Parallel Optimized Pearson Correlation Condition (PO-PCC) for Robust Cosmetic Makeup Facial Recognition

Kanokmon Rujirakul and Chakchai So-In
Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand

Abstract: *Makeup changes or the application of cosmetics constitute one of the challenges for the improvement of the recognition precision of human faces because makeup has a direct impact on facial features, such as shape, tone, and texture. Thus, this research investigates the possibility of integrating a statistical model using Pearson Correlation (PC) to enhance the facial recognition accuracy. PC is generally used to determine the relationship between the training and testing images while leveraging the key advantage of fast computing. Considering the relationship of factors other than the features, i.e., changes in shape, size, color, or appearance, leads to a robustness of the cosmetic images. To further improve the accuracy and reduce the complexity of the approach, a technique using channel selection and the Optimum Index Factor (OIF), including Histogram Equalization (HE), is also considered. In addition, to enable real-time (online) applications, this research applies parallelism to reduce the computational time in the pre-processing and feature extraction stages, especially for parallel matrix manipulation, without affecting the recognition rate. The performance improvement is confirmed by extensive evaluations using three cosmetic datasets compared to classic facial recognitions, namely, principal component analysis and local binary pattern (by factors of 6.98 and 1.4, respectively), including their parallel enhancements (i.e., by factors of 31,194.02 and 1577.88, respectively) while maintaining high recognition precision.*

Keywords: *Cosmetic, facial recognition, makeup, parallel, pearson correlation*.

Received September 1, 2015; accepted September 26, 2016

1. Introduction

Facial Recognition Systems (FRSs) have been widely used in many real-world applications, such as authentication and identification systems, surveillance and tracking systems, and person verification and investigation [21]. FRSs developed by commercial and research firms primarily focus on improving the recognition rate or achieving high accuracy [16].

FRSs are one of the main computer applications used for person identification using a facial image from a digital picture or video frame. Two main components of an FRS are facial detection [19] and facial identification [3]. The first component is used to search and identify a face in a particular image. The face discovered by the first component will then be verified by the second component.

Although previous studies focused on face location identification given explicit facial characteristics, such as skeleton, shape, and border [8], the development of face verification is ongoing [16, 39], including studies to improve its precision and computational time and complexity, especially once it is applied in practical cases, i.e., real-time/online facial recognition [16].

In addition to the baseline face, several factors and features have a direct impact on the recognition precision [33], including illumination, facial pose, occlusion, and expression (including makeup changes, such as the application of cosmetics). Although several techniques have been proposed to improve the recognition of the first three factors [39]; few research studies consider the challenge of the fourth factor, i.e., makeup changes. This factor was recently found to affect the verification accuracy due to a high degree of change in the facial appearance [13].

There are several differentiations due to the use of makeup or cosmetics, such as skin color and shade [10]. The use of foundation, toner, and concealer can increase the illumination of skin and remove wrinkles as well as spots on the original face. Foundation can also cause the appearance of facial texture to change. In fact, makeup can change the main facial features, such as shape, size, color, and position of different facial areas-lip size, cheek color, and position of the eyebrows (from lipstick, brush-on makeup, and eye shadow), all of which can mislead the interpretation due to the diversity of extracted features [5].

Despite the potential improvement in recognition accuracy, real-world applications are also constrained by feasibility, i.e., computational complexity. Several techniques are available to reduce the computational complexity of FRSs [28, 30, 31]; however, no facial recognition research considers the challenge of makeup changes, especially with the combination of these two matrices.

Thus, this study proposes a robust FRS; in particular, the system focuses on the makeup/non-makeup facial images to achieve high recognition accuracy with low computational complexity. The system is based on the relationship between the training and testing images, i.e., color-scale selection and statistical data.

Our contributions can be divided into three stages:

1. In the pre-processing stage, a proper condition is selected to determine a suitable color channel, including the use of Histogram Equalization (HE) to spread the intensity in the image.
2. In the feature extraction stage, Pearson Correlation (PC) is used to calculate the relationship of the images, including its optimization; to improve its precision further, the Optimum Index Factor (OIF) is also used to determine the candidates.
3. In the image identification stage, a concept of majority voting is applied to these candidates.

