T x_j\right\} \quad (3.11)$$

In SVM, we finally find the separating plane that maximizes the margin by solving equation (3.11).

3.4.3. Recognition Using Support Vector Machine (SVM)

The basic structure of SVM is a linear threshold element, as shown above, but this cannot be applied to data that are not linearly separable, and the range of application of SVM is extremely limited. One way to enable nonlinear classification by SVM is to increase dimensionality. In this method, the original input data are mapped to a high-dimensional feature space using nonlinear mapping ∅, and linear separation is performed in the feature space. As a result, as shown in Fig. 3.11, the classification is nonlinear in the original input space.

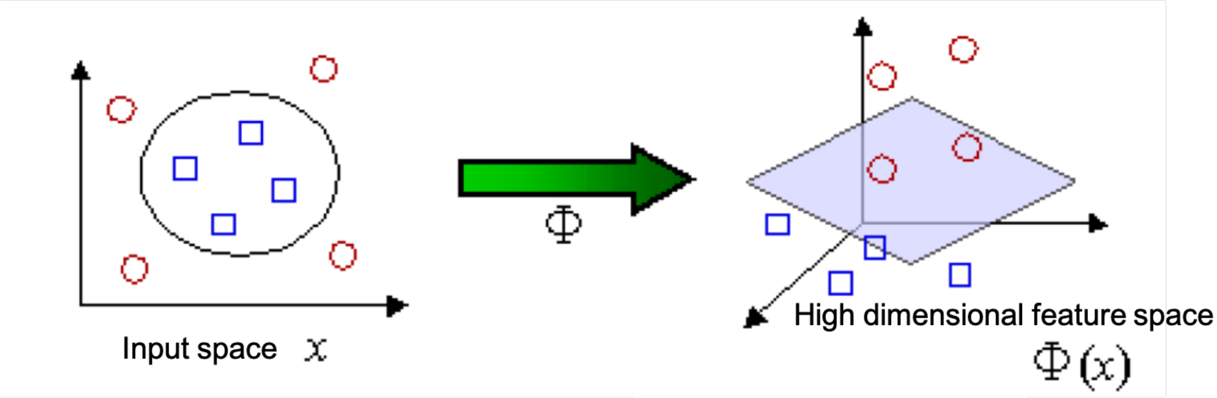


Fig. 3.11: Increasing dimensionality by nonlinear mapping

However, when we implement this, we do not compute ∅ but replace it with the computation of the kernel function. This is called the kernel trick. Using the kernel trick, SVM avoids computing ∅ directly and overcomes the computational difficulty. In addition, nonlinear separability enables multiclass classification in high-dimensional space.

Fig. 3.12: below shows an example of a result of nonlinearly classifying 2D data.

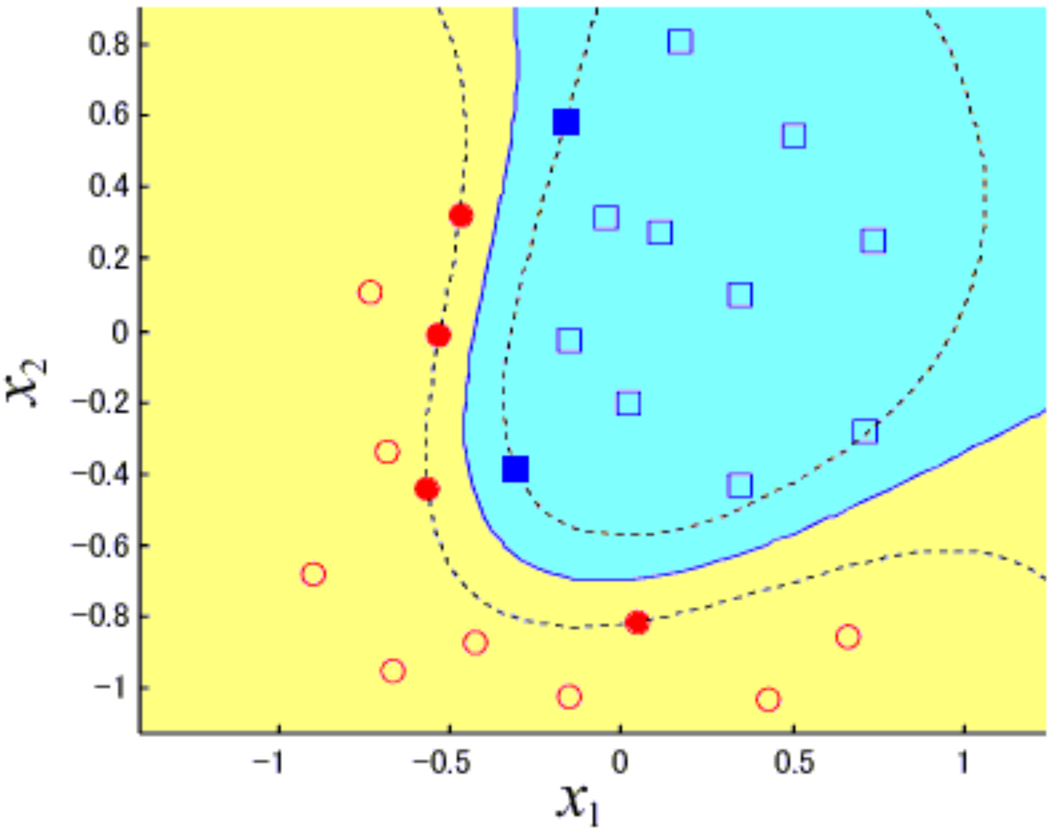


Fig. 3.12: Higher dimensionality by nonlinear mapping

The solid line in the figure represents the separating hyperplane (in this case, a curve), and the dashed line represents the margin. The points on the margin are called support vectors (filled points in the figure). Support vectors are solely used to determine the separation hyperplane, and the other points do not contribute to the construction of the separation hyperplane.

In this experiment, we employed the SVM of (2), which has high generalizability in high-dimensional multiclass, and evaluated the conventional and proposed methods.

3.5 Preliminary experiments

We experimented to see whether the system recognizes the correct pattern using the conventional method.

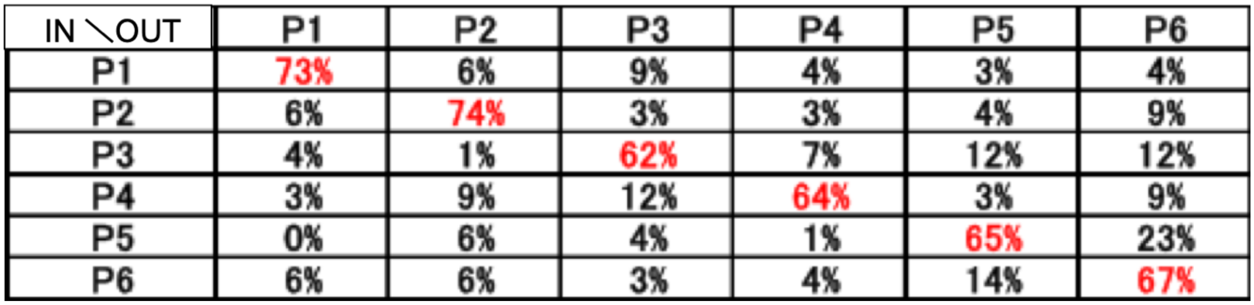
As described in Chapter 2, six patterns of each action for proper handwashing were P1: rubbing the palm; P2: rubbing the back of the hand; P3: rubbing the fingertips and nails; P4: rubbing between the fingers; P5: twisting the opposite hand around the thumb; P6: turning the wrist with the opposite palm.

For learning videos, we prepared videos (approximately 5 s for each pattern) of the six correct handwashing patterns described in Chapter 3 and used them as the teacher data. A classification model is developed by taking feature values from the videos.

We used 66 input videos, all of which show the correct handwashing patterns. Each video was converted into multiple input frames, and features were extracted to determine whether the pattern was correct.

The results are shown in Table 3.3.

Table 3.3: Recognition rate by the conventional method



IN ＼OUT	P1	P2	P3	P4	P5	P6
P1	73%	6%	9%	4%	3%	4%
P2	6%	74%	3%	3%	4%	9%
P3	4%	1%	62%	7%	12%	12%
P4	3%	9%	12%	64%	3%	9%
P5	0%	6%	4%	1%	65%	23%
P6	6%	6%	3%	4%	14%	67%

On average, 33% of the videos could not be recognized as correct hand movement patterns by the conventional method.

Fig. 3.13 shows an optical flow image generated from video frames. It was obtained from successive frames in P1,i.e., palm rubbing motion, in the order of time series t = 1 to 9.

