

Enhanced Human Hitting Movement Recognition Using Motion History Image and Approximated Ellipse Techniques

I Gede Susrama Mas Diyasa ^{a,1,*}, Made Hanindia P ^{b,2}, Mohd Zamri ^{c,3}, Agussalim ^{a,4},
Sayyidah Humairah ^{b,5}, Denisa Septalian A ^{d,6}, Faikul Umam ^{e,7}

^a Department of Magister Information Technology, Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Surabaya, Indonesia

^b Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Surabaya, Indonesia

^c Faculty of Electrical and Electronics Engineering Technology, University Malaysia Pahang Al-Sultan Abdullah, Pahang, Malaysia

^d Department of Data Science, Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Surabaya, Indonesia

^e Department of Information System, Faculty of Engineering, Universitas Trunojoyo Madura, Indonesia

¹ igsusrama.if@upnjatim.ac.id; ² madehanindia.fik@upnjatim.ac.id; ³ zamri@ump.edu.my;

⁴ agussalim.si@upnjatim.ac.id; ⁵ 20081010047@student.upnjatim.ac.id; ⁶ 21083010113@student.upnjatim.ac.id;

⁷ faikul@trunojoyo.ac.id

* Corresponding Author

ARTICLE INFO

Article history

Received September 09, 2024

Revised December 01, 2024

Accepted December 14, 2024

Keywords

Human Hitting Movement;

Recognition;

Motion History Image;

Motion Detection;

Video Analysis;

Approximated Ellipse

ABSTRACT

Recognition of human hitting movement in a more specific context of sports like boxing is still a hard task because the existing systems use manual observation which could be easily flawed and highly inaccurate. However, in this study, an attempt is made to present an automated system designed for this purpose to detect a specific hitting movement commonly known as a punch using video input and image processing techniques. The system employs Motion History Image (MHI) to model trajectories of motions and combine them with other parameters to reconstruct movements which tend to have a temporal component. Thus, CCTV cameras set at different positions (front, back, left and right) enable the system to identify several types of punches including Jab, Hook, Uppercut and Combination punches. The most important aspect of this work is the proposal of MHI and the Ellipse approximation which is quicker in the integration of both than other sophisticated systems which take a considerable duration in computations. Therefore, the system classifies C_motion, Sigma Theta, and Sigma Rho parameters to distress hitting from non-hitting movements. Evaluation on a dataset captured from multiple viewpoints establishes that the system performs well achieving the goal of 93 percent when detecting both the hitting and the non-hitting motion. These results demonstrate the system's superiority to the system based such detection methods. This study paves the way for other applications in real-time such as sports analysis, security surveillance, and healthcare requiring greater efficiency in and accuracy of human movement assessment. The focus of future work may be in the direction of improving the recognition of slower movements, also modifying the system for more dynamic conditions in the future.

This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

In a normal or abnormal activity such as a fight or boxing match and others, of course, there is a hitting movement, kicking movement or other attacking and defending movements [1], [2]. In the process of executing hitting movement or other attack techniques, one of the ways to perform this is by utilizing hand. Consequently, there arises a necessity to perform human hand-hitting motion detection in a fight [3], [4]. This will make detecting hitting movement easier to be detected by the system through video surveillance as a video capture tool compared to the manual observation that has been used [5], [6].

The hitting movement often used in fights is the punching movement which aims to knock down an opponent in a boxing match. For this reason, in this study, the recognition of hitting movements in humans was carried out by classifying the types of punches that exist in boxing matches for testing. The hitting movements include Jab moves, Uppercut moves, Hook moves, and Cross moves [7]. Hand gestures are one of the most natural interaction approaches of human beings and have few physical and mental limitations [8], [9].

Motion History Image (MHI) and Approximated Ellipse became the means of processing images in order to identify movements made by a person [10], where a recorded video is perceived and taken apart into several images so as to create sequences of images. Subsequently, many of these images can be processed to determine an object's movements based on the shifting in the pixels. In the context of movement in normal activities, the image pixels do not experience big shifting, while on the other hand, irregular movements and hitting in particular will result in rapid pixel shifting [13]. Video data from CCTV cameras will be processed using Motion History Image (MHI) [14], which later goes further into a test using the parameters C_{motion} , Sigma Theta, and Sigma Rho that serve as points of reference to identify human hand punching activities [15].

There are several studies related to the detection of human movement, including research conducted by Rougier, C. et al., [16], this research focuses on identifying a person's fall as perceived by a computer. The processing technique, namely Motion Analysis and Multiple Shape Feature analyzes shifting in the pixels of the image sequence to be able to identify a person's movements, and when a certain movement appears to be swift and stops right after, the person is then recognized by the computer system to have fallen [17]-[20]. This study has proved that a system run in a computer is able to identify a certain movement such as people falling and distinguish them from everyday movements. Other researchers Cheng Xu, et al., also conducted a study regarding the introduction of human movement with the Recurrent Transformation Prior Knowledge-based Decision Tree (RT-PKDT) model, where the test results using the RT-PKDT method obtained an accuracy of 96.68% [21]. With high accuracy in this study, it will be used to classify the types of punches and kicks to the types of fights [22].

Rafet Durgut and Oğuz Findik, About Human Gesture Recognition, an action recognition method is proposed using the bone joint information obtained by the Microsoft Kinect sensor. Separation of the keyframe is accomplished through utilizing the frame connection information [23]. This method calculates the weight value of each keyframe, and the temporal discrepancy issue occurring when comparing test and reference actions was resolved by Dynamic Time Warping (DTW). The results obtained from DTW are classified using k-neighbor algorithm. In a parallel vein, a study conducted by V. Hany El-Ghaish, et al. acknowledges that human action recognition is a complicated problem since it must detect the movement of the actor from each actor and from viewpoint variation [24]. This study uses 3 models, namely 3-D skeletons, body part images, and motion history image (MHI). While 3D skeleton modeling serves to capture the pose of the actor adeptly, the shortcomings in this modeling in manipulating the body shape information prompt the necessity to add MHI [25]-[27].

Based on some of the studies above, the drawbacks regarding the slow computational time will be a problem when applying it in real-time. Therefore, in this study, a simple method was used by combining motion history images, Frame Difference, and Approximated Ellipse. This method is performed by dividing the video into several frames so that the computation time required is shorter to detect and classify hitting movements in humans. This research begins with an introduction,

followed by developing the method used and carrying out trials to obtain the maximum results in detecting hitting movements in humans.

2. Method

To enhance focus during the research process, a comprehensive block diagram is developed. This diagram serves as a visual representation of the research workflow, effectively illustrating the interconnections among various research components. By adopting this systematic approach, we ensure clarity, coherence, and alignment with the overarching research objectives.

The initial stage of the research is to capture motion images as input data from a CCTV video surveillance that has been set to carry out the trial process as shown in the block diagram of the hitting motion detection process in Fig. 1. The input of this hitting motion detection system is video data that has been recorded on the previous CCTV camera. The input to the system can be processed if two videos are entered in one recording, where the time and place are in the same position as though two CCTV opposites or right and left of the object. The video input contains one person making a hitting motion, the video is already in the process of being cut to 2 seconds per video in MP4 format.

Hitting data is taken in a room with sufficient light and is not blocked by anything, video data is also taken on objects (humans) with different types of strokes. Hit data will be entered alternately in the python program to build parameters that will be used as a reference for detecting hits on the system [28].