These three stages are used to gain higher precision; however, due to the time complexity involved, the concept of parallelism is also integrated into the first two stages.

The remainder of this article is organized as follows. Section 2 provides a brief overview of the background of facial recognition. Section 3 presents a comparative survey of recent related works. Section 4 presents our approach, Parallel Optimized Pearson Correlation Condition (PO-PCC), with an emphasis on cosmetic facial images. Section 5 discusses the performance of our proposed approach. Finally, section 6 presents the conclusions and avenues for future work.

2. Face Recognition Systems

In general, there are two stages in an FRS: training and testing. Each of these two stages consists of three main steps [2], namely, pre-processing, feature extraction, and image identification.

With training, the system will acquire the facial images as the training image from various sources, such as a video frame, a camera, or photo scanning, and then perform the pre-processing; many pre-pre-processing approaches have been proposed, i.e., illumination normalization, background removal, gray-scale conversion, and HE, all of which are used to normalize the facial image. Next, the system will extract the features of the normalized facial images, thereby collecting their feature vectors in training sets for the purpose of comparison in the testing process.

Similarly, the testing image will pre-process the images to obtain normalized images. The feature vector of the testing will likely be determined at this point. Subsequently, the normalized test image will be classified by comparing its vector against the vectors from the training process.

Several well-known feature extraction methods, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA), exist [16]. PCA is employed as a baseline and is extensively employed for facial recognition [2, 3, 23, 34]. In addition, regarding the abnormal characteristics of a face, i.e., cosmetics, Local Binary Pattern (LBP) [13] is one of the feature-based techniques resulting in high accuracy [1, 15, 20, 27]. However, LBP requires substantial computational time, especially in makeup change applications [9, 10].

2.1. Principal Component Analysis for Facial Recognition

PCA is an appearance-based feature extraction approach that is generally employed for facial recognition [25]; the key objective of PCA is to reduce large-dimensional data while retaining only the major features [23]. PCA for facial recognition consists of two main stages, as described below.

1. Feature extraction: there are four components in this step:
   a) Computing a mean over training images by normalizing all training data by subtracting the average from each datum.
   b) Calculating a covariance matrix from the normalized data and determining an eigendecomposition to obtain the eigenvalues and eigenvectors from the covariance matrix.
   c) Sorting the eigenvectors in descending order of their eigenvalues and removing the eigenvectors that have an eigenvalue of zero.
   d) Projecting the training image via the multiplication of the normalized data with the sorted eigenvectors.
2. Image identification: given the testing image, the system will then calculate the normalized testing image by subtracting the mean of the training images from the testing image and then projecting the normalized testing image into the eigenspace via multiplication with the eigenvectors from the training section. Finally, the training projection and testing projection will be compared to determine the minimum difference.

Although PCA can reduce high-dimensional data, which in turn prepare a proper feature for further classification, there remain some limitations, such as high computational complexity [18] and low recognition precision with some facial variations, such as illumination, occlusion, facial pose, expression, and makeup changes [11, 33].

2.2. Local Binary Pattern for Facial Recognition

LBP is one of the most widely used feature extraction approaches [15], with its key advantage being high recognition precision, especially with texture property extraction [1, 15, 20, 27]; however, LBP also has a high computational complexity.

The LBP approach is primarily used to calculate a representative of the defined area, generally with a size of 3×3 pixels. Next, the center point of the area will be determined as the reference point. Every value in the pixel around the center point will be compared with the value in the center point to search for the threshold. If the value is less than the center point's value, then the value will be updated to 0; otherwise, the value is 1. The binary representation will be converted into histogram format as the image texture property. Subsequently, a multiplication of the threshold and its weight is performed to determine a representation of LBP for the training and testing images. The Euclidean distance will be then used to select the actual image.

2.3. Pearson Correlation for Facial Recognition

Most FRSs apply appearance-based feature extraction (PCA and LBP, including their derivatives); however, certain facial image characteristics, one of which is the makeup (cosmetic) changes, affect the accuracy of these approaches, and their high computational complexity is not suitable for real-time facial recognition.

