Designing and Implementing System of Real Time Face Detection and Recognition Based on RBF

Eri Prasetyo Wibowo, Bonaventura Pinandito, Bheta Agus Wardijono and Akeda Bagus

Abstract—Face recognition is the one of many interesting researches concerned by biometric system since several years ago. This interest is caused by the easy process to obtain the research data (face images), although there are still many factors that can affect a failure rate of identification in pattern recognition of face. The research about face recognition develops continuously with the growing human needs in the security field, human computer interaction, and so on. Today, the object of research is not only the static image, but also begins to the real-time moving images.

The paper explores the implementation of Radial Basis Function (RBF) method in making real-time face recognition application. The input data for the RBF network is obtained from the face localization process using Haar-like feature method, egmentation, feature extraction, and normalization using image processing algorithm to get the fiducial points of face; representing the characteristics of each human face. The RBF values are then processed to calculate the difference level with each target; resulting of network learning process. The method that is used in learning process network is LMS (Least-Mean-Square) and to measure the difference level used Euclidean Distance method.

From the test results of 10 respondents with several conditions, RBF method has been able to do face recognition with the recognition success rate of 90% - 100% with a record in database, 81.6667% - 94.2262%, with 3 records in database, and 80% - 86.3095% with 5 records in database. The conditions of sampling and testing time affect the percentage of success rate.

Index Terms—RBF, Recognition, Face, Real-Time, ANN.

I. INTRODUCTION

FACE recognition has drawn considerable interest and attention from human computer interaction, biometric, and security researchers for the two last decades. Begun in 1970 with semi-otomatic system; by finding main features of human face (eyes, ears, nose and mouth) in photograph, calculating the distance and ratio from one point to global reference point and then compare the distance with references; face recognition has been grown [2]. There are a lot of methods used from Eigenface PCA, Fisherface FLD, Support Vector Machine, etc; invented to increase the accuracy process on identifying face with different challenges such as various exposures, face poses, expressions, obstacle objects and face components that may be exists or not; both on still images or real-time moving images.

Conditions in real-time human face recognition can occur if the process happens rapidly identifying or equal the actual process in the real world. In its application, a tool such as a webcam has been widely used for implement realtime conditions such. Besides tools used, there are several approaches that can be done to optimize the face recognition process real-time and one of them by using the method Radial Basis Function (RBF)[4][5].

Many researchers use the RBF method to develop a system face recognition in real-time, because besides having strong network to the conditions that allow decrease in the level of identification accuracy, this method also has a simple calculation[4][6].

II. FUNDAMENTAL THEORY

Radial Basis Function (RBF) was first introduced in real problem solving multivariate interpolation[1]. RBF is a method that has ANN feedforward (all input neurons connected to hidden neuron in which connected to the output neurons without feedback) that consists of input, hidden and output layers. Input layer has N units for an N-dimensional input vector. Unit input fully connected to hidden layer units I, which is also connected to the output layer unit J, where J is the number of output classes [4]. Architecture RBF network can be seen in Figure 1.

RBF network can be categorized as kernel. Each units or hidden node calculates kernel function of data input, and output layer reach a weight of kernel functions. Each unit or node is characterized by 2 values important parameters, RBF center and width. Node Hidden providing high value and input vector for center and the output value decreased with increasing center distance[4].

Sep 2011 ATC-40101045©Asian-Transactions 1
RBF function can be seen as a process that maps each p-dimensional input vectors (patterns) of inputs into the living room m-dimensional decision. Thus, the function non-linear (kernel function activation) and function used in the linear hidden layer and output respectively. Conventionally, the hidden layer, function is used as a non-linear function is a function Gaussian, which is defined as follows:

\[
\phi_j(x_i) = \exp\left(-\frac{d(x_i, c_j)^2}{\sigma_j^2}\right)
\]

\[j = 1,2,\ldots,n; i = 1,2,\ldots,N\]

where \(c_j\) and \(\sigma_j\) is the center and width of the receptive field of the Gaussian function associated with \(j\) hidden layer neurons, each \(n\) and \(N\) is the number of hidden layer neurons and the number of input patterns. In a recent study, the values of \(n\) and \(N\) is the same and equal to the number of input patterns or samples training[3].