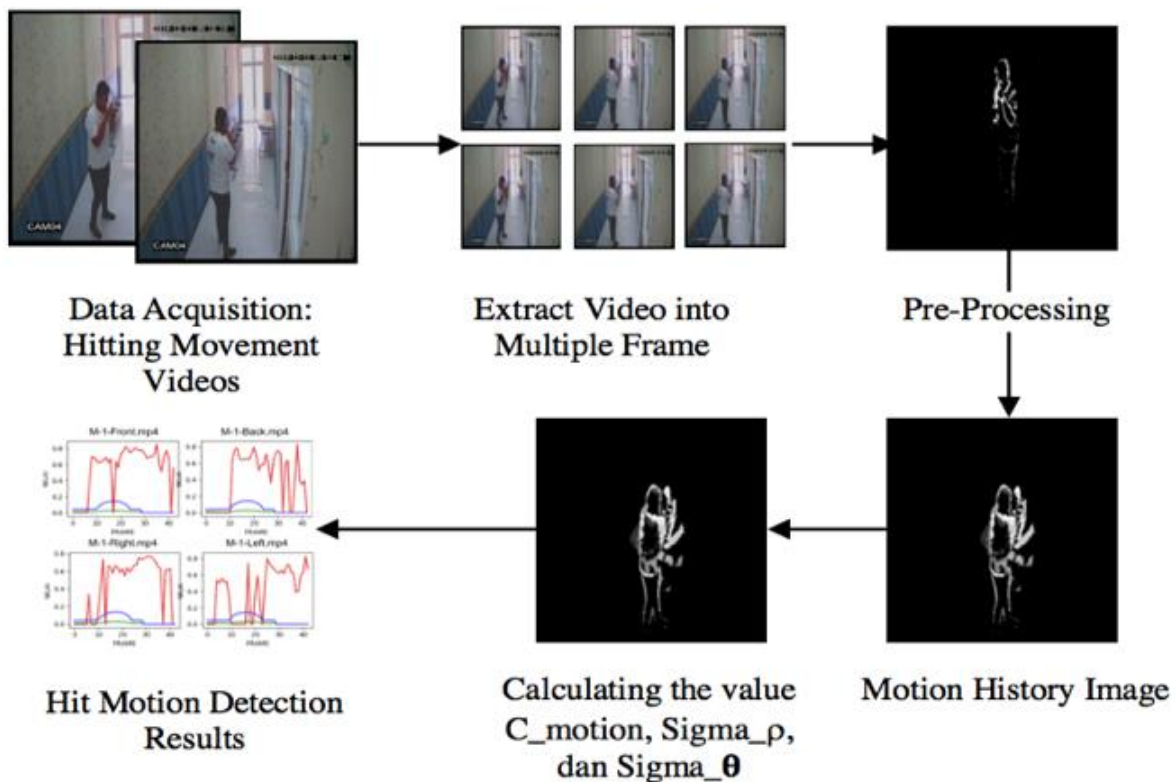


Fig. 1. Block diagram of the hitting motion detection process

2.1. Huge Data Acquisition Hits

The research begins with data acquisition from CCTV video surveillance, capturing hitting motions from individuals in controlled environments. The file entered for this research is a video in MP4 format, before entering the data for processing, the video data must be extracted into image frames individually based on the number of frames per second the video has, for example if a video has a quality of 60 fps (frames per one second) then an extraction from that video would produce 60 fps, and so on. As shown in Fig. 2, the workflow on the hitting motion detection system for the process

of extracting video into frames is done by playing back the video that has been provided, then determining the desired video duration, in this study several extra 60 fps – 120fps were carried out and video capture per frame, up to the duration the specified time is complete, this process will get frames for hitting motion detection which will be analysed in the next process [29], [30].

2.2. Convert RGB Image to Grayscale

RGB to Gray-scale conversion can be done using a simple transformation. Gray-scale conversion is the first step in some image analysis algorithms, because it basically simplifies (reduces) the amount of information in the image [31].

$$X = \frac{R + G + B}{3} \quad (1)$$

As shown in Equation (1), the RGB to Gray-scale conversion reduces the red, green, and blue channels in an image by calculating each pixel's mean. Although Grayscale contains less information than color images, some important information, information related to features is retained such as edges, areas, blobs and so on. RGB to Gray-scale conversion also can be done with the equation that contains I is perceived as intensity, WR acts as weight factor of R, WG acts as weight factor of G, and WB acts as weight factor of B. It is imperative to provide an equation (1) in which the weight factor has to end in one, namely WR+WG+WB=1. Noticing same three colors are also of importance and therefore as follows, WR=WG=WB=1/3 [27], [32], [33].

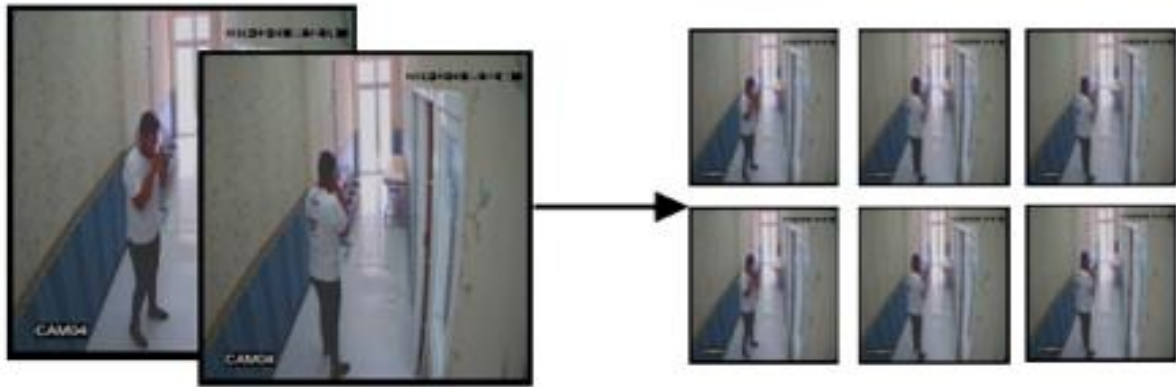


Fig. 2. Block diagram of the data acquisition process

2.3. Frame Difference Process

The frame difference process plays a critical role in isolating motion within video sequences by leveraging a background model derived from the average of multiple frames, shown in equation (2).

$$B(x, y) = \frac{1}{N} \sum_{i=1}^N F_i(x, y) \quad (2)$$

Where:

$B(x, y)$ = Background model

$F_i(x, y)$ = Intensity at pixel (x, y) in the i -th frame

N = Total number of frames used to compute the average

To identify moving objects, each frame extracted from the video is compared against the background model. The difference between the frame and the background is calculated pixel by pixel, shown in equation (3).

$$D(x, y) = |F_t(x, y) - B(x, y)| \quad (3)$$

Where:

$D(x, y)$ = Frame difference

$F_t(x, y)$ = Frame extracted from the video

In this process, a new image will be generated which contains differentiating objects between the background model and the frames from the video extraction. The resulting new image is still in grayscale form (f), the background model is taken from the average frame which will then be compared with the extracted video frames. In this process, a new image will be generated which contains differentiating objects between the background model and the frames from the video extraction. The resulting new image is still in grayscale form (f) [34], [35].

2.4. Grayscale to Binary Process

Binary images are widely known as B&W (black-and-white) or monochromatic images, and each pixel in a binary image is depicted in 1 bit only. Constructing a binary image takes a certain colorimetry or scale in shades of gray as a reference [36]. For example, pixels in a more intense shades of gray are valued as 1 and pixels in a less intense scale are valued as 0. In converting grayscale images into binary images, same equation applies with the equation (2).