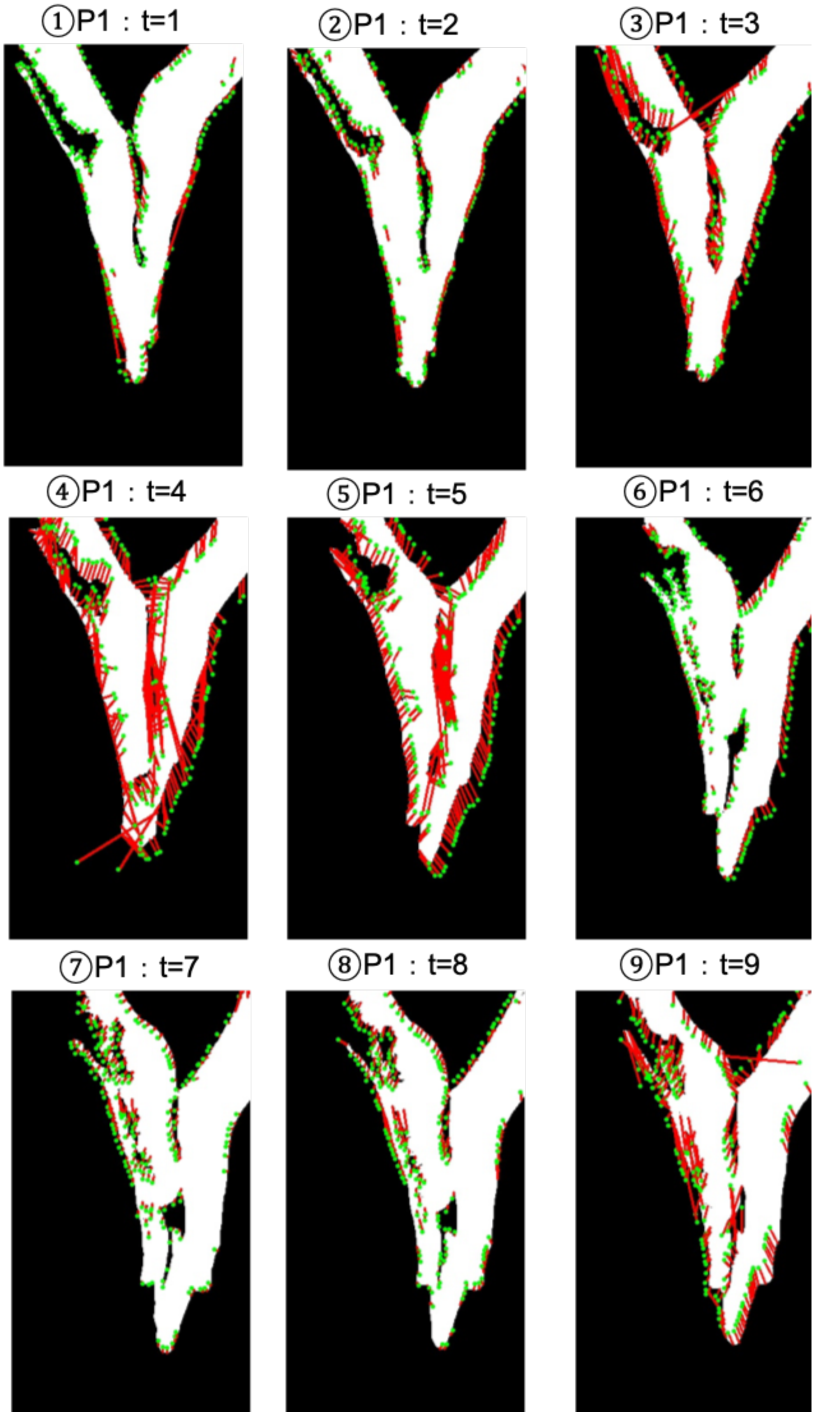


Fig. 3.13: Optical flow diagram obtained from the video of P1

3.6 Problems with conventional methods

There are two problems with the current feature extraction method: First, due to changes in lighting, areas that are not skin tone are judged as skin tone and noise is introduced, making it difficult to obtain appropriate feature values.

The second is that some people shake their hands a lot when they wash their hands. Such movements cause optical flow vectors to move in various directions, making it difficult to obtain appropriate optical flows.

3.6.1. Increase in noise due to skin tone detection (Problem 1)

When extracting skin tone images using HLS, areas that are not hand areas are sometimes judged as skin tone and become noise. If the noise is small, it can be removed by contraction or expansion, but if the noise is large, it cannot be completely removed. The main reason for this is that the noise is judged to be similar to skin tone due to changes in lighting when the camera captures the image, introducing a large amount of noise, as shown in Fig. 3.14 below.

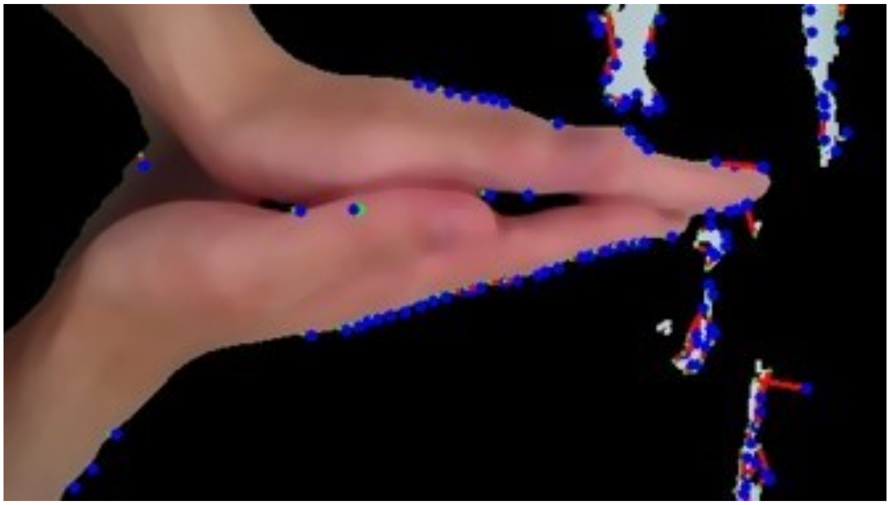


Fig. 3.14: Example of skin tone detection failure

As shown in Fig. 3.14, the background of the image is recognized as skin tone, and the noise remains. As a result, the optical flow and area features are also taken from the noise.

To eliminate this noise, we propose a noise reduction method based on labeling. A detailed explanation is provided in the proposed method in Chapter 4.

3.6.2. False detection of optical flow due to hand shake (Problem 2)

When washing hands, some people move their hands in a way that causes a large amount of hand shake. This movement causes vector angles to point in various directions when taking the optical flow, making it difficult to obtain an appropriate optical flow. For example, in the case of Pattern 1, most people move in a constant direction, as shown in Fig. 3.14, but in rare cases, some people move with additional movement in other directions, as shown in Fig. 3.15.

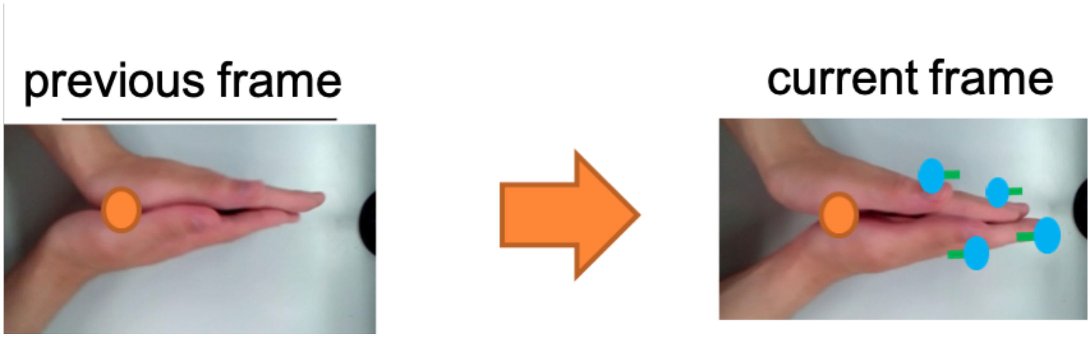


Fig. 3.14: Example of successful optical flow acquisition

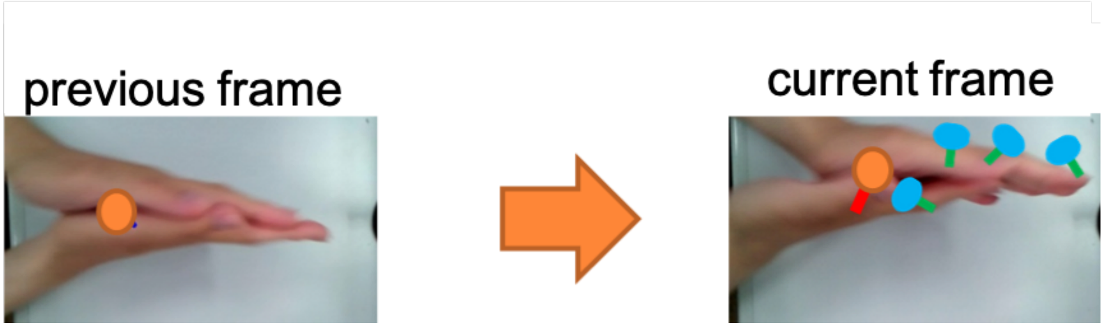


Fig. 3.15: Failure of optical flow acquisition

The movement shown in Fig. 3.15 is extremely different from the optical flow vector of the correct hand-washing, which is the original purpose of the system and is thought to affect the frequency and magnitude of features.

As an improvement of this false detection, we propose an optical flow correction method that focuses on the motion of the center of gravity of the skin tone area. A detailed explanation is provided in the proposed method in Chapter 4.

4. Proposed Method

In the existing method, recognition accuracy is degraded by noise caused by shaking of the hand during hand-washing and changes in illumination.
In this study, we removed the cause of the noise and corrected images by considering the center of gravity of the hand when capturing the characteristics of the motion, thereby tackling the degradation of recognition accuracy caused by hands haking.

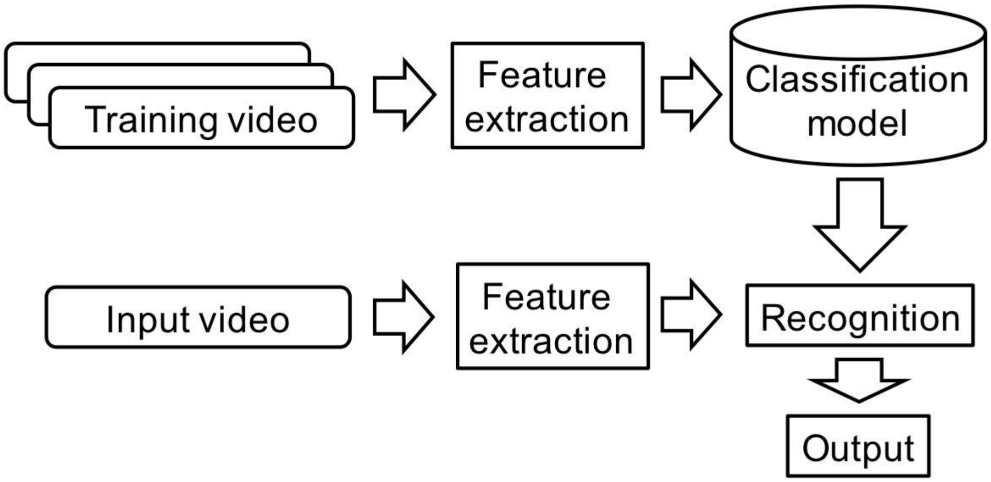


Fig. 3.1: Outline of hand-washing support system using image processing techniques. (reprinted))

The system overview is reproduced in Fig. 3.1. The system extracts features from the training video and creates a classification model without changing the conventional method. The input video is converted into input frames, and features are extracted. The system uses a discriminator to identify the input frames and the classification model and outputs the results.