\(d(x_i, c_j)\) is the euclidean distance between vectors input \(x_i\) and the center vector \(c_j\) and is usually regarded as a size difference, which is defined as follows

\[
d(x_i, c_j) = \left(\sum_{u=1}^{p} |x_{iu} - c_{ju}|^2\right)^{1/2}
\]

Compared with euclidean distance, the number of absolute differences of the components of input vector \(x_i\) and center \(c_j\) can also be used as size difference, which is defined as follows

\[
d_i(x_i, c_j) = \sum_{u=1}^{p} |x_{iu} - c_{ju}|
\]

Other simple form about distance, that’s only consider absolute difference between \(x_{iu}\) and \(c_{ju}\) which is greater than \(\delta\) threshold, can be used in Gaussian function and defined as follows

\[
d_i(x_i, c_j) = \sum_{u=1}^{p} |x_{iu} - c_{ju}| \geq \delta
\]

Another approach is by using simple activation function than Gaussian function. Heaviside function can be used for this purpose which defined as follows

\[
\phi_j(x_i) = \begin{cases} 
1; & d(x_i, c_j) \leq \sigma_j \\
0; & d(x_i, c_j) > \sigma_j 
\end{cases}
\]

Output from k-th neuron layer determined by following linear function

\[
z_{ik} = \sum_{j=1}^{n} \phi_j(x_i) \omega_{kj} + b_k \omega_k
\]

where \(\omega_{kj}\) is a weight between \(j\) hidden neuron layer and k-th output neuron layer, \(b_k\) and \(\omega_k\) are positive bias unit and weight to k-th output neuron layer from each neuron bias

III. DESIGN AND IMPLEMENTATION

A. Flowcharts of system

In general, this real-time face recognition application consists of three steps. First step is pre-processing consisting of face localization, segmentation, features extraction, and normalization. In this step all results were acquired using artificial neural network algorithm and image processing. The result from face localization and segmentation can be seen on figure 2 and for feature extraction can be seen on figure 3.
used to learn a network. In learning a network, a calculation for weight and bias is performed. One of algorithm that can be used to calculate those values is LMS (Least-Mean-Square)[7]. The results from learning are center, width, weight, bias and target. The final step is post-processing stage. In this step, dissimilarities between target from learning process and RBF value that already multiplied by weight and bias are determined. The dissimilarities are determined by using euclidean distance. Figure 4 shows two processes that can be happened in this application, that’s, learning process and face recognition process.

**B. Radial Basis Function Network**

As explained in fundamental theory, Radial Basis Function (RBF) has 3 layers architecture, namely input layer, hidden layer, and output layer. Each layer has its own algorithm to calculate input and output values. In input layer, input values for network were acquired from several processes that has been explained on processing stage. Input layer has total 120 values, where each value represents reference coordinates from each components in face. These values will be transmitted to hidden layer, where RBF values will be determined. There are several parameters to determine RBF values, including input values (120 coordinates that already calculated), center and width RBF. Total center and width is 7 according to number of cluster that already defined. Total number of cluster was defined from total face components that may represent human face characteristics. The characteristics are distances between one component with other components in human face. Clusters on face:

- left and right eyebrows,
- left and right eyes,
- center area between eyes and nose,
- nose, and
- mouth.

Because there are 7 values for center and width for each cluster, then there will be 840 RBF values. These RBF values will be multiplied with theirs weight. The weights were determined from learning the network of 8 images of human faces. The result will be summed and this sum will be summed again with bias values that were acquired from learning the network. The final result from these processes will be compared with each target value of sample in database. This comparison was done by using euclidean distance. If the comparison has dissimilarity between 0 - 0.5 then the application successfully recognize or identify a human face that acquired by webcam, even though there maybe a failure in identification process.

Figure 5 shows Radial Basis Function network architecture on identifying human face in real-time.

**Figure 5 description:**

- $x$ is input acquired from final result on pre-processing.
- $\phi$ are Radial Basis Function values determined on processing.
- $n$ is total $x$, which is 120.
- $m$ is total cluster, which is 7.
- $b$ is bias value from learning process.
- $w$ is weight from learning process.
- $z$ is final result from each cluster. This value determined by using the following formula

$$Z_m = \sum_{m=1}^{120} \phi_m x_m + b_m$$

(7)
C. Determine the Center of Radial Basis Function (c)

Center RBF is one of parameter needed to determine the RBF itself. This center value is a coordinate which acquired by calculating center point between nearest distances and farthest distances that owned by x and y on each cluster. For example, on eyes cluster that consists of 16 points with each coordinate value. First, determine the smallest x and y (x₀ and y₀) then determine the biggest x and y (x₁ and y₁) from each reference point on this cluster. The center point of x is the difference between x₀ and x₁ divided by 2. The center point of y is the difference between y₀ and y₁ divided by 2. These center points are called center.