2.5. Motion History Image

The motion history image is built using binary images that have gone through preprocessing before, then later image is utilized as input data to build MHI [37]-[39]. Where to look for bright values according to newer movements [40]. In order to get MHI, one thing to do is to pull binary motion sequence of an object $D(x, y, t)$ from its first taken image $I(x, y, t)$ by utilizing an image-differencing method. Furthermore, each pixel of the Motion History Image H_τ acts as one function in a certain time range of $t(1 \text{ n})$, as shown in equation (2) [41], [42].

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t - 1) - 1) & \text{otherwise} \end{cases} \quad (4)$$

Where:

x = Location of a targeted pixel at x-coordinate.

y = Location of a targeted pixel at y-coordinate.

$H_\tau(x, y, t)$ = Motion History Image (MHI).

$D(x, y, t)$ = Binary arrangements of an object's movement.

Objects are predicted utilizing an ellipse using moments. Ellipse is specified by the centroid (x, y), orientation, and major semi-axis a and minor semi-axis b . Therefore, the moments' value of sequential images can be achieved by an equation as follows [43]-[45].

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (5)$$

Where:

m_{pq} = Moment's value.

Value of p, q = 0, 1, 2

The centroid of an ellipse can be achieved by measuring coordinates at the core of mass with first and zero spatial order moments [46]-[48]:

$$\bar{x} = \frac{m10}{m00}; \bar{y} = \frac{m01}{m00} \quad (6)$$

Where:

\bar{x} = Centroid on the x-axis coordinate.

\bar{y} = Centroid on the y-axis coordinate.

Centroid (x, y) is used to measure the central moments [49], [50]:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (7)$$

Where:

μ_{pq} = Central moments.

An angle in the middle of object's principal axis and x axis provides orientation of the ellipse, that can be measured by central moments of order 2 [51]-[53].

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \quad (8)$$

Where:

θ = Orientation ellipse.

To get the major semi-axis a and the minor semi-axis b from the ellipse, we must calculate I-min and I-max, i.e. the smallest moment of inertia and the largest moment of inertia. This value can be calculated by evaluating the eigenvalues of the covariance matrix [54]-[56]:

$$J = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix} \quad (9)$$

The eigenvalue from matrix J is I_{\min} and I_{\max} which is measured by:

$$I_{\min} = \mu_{20} + \mu_{02} - \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \quad (10)$$

$$I_{\max} = \mu_{20} + \mu_{02} - \frac{\sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \quad (11)$$

Where:

J = covariance matrix

I_{\min} = smallest moment of inertia

I_{\max} = greatest moment of inertia

With a and b, we can determine the ratio of the ellipse:

$$\rho = \frac{a}{b} \quad (12)$$

Where:

ρ = Ratio ellipse.

a = Semi-minor axis of a.

b = Semi-minor axis of b.

3. Results and Discussion

In this section, it is explained the results of research and at the same time is given the comprehensive discussion.

3.1. Trial Scenario

The test scenario is carried out to get results that are in accordance with the design, but before going to the stage of the system test scenario, first perform a functional test of the components that are on the device that has been prepared. After carrying out the functional test of the components, it is continued by compiling the system using the python programming language. So, to do a punch detection system using the Motion History Image method.

3.2. CCTV Camera Functioning Testing

CCTV testing to determine the performance and functionality of the CCTV camera whether it works well or not. The steps of testing a CCTV camera are as follows:

- Prepare 2 CCTV cameras with 944x1080p image quality.
- Setting up a DVR to retrieve video data from a CCTV camera.
- Prepare 2 tripods to support 2 CCTV cameras.
- Setting up the monitor and mouse to monitor and move the pointer on applications found on the DVR
- Inserting a flash/hard disk for video data retrieval on the DVR

3.3. System Test

The system testing process aims to determine whether the system can implement and produce output in accordance with the design discussed earlier. Accuracy results from the Motion History Image testing process with parameters C_motion , $Sigma\ theta$ and $Sigma\ rho$ to distinguish between hitting and not hitting. Based on the results of the accuracy test, by comparing the accuracy using the program, it has been found with the appropriate output results, the results from the above test will be converted into a python programming application as a program to give an output command in the form of a value considered hitting from the image that has been taken via a CCTV camera.

3.4. Testing the Calculation of the Value of the Hitting Movement

The steps in the process of finding a value for calculating the value of a stroke are as follows:

- Prepare test data in the form of videos that have been cut into 2 seconds per video, either hitting or not hitting
- Using the Motion History Image method to get the per frame value, then looking for the C_motion , $Sigma\ theta$ and $Sigma\ rho$ values to get the hitting value and also not hitting it as a reference for parameter values.
- After getting the best value, it is continued by testing the punch detection system with that value by using 2 videos in one incident on the right and left sides.

3.5. Data Acquisition Setup






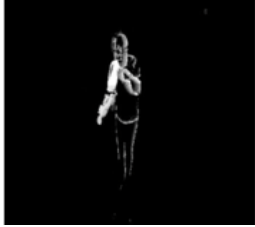






Data collection for this study used 2 CCTV cameras that had been previously adjusted for distance to get good pictures. Video data retrieval is carried out in a closed room with one object (human) performing 4 types of strokes and non-hitting movements [14]. The initial stage of the research is to capture motion image as input data for punch detection by a CCTV video surveillance that has been set up for testing. The input of this punch detection system is video data that has been recorded on the previous CCTV camera [57].

3.6. Video Extraction

In the video extraction process, the data is separated into frames by opening the video file after which it captures each frame in the video, with a video duration of two seconds so 43 frames are obtained. The result of this video extraction process is a collection of frames from the video. The process in capturing frames is to read the video and then read the first frame and then scale the resize. calculate the length and width of the frame dimensions, then the first frame as the previous frame,

initiate MHI as matrix 0 size wxh then save the number of frames. The following are some examples of the results of the video extraction process, as shown in Table 1 (a) and Fig. 3 is the source code for the Extraction process.

Table 1. Video extraction results of hitting movement, pre-processing results, and MHI results

	Hitting movement videos (a)	Preprocessing results (b)	MHI results (c)
Back side			
Front side			
Right side			
Left side			

3.7. Pre-Processing

The preprocessing process changes the image frames from RGB to grayscale [58], followed by the background subtraction process by modifying the frame difference process to converting the resulting frame difference image to a binary image, the pre-processing flow is as shown in Fig. 4.

3.8. Motion History Image

Motion History Image is built using an image that has previously been refined, the image is then utilized to be an input data to make a Motion History Image [59], [60]. Process Motion history image will be explained in accordance with Fig. 5.

To build a Motion History Image [61], [62] using the equation described in equation (1), the fg variable stores the frame while t stores the highest value motion history image, the first frame is taken and then stored in the frame dimensions then multiply each element in fg by t after that loop the second to the last frame then save the original value of the current frame in the frame variable after that loop based on the frame size and save the value to be manipulated into the variable matrix, if the original value of the pixels y and x is 225, change the value to t, otherwise save the value of the previous frame (which has been changed) into mtx, if the pixel value of mtx on y and x is more than one, the pixel value of the current frame is equal to the previous frame minus one otherwise the pixel value of the

current frame is equal to 0. Process Motion history image will be explained in accordance with Fig. 5. And Table 1 (c) is an example of the process results motion history image.

```
cam = cv2.VideoCapture(f)
ret, frame = cam.read()

scale = .4
h, w = frame.shape[:2]
rw = int(w * scale)
rh = int(h * scale)
prev_frame = cv2.resize(frame.copy(), (rw, rh))
motion_history = np.zeros((rh, rw), np.float32)
timestamp = 0
```

Fig. 3. Source code for video extraction in hitting motion detection

3.9. Specifying C_Motion Value

After receiving the MHI image, the next stage is to look for the value of C_{motion} as C_{motion} exhibits the speed of movement of objects that are scored in a scale of 0-1, which 0 means idling while 1 means moving. Finding the C_{motion} value involves the process of taking the number of frames, initializing a list of 0 along the number of frames, loop for each frame and define the highest value 15.