Thus, this research proposes another candidate to focus on the problem of facial recognition with cosmetic images, i.e., the application of PC. PC is one of the statistical methods used to calculate the relationships between two datasets [32]. This relationship is given as the correlation coefficient (CC), or *r*, as shown in Equation (1) [18]. Here, $\overline{X}$ and $\overline{Y}$ are the average of the first dataset and the second dataset, respectively. *r* ranges between -1.00 and +1.00.

$$r = \sum_{i=1}^{n} (X_i-\overline{X})(Y_i-\overline{Y}) \div \sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\overline{Y})^2} \qquad (1)$$

In this Equation, a value of *r* closer to +1.00 means that the two datasets have the same directional relationship. If the first dataset increases, then the other dataset will also increase. In contrast, for a value of *r* near-1.00, one can assume that the two datasets have the inverse relationship, i.e., if the first dataset increases, then the second dataset will decrease. If *r* is close to 0, the two datasets are not significantly correlated. Thus, *r* values closer to-1.00 or+1.00 indicate a closer relationship between the two datasets.

When applying the traditional PC to the facial recognition process, the system first performs the pre-processing step in both the training and testing stages. Next, the average of each image is calculated, and all images are normalized by subtracting each image with its mean. Subsequently, Equation (1) is used to obtain the *r* value. Finally, the maximum *r* value is obtained.

Algorithm 1: Traditional Pearson Correlation

```
Input: Matrix input[N, M]
Output: Matrix answer[N, N]
 1.  i = 0, j = 0, k = 0
 2.  while i < N do
 3.    while j < N do
 4.      while k < M do
 5.        answer[i,j]+=input[i,j]× input[j,i]
 6.        sumX[i,j]+= input[j,k]× input[j,k]
 7.        sumY[i,j]+= input[i,k]× input[i,k]
 8.        k++
 9.      endwhile
10.      j++
11.    endwhile
12.    i++
13.  endwhile
14.  i = 0, j = 0
15.  while i < N do
16.    while j < N do
17.      answer[i,j]/=(√sumX[i,j] ×√sumY[i,j])
18.    endwhile
19.  endwhile
20.  return answer
```

Algorithm 1 shows a traditional PC when *N* and *M* are the amount of data and their corresponding dimensions, respectively. The *input* matrices contain the color value in the image, the average of which is subtracted. After the computation, the *answer* matrix, which consists of the list of *r* values for each image pair, is generated. Lines 1 to 13 illustrate how to calculate the summation of the input data for the second stage (to determine *r*). Lines 5 to 7 show the accumulative summation of $(X_i\text{-}X)$ $(Y_i\text{-}Y)$; $(X_i\text{-}X)^2$ and $(Y_i\text{-}Y)^2$-*sum X* and *sum Y*. Lines 15 to 19 are the computations based on Equation (1). Here, Line 17 is the actual calculation of *r*.

3. Related Work

Although several facial recognition techniques have been proposed [7, 11, 15, 16, 18, 19, 21, 25, 27, 33, 39], this study focuses on PCA derivatives and, in particular, noise-infected images and cosmetic/ makeup changes for women.

PCA-based techniques have been widely used for facial recognition. For example, in 2003, Zhao *et al.* [38] provided a comprehensive survey of facial recognition techniques and noted that the performance of PCA was outstanding [34]. The recognition rate was improved by Gumus *et al.* [12] by applying a hybrid approach of PCA and wavelets to extract facial features. In 2013, Bansal and Chawla [2] proposed a normalized PCA to remove lighting variations. Singular value decomposition was also applied to eliminate background effects. Recently, in 2015, So-In and Rujirakul [35] proposed a fast FRS using a fixed point algorithm considering the weighted factor, resulting in comparative accuracy with reduced complexity.

PCA and its optimizations provide improved precision; however, aside from the complexity trade-off, some limitations remain, one of which is the limitation of the application of PCA to makeup changes. Some researchers have studied these limitations and proposed solutions. For example, in 2005, Pamudurthy *et al.* [26] proposed a novel method using skin correlation by tracking the position of skin pores on the face with different expressions to obtain the face vector, which is then used for identification. However, this method is only suitable for high-resolution images.