In the current method, we have improved the feature set.

4.1. Improvements to the Existing Method

The proposed method aims to solve the two problems limiting the existing method. For the first problem, we attempted to solve it by denoising with labeling; for the second problem, we attempted to solve it by correcting the engineering image using the center-of-gravity vector. The second problem was solved by correcting the engineering image using the center-of-gravity vector because it was found that the center of gravity of the hand also shifts when the hand shakes significantly. Fig. 4.1 shows the improvement of feature extraction in the overall system.

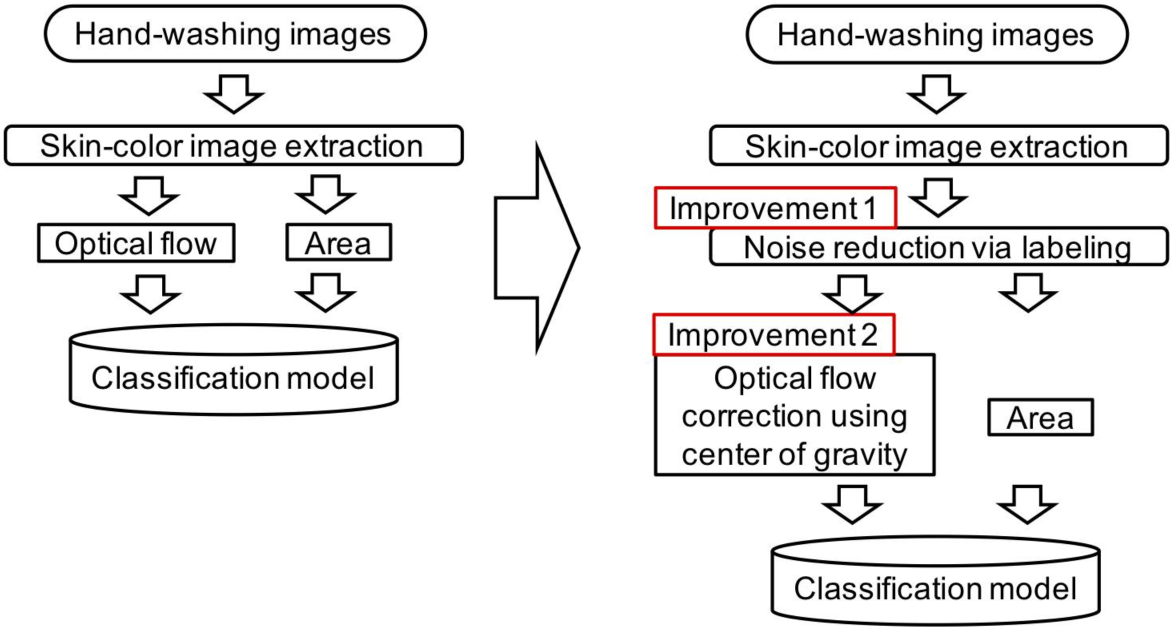

Hand-washing images
Skin-color image extraction
Optical flow
Area
Classification model
Hand-washing images
Skin-color image extraction
Improvement 1
Noise reduction via labeling
Improvement 2
Optical flow correction using center of gravity
Area
Classification model

Fig. 4.1: Improvements in feature extraction

4.2. Hand Shape Recognition Based on Labeling Removal Region

As a solution to the problem of increasing noise caused by skin tone detection, where non-skin tone areas in an image are detected as skin tone, we explain below a method for removing noise by labeling.

4.2.1 Labeling

Labeling is the process of assigning the same number to consecutive white (or black) pixels in a binarized image, as shown in Fig. 4.2. Typically, the area (number of pixels), width, height, and other features of each number are obtained and used for defect inspection and classification.

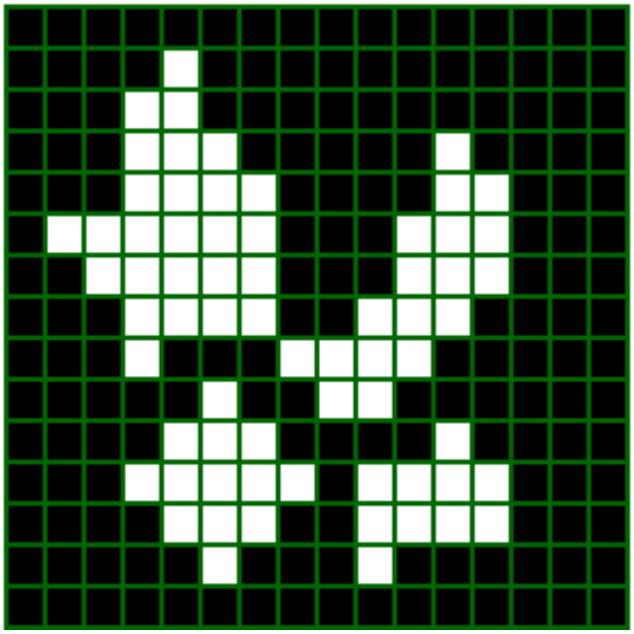
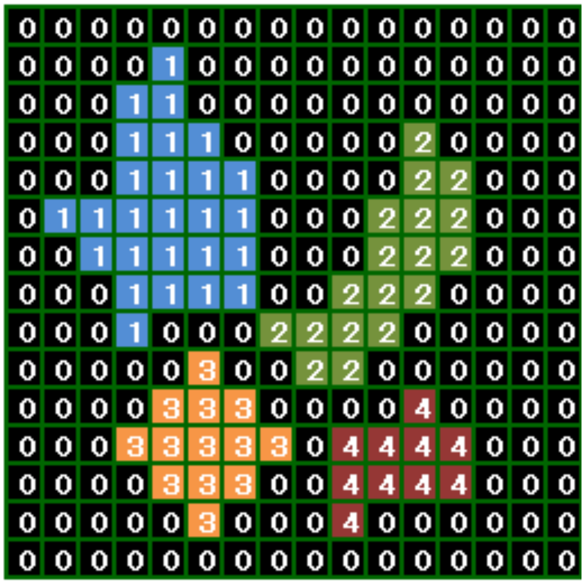


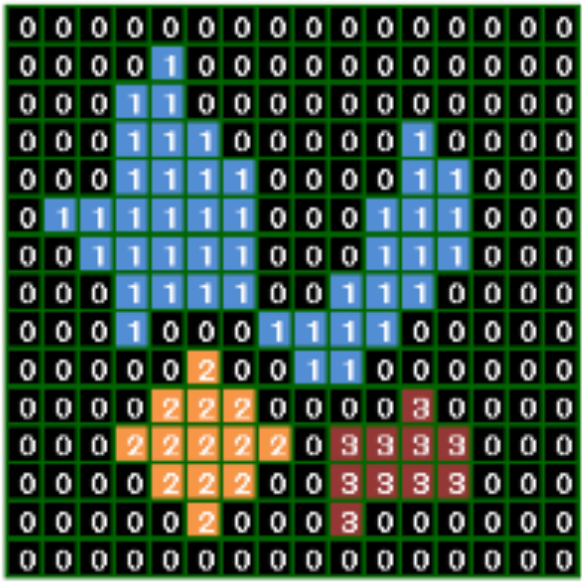
Fig. 4.2: Example of valorized image

There are two types of labeling processes: 4-connect (four nearest neighbors), where the consecutive portions in the vertical and horizontal directions of the binarized image are given the same label, and 8-connect (eight nearest neighbors), where consecutive portions in the vertical, horizontal, and diagonal directions are given the same label. Fig. 4.3 and Fig. 4.4 below show examples of 4- and 8-connections.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	2	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	2	2	0	0	0
0	1	1	1	1	1	1	0	0	0	2	2	2	0	0	0
0	0	1	1	1	1	1	0	0	0	2	2	2	0	0	0
0	0	0	1	1	1	1	0	0	2	2	2	0	0	0	0
0	0	0	1	0	0	0	2	2	2	2	0	0	0	0	0
0	0	0	0	0	3	0	0	2	2	0	0	0	0	0	0
0	0	0	0	3	3	3	0	0	0	0	4	0	0	0	0
0	0	0	3	3	3	3	3	0	4	4	4	4	0	0	0
0	0	0	0	3	3	3	0	0	4	4	4	4	0	0	0
0	0	0	0	0	3	0	0	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4.3: Example of 4-coupling



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0
0	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	1	1	1	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	2	0	0	1	1	0	0	0	0	0	0
0	0	0	0	2	2	2	0	0	0	0	3	0	0	0	0
0	0	0	2	2	2	2	2	0	3	3	3	3	0	0	0
0	0	0	0	2	2	2	0	0	3	3	3	3	0	0	0
0	0	0	0	0	2	0	0	0	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 4.4: Example of 8-coupling

4.2.2Denoising using labeling

Using the HSL color space, we cut out the skin tone region as the hand region. In our proposed method, we used an 8-coupled labeling process for denoising.

When the labeling process is performed and the region with the largest area (number of pixels) is left, only the skin-colored area is left clean. As shown in Fig. 4.5, we attempted to remove noise by subtracting, enlarging, and labeling the cut-out skin tone area.