Loop for each pixel point in the frame if the value at this point is equal to max add the number of white pixels, if the value is greater than 0 and less from max add one for the number of ash values based on this point value, loop up to the max value if the current total ash value is less than $0.65 \times$ the number of white points add the number of ash value points with the ash value, if the ash value is not empty set C_{motion} the current frame becomes (number of gray points/(number of white points + number of gray points)), if the value already obtained is not a number set it to a value of 0 then return the value C_{motion} . Result of value C_{motion} can be seen in Fig. 6.

From Fig. 6 it can be seen that the value on the y line as a result of the C_{motion} value is colored red while the frame on the x line is the number of frames. It can be seen in the videos that M-1-Front.mp4, M-1-Back.mp4, M-1-Right.mp4 and M-1-Right.mp4 have the highest C_{motion} value of 0.848. For the number of frames per video, there are forty-three (43) frames in two seconds of video that have been entered into the system. Seen on the x line there are multiples of 10 where the number describes the number of frames that exist. In the graph the value on the y line describes the C_{motion} value in each frame, namely the value from 0.0 to more than 0.8 which is depicted in the output results run by the program.

3.10. Determining Sigma Theta and Sigma Rho Values

After taking an MHI image, the next stage is to figure out $Sigma\ Theta$ and $Sigma\ Rho$ values. The $Sigma\ Theta$ and $Sigma\ Rho$ values exhibit movements of an object's shape taken from several experiments. If the highest value of $sigma\ theta$ multiplied by 0.9 is less than equal to $sigma\ rho$, then it is considered a movement. The value of $sigma\ theta$ and $sigma\ rho$ is obtained by taking the length of theta value and then initializing sigma with a list of length s, loop as many as elements in theta if the theta value at index q is not a number set the theta value to be the value at the previous index or 0 if the index is equal to 0, loop from 14 to s then set value with standard deviation. From Fig. 6, it can be seen that the value on the y line as a result of the $sigma\ theta$ value is colored green while the $sigma\ rho$ is colored blue for the frame on the x line as the number of frames. It can be seen in the video M-

1-Front.mp4, M-1-Back.mp4, M-1-Right.mp4, and M-1-Right.mp4 have *sigma theta* values of 0.027 and *sigma rho* 0.144.

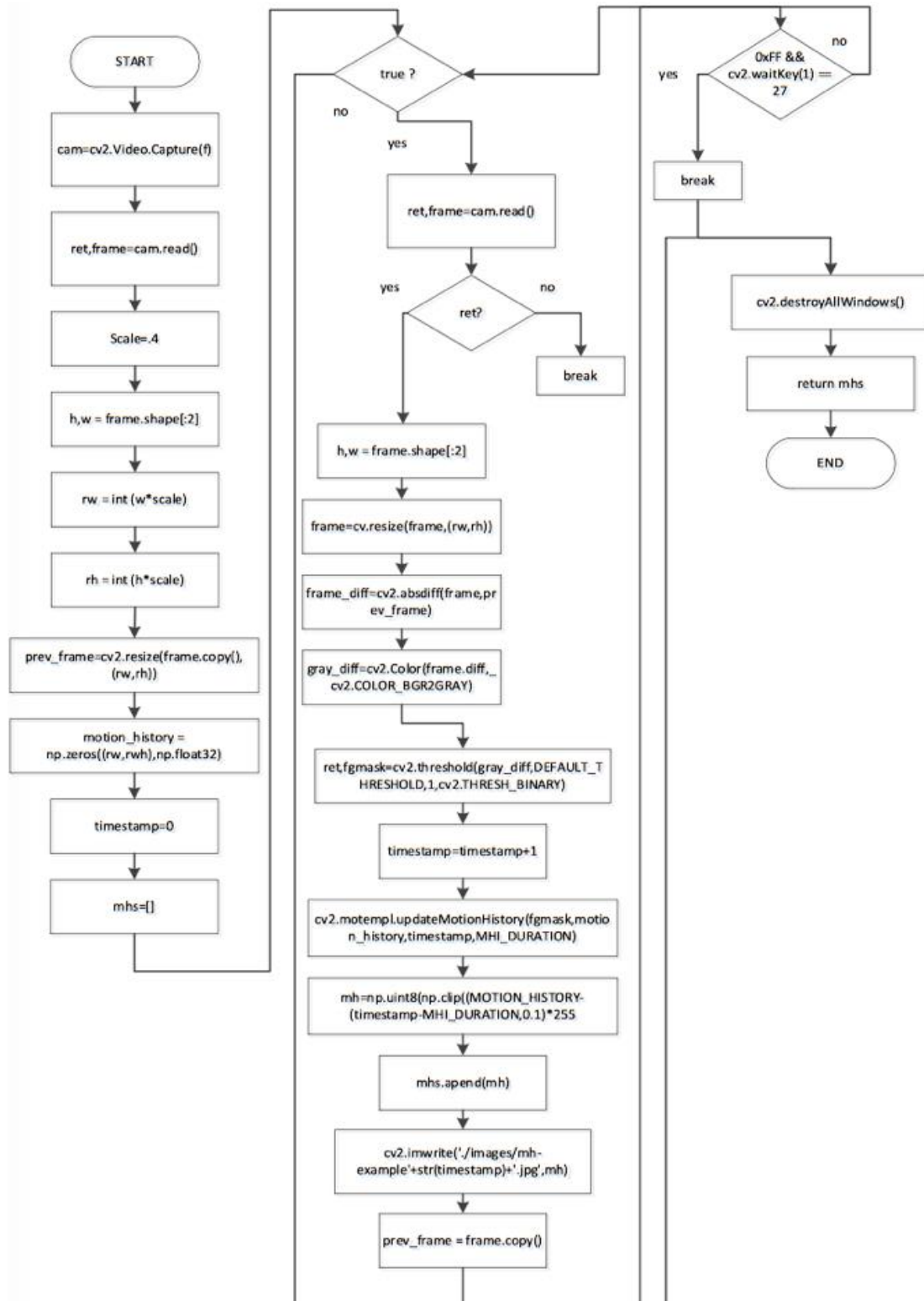


Fig. 4. Pre-processing flow

3.11. Detecting Hitting Motion

Upon getting the values of C_{motion} , $Sigma\ Theta$, and $Sigma\ Rho$, these values are then processed.