In 2010, Ueda and Koyama [37] also studied the effect of makeup in two scenarios:

1. From the make-up images to the no cosmetics images.
2. From normal images to their make-up images. This analysis was provided by human interaction instead of image processing.

With the aid of image processing, in 2012, Moriguchi *et al.* [24] proposed a subspace-based method using PCA for quantitative analysis of the facial appearance. Two of the subspaces were divided into a global (using skin color) and local (using skin texture) subspace, with either high or low eigenvalues corresponding to its eigenvector. The classification results are comparable to a psychological test.

In 2013, Hu *et al.* [14] proposed another facial recognition approach for makeup changes using PCA; however, PCA was applied only for feature extraction. They adopted Canonical Correlation Analysis (CCA) to learn a meta feature subspace to project facial samples, both with and without makeup, into a common feature space to determine makeup invariance, which are then fed as the inputs for the final classification using a support vector machine.

In 2014, Guo *et al.* [13] implemented a system for facial recognition with makeup using correlation factors mapping between the explicit features, with nine local patches of makeup and non-makeup faces. In addition to recognition, a makeup detection process was performed using the characteristics of cosmetics, including skin color tone, skin smoothness, texture, and highlights. A marginal fisher analysis was used to combine selected patches of skin color and skin smoothness. Next, PCA was employed as a means of feature extraction, and partial least squares was used to map non-makeup and makeup features. Although they reported an increase in accuracy to 80% with 500 pairs of (non) makeup, the main limitation of the proposed approach was its computational complexity.

High-precision facial recognition has generally been achieved with high complexity [24]; thus, the concept of parallelism was also implemented for real-time applications. For example, in 2003, Chunhong *et al.* [6] applied a distributed system concept to facial recognition. The training database was divided into five sub-databases; each database computes in parallel. This approach is called Parallel PCA (P-PCA). Using multi-core processors, Rujirakul *et al.* [28] investigated how to speed-up PCA in parallel with a feature extraction optimizer using a fixed point. However, that work did not consider cosmetics.

In 2015, Rujirakul *et al.* [30] proposed to apply a statistical model using PC to increase the recognition rate for images with makeup changes. This proposal also applied parallelism to decrease the computational time. However, the precision of their approach can be improved further.

4. Parallel Optimized Pearson Correlation Condition

Figure 1 shows the total process of our proposed Approach-Parallel Optimized Pearson correlation (PO-PCC)-which consists of three main phases: pre-processing, feature extraction, and image identification. In this figure, based on our extensive evaluation, the parallelism will be applied only to the first two states, i.e., each pixel in the facial image will be processed in parallel, and the parallel computation of the PC will apply to parallel matrix manipulation, i.e., each matrix element will be calculated at the same time.

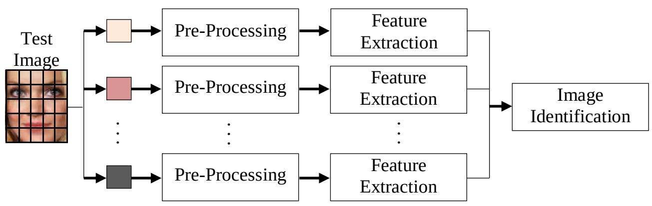


Figure 1. Our proposed method: Parallel Optimized Pearson Correlation Condition (PO-PCC), face recognition.

4.1. Parallel Pre-Processing

In general, cosmetics can change the appearance of color on the actual face compared to that of the ordinary face, thus directly impacting the recognition rate. To mitigate this effect, this research applies the HE as the pre-processing step. Based on our previous work [30], channel selection is found to change the precision. For example, the channel with the lowest standard deviation or fewer outliers will achieve a higher recognition rate. Therefore, in this research, the following two main steps of pre-processing are used:

1. *Channel selection*: this step was used to convert the test image into four channels, i.e., red/green/blue channels and a gray-scale image. Next, the system will calculate a standard deviation of each individual channel using Equation (2). Subsequently, the proper channel with the lowest standard deviation will be selected, as a lower standard deviation indicates fewer outliers.