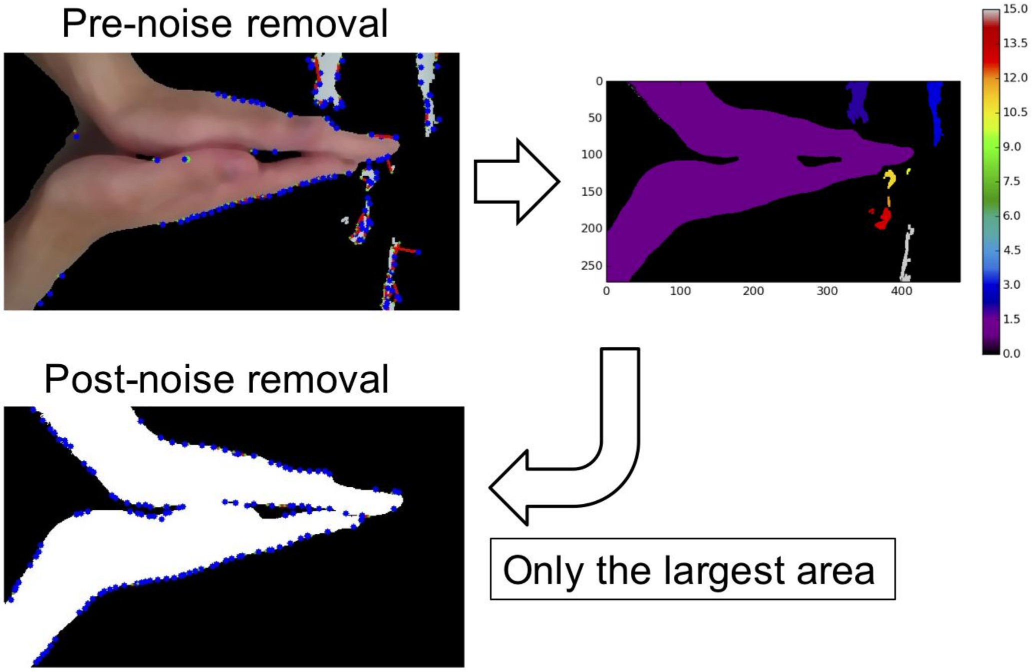
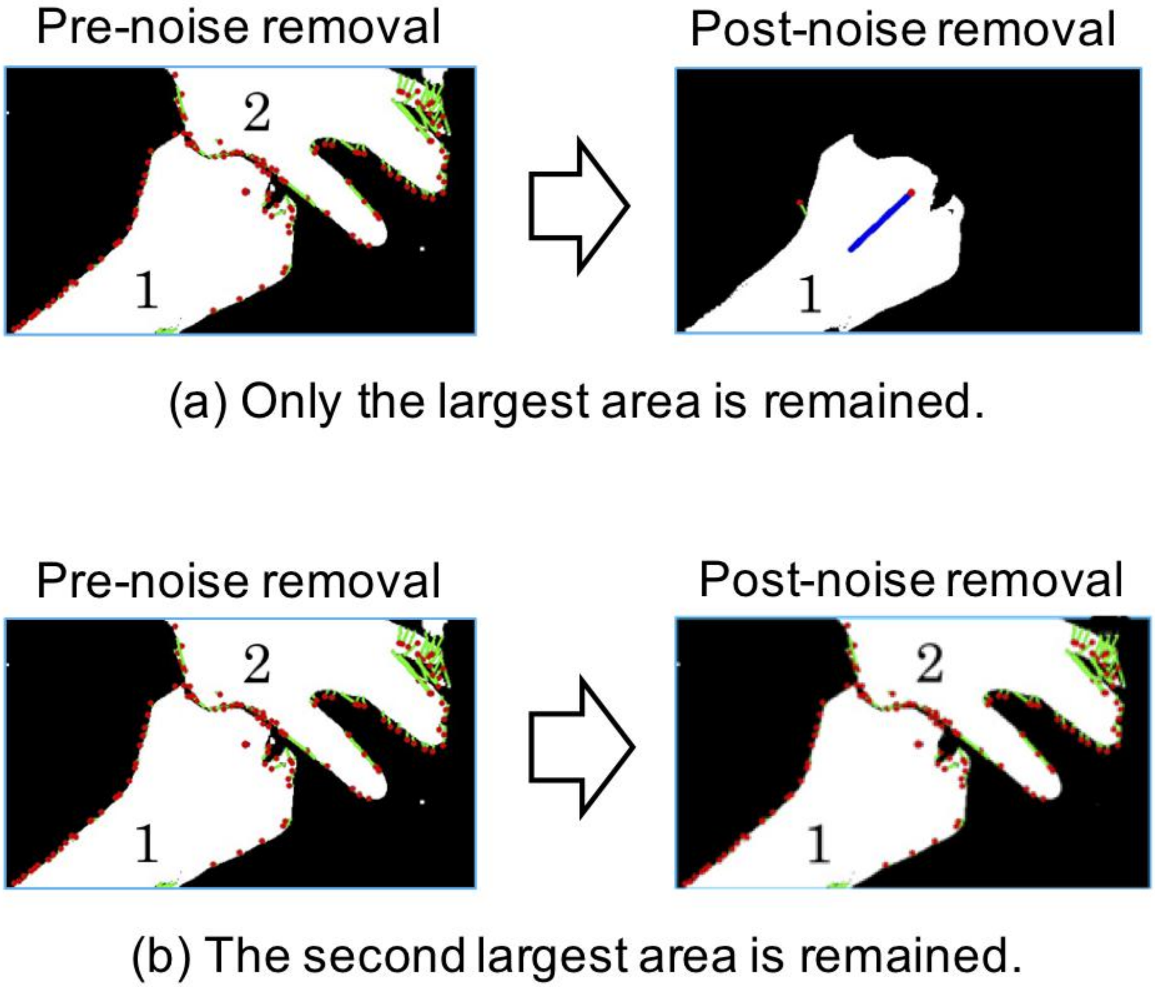


Fig.4.5: Labeling of skin tone area

Based on the labeling, we could remove the noise so that only the largest area in the image remains. However, by excluding the largest area from the labeling process, if the skin tone areas are not properly connected, the areas to be detected may be separated, as shown in Fig. 4.6(a).

Since this study aims to detect hygienic hand-washing using alcohol, etc., the maximum number of skin-colored areas is two. Therefore, the threshold is applied to the second largest area.

Therefore, when we removed noise by labeling, we retained the second largest area by thresholding in addition to the largest area, allowing us to capture both hand regions, even if we separated hand 1 and hand 2. The threshold of this area was determined by repeating the experiment. This is shown in Fig. 4.6(b).



(a) Only the largest area is remained.

(b) The second largest area is remained.

Fig. 4.6: Image with noise due to skin-color identification.

Then, feature extraction is performed, as in the conventional method. A quadrant was created, and the average of the areas was used as the feature value.

4.3. Hand Shape Recognition with Center-Based Optical Flow Correction

As a solution to the problem of not being able to obtain correct optical flow due to the vector angles of the optical flow being oriented in various directions due to handshake, the correction of the optical flow focusing on the motion of the center of gravity of the hand is described below.

In the previous section, we extracted the optical flow from the frames denoised by labeling. The motion of the center-of-gravity point (center-of-gravity vector) is obtained between each frame of the obtained optical flow images.

4.3.1 Center-of-gravity vector

Obtain the distance D between the center-of-gravity point (x, y) of the previous frame and the center-of-gravity point of the current frame (x', y').

$$\mathrm{D} = \sqrt{(x' - x)^2 + (y' - y)^2}$$

The magnitude (distance) and angle of the center-of-gravity vector are used to correct the optical flow. The center of gravity vector at each point can be calculated from the motion of the center of gravity of the current and a previous frame.

The vector difference is represented by c = a - b, where a represents the vector from the origin to point (1, 2) and b represents the vector from the origin to point (3, 1), as shown in Fig. 4.3.1 below.

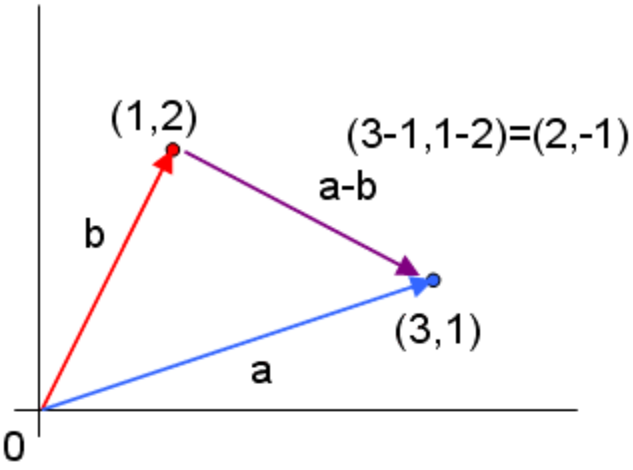


Fig. 4.7: Vector difference

4.3.2 Obtaining the correct optical flow using the center-of-gravity vector

From the area of the hand region, the center of gravity of the cluster region was obtained. By taking the vector difference between the obtained center-of-gravity vector and optical flow, we can obtain the corrected optical flow. An example is shown in Fig. 4.8.