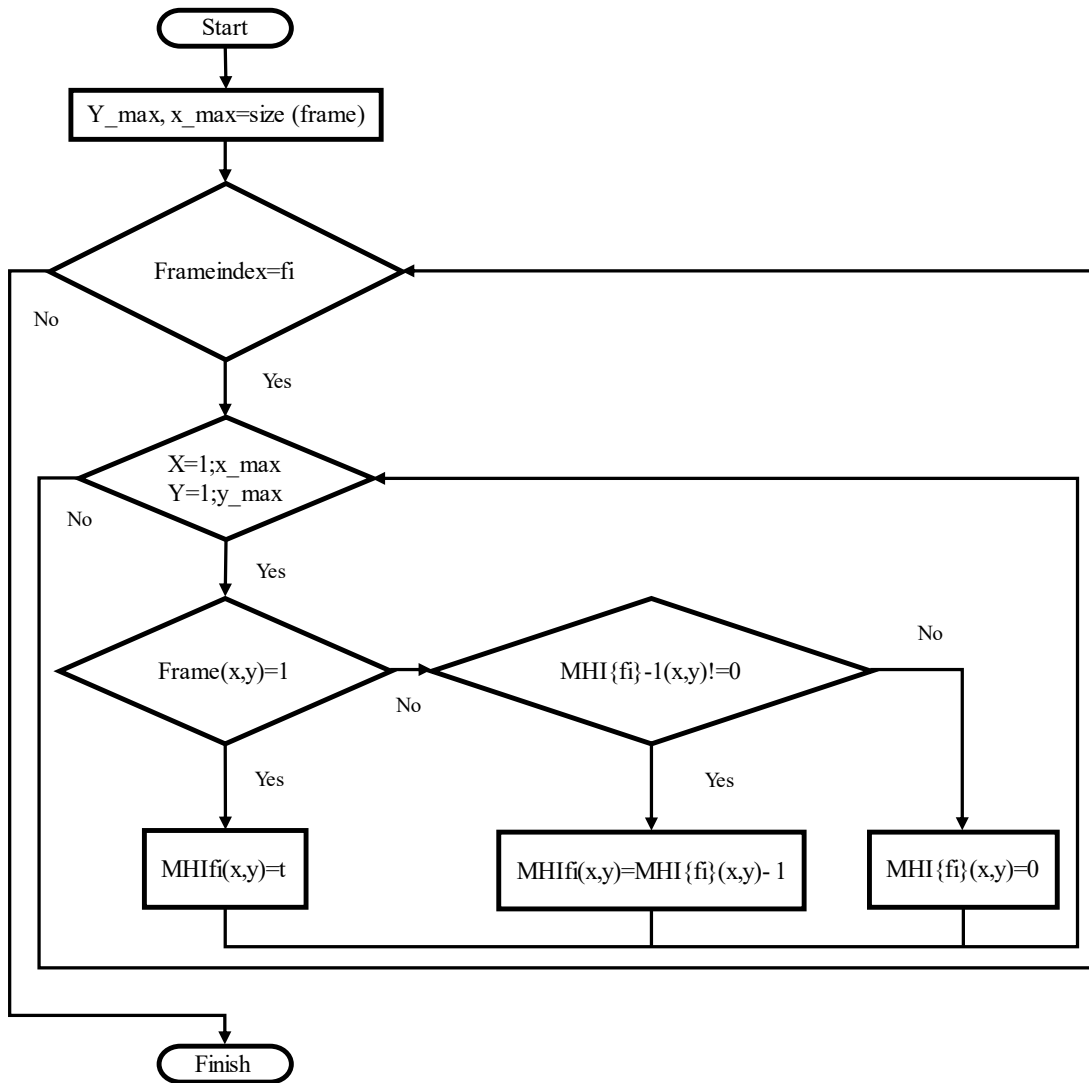


Fig. 5. Motion history image flow

Table 2. Trial result of hitting motion video for 62 data

No	Videos	C_motion	Sigma theta	Sigma rho
1	Video 1	0.853	0.027	0.146
2	Video 2	0.867	0.027	0.142
3	Video 3	0.819	0.027	0.144
4	Video 4	0.844	0.027	0.144
5	Video 5	0.863	0.027	0.144
6	Video 6	0.865	0.027	0.145
7	Video 7	0.827	0.027	0.145
8	Video 8	0.859	0.027	0.146
9	Video 9	0.848	0.027	0.144
10	Video 10	0.836	0.027	0.145
11	Video 11	0.876	0.027	0.146
12	Video 12	0.821	0.027	0.146
13	Video 13	0.882	0.027	0.146
14	Video 14	0.843	0.027	0.142
15	Video 15	0.852	0.027	0.144
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
60	0.873	0.027	0.143	0.873
61	0.871	0.027	0.146	0.871
62	0.822	0.027	0.142	0.822

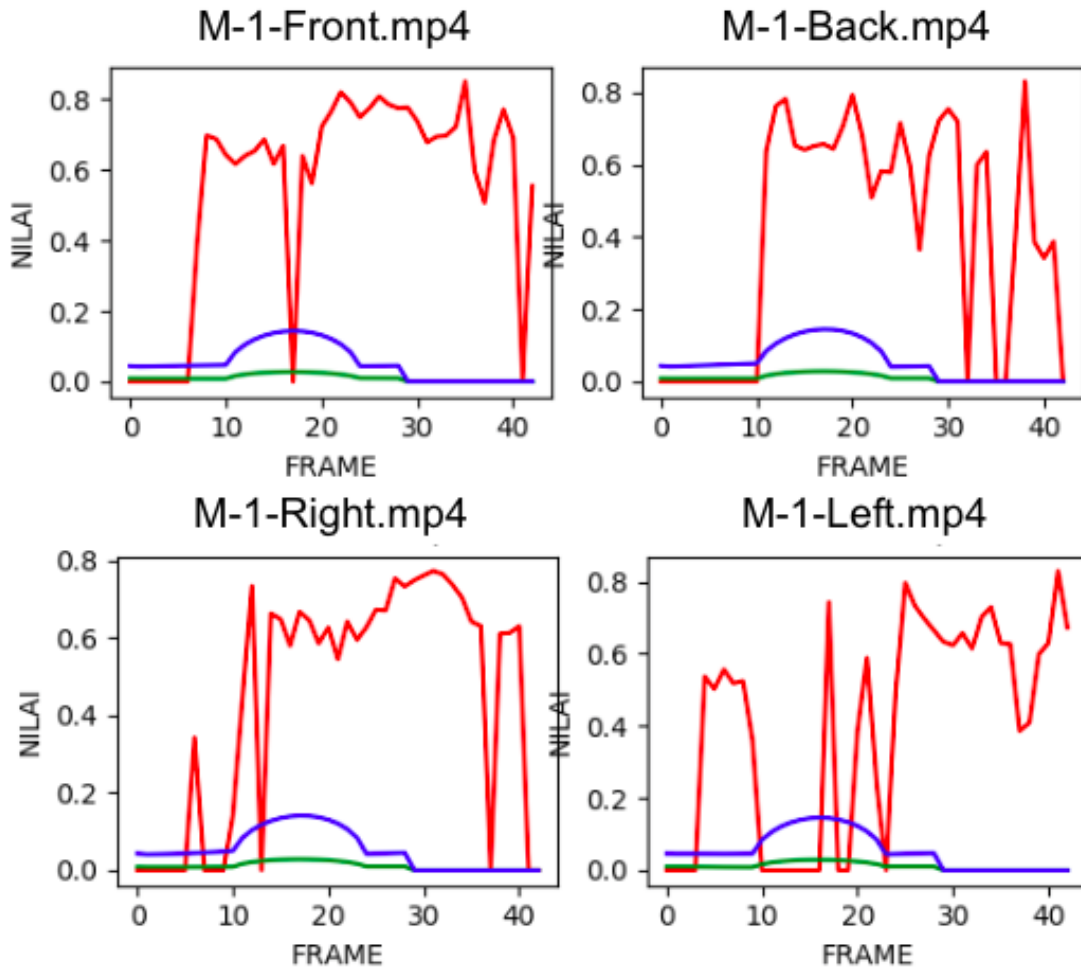


Fig. 6. Graph results of the C_{motion} , Sigma Theta, and Sigma Rho values from hitting movement detection in front, back, right, and left side (Nilai = Value)

If the value of $C_{motion} > 0.8$ and the highest value of σ_{theta} multiplied by 0.9 is less than equal to σ_{rho} , it will be considered to have made a hitting movement, while if the value is otherwise, it will be considered not to make a hitting move. To get the hitting motion detection result, if C_{motion} is more than 0.8 and the highest value of σ_{theta} multiplied by 0.9 is less than equal to σ_{rho} then if the value is appropriate, it will display detected hitting motion, while otherwise, it displays no hitting motion detected. From the test results, as shown in Fig. 6, the value on the y line as a result of the C_{motion} value is colored red while the σ_{theta} and the σ_{rho} are green and blue, while the frames on the x line are the number of frames.