$$\textit{Standard Deviation (SD)} = \sqrt{\sum(X - Mean)^2 \div N} \qquad (2)$$

2. *Parallel HE*: given the appropriate channel from the previous step, the system converts both training

and testing images into the proper channel. HE will be then performed on each image in parallel to further reduce the computational time. The details of HE calculation are as follows:

a) *Counting table generation*: the system first creates the counting table used to collect the frequency of the colors appearing in each image (one table per image). The colors are in the range between 0 and 255, where 256 is the table size. The system investigates the color in each image, and then, the accumulative counter in the stored table is increased.

b) *Cumulative counter*: given the generated table, the cumulative frequency of each image will be calculated using Equation (3).

$$\text{Cumulative Frequency } [j] = \sum_{i=0}^{j} (\text{Frequency } [i]) \quad (3)$$

c) HE: the last step is to update the new color value, which is calculated using Equation (4) [30]. Here, cf is the cumulative frequency of the $M \times N$ image, v is the observed color, and L is the overall color.

$$h[v] = round\,((cf[v]\text{-}cf_{min})\div((M\times N)\text{-}cf_{min}))\times(L\text{-}1) \quad (4)$$

Algorithm 2 shows how to apply parallelism for HE [30, 31]; here, *img* is the matrix of each color pixel, and *result* is the matrix of generated colors as the output. Lines 1 to 7 are used to determine the color in each pixel to count the frequency in the table (*freqMat*) for a given image width and height (*imgWidth* and *imgHeight*, respectively).

In addition, Lines 8 to 16 illustrate how to compute the accumulative frequency (*cumuMat*) based on *FreqMat*. Lines 11 to 14 show the actual computation for the lowest frequency with its position (*min* and *minPos*). Lines 17 to 23 show the determination of the new color based on Equation (4) (line 20).

For example, suppose that there are 4 CPU cores. The four observed colors will be checked, and their frequency will be updated in the counting table in parallel. Once every pixel in all images is counted, the system will then compute the cumulative frequency using Equation (3) in sequence. Finally, the new color will be updated using Equation (4) in parallel.

Algorithm 2: Parallel Histogram Equalization [31]

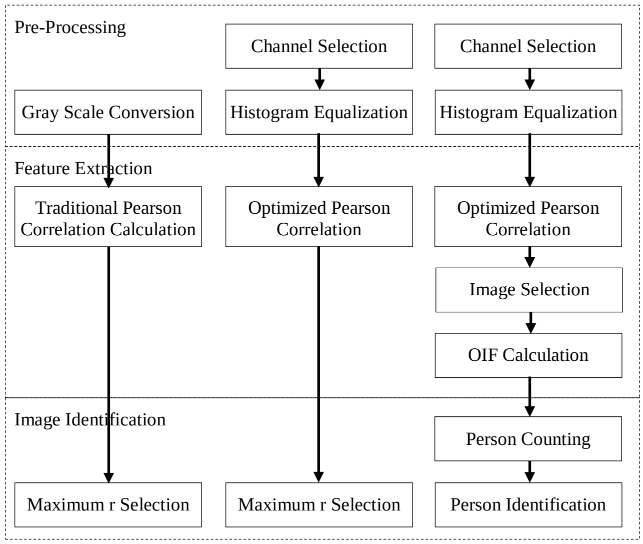
Input: Matrix img$_{[imgWidth,\ imgHeight]}$
Output: Matrix answer$_{[imgHeight\times\ (1+imgWidth)]}$

```
1.  parallel for i from 0 to imgWidth
2.    j = 0
3.    while j < imgHeight
4.      freqMat[img[i,j]]=freqMat[img[i,j]]+1
5.      j++
6.    endwhile
7.  endfor
8.  j = 0
9.  while j < 255
10.   cumuMat[j]+=freqMat[j]
11.   if ((cumuMat[j]>0)&&(cumuMat[j]<min))
12.     min=cumuMat[j]
13.     minPos=j
14.   endif
15.   j++
16. endwhile
17. parallel for i from 0 to imgWidth
18.   j = 0
19.   while j < imgHeight
20.     answer[j+imgHeight×i]=(cumuMat[img[imgHeight
        ×i+j]]cumuMat[minPos])÷(((imgWidth×imgHeight)
        -cumuMat[minPos])×255)
21.       j++
22.   endwhile
23. endfor
24. return answer
```

4.2. Parallel Feature Extraction

In this research, feature extraction is enhanced from our previous work [30], which only applied PC; here, a new ranking technique is applied, namely, the Optimum Index Factor (OIF) [4].