There’s special case to determine the center value between right eye, left eye and nose, that’s, by calculating the smallest and the biggest values of x and y from references points if right eye, left eye, and nose with the center point of nose. This calculation is done by calculating the distance between mid area of right eye, left eye and nose. After these values have been acquired by calculating the distance between each reference points of eyes with the center of right and left ey

\[ ||x_i - c||, j = 1,2,.....n \text{ and } i \neq j \]  

where this \( \beta \) is a constant between 1-3. In this application, \( \beta \) is 1.75. There’s special case to determine the width in mid clusters of right eye, left eye and nose; the distance calculation between these clusters with the center of right and left eyes as performed. The smallest value from both distances is the width value. The width of nose’s cluster was determined by calculating the distance between mid area of right eye, left eye, and nose with the center point of nose. This calculation was done by using euclidean distance.

D. Determine the Width of Radial Basis Function (σ)

One of parameter that’s used to calculate RBF is width. There are 7 widths on each images of human face in this application. These width were acquired by calculating the distance between one center point with other center point then multiplied by \( \beta \) [3].

\[ \sigma_j = \beta \cdot \min ||x_i - c||, j = 1,2,.....n \text{ and } i \neq j \]  

where \( \alpha \) and \( \beta \) are two parameters that must be defined, that’s, learning rate parameter \( \alpha \) and threshold. In this application the value of \( \alpha = 0.001 \) and the threshold \( \delta = 0.0005 \).

As this learning is supervised there must be a chosen target to learn. The total target is proportional to total samples used in learning process. Each sample will be a target in the learning process. To determine the target from first sample to eighth sample was chosen in order. Each target has 7 sub targets. These sub targets contain values from sum of RBF values in each cluster. The following is the algorithm used in the learning process:

1. As long the failure rate < threshold or the failure rate > Threshold
   a. Set input activation (RBF values)
   b. Determine the output response
   \[ y = f(\text{net}) = \text{net} = \sum x_i \alpha_i + b \]  

c. Fix weight and bias with following formula:
   \[ \alpha_i(\text{new}) = \alpha_i(\text{old}) + \alpha(t - y) x_i \]  
   \[ b(\text{new}) = b(\text{old}) + \alpha(t - y) \]  

2. Replace target with next target, loop step 0 and 1 until all samples become target.
The sample with smallest errors of the overall calculation of the target samples will be used as a reference for choosing weights, bias, target, center and width that will be saved and will be used to search the RBF and the results of processing stages in the core network of face recognition in real-time.

G. Matching

To perform the matching process there are many ways to accomplish and one of them is to calculate the degree of dissimilarity. The calculation is done by comparing the output of processing stage with target values generated in the learning process which stored in the database. To perform these processes, this application uses euclidean distance algorithm.

\[
d(p-q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \ldots + (p_7-q_7)^2}
\]

\(p_1, p_2, p_3, \ldots, p_7 = \) output from processing stage
\(q_1, q_2, q_3, \ldots, q_7 = \) target sample from database

The results of dissimilarity through Euclidean distance algorithm will be checked whether the resulting value lie between 0-0.6 value or not. If the value is contained within the range of 0-0.6, the image contained within the frame was successfully recognized. The success of facial recognition was marked by the emergence of the name of the person concerned his/her face in the upper left of the ROI detected face. The result of face recognition can be seen on figure 7.

Fig. 7. Face successfully recognized

IV. TESTING AND ANALYSIS

The system was tested by various conditions, including:
- Consists of 1, 3, and 5 records in the database.
- Dark and light conditions.
- The distance between 36 - 55 cm and 56 - 75 cm.

In this test, the author also analyzes the influence of sampling on the face of dark and light conditions on the final result obtained.

Hardware specifications which used in the testing system:
- Processor : Intel(R) Celeron(R) M Processor 1.30 GHz
- Memory : 1,2 GB RAM
- VGA Card : 64 MB
- Harddisk : 40 GB

And the specification of the webcam as follows:
- Sensor Type: CMOS SXGA sensor with NightVision Technology
- Resolution : SXGA 1280 x 1024
- The maximum resolution Snapshot : Up to 2582 x 1944 (5M)
- Frame Rate : Up to 30 fps(VGA)
- Viewing angle : 71° horizontal
- Image Format : 420, RGB24
- Dimension : 60 x 70 x 75 mm

A. Testing Procedure

In testing procedure, the steps are performed into 2 phases, namely sampling phase, and face recognition stage.

