New Optimization Technique to Extract Facial Features

Erwin, *Member, IAENG*, Saparudin, *Member, IAENG*, Muhammad Fachrurrozi, *Member, IAENG*,
Arief Wijaya, Muhammad Naufal Rachmatullah

***Abstract*— The improved harmony search algorithm (IHSA), harmony search algorithm (HSA) and genetic algorithm (GA) are applied to extract facial features based on edges in the horizontal direction. The study used 100 images from 20 different faces from the California Institute of Technology dataset. The results show that IHSA is better in terms of accuracy and speed compared to HSA and GA, with average accuracy of 77.055% in an average time of 0.077 seconds compared to HSA at 75.176% in an average of 0.105 seconds and GA at 74.931% in an average of 3.627 seconds.**

***Index Terms*— facial-feature extraction, genetic algorithm, harmony search algorithm, improved harmony search algorithm**

I. Introduction

Facial-feature extraction is important to applications such as human face recognition, surveillance systems and video conferencing [1]. The main challenge of facial-recognition systems is to extract features, including the eyes, eyebrows, nose and mouth from the detected face region. In the positioning method, facial features are divided into two parts, with texture corresponding to the pixel value and geometry corresponding to prior information [2]-[3].

Much research has concerned the extraction of facial features using the Active Shape Model [4], Discrete Cosine Transform [5], Principle Component Analysis [6], Support Vector Machine [7], Neural Network [8] and Conditional Regression Forests [9]. These methods provide variable accuracy depending on various aspects of the image, such as background complexity, image resolution, light, head pose, facial accessories and facial expressions. Variable accuracy is one reason that the extraction of facial features is still an interesting research topic. A robust algorithm under various conditions is still certainly required to improve performance compared to those applied in previous studies.

One study used the Support Vector Machine to detect facial features [10], but this method only detects eyes. Another method and more time is required to detect more facial features. [7] reported using a Support Vector Machine to extract facial features, utilizing the image resolution to learn and find facial characteristics, with performed tests yielding 95% accuracy. Another study of facial-feature extraction [9] used a Conditional Regression Forest, with machine learning producing better accuracy in the detection of facial features.

In other side, Luo et al. [11] proposed a method for extracting facial features that combines Principal Component Analysis with the Local Binary Pattern method. However, this produces overall facial characteristics, not distinctive features such as eyes, nose or mouth. Thus, in certain cases, distinction between faces is challenging.

In [12], feature extraction was performed using a separate morphological operation and a rule-based method. Templates also specify the region of features to be searched. This method performs sequential searches that can slow down the computation process.

The present study likewise extracts features using a template that starts from the edges found by the segmentation process to obtain characteristic facial features. Templates work by comparing input images, so each facial feature can have different templates [12]. Features can be extracted by using templates as an objective function of the optimization algorithm. The harmony search algorithm (HSA), a modified optimization algorithm, has proven to be better than genetic algorithms (GA) in various cases, such as visual tracking systems, math, image compression, feature selection and other optimization problems [13]–[15].

A similar study in [16] compared two methods, namely HSA and principal component analysis (PCA), to optimize facial recognition by improving the extraction of facial features. The results showed that the features obtained using HSA produce better recognition accuracy than those obtained using PCA.

Therefore, the present study uses HSA, modified [13] to find the location of facial features in image coordinates, which represent optimization problems measured by objective functions based on a bounding box of each feature. The purpose of this study is to prove that the modified HSA improves on traditional HSA and GA for feature extraction in terms of accuracy, computation time, stability, and convergence rate.

Manuscript received July 06, 2017; revised August 25, 2018.

Erwin is with the Department of Computer Engineering, Faculty of Computer Science, Sriwijaya University, Indralaya – Ogan Ilir 30662, Indonesia (e-mail: erwin@unsri.ac.id).

Saparudin, M. Fachrurrozi, A. Wijaya, and M. Naufal Rachmatullah are with the Department of Informatic Engineering, Faculty of Computer Science, Sriwijaya University, Indralaya - Ogan Ilir 30662, Indonesia (e-mail: saparudinmasyarif@gmail.com; obetsobets@gmail.com; naufalrachmatullah@gmail.com; ariefwiijaya@gmail.com).

II. Facial Feature Extraction

In human face-recognition systems, the feature extraction process is performed after face detection. This section discusses the extraction of features found in the facial image.

A. Face Detection

Detecting a face in an image can be approached as an ellipsoid-detection problem because of the generally assumed shape of the human face. This method can solve images with complex backgrounds, with unnecessary body parts, like hands or hairs, or with tilted heads. An elliptical search can also be performed using an optimization algorithm for five-parameter problems, using the following five parameters: the midpoint of the ellipse (x and y), radius x (R_x), radius y (R_y) and the degree of tilt (θ). The ratio of the resulting face size is 1:1.5, and the size of the face image used for feature extraction is 200 × 300 pixels.

B. Face Segmentation

The optimization algorithm performs facial feature searches based on the edge thickness resulting from the horizontal Sobel operator (see Figure 1). The input is converted to a grey image, sent through a noise-reduction process to produce high-quality edges with minimal noise, and converted into a binary image to produce a segmented image. Using horizontal edges can eliminate features that are unrelated to the face so that facial characteristics can be divided.