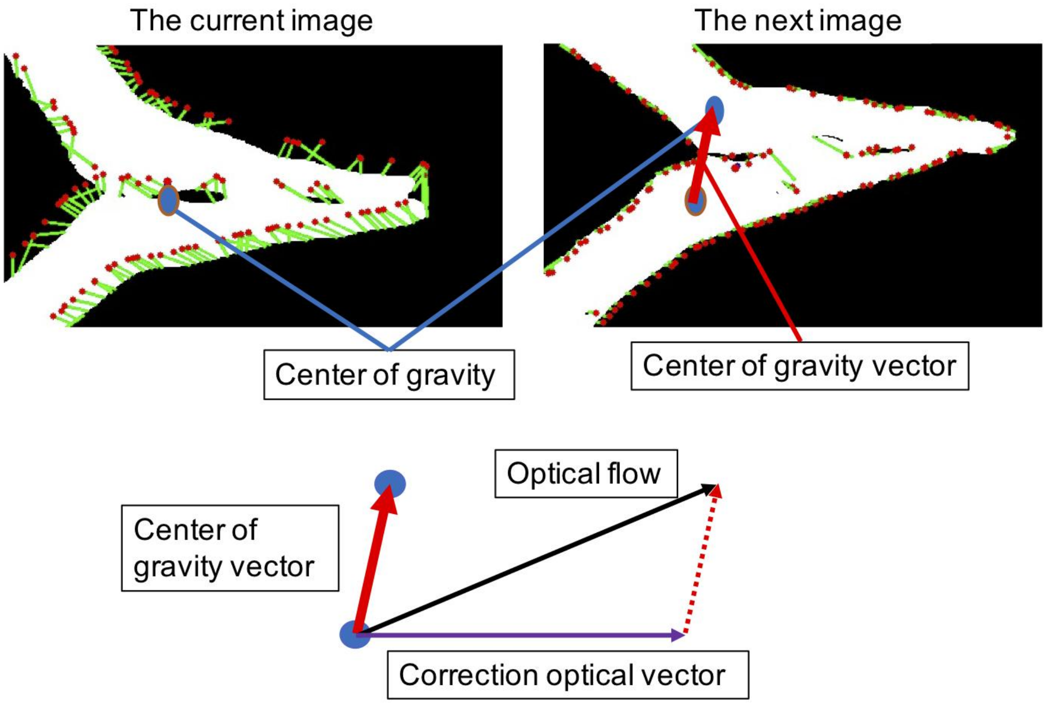


Fig. 4.8: Center-based optical flow correction

If we replace the formula for calculating the vector difference with the center-of-gravity vector taken from the motion of the center-of-gravity point and the optical flow, the vectors are far apart, as shown in Fig. 4.9. The vectors are too far apart, as shown in Fig. 4.9. Therefore, we shifted the center-of-gravity vector parallel to the optical flow to obtain the vector difference.

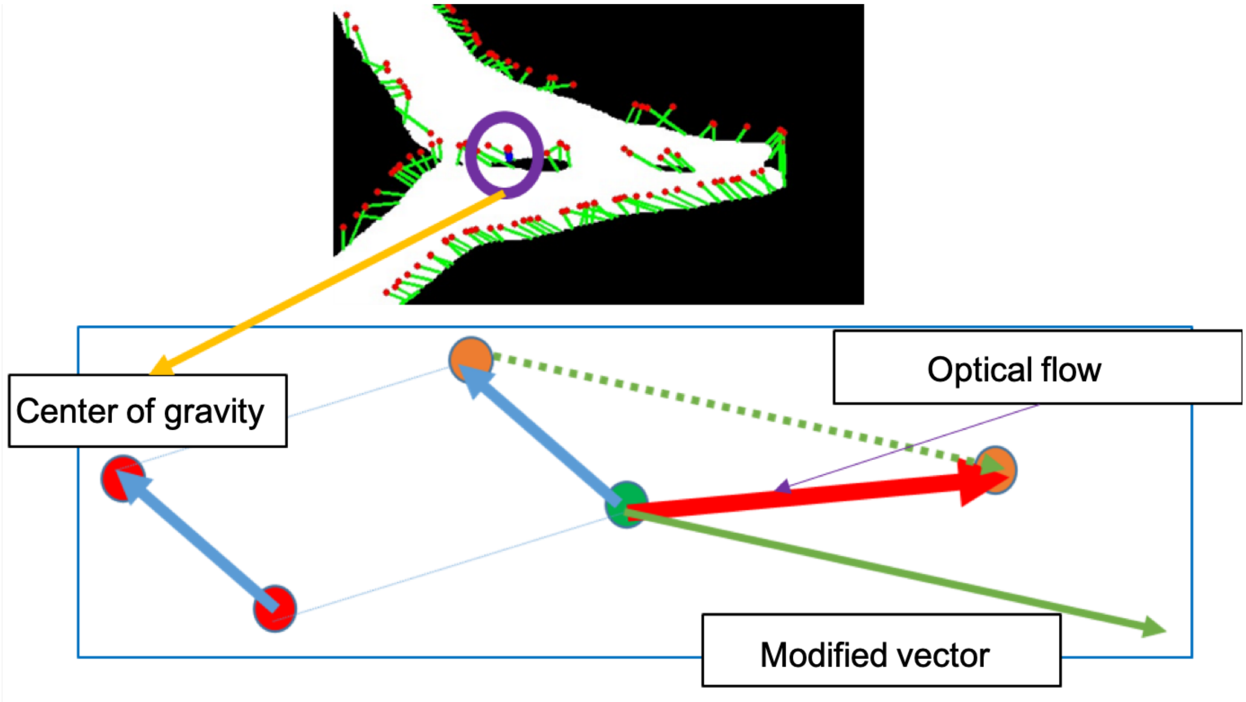


Fig. 4.9: Correction of the optical flow by the center of gravity

Fig. 4.10 shows the result of the correction using the vector difference between the center-of-gravity vector and the optical flow. The optical flow, which had been uneven, has settled down.

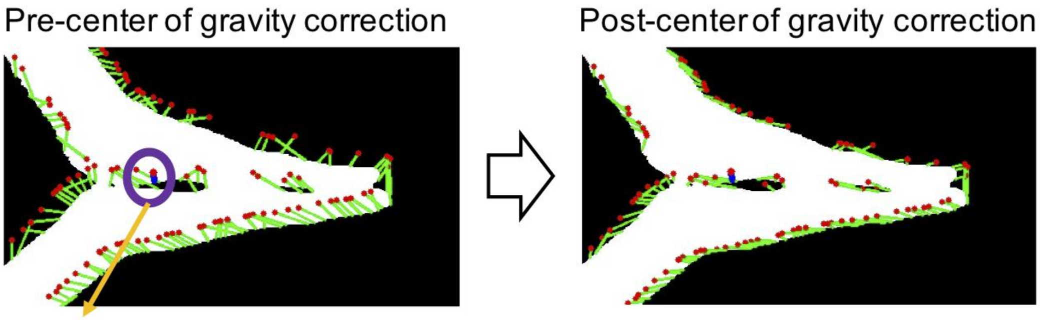

Pre-center of gravity correction
Post-center of gravity correction

Fig. 4.10: Change before and after correction.

Using this corrected optical flow, we extract features using the optical flow segmentation method described in the previous section. The image was divided into eight areas, and the frequency of occurrence, average size, and variance of the vectors were used as feature values. For the area, the feature values were extracted using the area segmentation method described in the conventional method in Chapter 3 for the image denoised through the labeling process described above.

A classification model was developed on the basis of the optical flow and area extracted by each method, and the experiments described in Chapter 6 below were conducted.

5. Evaluation

We investigated the effectiveness of our method for feature extraction. In the experiment, only hand movements were recorded to eliminate external factors that interfere with the features as much as possible. External factors of hand-washing include soap bubbles, water from the faucet, and the environment around the washbasin. Therefore, for both input and training videos, we placed a camera at a certain height on a table and filmed multiple people washing their hands from directly above. As a premise of the experiment, proper hand-washing is defined as performing six patterns for at least 30 s (at least 5 s for each pattern).

For the learning videos, a pair of videos of six patterns of proper hand-washing (approximately 5 s each) was shot for each person 11 times each. The classification model was developed on the basis of a database of 66 sets of 396 videos.

5.1. Experimental Environment

5.1.1 Library structure

We developed the library in two languages, C++ and Python. We used the Opencv library to obtain optical flow features, and the differences between the Opencv libraries in C++ and Python are shown below.

5.1.2 Opencv library in C++ (optical flow)

void calcOpticalFlowPyrLK (const Mat & prevImg , const Mat & nextImg , const vector textless Point2f textgreater & prevPts , vector textless Point2f textgreater & nextPts, vector textless uchar textgreater & status, vector textless float textgreater & err, Size winSize=Size (15, 15), int maxLevel= 3 , TermCriteria criteria=TermCriteria (TermCriteria::COUNT+TermCriteria::EPS , 30 , 0.01), double derivLambda=0.5 , int flags=0)

The above library computes the optical flow for a sparse feature set by iterating the LK method using image pyramids, as described in Chapter 2.

Parameter setting (C++ optical flow)

- prevImg: The first input, 8-bit, single-channel, or 3-channel
- nextImg: The second input image of the same size and the same type as prevImg.
- prevPts: The vector of (feature) points for which the flow needs to be detected.
- nextPts: The vector of output (feature) points that will contain the newly computed positions of the feature points in the second input image.
- Status: The output status vector. If the flow of a feature point is detected, the corresponding element of this vector is set to 1; otherwise, it is set to 0.
- err: The output vector containing the difference between the surrounding area of the feature points before and after the move.
- winSize: Size of the search window at each pyramid level.
- maxLevel: Maximum number of levels in the image pyramid (0-based). If it is 0, the image pyramid is not used (level 1); if it is 1, the image pyramid with level 2 is used.
- Criteria: Criteria for stopping the iterative search algorithm (either the specified maximum number of iterations criteria. maxCount is reached or the search window travels less than criteria. epsilon).
- derivLambda: Relative weights of the spatial derivatives of the image that affect the estimation of the optical flow. If derivLambda=0, only the luminance values of the image are used; if derivLambda=1, only the derivative values of the image are used. Both luminance and derivative values are used between 0 and 1 (in the respective specified ratios).
- flags: Processing flags: OPTFLOW_USE_INITIAL_FLOW, the initial estimates stored in nextPts are used. If this flag is not set, nextPts←prevPts is first used.

5.1.3 The Opencv library in Python (optical flow)

CalcOpticalFlowPyrLK (prev, curr, prevPyr, currPyr, prevFeatures, winSize, level, criteria, flags, guesses=None) - textgreater (currFeatures, status, track_error)
The above library is also based on the LK pyramid method, which is the same as the C++ method described in Chapter 2.