The first step is to do sampling, are as follows,
1. Running the application, select the training menu, and enter the sample name as the identity of the data to be entered into a database.
2. Set the distance and position of the face to webcam.
3. Taking eight frontal face positions to be used as input data in the learning process, by pressing the ‘S’ or ‘s’ on the keyboard. If the 8th face image has been taken, then the sampling step can be continued by pressing the[Training].
4. The program will learn the data input and store information like the name of the sample, bias, target, center RBF, RBF width, and weight into the XML database.

The second step is face recognition. This step is an important part, where application will perform the face recognition process to every faces that was captured by the webcam. At this stage the steps taken is as follows.
1. Set the distance and position of the face of people who were tested on a webcam.
2. Running the application.
3. Counting time for 90 seconds, if within 90 seconds the face is recognized, then:
   a. start counting time for 10 seconds.
   b. save the image on the frame successfully recognize faces.
   c. back to point 1.
   if application wasn’t able to recognize the face then facial recognition value = 0.

B. Data Source

Data used in this application comes from a face image retrieval performed in real-time against 10 people. Each person was captured as many as 8 samples for the learning process. Taking the image of the sample performed at different places but in 2 general conditions, that is, light and dark.
conditions. In bright conditions, sampling done with the help of sunlight and some samples with the aid of light in the room. In dark conditions, sampling is done by a little sunlight and specific to conditions that are too dark, use a little help from the webcam light. Sampling was also performed with the provisions that face must be in frontal conditions with little expressions.

### Table I

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Name</th>
<th>LP (ms)</th>
<th>RNDK</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barli</td>
<td>566 828</td>
<td>168 475</td>
<td>Light</td>
</tr>
<tr>
<td>2</td>
<td>Bonaventura</td>
<td>317 422</td>
<td>90 5672</td>
<td>Dark</td>
</tr>
<tr>
<td>3</td>
<td>David</td>
<td>208 609</td>
<td>80 2892</td>
<td>Dark</td>
</tr>
<tr>
<td>4</td>
<td>Daya</td>
<td>468 391</td>
<td>88 8434</td>
<td>Dark</td>
</tr>
<tr>
<td>5</td>
<td>Dick</td>
<td>564 372</td>
<td>71 8676</td>
<td>Dark</td>
</tr>
<tr>
<td>6</td>
<td>Hadi</td>
<td>417 329</td>
<td>98 8027</td>
<td>Dark</td>
</tr>
<tr>
<td>7</td>
<td>Faith</td>
<td>366 484</td>
<td>89 652</td>
<td>Dark</td>
</tr>
<tr>
<td>8</td>
<td>Michael</td>
<td>379 360</td>
<td>57 4068</td>
<td>Dark</td>
</tr>
<tr>
<td>9</td>
<td>Ryan</td>
<td>310 406</td>
<td>116 892</td>
<td>Light</td>
</tr>
<tr>
<td>10</td>
<td>Zhilal</td>
<td>180 578</td>
<td>67 9728</td>
<td>Dark</td>
</tr>
</tbody>
</table>

Description Table:

- **LP (ms)** = Long Learning (milliseconds)
- **RNDK** = Average value of gray Degrees

### C. Learning the Sample Data

In the process of learning 8 sample images of each person, a timer that measures the learning process is activated. This time calculation is done implicitly by the program. The value of the time required for a process of learning produces a quantity which varies between the first face learning with others. Although the amount of time learning process that produced a value different from one another, but in general these values are within range of 180578 up to 566828 ms. Table I shows the results of this learning process.

The average gray degree value is used as a parameter to measure the light level at the time of testing. If the calculation of average gray degree value greater than or equal to 110 then the environment at the time of testing can be categorized as a bright environment, while if the average gray degree value smaller than 110 then categorized dark testing environment. From table I can be calculated the average network learning (RWP) for:

\[
RWP = \frac{\sum_{n=1}^{10} \text{long learning}_n}{10} = \frac{3779782}{10} = 377978.2\text{ms}
\]

### D. Tests with 1, 3, and 5 Records in the Database

This test aims to measure the influence of the number of data samples in the database on the level of success and failure in the process of face recognition. Testing is also done on some other condition that is a combination of dark and light conditions and distances 36-55 and 56-75 distances.

Testing with a record in the database is first test done by each person tested. The goal of this testing is to see how the results obtained if the database only consists of a record, record of each person, and there is no other record which could result in a face recognition error.