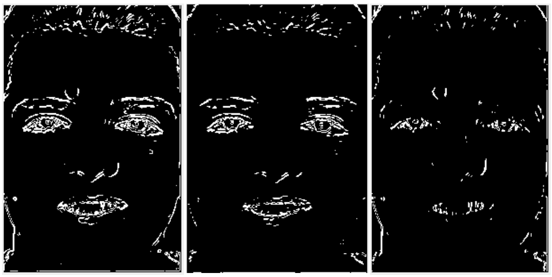


Fig. 1. Segmented face based on edge direction. (left) horizontal and vertical, (middle) horizontal, (right) vertical.

C. Determining Facial Feature Region

The nose region (R_n) is determined after the right eye (R_r), left eye (R_l), and mouth (R_m) have been found using optimization algorithms. Therefore, the search process is performed four times to obtain every region. The facial feature area is determined based on the feature template shown in Figure 2, with a face-size ratio of 1:1.5. The facial features are divided into the right eye (R_r), left eye (R_l), nose (R_n), and mouth (R_m) regions, each bounded by four coordinates x1, x2, y1, and y2.

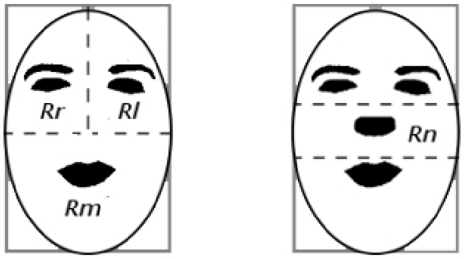


Fig. 2. Template for facial feature region

The facial features are determined as follows.

1. Coordinates of the left-eye region (Figure 3) can be formulated as in equations (1) – (4):

$$R_l x1 = \frac{width}{2} \tag{1}$$
$$R_l x2 = width \tag{2}$$
$$R_l y1 = 0 \tag{3}$$
$$R_l y2 = \frac{height}{2} \tag{4}$$

2. Coordinates of the right-eye region (Figure 3) can be formulated as in Equations (5) – (8):

$$R_r x1 = 0 \tag{5}$$
$$R_r x2 = \frac{width}{2} \tag{6}$$
$$R_r y1 = 0 \tag{7}$$
$$R_r y2 = \frac{height}{2} \tag{8}$$

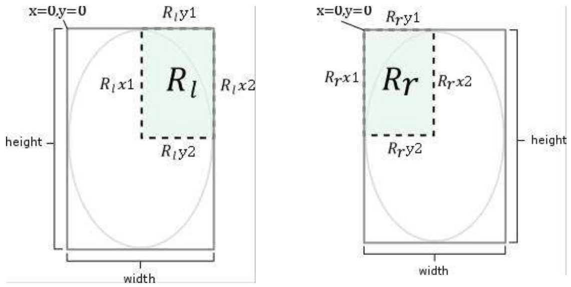


Fig.3. Left and right eye regions.

3. Coordinates of the mouth region (Figure 4) can be formulated as in Equations (9) – (12):

$$R_m x1 = 0 \tag{9}$$
$$R_m x2 = width \tag{10}$$
$$R_m y1 = \frac{height}{2} \tag{11}$$
$$R_m y2 = height \tag{12}$$

4. Coordinates of the nose region (Figure 4) can be formulated as in Equations (13)–(17):

$$R_n x1 = 0 \tag{13}$$
$$R_n x2 = width \tag{14}$$
$$R_n y1 = LE_{y2}\ if\ LE_{y2} > RE_{y2} \tag{15}$$
$$R_n y1 = RE_{y2}\ if\ RE_{y2} > LE_{y2} \tag{16}$$
$$R_n y2 = M_{y1} \tag{17}$$

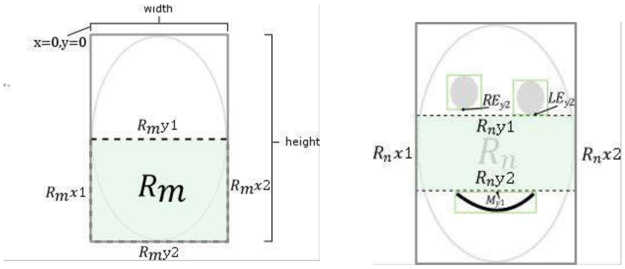


Fig. 4. Mouth and nose regions.

D. *Matching Facial Feature*

Searching for each feature region requires a template or bounding box. The size of the bounding box varies for each feature, set to fit the size of typical facial features: eyes measuring 1.6 : 1, nose 3 : 1, and mouth 2.125 : 1. Since the facial image is sized 200 × 300 pixels, the eyes are 80 × 50 pixels, nose 60 × 20 pixels, and mouth 85 × 40 pixels. The bounding box is just an initial assumption that will be revised to the most appropriate size for each feature.