Parameter settings (Python optical flow)

- prev (CvArr): The first frame at time t.
- curr (CvArr): The second frame at time t + dt.
- prevPyr (CvArr): Buffer of the image pyramid for the first frame. If the pointer is not NULL, the buffer must be sufficiently large to hold the image pyramids. A total of (image_width+8) * image_height/3 bytes is a sufficient size.
- curr_pyr: Similar to prevPyr. It will be used for the second frame.
- prevFeatures (CvPoint2D32f): Array of (feature) points needed to detect the flow.
- currFeatures (CvPoint2D32f): Array of 2D coordinate points. The new positions of the input features in the second frame are computed and stored here.
- winSize (CvSize): Size of the search window at each level of the image pyramid.
- level (int): Maximum number of levels in the image pyramid. If it is 0, the image pyramid is not used (1 level); if it is 1, up to 2 levels of the image pyramid are used. The same applies thereafter.
- status (str): Array. If the flow of feature points is detected, the corresponding element is set to 1; otherwise, it is set to 0.
- track_error (float): Optional. track_error (float): Optional array containing the difference between the perimeter of the feature points before and after the move. It can be NULL.
- criteria (CvTermCriteria): Criteria for stopping the iterative computation of the flow for each feature point in the image pyramid at each level.
- flags (int): Miscellaneous flags: CV_LKFLOWPyr_A_READY, the image pyramid for the first frame has been computed in advance (before calling the function). CV_LKFLOWPyr_B_READY, the image pyramid for the second frame has been computed in advance (before calling the function). CV_LKFLOWPyr_B_READY, the image pyramid for the second frame has been computed in advance (before calling the function).
- guesses (CvPoint2D32f): Optional array of estimated (initial) feature coordinates for the second frame, the same length as prevFeatures.

The main difference between C++ and Python is the library mentioned above. In C++ we need to detect the flow in advance and use the void goodFeaturesToTrack () library to detect the strongest corners in the image and use them as input values for the prevPts.

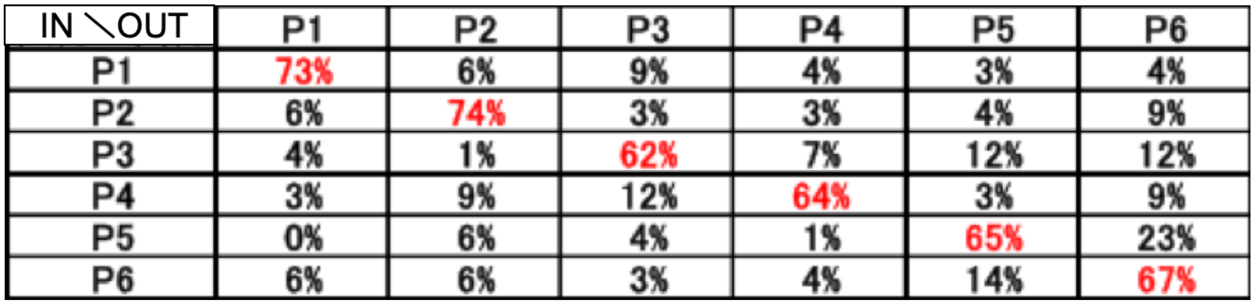
However, processing in Python is overwhelmingly more time-consuming and lacks real-time performance compared to processing in C++.

5.1.4 Usefulness of the library for the proposed method

For each pattern, we compared the identification results of the C++ library and the Python library. For comparison, the results of the conventional and proposed methods are shown in the table below.

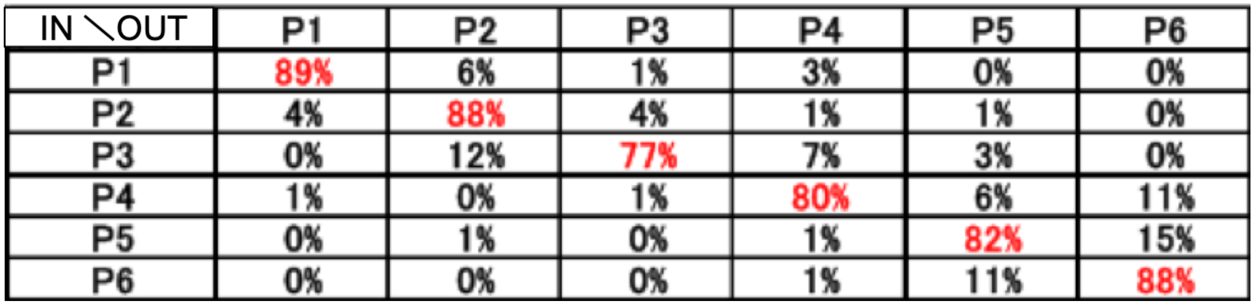
- **Experimental results for C++**

Table 5.1: Conventional method



IN ＼OUT	P1	P2	P3	P4	P5	P6
P1	73%	6%	9%	4%	3%	4%
P2	6%	74%	3%	3%	4%	9%
P3	4%	1%	62%	7%	12%	12%
P4	3%	9%	12%	64%	3%	9%
P5	0%	6%	4%	1%	65%	23%
P6	6%	6%	3%	4%	14%	67%

Table 5.2: Proposed method

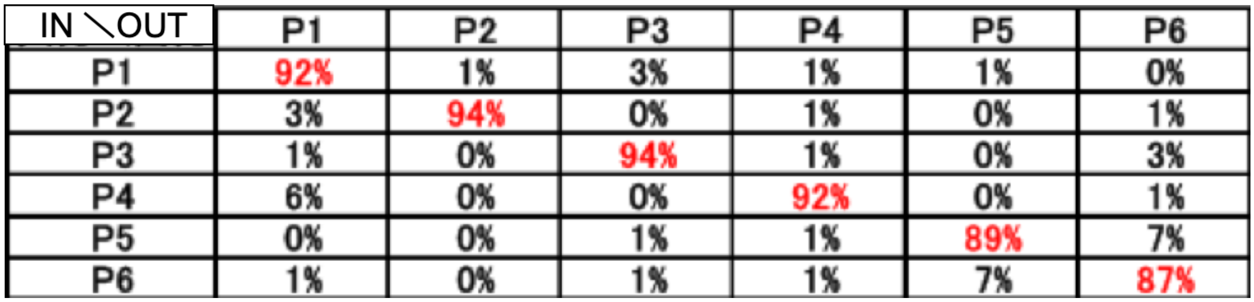


IN ＼OUT	P1	P2	P3	P4	P5	P6
P1	89%	6%	1%	3%	0%	0%
P2	4%	88%	4%	1%	1%	0%
P3	0%	12%	77%	7%	3%	0%
P4	1%	0%	1%	80%	6%	11%
P5	0%	1%	0%	1%	82%	15%
P6	0%	0%	0%	1%	11%	88%

Classification accuracy of Table 5.1: Average approximately 67.5%.
Classification accuracy of Table 5.2: Average approximately 84%.

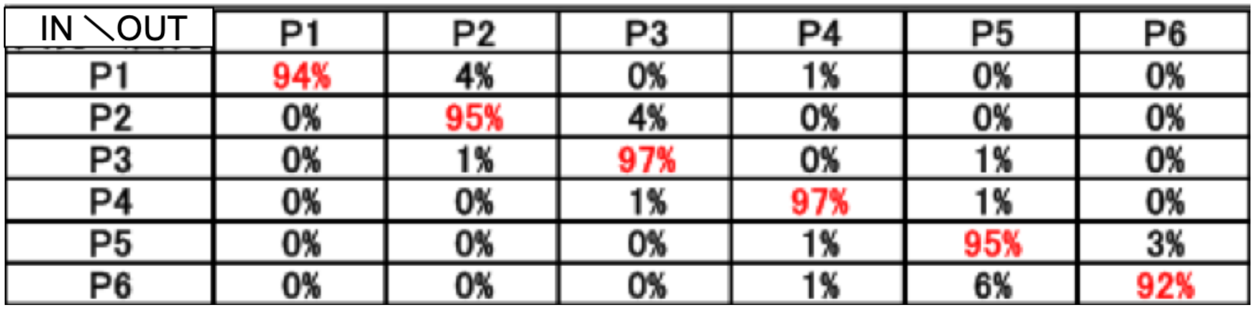
- **Experimental results for Python**

Table 5.3: Conventional method



IN ＼OUT	P1	P2	P3	P4	P5	P6
P1	92%	1%	3%	1%	1%	0%
P2	3%	94%	0%	1%	0%	1%
P3	1%	0%	94%	1%	0%	3%
P4	6%	0%	0%	92%	0%	1%
P5	0%	0%	1%	1%	89%	7%
P6	1%	0%	1%	1%	7%	87%

Table 5.4: Proposed method



IN ＼OUT	P1	P2	P3	P4	P5	P6
P1	94%	4%	0%	1%	0%	0%
P2	0%	95%	4%	0%	0%	0%
P3	0%	1%	97%	0%	1%	0%
P4	0%	0%	1%	97%	1%	0%
P5	0%	0%	0%	1%	95%	3%
P6	0%	0%	0%	1%	6%	92%

Classification accuracy of Table 5.3: Average approximately 91.3%.
Classification accuracy of Table 5.4: Average approximately 95%.

The above experimental results (1) show that the proposed method has high classification accuracy in both C++ and Python.

- Identification using the C++ library
  Comparing the results of the conventional and proposed methods, the classification accuracy of the proposed method is improved by 16.5%. In addition, when comparing the results of the proposed method and the results after denoising, the proposed method achieved higher accuracy.

- Identification using the Python library
  The identification results were high. Comparing the results of the conventional and proposed methods, the classification accuracy of the proposed method was only slightly better at 3.7%, but the classification accuracy was extremely high at approximately 95%. However, when we compare the results of the proposed

method with the results of the denoised method, the accuracy of P3 is slightly lower than that of the proposed method.

As mentioned above, both methods were found to be useful. To measure the usefulness of the proposed method accurately, we conducted an experiment using the C++ library.

5.2. Experiments on Pattern Identification

In this study, to demonstrate the effectiveness of features, we experimented to determine whether we could correctly identify each pattern. The methods and results of the experiments are described below.