Table 2 is the result of the analysis carried out, there are 62 datasets of stroke movements. From the experimental results, it can be seen that the value of C_{motion} the highest out of 62 videos that contained hitting movements was 0.887 with the smallest value of 0.802. Out of 62 hitting videos scored $C_{motion} > 0.8$ totaling 62 data. From this analysis, it can be concluded that if the C_{motion} value is above 0.8 then in the video it is determined that there is a hitting motion. As for the value of σ_{theta} and σ_{rho} in the hitting movement with a θ value of 0.027 and the highest ρ value of 0.146 while the lowest value of 0.144. Score of C_{motion} represents a value of the object's velocity while the value of σ_{theta} with σ_{rho} is to represent the value of motion in the object's body.

From Table 3, analysis is carried out using 15 data with the composition of 10 videos hitting and 5 videos not hitting. There is an error in the data not making a hit, namely when the object waves both hands simultaneously, and the object is at high speed, the movement causes the parameters to change similar to hitting, so the accuracy of the method used can be calculated as Table 4.

Table 3. Test results of the hitting movement and non-smacking movement video

No	Activity	Cam 1	Cam 2	Cam 3	Cam 4
1	Hit	Yes	Yes	Yes	Yes
2	Hit	Yes	Yes	Yes	Yes
3	Hit	Yes	Yes	Yes	Yes
4	Hit	Yes	Yes	Yes	Yes
5	Hit	Yes	Yes	Yes	Yes
6	Hit	Yes	Yes	Yes	Yes
7	Hit	Yes	Yes	Yes	Yes
8	Hit	Yes	Yes	Yes	Yes
9	Hit	Yes	Yes	Yes	Yes
10	Hit	Yes	Yes	Yes	Yes
11	Not hitting	Yes	Yes	Yes	Yes
12	Not hitting	Yes	Yes	Yes	Yes
13	Not hitting	Yes	Yes	Yes	Yes
14	Not hitting	No	No	No	No

Table 4. Test results to determine the accuracy of video hitting and not hitting movement outcomes

		Current	
		Hitting Movement	Not Hitting Movement
Prediction	Positive	TP (10)	FP (1)
	Negative	FN (0)	TN (4)

Where:

- True Positive (TP) If actual hit and predicted result hit.
- False Positive (FP) If Actual doesn't hit and predicted result hit.
- True Negative (TN) If Actual doesn't hit and predicted result doesn't hit.
- False Negative (FN) If the actual hits and the predicted result doesn't hit.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% = 93\%$$

$$Precision = \frac{TP}{TP + FP} = 0.9$$

3.12. Comparison of the Accuracy of Each Method

In this study, a comparison of several methods was carried out to measure accuracy in gesture recognition. Table 5 shows the results of the accuracy comparison of the various methods used. The RT-PKDT (Recurrent Transformation Prior Knowledge-based Decision Tree) method produced the highest accuracy of 92.68%, although with high computational complexity, making it less suitable for real-time applications. In addition, the DTW (Dynamic Time Warping) and 3D Skeleton Modeling methods obtained accuracies of 91% and 92% respectively, with DTW showing superiority in handling temporal variations. Meanwhile, the method used in this study, a combination of MHI (Motion History Image) and Approximated Ellipses, achieved an accuracy of 93%. While not as superior as RT-PKDT, this method offers an optimal balance between accuracy and processing efficiency, making it a more practical choice for real-time applications.

Table 5. Comparison of the accuracy of each method

No	Method	Accuracy
1	RT-PKDT	92.68%
2	DTW	91%
3	3D Skeleton Modeling	92%
4	MHI + Approximated Ellipse	93%

4. Conclusion

This study acknowledges that the Motion History Image method can detect hitting movements with a C_motion value of more than 0.8. And if the highest σ_{θ} value multiplied by 0.9 is less than the same as σ_{ρ} , then it is considered to be moving. The Motion History Image method using the C_motion parameter can be used to test hitting activities if the hitting activities are in fast and medium speed but if in slow conditions it cannot be detected, so the accuracy of the results of this test is 93% with a precision level of 0.9.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Acknowledgment: Thank you to Ministry of Research and Technology/National Research and Innovation Agency; Ministry of Education and Culture, and LPPM UPN Veteran Jawa Timur, for the research that has been given, so that we can publish papers in several journals.

Conflicts of Interest: The authors declare no conflict of interest.

References

- [1] P. -H. Chiu, P. -H. Tseng and K. -T. Feng, "Interactive Mobile Augmented Reality System for Image and Hand Motion Tracking," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9995-10009, 2018, <https://doi.org/10.1109/TVT.2018.2864893>.
- [2] K. Host and M. Ivašić-Kos, "An overview of Human Action Recognition in sports based on Computer Vision," *Heliyon*, vol. 8, no. 6, p. e09633, 2022, <https://doi.org/10.1016/j.heliyon.2022.e09633>.
- [3] Z. Dong, F. Li, J. Ying and K. Pahlavan, "A Model-Based RF Hand Motion Detection System for Shadowing Scenarios," *IEEE Access*, vol. 8, pp. 115662-115672, 2020, <https://doi.org/10.1109/ACCESS.2020.3004513>.
- [4] L. Zhang *et al.*, "KaraKter: An autonomously interacting Karate Kumite character for VR-based training and research," *Computers & Graphics*, vol. 72, pp. 59-69, 2018, <https://doi.org/10.1016/j.cag.2018.01.008>.
- [5] P. Washington *et al.*, "Activity recognition with moving cameras and few training examples: Applications for detection of autism-related headbanging," *CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, no. 3, pp. 1-7, 2021, <https://doi.org/10.1145/3411763.3451701>.
- [6] W. Pouw, J. P. Trujillo and J. A. Dixon, "The quantification of gesture–speech synchrony: A tutorial and validation of multimodal data acquisition using device-based and video-based motion tracking," *Behavior Research Methods*, vol. 52, pp. 723–740, 2021, <https://doi.org/10.3758/s13428-019-01271-9>.
- [7] T. Menzel and W. Potthast, "Application of a Validated Innovative Smart Wearable for Performance Analysis by Experienced and Non-Experienced Athletes in Boxing," *Sensors*, vol. 21, no. 23, p. 7882, 2021, <https://doi.org/10.3390/s21237882>.
- [8] F. Zhou, X. Li and Z. Wang, "Efficient High Cross-User Recognition Rate Ultrasonic Hand Gesture Recognition System," *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13501-13510, 2020, <https://doi.org/10.1109/JSEN.2020.3004252>.
- [9] X. Pan and, A. F. d.C. Hamilton, "Why and How to Use Virtual Reality to Study Human Social Interaction: The Challenges of Exploring a New Research Landscape," *British Journal of Psychology*, vol. 109, no. 3, pp. 395-417, 2018, <https://doi.org/10.1111/bjop.12290>.
- [10] M. B. frasetyo, E. S. Wahyuni and H. Setiawan, "Comparison of Motion History Image and Approximated Ellipse Method in Human Fall Detection System," *Indonesian Journal of Computing and Cybernetics Systems*, vol. 13, no. 2, pp. 199-208, 2019, <https://doi.org/10.22146/ijccs.43632>.
- [11] S. N. Nwe Htun, T. T. Zin and P. Tin, "Image Processing Technique and Hidden Markov Model for an Elderly Care Monitoring System," *Journal of Imaging*, vol. 6, no. 6, p. 49, 2020, <https://doi.org/10.3390/jimaging6060049>.