Some coordinates in the feature regions are matched using a bounding box, calculating the number of edges that are on the bounding box. The more edges that are located at the coordinates of the bounding box, the more the box is considered to be characteristic of the face in the region. The level of facial features is measured using Equation (18).

$$f(x,y) = \frac{1}{h\ x\ w}\sum_{i=x}^{h}\sum_{j=y}^{w} T(i,j) \tag{18}$$

The height of the bounding box is symbolized by h, and w is the width. $f(x,y)$ is a coordinate at the top left of a *bounding box*. $T(i,j)$ is a binary value, with 1 representing white pixels and 0 black at coordinate (i,j). The black pixel (or the zero value) is not included in the calculation to accelerate computation.

III. Improved Harmony Search Algorithm

The Improved Harmony Search Algorithm (IHSA) is a modification of the HSA that improves its accuracy and rate of convergence in generating vectors [13]. HSA is a metaheuristic algorithm inspired by musicians who search for the best harmony by continuing to play a tone based on experience or randomly until they obtain beautiful music [17].

The difference between the IHSA and HSA lies in the value of the Pitch Adjustment Rate (PAR) and Bandwidth (BW) for each iteration. In the HSA, these values are fixed, requiring a higher number of iterations to find the optimal solution [13]. The IHSA adds four new parameters—PAR_{min}, PAR_{max}, BW_{min}, and BW_{max}, , —to obtain the values for PAR and BW dynamically according to the number of generations, using Equation (19) as follows:

$$PAR(gn) = PAR_{min} + \frac{(PAR_{max}-PAR_{min})}{NI} x\ gn \tag{19}$$

$PAR(gn)$ is a new *pitch-adjustment rate* in every generation. The range for PAR is specified using PAR_{min} and PAR_{max}. NI is the number of solution vectors, and gn is the generation. $BW(gn)$ values are obtained using the following Equations (20) and (21):

$$BW(gn) = BW_{max}\exp(c.gn) \tag{20}$$

$$c = \frac{Ln(\frac{BW_{min}}{BW_{max}})}{NI} \tag{21}$$

where $BW(gn)$ is a bandwidth in each generation, and the range of PAR is specified using BW_{min} and BW_{max}.

In the optimization process, the IHSA algorithm has several stages, as follows.

1) *Initializing parameters of IHSA*

This stage initializes the parameters and the required optimization problems, beginning with determining the objective function, the number of decision variables and their values and required parameters, namely, the number of Harmony Memory (HM), HM Consideration Rate (HMCR), PAR_{min}, PAR_{max}, BW_{min} , and BW_{max}.

2) *Initialization of Harmony Memory*

Harmony Memory (HM) is the initial solution vector N obtained randomly, then sorted based on the value of the objective function.

3) *Improvisation of new harmony*

At this stage, the value of the PAR and the bandwidth are adjusted using equations (19) and (20) for each iteration. HM is also adjusted to meet the predetermined rules, and HMCR and PAR are compared. A new vector solution is then obtained.

4) *HM update*

The new harmony is checked to determine if it is better than the previous minimum harmony. If so, the harmony is inserted in the HM.

5) *Termination criterion*

Return to step 3 if the termination criterion has not been met; stop when it has been reached.

After the optimization process is complete, the HM with the most optimal value of the objective function (maximum or minimum) is the selected solution.

IV. Facial Feature Extraction Using the IHSA

Figure 5 shows a flow diagram of the overall process, and the Figure 6 represents the architecture of the extraction process for each feature using IHSA.

z

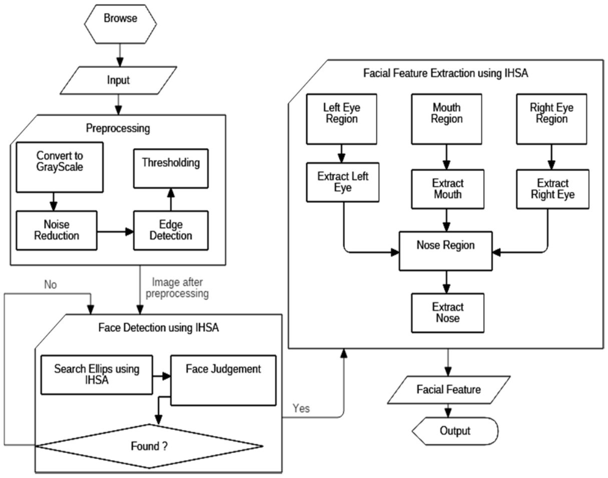


Fig. 5. Overall face detection and facial-feature extraction process.

A. *Solution vector*

The solution vector represents harmony in the IHSA. The number of decision variables in facial-feature extraction of vector solution comprises two variables, x and y.

B. Range of decision variable

Each value of the decision variables is limited by PVB_{lower} and PVB_{upper} for each facial feature, as shown in the following equations.

$$PVB_{lower}(x) = x1 \tag{22}$$
$$PVB_{upper}(x) = x2 - W_{bb} \tag{23}$$
$$PVB_{lower}(y) = y1 \tag{24}$$
$$PVB_{upper}(y) = y2 - H_{bb} \tag{25}$$

$x1, y1$ is the minimum value of x, y coordinates in the feature region and $x2, y2$ are the maximum value of x, y coordinates in the feature region. Meanwhile, W_{bb} is the width of a bounding box and the H_{bb} is the height of a bounding box.