1. Input: 1 pair (1 video each of 6 patterns) of 66 pairs (396 videos) used as training videos.

2. Learning: classification model is developed using the remaining 65 pairs (390 videos).

3. Features: We experimented with three types of features: the conventional method, after denoising (before correction of optical flow), and the proposed method.
   ① Conventional method: Optical flow + area
   ② After denoising: After denoising by labeling, optical flow + area
   ③ Proposed method: After removing noise by labeling process, i.e., correction of optical flow by the center of gravity + area

4. Discriminator: six patterns using SVM.

5. Number of experiments: A total of 66 experiments by changing the input pair by pair using cross-validation.

Cross-validation is a way of experimenting by changing one pair of input and 65 pairs of training in the above experiment method. The total number of experiments is 66. For example, A (a1 to a6) and B (b1 to b6). If A is the input, the others B to L are used as the classification model. If B is the input, the other B is used as the classification model, and if C is the input, the other C is used as the classification model.

The experimental results of the conventional method, the denoised method, and the proposed method are shown in the Confusion Matrix table below.

5.3. Is Each Pattern Properly Recognized?

We conducted experiments to determine whether each pattern is properly categorized. Thus, we used one set (6 videos) out of the 66 sets (396 movies) as input data and the remaining 65 sets (390 movies) as training data. Through cross-validation, by switching the input data set by set with the training data, we performed 66 experiments. We employed SVM as a recognition classifier, where the radial basis function is used as the kernel function, and the parameters C and gamma were set to 100 and 0.1, respectively.

Three types of methods were used to output data to compare the traditional method with the proposed method, namely, the traditional method, post-noise removal (before center-of-gravity correction), and the proposed method. The results of the experiments using these methods are summarized in Table 1. The vertical and horizontal axes show the inputs and results of recognition, respectively.

The closer the results of the diagonal axis to 100%, the higher the categorization accuracy

Table 5.5: The accuracy of each washing pattern using the traditional method.

Input *n* Output	P1	P2	P3	P4	P5	P6
P1	73%	6%	9%	4%	3%	4%
P2	6%	74%	3%	3%	4%	9%
P3	4%	1%	62%	7%	12%	12%
P4	3%	9%	12%	64%	3%	9%
P5	0%	6%	4%	1%	65%	23%
P6	6%	6%	3%	4%	14%	67%

Table 5.6: The accuracy of each washing pattern using the noise removal method (before center-of-gravity correction).

Input *n* Output	P1	P2	P3	P4	P5	P6
P1	80%	17%	0%	1%	1%	1%
P2	11%	82%	4%	1%	1%	0%
P3	0%	9%	82%	3%	4%	1%
P4	1%	0%	3%	82%	4%	9%
P5	0%	0%	4%	1%	79%	15%
P6	1%	0%	1%	6%	18%	73%

Table 5.7: The accuracy of each washing pattern using the proposed method.

Input *n* Output	P1	P2	P3	P4	P5	P6
P1	89%	6%	1%	3%	0%	0%
P2	4%	89%	4%	1%	1%	0%
P3	0%	12%	77%	7%	3%	0%
P4	1%	0%	1%	80%	6%	11%
P5	0%	1%	0%	1%	82%	15%
P6	0%	0%	0%	1%	11%	88%

Using the results in Tables 1 to 3 to compare the overall proper recognition rate of the traditional and proposed methods, the average correct recognition rate of the former and latter were 67.5% and 84%, respectively. The proposed method improved recognition accuracy, showing a higher correct recognition rate. Then, we conducted the following experiment.

5.4. Experimental Methods for Hand-washing

Next, we conducted an experiment on the hand-washing inspection method as follows.

In this database, six patterns (each pattern is approximately 5 s) included in one person's hand-washing video are considered as one set (6 videos), and 11 sets for six people were prepared. A total of 66 sets (396 videos) were used. For each video, feature extraction was performed and a classification model was developed. C++ (OpenCV v.2.4.9) was used as the language of the development system. A web camera that can acquire images in real time was used as the development equipment.

1. Input: For each person, a movie (approximately 30 s) of the correct washing method (approximately 5 s for each pattern) is taken once, in the order of P1 to P6. A total of five experiments will be conducted.

2. Learning: A learning model is developed using the 396 learning videos described in the experimental environment above.

3. As in experiment (1), we used the conventional method, after denoising (before correction of optical flow), and the proposed method.
   ① Conventional method: Optical flow + area
   ② After denoising: Optical flow + area after denoising by labeling process.
   ③ Proposed method: After removing noise by labeling process, i.e., correction of optical flow by the center of gravity + area

4. Discriminator: Classified into 6 patterns using SVM.

Discriminator: Classification into six patterns using SVM. The experimental results and discussion of the conventional method, after denoising (before optical flow correction), and the proposed method are described below.

5.5. Hand-washing Examination for Unknown Data

We conducted experiments for the hand-washing examination system using unknown data. A total of 66 sets (396 videos) in the database were used as the training data. Unknown data were used for the input data. For the unknown data, five people were videoed while washing their hands using patterns 1–6 in succession for approximately 30 s.

These videos were divided into video slices, and the window width of each video slice was set to 120 frames by moving 30 frames as the slide width (Fig. 5.1). Afterward, all patterns of video slices were recognized using SVM, as was the case with experiments in the previous section. Output data were acquired using the traditional method, pre-center-of-gravity correction (post-noise removal), and the proposed method.

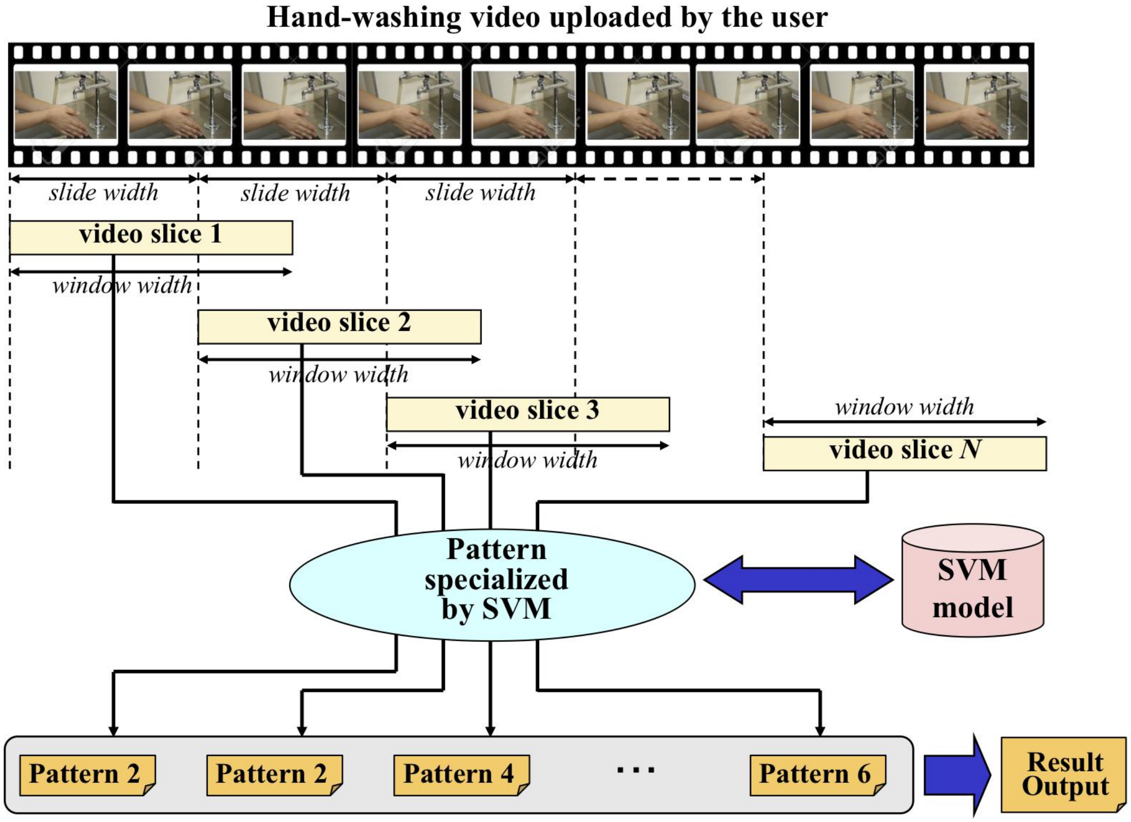


Fig5.1: Outline of the window-based hand-washing pattern specialized by SVM.

We calculated the rate of correct recognition for input videos 1–5. Table 5.8 summarizes the final results.

Table 5.8: Final accuracy for test video data using each method.

Methods *n* Data	Video 1	Video 2	Video 3	Video 4	Video 5	Average
Traditional Method	46%	36%	45%	27%	33%	37.4%
Post-Noise Reduction	50%	50%	73%	41%	83%	59.4%
Proposed Method	75%	68%	67%	68%	67%	69.0%

From the above experimental results, comparing the average of the correct identification rates of the conventional and proposed methods, the proposed method showed an improvement in recognition accuracy.

For each pattern from P1 to P6, each pattern was always identified by the correct washing method.

Comparing the results of the conventional and proposed methods, the proposed method showed an improvement of 33% in the average of the correct identification rate. Comparing the results of the proposed method with those of the denoised method, the proposed method showed higher accuracy, but the accuracy of P3 and P4 was considerably reduced in the proposed method.

The following is a discussion of the overall experiment regarding the problems from the above experiments on the identification of each pattern and the hand-washing inspection method.

6. Considerations

Comparing the average of the correct identification rate of the conventional and proposed methods based on the experimental results, the proposed method showed an improvement in recognition accuracy, which indicates the effectiveness of the proposed method in this research.

Comparing the results of the conventional and proposed methods, the proposed method showed an improvement of 33% in the average of the correct identification rate, but the overall result was not high with an average of 68%.

Comparing the results of the proposed method and the results after noise reduction, the proposed method showed higher accuracy, but in videos 3 and 5, our proposed method showed a lower correct identification rate than after noise reduction, probably due to the lower recognition rate of P3 and P4 (Table 5.3.2 and Table 5.3.3). In these hand motions, the center-of-gravity correction is not extremely effective, which may degrade recognition accuracy.

First, with respect to the hand-washing method movements of P3 and P4, we did not see a significant change in the center-of-gravity point because the movements were less violent and the hand was washed in that position than the other hand-washing method patterns. Therefore, the method for correcting the optical flow by the movement of the center of gravity may not be effective. Therefore, we need to consider how to decide whether to correct the optical flow before or after the correction.