### Table II

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Name</th>
<th>Dark</th>
<th>Light</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Range 36-55</td>
<td>Range 56-75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PWB</td>
<td>PWS</td>
</tr>
<tr>
<td>1</td>
<td>Barli</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Bonaventura</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>David</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>Dayat</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>Dick</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Hadi</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Iman</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>Michael</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>Ryan</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>Zhilal</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Description Table:**

- **PWB** = Successful face recognition
- **PWS** = Failure face recognition

Results obtained from this test can be seen in table II with the final average gray degree value (obtained from the average add value degrees of gray each image in the category, divided by the number of images that successfully recognize someone’s face) on table V for the case of a record in the database, table III with an average final value of the degree of gray in the table VI for case 3 records in the database, and table IV with an average final value of the degree of gray in the table VII for case 5 records in the database.
To calculate the percentage of success in recognizing facial applications the following formula was used:

\[ P_{success} = \frac{\sum PWB}{n} \times 100\% \quad (15) \]

\[ P_{success} = 0\% \quad PWB = 0 \quad (16) \]

As for the formula used to calculate the percentage of failures:

\[ P_{failure} = \frac{\sum PWS}{n} \times 100\% \quad (17) \]

\[ P_{success} = 0\% \quad PWS = 0 \quad (18) \]

Table VIII displays the results of the overall calculation of the percentage of successes and failures of real-time face recognition on this application.

E. Testing the Effect of Light on the Accuracy of the Sampling Application

The test is conducted to see whether the effect of light conditions (dark or light) to the value of the percentage of success and failure in some circumstances. For dark conditions, the authors took samples Bonaventura, David, Daya, Dicky, Hadi, Iman, Michael, and Zhial. As for the conditions, the authors took samples Barli and Ryan. The source of light used in the sampling condition is a little dark light of the webcam, while for sampling conditions, the light source used was sun light.

To calculate the percentage of successes and failures of applications in testing the effect of light during this sampling, the author uses the formula 15, 16, 17, and 18.

Table IX is the result of calculating the percentage of success and failure of application of the results of sampling dark conditions. While Table X is the result of calculating the percentage of success and failure of application of the sample conditions.

Table IX displays the results of the overall calculation of the percentage of successes and failures of real-time face recognition on this application.

V. CONCLUSIONS

- This study explores the implementation of the method of Radial Basis Function (RBF) in the manufacture of face recognition applications in real-time.
- The system of dividing the application process into 3 stages, namely pre-processing stage, processing, and
postprocessing, specifically for the implementation of the RBF method performed at this stage of processing.

- From the test results against 10 people on several conditions, RBF method has been able to do facial recognition with the introduction of the success percentage of 90% - 100% with 1 record in the database, 81.6667% - 94.2262% with 3 records in the database, and 80% - 86.3095% with 5 records in the database.
- More and more data samples in the database, the smaller the level of success in face recognition.
- Conditions at the time of sampling and testing as a major influence small percentage of successful applications.

REFERENCES


Eri Prasetyo Wibowo was born in Kendal, Indonesia, in 1966. He received the B.S degree in Electronics and Instrumentation from Gadjah Mada University, Indonesia, in 1992, the M.S. degree in Information System from Gunadarma University, Indonesia, in 1994 and the Ph.D. degree in Electronics Informatics from the Burgundy University, France, in 2005. He is member of IEEE, Professional Organization in the field of Information and Computer Technology (IPKIN). He was as Secretary of Doctoral Program in Information Technology at Gunadarma University. He was member of EACOVIROE Project to promoted Master VIBOT which founded by Erasmus Mundus. His current interests are in System On CHIPS Design

Bonaventura Pinandito born in Depok on October 20, 1988 as the couple’s second child Ign Djoko Irianto and MM. Menik Saryati. The author has completed primary education in 2000 at SD Santa Theresia Depok, junior high school in 2003 in SMPN 2 Depok, and high school in SMA Mardi Yuana Depok. The author has finished writing Gunadarma science at the University in 2009 with the title of the writing of "MAKING THE WEB SITE BY USING THE CMS NATIONAL FRAMEWORK DEPOK CodeIgniter 1.7.1" to obtain a baccalaureate degree. In 2010, the author managed to complete college with a degree in Bachelor of Engineering majoring in Computer Science. The author lives in Depok. Author Email Address: bona.dvent@gmail.com.

Bheta Agus Wardijono was born in Jakarta, November 3-rd 1970. The author has finished the bachelor from University of Indonesia with Physics as the major. In 2001 he has finished post graduate and got the Master of Technique from Tri Sakti University. In the year 2010, he has finished Doctor degree from Gunadarma University. He is a member of Professional Organization in the field of Information and Computer Technology (IPKIN) in Indonesia. Currently he works as a lecture in Gunadarma University and High School of Computer and Informatics Jakarta STIK. The author lives in Jakarta and can be reached via email address: bheta@staff.gunadarma.ac.id or bheta@jak-stik.ac.id