C. Objective function

The value of the objective function or fitness harmony is calculated using Equation (17) for feature extraction. The harmony with the highest fitness value after improvisation has finished is selected as the preferred solution.

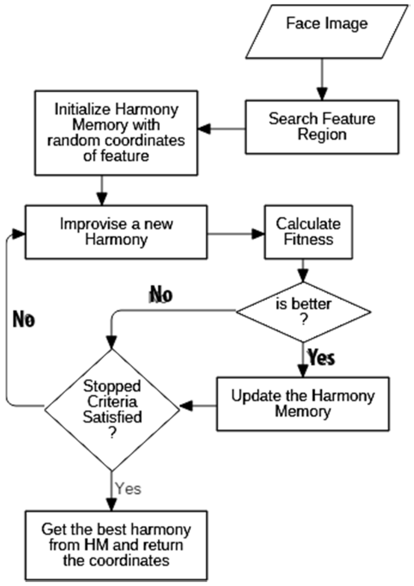

Face Image
Search Feature Region
Initialize Harmony Memory with random coordinates of feature
Improvise a new Harmony
Calculate Fitness
is better ?
No
Yes
No
Stopped Criteria Satisfied ?
Update the Harmony Memory
Yes
Get the best harmony from HM and return the coordinates

Fig. 6. Facial-feature extraction procedure using IHSA.

V. Experimental Results and Discussion

The used test images are 200 × 300 pixels. As many as 100 images from 20 different people with different backgrounds, expressions and lighting conditions were obtained from the dataset of California Institute of Technology [18]. Edges are detected using Sobel with a 3 × 3 kernel, and noise reduction uses a median filter with a kernel of the same size as Sobel and one iteration. Segmentation is obtained using a threshold of 90. For this test, IHSA algorithmic parameters on feature extraction are listed in Table I, determined based on parameters generated in previous research [13], [14].

TABLE I
Parameters of Improved Harmony Search Algorithm

Parameter	Value
NI (Number of Improvisations)	100
HMS (Harmony Memory Size)	7
HMCR (HM Consideration Rate)	0.95
PAR Minimum (Pitch Adjustment Rate)	0.35
PAR Maximum (Pitch Adjustment Rate)	0.99
Bw Minimum (Bandwidth)	1
Bw Maximum (Bandwidth)	12

Tests were performed using IHSA, HSA, and GA, identifying the accuracy percentage and computation time for each algorithm and investigating the effect of modifying HSA parameters and number of iterations for HSA and IHSA.

Various experiments were conducted on all test data to measure IHSA's level of accuracy and computational time and to compare the method with HSA and GA.[1] Figure 7 compares the accuracy of the algorithm based on trendline, and Figure 8 compares computation time. The highest average accuracy of all test data is 77.055%, obtained by IHSA; GA is 75.176% and HSA is 74.931%.

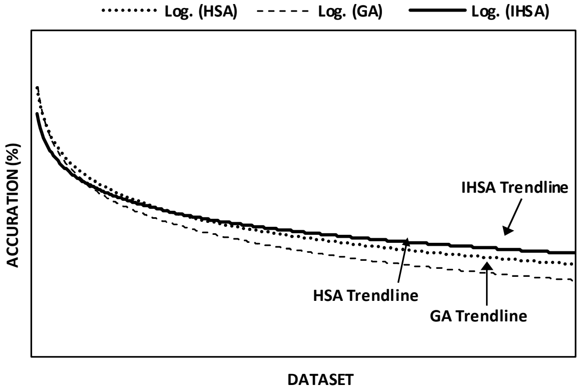


Fig. 7. Accuracy Comparison between IHSA, HSA and GA.

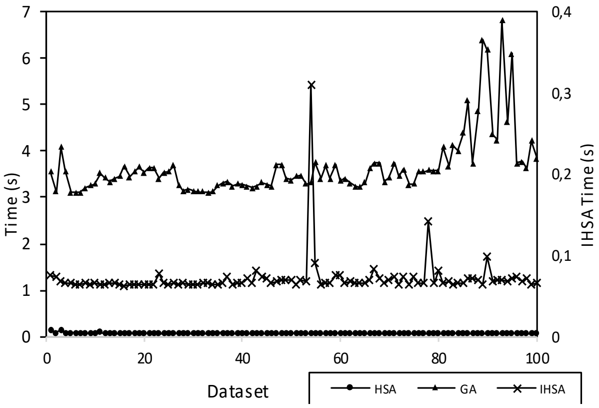


Fig. 8. Speed Comparison between IHSA, HSA and GA.

[1] Computer environment: Intel Pentium(R) Dual-Core CPU E5400 @ 2.70 GHz, 3.00 GB; Programming Language: Java; Software: Netbeans IDE 8.2.

Each algorithm is close in accuracy, but not in speed. IHSA is fastest at 0.077 seconds, compared 0.105 seconds for HSA and the much-slower GA at 3.627 seconds on average. Besides being slower, GA also takes unstable amounts of computation time ranging from 3.0 to 4.5 seconds, which makes GA less precise when applied to real-time systems.