In addition, to improve the overall recognition accuracy, it seems necessary to take measures against videos with low recognition accuracy (e.g., Fig. 6.1) in which skin color is not detected correctly.

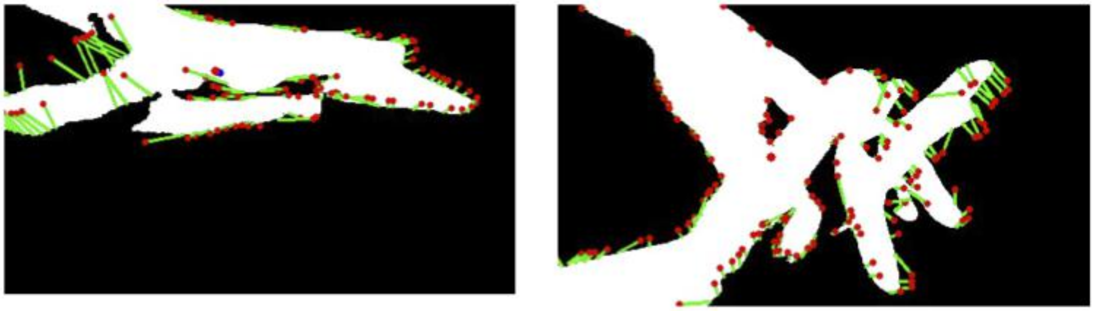


Fig. 6.1: Cause of decrease in accuracy (skin-color detection).

There are cases, where skin tone feature values are not properly obtained because of changes in lighting, etc. When such images are mixed, the appropriate feature values cannot be obtained.

One possible solution is to use supervised learning to obtain skin tone regions and background subtraction to preserve hand regions.

In this study, we developed our system using both the C++ and Python languages. The differences between the two languages are described below.

- C++ has a high search speed and is suitable for real-time processing, making it suitable for our future goal of real-time hand-washing inspection. However, we need to improve classification accuracy.
- Python showed an extremely high classification accuracy. However, the processing speed is extremely slow, and it is unsuitable for real-time processing. In terms of development, Python is extremely easy to use and can be programmed quickly, so it is a good choice when trying out new methods in experiments.

7. Conclusion

7.1 Conclusion

In this study, we conducted an experiment to verify the effectiveness of the features by comparing our proposed method with the methods found in existing academic papers. On the basis of the experimental results, we confirmed the effectiveness of the features of the proposed method because each pattern can be recognized.

In this study, we improved a hand-washing method for correct hand-washing against alcohol disinfection and demonstrated the effectiveness of our method. In addition, we improved the hand-washing method for correct hand-washing against alcohol disinfection by removing noise through labeling and correcting optical flow focusing on the center of gravity. As experiments, we conducted tests to determine whether each pattern was identified by the conventional method, after noise removal (before correction of the optical flow by the center of gravity), and the proposed method, as well as tests on a hand-washing inspection system, to confirm the accuracy and identification results of each experiment.

(Problem 1) because of changes in lighting and other factors, the skin color features cannot be extracted properly, and images with incorrectly shaped hands are mixed into the overall training and input, degrading accuracy. As a solution to this problem, methods such as skin color detection and background subtraction using supervised learning can be considered.

(Problem 2) For P3 and P4, the effectiveness of the labeling process in removing noise was observed, but the effectiveness of the correction of optical flow due to the center of gravity was not great. The reason for this was that the center-of-gravity point did not move significantly for P3: washing fingertips and fingernails; P4: washing between fingers; thus, the optical flow was not corrected. A method for determining whether to correct the optical flow before and after correction is required.

7.2 Future work

For future work, we need to address the problem of hand detection by skin color detection and the problem of patterns that do not show the effectiveness of optical flow correction. In addition, it is necessary to consider a frame segmentation method that can automatically identify the breaks in hand-washing. The frame segmentation in this study is based on the method described in Chapter 5, where the interval between frame segments is fixed. In this study, we used the method described in Chapter 5 for frame segmentation.

In addition, to expand the range of adaptation to daily hand-washing, experiments using features for external factors such as soap bubbles and water should be conducted.

Acknowledgments

We would like to thank everybody who helped us in this research. We would like to thank the reviewers for their constructive comments.

We would like to express our sincere gratitude to Professor Masamiki Shishibori, Graduate School of Socio-Techno-Science, The University of Tokushima, for his long hours of guidance as our advisor.

This work was supported in part by NTT Plala Inc. and a Grant-in-Aid for Scientific Research (No. 18K11573) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. "Summary of the 2012 Annual Report of the Monthly Vital Statistics (Approximate)," Ministry of Health, Labour and Welfare, http://www.mhlw.go.jp/toukei/saikin/hw/jinkou/geppo/nengai12/, (reference 2014-02-02)

2. "Infection status in the world", NHK (Japan Broadcasting Corporation), https://www3.nhk.or.jp/news/special/coronavirus/world-data/, (reference 2021-07-08)

3. Tamito Fukada, "Comparison of the incidence of surgical site infection between rubbing and scrubbing methods for hand washing during surgery", Journal of the Japanese Society for Surgical Infectious Diseases, 13495755, 2006-11, 3 4 515-519, Japanese Society for Surgical Infectious Diseases

4. Hisako Yano, Hiroi Kobayashi, "Hygienic Handwashing Behavior of Nurses", Environmental Infection, 1995, vol. 10, no. 2, p. 40-43, published July 21, 2010, Japanese Society for Environmental Infection Research

5. Yoshida Pharmaceutical, "Y's Square: Academic information on hospital infection and nosocomial infection control (2) Healthcare workers", http://www.yoshida-pharm.com/2012/text03_01_02/, (reference 2014-02-02)

6. Asako Terashima, Tomoko Takemura, Kayoko Maezawa, Noriko Kobayashi, Junko Kizu, "Effectiveness of hand washing education for pharmacy students before going to clinical practice", Japanese journal of environmental infections 24(6), P425-431, 2009-11- 25

7. M. Isogai, T. Nishikawa, H. Isogai, N. Isogai, Y. Kurebayashi, and T. Hayashi, "Handwashing Effectiveness for Sterilization in the Home and Detection of Bacteria from Environmental Surfaces," Environmental Infection, Vol. 22 (2007) No. 3, pp. 175-180, 2010-07-21, Japanese Society for Environmental Infection Researc

8. H.Sawada, S. Hashimoto and T. Matsushita. A Study of Gesture Recognition Based on Motion and Hand Figure Primitives and Its Application to Sign Language Recognition, *Transactions of Information Processing Society of Japan*, 39(5), pp.1325{1333, 1998 (Japanese).

9. Shinpei Inokari, Naohiro Fukumura, "Recognition of Arm Movement Patterns of Sign Language Words Based on Jerk Minimal Model for Sign Language Translation", IEICE Transactions D, Vol.J98-D No.3, pp.437-447, 2015-03-01

10. D. Maehata, M. Nishida, Y. Horiuchi, and S. Kuroiwa, "Recognition of Sign Language Words by HMM Focusing on Hand Position and Motion," IEICE Technical Report PRMU, Pattern Recognition and Media Understanding 108(94), p7-12, 2008-06-12

11. S. Igari, N. Fukumura. Recognition of Japanese Sign Language Words Represented by Both Arms Using Multi-Stream HMMs, In *Proceedings of the 7-th International Multi-Conference on Complexity, Informatics and Cybernetics*, pp.157{162, 2016.

12. M. Yamamoto and K. Kaneko, "Parallelization of Vanishing Point Trajectory Estimation from Moving Camera Image Database Using MapReduce," DEIM Forum 2011, E9-2

13. B. D. Lucas, T. Kanada. "An Iterative Image Registration Technique with an Application to Stereo Vision", In *Proceedings of the 7-th International Joint Conference on Artificial Intelligence*, 2, pp.674{679, 1981.

14. Stan Birchfield, “Derivation of Kanade-Lucas-Tomasi Tracking Equation”, January 20, A. D. 1997 http://citeseerx.ist.psu.edu/viewdoc/download (viewed July 08, 2021)

15. T. Tsuchida, M. Ogino and C. Takeda. A Study of Verification for Learning Effects of a New Hand Washing Learning System with Gamification Using a Three-Axis Acceleration Sensor, Japanese journal of nursing research, 36(4), pp.19{27, 2013 (Japanese).

16. "Development of video recognition AI technology to determine correct hand washing behavior."Fujitsu Laboratories Ltd, May 26, 2020, https://pr.fujitsu.com/jp/news/2020/05/26.html, (viewed July 08, 2021)

17. J. A. K. Suykens, J. Vandewalle. Least Squares Support Vector Machine Classifiers, Neural Processing Letters, 9, pp.293{300, 1999.

18. K. Cho, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp.1724{1734, 2014.

19. H. Zen and H. Sak. Unidirectional Long Short-Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis, In Proceedings of International Conference on Acoustics, Speech, and Signal Processing, pp.4470{4474, 2015.

20. Seiji Hotta, Chiya Kiyasu, and Sueharu Miyahara, "Pattern Identification Based on the Distance between Unknown Pattern and Category k Neighborhood Mean," IEICE

Transactions D, Vol. J88-D2 No. 8, pp. 1357-1366, 2005-08-01

21. Takahiro Okabe and Yoichi Sato, "Application of Support Vector Machine to Object Recognition with Illumination Change," Transactions of Information Processing Society of Japan, Vol.44 No.SIG5(CVIM6), pp.22-29, 2003-04

List of Papers Published

Main Thesis

1) Katsumi Nagata, Masaki Oono, Masami Shishibori, “The Development of a Hand-Washing Support System Using Image Processing Techniques”, International Journal of Advanced Intelligence, Vol.11, Num.1, pp. 1-13, April, 2020.

Submain Thesis

1) Katsumi Nagata, Masaki Oono, Masami Shishibori, “The Development of a Hand-Washing Education System”, Proceedings of the 27-th International Conference on Computers in Education, Vol.1, pp. 225-229, December, 2019.