-
- [12] A. M. Husein, Calvin, D. Halim, R. Leo, and William, "Motion detect application with frame difference method on a surveillance camera," *Journal of Physics: Conference Series*, vol. 1230, no. 1, p. 012017, 2019, <https://doi.org/10.1088/1742-6596/1230/1/012017>.
- [13] O. Mercanoglu Sincan and H. Y. Keles, "Using Motion History Images With 3D Convolutional Networks in Isolated Sign Language Recognition," *IEEE Access*, vol. 10, pp. 18608-18618, 2022, <https://doi.org/10.1109/ACCESS.2022.3151362>.
- [14] I. G. S. M. Diyasa, A Fauzi, M Idhom and A Setiawan, "Multi-face Recognition for the Detection of Prisoners in Jail using a Modified Cascade Classifier and CNN," *Journal of Physics: Conference Series*, vol. 1844, p. 012005, 2021, <https://doi.org/10.1088/1742-6596/1844/1/012005>.
- [15] E. Sukma Wahyuni, M. Brado Frasetyo, and H. Setiawan, "Combination of motion history image and approximated ellipse method for Human Fall Detection System," *International Journal of Simulation: Systems, Science and Technology*, vol. 19, no. 3, pp. 1-13, 2019, <https://doi.org/10.5013/IJSSST.a.19.03.13>.
- [16] K. Sehairi, F. Chouireb and J. Meunier, "Elderly fall detection system based on multiple shape features and motion analysis," *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1-8, 2018, <https://doi.org/10.1109/ISACV.2018.8354084>.
- [17] L. C. Ngugi, M. Abelwahab, M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition—A review," *Information processing in agriculture*, vol. 8, no. 1, pp. 27-51, <https://doi.org/10.1016/j.inpa.2020.04.004>.
- [18] Z. Liu, M. Yang, Y. Yuan and K. Y. Chan, "Fall Detection and Personnel Tracking System Using Infrared Array Sensors," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9558-9566, 2020, <https://doi.org/10.1109/JSEN.2020.2988070>.
- [19] S. Z. Gurbuz and M. G. Amin, "Radar-Based Human-Motion Recognition with Deep Learning: Promising Applications for Indoor Monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16-28, 2019, <https://doi.org/10.1109/MSP.2018.2890128>.
- [20] N. Lapierre, A. St-Arnaud, J. Meunier, and J. Rousseau, "Implementing an intelligent video monitoring system to Detect Falls of older adults at home: A multiple case study," *Journal of Enabling Technologies*, vol. 14, no. 4, pp. 253–271, 2020, <https://doi.org/10.1108/JET-03-2020-0012>.
- [21] C. Xu, J. He, X. Zhang, "Hierarchical Decision Tree Model for Human Activity Recognition Using Wearable Sensor," *Advances in Intelligent Systems and Interactive Applications, Advances in Intelligent Systems and Computing*, vol. 686, pp. 368-372, 2019, https://doi.org/10.1007/978-3-319-69096-4_51.
- [22] T. Mahalingam and M. Subramoniam, "A robust single and multiple moving object detection, tracking and classification," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 2–18, 2021, <https://doi.org/10.1016/j.aci.2018.01.001>.
- [23] A. Srivastava, T. Badal, A. Garg, A. Vidyarthi, and R. Singh, "Recognizing human violent action using drone surveillance within real-time proximity," *Journal of Real-Time Image Processing*, vol. 18, no. 5, pp. 1851–1863, 2021, <https://doi.org/10.1007/s11554-021-01171-2>.
- [24] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca and F. Brémond, "Self-Supervised Video Pose Representation Learning for Occlusion- Robust Action Recognition," *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pp. 1-5, 2021, <https://doi.org/10.1109/FG52635.2021.9667032>.
- [25] H. El-Ghaish, M. E. Hussien, A. Shoukry and R. Onai, "Human Action Recognition Based on Integrating Body Pose, Part Shape, and Motion," *IEEE Access*, vol. 6, pp. 49040-49055, 2018, <https://doi.org/10.1109/ACCESS.2018.2868319>.
- [26] I. G. S. M. Diyasa, E. Y. Puspaningrum, M. Hatta and A. Setiawan, "New Method For Classification Of Spermatozoa Morphology Abnormalities Based On Macroscopic Video Of Human Semen," *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 133-140, 2019, <https://doi.org/10.1109/ISEMANTIC.2019.8884348>.
- [27] H. Zhang *et al.*, "Deep Multimodel Cascade Method Based on CNN and Random Forest for Pharmaceutical Particle Detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 7028-7042, 2020, <https://doi.org/10.1109/TIM.2020.2973843>.
-

-
- [28] K. A. Lambert, "Fundamentals of Python: First Programs," *Cengage Learning*, 2019, <https://ggnindia.dronacharya.info/Downloads/Sub-info/RelatedBook/PYTHON-TEXT-BOOK-1.pdf>.
- [29] B. O. Sadiq, H. Bello-Salau, L. Abduraheem-Olaniyi, B. Muhammad and S. O. Zakariyya, "Towards Enhancing Keyframe Extraction Strategy for Summarizing Surveillance Video: An Implementation Study," *Journal of ICT Research and Applications*, vol. 16, no. 2, pp. 167-183, 2022, <https://doi.org/10.5614/itbj.ict.res.appl.2022.16.2.5>.
- [30] H. R. Sinulingga and S. G. Kong, "Key-Frame Extraction for Reducing Human Effort in Object Detection Training for Video Surveillance," *Electronics*, vol. 12, no. 13, p. 2956, 2023, <https://doi.org/10.3390/electronics12132956>.
- [31] F. Pan *et al.*, "Accuracy of RGB-D camera-based and stereophotogrammetric facial scanners: a comparative study," *Journal of Dentistry*, vol. 127, p. 104302, 2022, <https://doi.org/10.1016/j.jdent.2022.104302>.
- [32] S. A. H. Alrubaie and A. H. Hameed, "Dynamic Weights Equations for Converting Grayscale Image to RGB Image," *Journal of University of Babylon for Pure and Applied Sciences*, vol. 26, no. 8, pp. 122-129, 2018, <https://doi.org/10.29196/jubpas.v26i8.1677>.
- [33] S. Ali *et al.*, "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," *Information Fusion*, vol. 99, p. 101805, 2023, <https://doi.org/10.1016/j.inffus.2023.101805>.
- [34] M. S. Zaharin, N. Ibrahim, T. M. A. T. Dir, "Comparison of human detection using background subtraction and frame difference," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 345-353, 2020, <https://doi.org/10.11591/eei.v9i1.1458>.
- [35] A. J. Sathyamoorthy *et al.*, "VERN: Vegetation-Aware Robot Navigation in Dense Unstructured Outdoor Environments," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11233-11240, 2023, <https://doi.org/10.1109/IROS55552.2023.10342393>.
- [36] S. Nurdiani, M. Rezki, R. Dahlia, M. I. R. Ihsan, Frieyadi and S. Fauziah, "Comparison Of Apple Image Segmentation Using Binary Conversion And K-Means Clustering Methods," *PILAR Nusa Mandiri Journal*, vol. 17, no. 1, pp. 99-103, 2021, <https://doi.org/10.33480/pilar.v17i1.2256>.
- [37] M. K. Khinn Teng, H. Zhang and T. Saitoh, "LGNMNet-RF: Micro-Expression Detection Using Motion History Images," *Algorithms*, vol. 17, no. 11, p. 491, 2024, <https://doi.org/10.3390/a17110491>.
- [38] A. J. Sathyamoorthy *et al.*, "VERN: Vegetation-Aware Robot Navigation in Dense Unstructured Outdoor Environments," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11233-11240, 2023, <https://doi.org/10.1109/IROS55552.2023.10342393>.
- [39] A. Alavigharabagh, V. Hajihashemi, J. J. M. Machado and J. M. R. S. Tavares, "Deep Learning Approach for Human Action Recognition Using a Time Saliency Map Based on Motion Features Considering Camera Movement and Shot in Video Image Sequences," *Information*, vol. 14, no. 11, p. 616, 2023, <https://doi.org/10.3390/info14110616>.
- [40] O. Mercanoglu Sincan and H. Y. Keles, "Using Motion History Images With 3D Convolutional Networks in Isolated Sign Language Recognition," *IEEE Access*, vol. 10, pp. 18608-18618, 2022, <https://doi.org/10.1109/ACCESS.2022.3151362>.
- [41] K. K. Verma and B. M. Singh, "Deep Multi-Model Fusion for Human Activity Recognition Using Evolutionary Algorithms," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 2, pp. 44-58, 2021, <https://doi.org/10.9781/ijimai.2021.08.008>.
- [42] G. L. Sravanthi, M. V. Devi, K. S. Sandeep, A. Naresh, A. P. Gopi, "An Efficient Classifier using Machine Learning Technique for Individual Action Identification," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 513-520, 2020, <https://dx.doi.org/10.14569/IJACSA.2020.0110664>.
- [43] J. Cao and Y. Tanjo, "High-Accuracy Human Motion Recognition Independent of Motion Direction Using A Single Camera," *International Journal of Innovative Computing, Information and Control ICIC International*, vol. 20, no. 4, pp. 1093-1103, 2024, <http://www.ijicic.org/ijicic-200408.pdf>.
-