Stability of the accuracy achieved by each algorithm was tested by repeating the experiment 50 times using the same test data. Because the optimization algorithm is probabilistic, the resulting accuracy is not exactly the same in each experiment. Figure 9 shows the trendline stability of accuracy. IHSA appears to more stably maintain accuracy than either HSA or GA. IHSA achieved the highest accuracy on the 45th iteration, with an average accuracy of 86.703%. HSA has an average of 85.564% and GA has 84.864% accuracy.

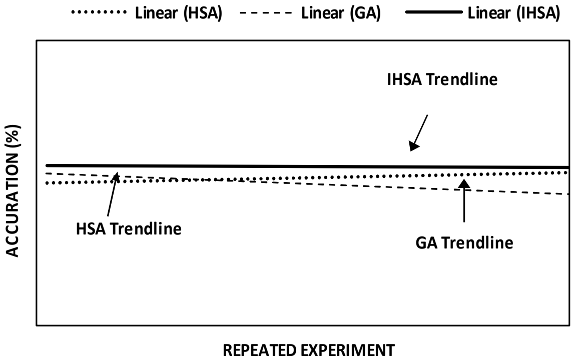


Fig. 9. Comparison of Stability between IHSA, HSA and GA.

In some experiments, IHSA could find facial features with fairly good accuracy over 80% after just a few iterations. The left and right eyes were found as early as the 17th iteration, the nose was found in the 20th iteration and the mouth was found starting from the 10th iteration. Convergence testing was performed based on accuracy, with results in Figure 10 showing the average convergence rates of IHSA, HSA and GA on each iteration. It appears that when using the same amount of memory (called Harmony Memory in HSA and Population in GA), IHSA converges faster than either HSA or GA. Compared to IHSA, HSA converges only as fast at the beginning before slowing down; IHSA has a rapidly increasing rate of convergence. IHSA requires small amounts of memory and time to obtain high accuracy. Meanwhile, GA convergence slows down and takes longer. This shows that the use of PAR and BW very effectively increases the rate of IHSA convergence for detecting facial features.

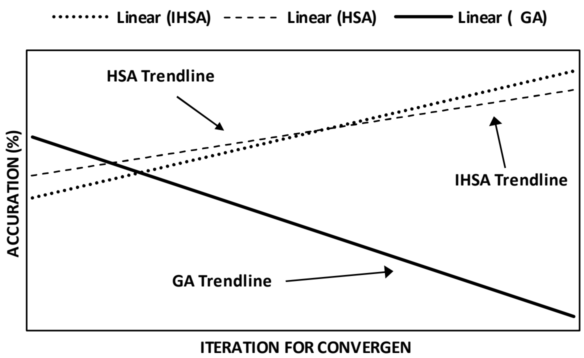
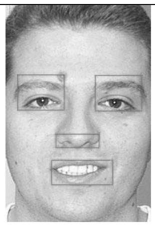
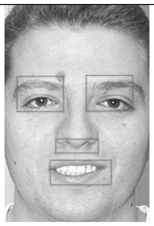
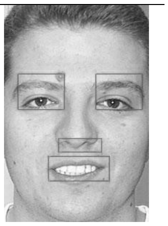
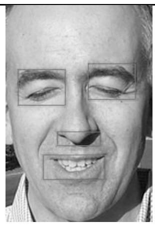
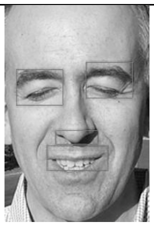
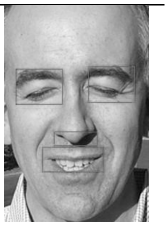
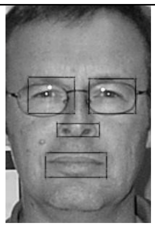
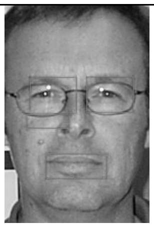
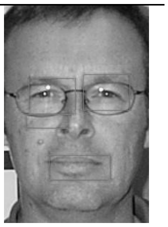


Fig. 10. Comparison of Convergence Rate between IHSA, HSA and GA.

Table II compares the results of the detection and extracted features, showing that IHSA has been successfully applied to detect and extract facial features with better performance than HSA and GA.

TABLE II
PERFORMANCE COMPARISON OF FACIAL-FEATURE EXTRACTION FROM DIFFERENT FACES AND UNDER DIFFERENT CONDITIONS

No	IHSA	HSA	GA
1			
2			
3			

Details of accuracy and speed, along with averages of the six trials, are provided in Table III; in addition to excelling in terms of accuracy, IHSA also outperformed HSA and GA in terms of speed.

TABLE III
ACCURACY RATE AND EXECUTION TIME FROM TABLE III

No	Accuracy (%)			Execution Time (s)		
	IHSA	HSA	GA	IHSA	HSA	GA
1	94.292	83.115	83.936	0.067	0.107	3.268
2	87.551	84.198	80.974	0.082	0.111	3.265
3	97.901	78.815	78.262	0.066	0.093	3.211
4	83.047	79.534	81.823	0.068	0.094	3.221
5	91.921	49.871	49.828	0.069	0.092	3.347
6	88.359	83.439	88.143	0.067	0.097	3.386
Avg.	90.512	76.495	77.161	0.07	0.099	3.283

Accuracy in detecting and extracting features actually is greatly impact by the results of facial segmentation. In this case, the detected edges can minimize errors in the obtained facial features. As shown in Figure 11, some facial features are not extracted properly because of poor lighting and detected edges of unnecessary objects. Even some edges of facial features are not well segmented, leading to errors in determining the characteristics.

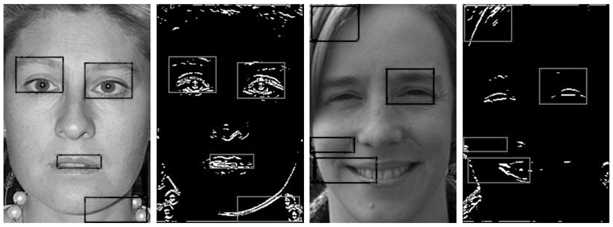


Fig. 11. Wrong Facial Features Extracted

VI. Analysis and Evaluation

The accuracies obtained in experiments with each algorithm show that IHSA is superior to HSA and GA, because the PAR and BW parameters for IHSA are dynamic. PAR parameters provide the possibility to find new solutions, while BW provides direction for the searching solution, here the coordinate feature direction stored in harmony memory. Both allow the distance from a possible solution to be usefully estimated and help IHSA search every possible solution thoroughly without being trapped by local optima. The presence of these parameters improves IHSA performance beyond that of both HSA and GA. Figure 12 presents a process visualization of the PAR and BW parameters.

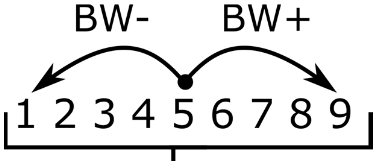


Fig. 12. Visualization of PAR and BW processes on harmony memory

HSA has lower accuracy than IHSA because its PAR and BW parameters are fixed and unchanging, which can get searches stuck on local optima. GA yields the lowest accuracy compared to IHSA and HSA due to the nature of its search method, in which the search is affected by the first randomly generated population. Crossover operations of poor parent chromosomes will retain at least some poor genes, even after mutations, which also causes the search to get stuck on local optima.

In this research, the solution vector used is represented by the following equation:

$$S_i = [x_i \ \ y_i] \tag{26}$$

Where S is a solution vector, i is the solution vector index of the memory harmony, and *x* and *y* comprise the solution vector element called the decision variable. Then, each element of the solution vector is initialized with the image coordinates between $\text{PVB}_{\text{lower}}$ to $\text{PVB}_{\text{upper}}$ from Equations (22) to (25).

$$x_i = rand[PVB_{lower}(x), PVB_{upper}(x)] \tag{27}$$
$$y_i = rand[PVB_{lower}(y), PVB_{upper}(y)] \tag{28}$$

Each solution vector with an initial value directly calculates its fitness value using Equation (18), obtaining the harmony memory as follows:

$$HM = \left[\begin{array}{cc|c} x_0 & y_0 & f(x,y)_0 \\ x_0 & y_0 & f(x,y)_1 \\ .... & .... & .... \\ x_{HMS-2} & y_{HMS-2} & f(x,y)_{HMS-2} \\ x_{HMS-1} & y_{HMS-1} & f(x,y)_{HMS-1} \end{array}\right] \tag{29}$$

HM is a matrix of harmony memory comprising solution vectors, HMS is the harmony memory size, and *f (x, y)* is the fitness value of the solution vector.

PAR and BW values for HSA are initialized first and do not change every generation or iteration, requiring more iteration to reach the optimal solution. According to [13], small PAR values with large BW can lead to poor algorithmic performance, and the number of iterations required increases when looking for optimal solutions. By contrast, large PAR values with small BW can improve the best solutions once converged at the end to the optimum solution vector. In other words, at the beginning, the BW should be greater to force the algorithm to increase the diversity of solution vectors, while a smaller BW at the end of a generation can improve the fine-tuned characteristics of the solution vector. IHSA solves this problem by modifying the PAR and BW parameters so that they may vary, according to the number of generations, using Equations (18) and (19).

The purpose of improvising new harmony in each generation is to obtain the best harmony that can be a solution to the problem. The harmony in question takes the form of a solution vector, as in Equation (28). When improvising, a random number between zero and one is generated as the degree of probability to affect HMCR, PAR, and BW. The new harmony is obtained using three rules, as follows.

a. Random selection

If the random number (R) generated is greater than the HMCR (or, R > HMCR), the new decision variable i is obtained randomly with a certain range of values, as in Equations (26) and (27).

b. Consideration of harmony memory

If the random number (R) generated is smaller than HMCR (or, R < HMCR), the new decision variable i will be derived from harmony memory:

$$D_1 = int[R \times HMS] + 1 \tag{30}$$
$$D_2 = HM(D_1, i) \tag{31}$$
$$xb(i) = D_2 \tag{32}$$

where D_1 is a random index on harmony memory, D_2 is the value of the decision variable *i* at index D_1 and *xb*(*i*) is a new decision variable *i*.

c. Pitch adjustment

Next, the random number (R) is considered again. If it is smaller than PAR (or, R < PAR), then *xb*(*i*) of the harmony memory will be adjusted to BW. This adjustment involves generating another random number (R); if R > 0.5, the new decision variable is adjusted to Equation (33). Otherwise, (if R < 0.5) it is adjusted to Equation (34).

$$D_3 = xb(i) + R \times bw \tag{33}$$
$$D_3 = xb(i) - R \times bw \tag{34}$$

The decision variable adjusted to *i* will then be verified. It cannot violate the limits on each of the permitted decision variables: if $D_3 \leq \text{PVB}_{\text{lower}}$, then $D_3 = \text{PVB}_{\text{lower}}$, and, vice versa, if $D_3 \geq \text{PVB}_{\text{upper}}$, then $D_3 = \text{PVB}_{\text{upper}}$.