-
- [44] J. Yang *et al.*, "Reinventing 2D Convolutions for 3D Images," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 3009-3018, 2021, <https://doi.org/10.1109/JBHI.2021.3049452>.
- [45] S. Bakheet and A. Al-Hamadi, "A deep neural framework for real-time vehicular accident detection based on motion temporal templates," *Heliyon*, vol. 8, no. 1, p. e11397, 2022, <https://doi.org/10.1016/j.heliyon.2022.e11397>.
- [46] J. S. Fowdur, M. Baum and F. Heymann, "An Elliptical Principal Axes-based Model for Extended Target Tracking with Marine Radar Data," *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, pp. 1-8, 2021, <https://doi.org/10.23919/FUSION49465.2021.9627039>.
- [47] M. Hu, H. Wang, X. Wang, J. Yang, R. Wang, "Video facial emotion recognition based on local enhanced motion history image and CNN-CTSLSTM networks," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 176-185, 2019, <https://doi.org/10.1016/j.jvcir.2018.12.039>.
- [48] A. Akdağ and Ö. K. Baykan, "Isolated sign language recognition through integrating pose data and motion history images," *PeerJ Computer Science*, vol. 10, p. e2054, 2024, <https://doi.org/10.7717/peerj-cs.2054>.
- [49] F. Hajabdollahi, K. N. Premnath, S. W. J. Welch, "Central moment lattice Boltzmann method using a pressure-based formulation for multiphase flows at high density ratios and including effects of surface tension and Marangoni stresses," *Journal of Computational Physics*, vol. 425, p. 109893, 2021, <https://doi.org/10.1016/j.jcp.2020.109893>.
- [50] P. A. Harris, L. Scognamiglio, F. Magnoni, E. Casarotti, E. Tinti, "Centroid moment tensor catalog with 3D lithospheric wave speed model: The 2016–2017 Central Apennines sequence," *Journal of Geophysical Research: Solid Earth*, vol. 126, no. 23, pp. 1–16, 2022, <https://doi.org/10.1002/essoar.10507862.1>.
- [51] X. Zhou, Q. Chen, S. Lyu and H. Chen, "Ellipse Inversion Model for Estimating the Orientation and Radius of Pipes From GPR Image," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8299-8308, 2022, <https://doi.org/10.1109/JSTARS.2022.3205889>.
- [52] K. Brown, H. Mathur, "Modified Newtonian dynamics as an alternative to the Planet Nine hypothesis," *The Astronomical Journal*, vol. 166, no. 4, pp. 168, 2023, <https://doi.org/10.3847/1538-3881/acef1e>.
- [53] H. Hangan, D. Romanic, C. Jubayer, "Three-dimensional, non-stationary and non-Gaussian (3D-NS-NG) wind fields and their implications to wind–structure interaction problems," *Journal of Fluids and Structures*, vol. 91, p. 102583, 2019, <https://doi.org/10.1016/j.jfluidstructs.2019.01.024>.
- [54] X. Zhang, L. Duan, Q. Gong, Y. Wang, H. Song, "State of charge estimation for lithium-ion battery based on adaptive extended Kalman filter with improved residual covariance matrix estimator," *Journal of Power Sources*, vol. 589, p. 233758, 2024, <https://doi.org/10.1016/j.jpowsour.2023.233758>.
- [55] Z. Yang, L. Fang, B. Shen and T. Liu, "PolSAR Ship Detection Based on Azimuth Sublook Polarimetric Covariance Matrix," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8506-8518, 2022, <https://doi.org/10.1109/JSTARS.2022.3211431>.
- [56] A. Barthelme and W. Utschick, "DoA Estimation Using Neural Network-Based Covariance Matrix Reconstruction," *IEEE Signal Processing Letters*, vol. 28, pp. 783-787, 2021, <https://doi.org/10.1109/LSP.2021.3072564>.
- [57] C. Gartner, J. K. Mathew, J. Desai, J. Sturdevant, E. D. Cox and D. Bullock, "Methodology for Georeferencing Roadside Pan-Tilt-Zoom CCTV Camera Views to Highway GPS Coordinates," *IEEE Access*, vol. 12, pp. 158140-158149, 2024, <https://doi.org/10.1109/ACCESS.2024.3487048>.
- [58] I. Ding, N. Zheng, "CNN deep learning with wavelet image fusion of CCD RGB-IR and depth-grayscale sensor data for hand gesture intention recognition," *Sensors*, vol. 22, no. 3, p. 803, 2022, <https://doi.org/10.3390/s22030803>.
- [59] K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," *2021 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pp. 1-6, 2021, <https://doi.org/10.1109/ICITIIT51526.2021.9399594>.
-

-
- [60] H. Komori, M. Isogawa, D. Mikami, T. Nagai, Y. Aoki, "Time-weighted motion history image for human activity classification in sports," *Sports Engineering*, vol. 26, no. 45, 2023, <https://doi.org/10.1007/s12283-023-00437-1>.
- [61] M. F. Amin, "Confusion matrix in binary classification problems: A step-by-step tutorial," *Journal of Engineering Research (ERJ)*, vol. 6, no. 5, pp. 1–12, 2022, <https://digitalcommons.aaru.edu.jo/cgi/viewcontent.cgi?article=1278&context=erjeng>.
- [62] R. Kosasih, A. Fahrurrozi, D. Riminarsih, "Implementation of random forest on face recognition using Isomap features," *Journal of Computing Engineering, System and Science*, vol. 7, no. 2, pp. 459–469, 2022, <https://doi.org/10.24114/cess.v7i2.34498>.