GA requires much more computation time than IHSA or HSA due to its more complex processes. HSA and IHSA use only simpler calculations of some of the rules of conditioning. In theory, IHSA should be slower than HSA due to its complex calculations of PAR and BW. However, this factor is not very influential in implementation.

Cuevas et al. [19], discussing the detection of circles on images, proved HSA produces better accuracy and faster calculation than GA, Randomized Hough Transform or Adaptive Bacterial Foraging. GA disadvantageously considers only two major vectors in generating new vector solutions. HSA can overcome these weaknesses by considering all the vectors that are inside when getting a new vector. HSA also requires a bit of mathematical calculation, does not require an initial solution candidate and has a high probability of identifying an area that has a solution. HSA does have difficulty in searching locally for a numerical application [13]. Visual system tracking, discussed by Fourie et al. [14], provides an alternative that can overcome the weaknesses of the HSA; that work proves that IHSA is faster and produces more precise accuracy in search of an optimal solution than the Particle Filter or the Unscented Kalman Filter. IHSA is also used by Karthigeyan et al. [20], who compared IHSA, HSA and BBO (Biogeography Based Optimization), also showing that IHSA has better performance than the HSA and BBO algorithms in terms of providing the best solution at minimal expense. These studies reinforce the research conducted by Mahdavi et al. [13], the founder of IHSA, who found that IHSA is better than HSA or GA in terms of performance and ability to find optimal solutions. This evidence provides reason enough to expect that IHSA can produce better performance and accuracy; it is better at extracting facial features under various conditions.

All three algorithms have broadly similar stability of accuracy, but IHSA is superior here, too, maintaining greater accuracy because possibilities for the PAR and BW parameters are aggressively searched to find every possible best available solution.

VII. Conclusion

This study successfully applied the improved harmony search algorithm (IHSA) to compare its performance in terms of accuracy and speed with the harmony search algorithm (HSA) and the genetic algorithm (GA). The results show that IHSA's accuracy and performance are better than HSA or GA and could be applied to real-time systems. However, some issues do affect IHSA’s accuracy, such as image brightness, beards, glasses and other similar attributes. This is due to IHSA’s poor segmentation, which leads the algorithm to miss some edges besides those delineating strong characteristics of the proper facial features.

A method for better facial segmentation is needed to retain the edge of the facial features, a method that is less affected by lighting and can discard the edges of unnecessary objects. A method of determining facial features after detection is also necessary to minimize errors in such detection. Then, automation of face-feature size is needed to improve the accuracy of the extracted facial features.

References

[1] R. C. Gonzales and R. E. Wood, *Digital Image Processing*. Prentice Hall, 2005.

[2] V. Mayya, R. M. Pai, and M. Pai M M, “Automatic Facial Expression Recognition Using DCNN,” *Procedia - Procedia Computer Science*, vol. 93, no. September, pp. 453–461, 2016.

[3] A. Abouyahya, S. El Fkihi, R. Oulad, H. Thami, and D. Aboutajdine, “Features Extraction For Facial Expressions Recognition,” in *5th International Conference on Multimedia Computing and Systems (ICMCS)*, 2016.

[4] Y. Kristian, H. Takahashi, I. K. E. Purnama, K. Yoshimoto, E. I. Setiawan, E. Hanindito, and M. H. Purnomo, “A Novel Approach on Infant Facial Pain Classification using Multi Stage Classifier and Geometrical-Textural Features Combination,” *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 112–121, 2017.

[5] Zahraddeen Sufyanu, Fatma S. Mohamad, Abdulganiyu A. Yusuf, and Mustafa B. Mamat, "Enhanced Face Recognition Using Discrete Cosine Transform," Engineering Letters, vol. 24, no.1, pp52-61, 2016.

[6] B. Poon, M. A. Amin, and H. Yan, “PCA Based Human Face Recognition with Improved Methods for Distorted Images due to Illumination and Color Background,” *IAENG International Journal of Computer Science*, vol. 43, no. 3, pp. 277–283, 2016.

[7] V. Rapp, T. Senechal, K. Bailly, L. Prevost, U. Pierre, and T. T. Paristech, “Multiple Kernel Learning SVM and Statistical Validation for Facial Landmark Detection,” *IEEE Face Gesture*, pp. 265–271, 2011.

[8] V. Radha and N. Nallammal, “Neural Network Based Face Recognition Using RBFN Classifier,” in *Proceedings of the World Congress on Engineering and Computer Science, WCECS 2011*, 2011, vol. I, pp. 555–560.

[9] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, “Real-time facial feature detection using conditional regression forests,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, 2012, pp. 2578–2585.

[10] S. R. Benedict, “Geometric Shaped Facial Feature Extraction for Face Recognition,” in *IEEE International Conference on Advances in Computer Applications (ICACA)*, 2016, pp. 275–278.

[11] Y. Luo, T. Zhang, and Y. Zhang, “Optik A novel fusion method of PCA and LDP for facial expression feature extraction,” *International Journal for Light and Electron Optics*, vol. 127, no. 2, pp. 718–721, 2016.

[12] S. Devadethan, G. Titus, and S. Purushothaman, “Face Detection and Facial Feature Extraction Based on a Fusion of Knowledge Based Method and Morphological Image Processing,” in *Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD)*, 2014.

[13] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[14] J. Fourie, S. Mills, and R. Green, “Harmony filter : A robust visual tracking system using the improved harmony search algorithm,” *Image and Vision Computing*, pp. 1–15, 2010.

[15] A. Ouaddah and D. Boughaci, “Harmony search algorithm for image reconstruction from projections,” *Applied Soft Computing*, vol. 46, pp. 924–935, 2016.

[16] A. Bansal, “A Comparative Study of Feature Selection in Face Recognition Using Harmony Search Algorithm and PCA,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 6, no. 7, pp. 998–1001, 2017.

[17] Z. W. Geem, C.-L. Tseng, and J. C. Williams, *Music-Inspired Harmony Search Algorithm*, vol. 191. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[18] M. Weber, “Face Database Collection,” *California Institute of Technology*, 2015. .

[19] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, “Circle detection by Harmony Search Optimization,” *Journal of Intelligent and Robotic System: Theory and Applications*, vol. 66, no. 3, pp. 359–376, 2012.

[20] P. Karthigeyan, M. S. Raja, R. Hariharan, S. Prakash, S. Delibabu, and R. Gnanaselvam, “Comparison of Harmony Search Algorithm, Improved Harmony Search Algorithm with Biogeography Based Optimization Algorithm for Solving Constrained Economic Load Dispatch Problems,” *Procedia Technology*, vol. 21, pp. 611–618, 2015.

Erwin was born in Palembang, Indonesian, in 1971. He received the Bachelor degree in Mathematics from the University of Sriwijaya, Indonesia, in 1994, and the M.Sc. degrees in Actuarial from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2002. He is studying Ph.D. degrees in Informatics Engineering at University of Sriwijaya. In 1994, he joined, University of Sriwijaya, as a Lecturer. Since December 2006, he has been with the Department of Informatics Engineering, University of Sriwijaya, where he was an Assistant Professor, became an Associate Professor in 2011. Since 2012, he has been with the Department of Computer Engineering, University of Sriwijaya His current research interests include image processing, and computer vision. Erwin, S.Si., M.Si. is a member of IAENG and IEEE.

Saparudin was born in Pangkal Pinang, Indonesia, in 1969. He received the Bachelor degree in mathemathic education from the University of Sriwijaya, Indonesia, in 1993, and the M.Tech. degrees in informatics from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2000 and Ph.D. degrees in computer science from the Malaysian University of Technology (UTM), Johor Bahru, Malaysian, in 2012. In 1995, he joined, University of Sriwijaya, as a Lecturer. Since December 2006, he has been with the Department of Informatics Engineering, University of Sriwijaya, where he was an Assistant Professor, became an Associate Professor in 2011, and a Professor in 2017. His current research interests include image processing, and computer vision. Drs. Saparudin, M.T., Ph.D. is a member of Institute of Advanced Engineering and Science (IAES) and IEEE.

Muhammad Fachrurrozi was born in Palembang, Indonesia, in 1980. He received the Bachelor's degree in Mathematics from the University of Sriwijaya, Indonesia, in 2002, and the M.Sc. degrees in Software Engineering from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2008. In 2008, Fachrurrozi joined the University of Sriwijaya, as a Lecturer. Since December 2008, he has been with the Department of Informatics Engineering, University of Sriwijaya, where he was an Assistant Lecturer. His research interests include image processing, computer vision and machine learning. Muhammad Fachrurrozi S.Si., M.T. is a member of Institute of Advanced Engineering and Science (IAES) and IEEE.

Arief Wijaya was born in Pangkal Pinang, Indonesia, in 1996. He is student at Department of Informatic Engineering, Faculty of Computer Science, Sriwijaya University, Indonesia. In 2016, he joined the Laboratory of Image Processing, University of Sriwijaya, as Assistant Lecturer. His current research interests include real time, mobile computing, artificial intelligence, pattern recognition, computer vision and, image processing.

Muhammad Naufal Rachmatullah was born in Surakarta, Indonesia in 1992. He has already been a research assistant at Department of Informatics, University of Sriwijaya Palembang, Indonesia since 2015. He received the Bachelor degree in Informatics from th e University of Sriwijaya, Indonesia, in 2015. Currently He is a master student at Bandung Institute of Technology (ITB), Bandung, Indonesia. His research interests include image processing, computer vision, and machine learning.

(Advance online publication: 7 November 2018)