For clarity, Figure 2 illustrates difference stages of facial recognition for different PC derivatives, i.e., PC, PCC [30], and O-PCC (our proposal). Here, in addition to the Optimized PC (OPC) stage, two additional steps are proposed, i.e., image selection and OIF calculation.



a. Traditional PC. b. PCC. c. O-PCC.

Figure 2. Facial recognition: pre-processing, feature extraction, image identification: traditional PC, PCC, and O-PCC.

Compared to a traditional PC using gray-scale conversion in the pre-processing step, PC derivatives apply four channels with its proper selection, including HE. Our proposed method applies two steps for image identification, i.e., Pearson counting and identification, as opposed to the other two methods, which only apply the maximum CC.

4.2.1. Optimized Pearson Correlation

Once PC is applied in facial recognition (traditional PC), the relationship among training images is not necessary because PC only requires the relationship between the training and testing images. As a result, in the proposed approach, the time complexity is optimized by removing the non-necessary iteration (2 *for loop*), as illustrated in lines 1 and 10 in Algorithm 1. In other words, the variable *i* will be replaced by 0. Here, the computational time will be reduced from $O(mn^2+n^2)$ to $O(mn+n)$.

Comparing the computational complexity of OPC considering gray-scale and RGB images, the final computational time will be $O(mn+n)$ for each channel-$O(4mn+n)$; however, only the best selection channel reduces its complexity to $O(mn+n)$.

4.2.2. Image Selection

Based on our observation from the intensive evaluation reported in [30], although PCC can improve the recognition rate based on the computation of all channels, this complex calculation will result in a high computational time. Thus, to avoid this complexity, we propose a method to limit the number of training images instead of using all of them. In this case, the system will consider the *r* (correlation coefficient) value as a channel selection criterion, i.e., the system will only retain the training images whose *r* values are greater than or equal to their averages.

4.2.3. OIF Calculation

After performing the above two steps, the result is a set of training images, all of which have *r* values greater than or equal to the mean *r*. In this step, the system will then calculate the *r* value of this set in each channel, i.e., gray-scale, red, green and blue, and then select a particular training image with the maximum *r* of each channel. As a result, only four images are used.

Based on the evaluation observation, including the discussion of related work [30], the recognition rate can be improved further if the standard deviation of the training images is considered. Thus, this research applies the OIF, a statistical value derived from the standard deviation that was first introduced by Chavez *et al.* [4], to improve the accuracy. The OIF can be calculated using Equation (5). Here, σ is the standard deviation of each training image and *r* is the CC derived from PC.

$$OIF = \sigma \div r \tag{5}$$

4.3. Parallel Optimized Pearson Correlation Condition

Parallelism is applied in our proposed approach to increase its speed; the resulting approach is called Parallel Optimized Pearson Correlation (PO-PCC). Here, the main advantages of parallel processing are parallel matrix manipulation of basic operations and its enhancement for PC.

4.3.1. Parallel Matrix Manipulation for Basic Operations

In this research, most of the calculations are based on matrix manipulation. From our observation, these calculations can perform in parallel to decrease the computational time. According to the parallel matrix method proposed by Rujirakul *et al.* [31], the five main matrix manipulations are as follows: subtraction, addition, multiplication, division, and transposition. These are the basic matrix operations consist of either matrix multiplication with another matrix or between a matrix and constant.

4.3.2. Parallel Matrix Manipulation for PC

In addition to basic matrix operations, Rujirakul *et al.* [30] recommended three additional operations used for PC in parallel: subtraction/multiply/division; sum each row; and power and square root.

Algorithm 3 also shows these calculations; here, the left is the input of an $M \times N$ matrix, but the right matrix is only one dimensional. Another input string is used to indicate the computational operation. The function *Parallel_for* [22] in .NET C# was used to lead the process work as concurrency.

Algorithm 3: Parallel Matrix Manipulation for Pearson Correlation

Input: Matrix left$_{[M, N]}$, Matrix right$_{[M]}$, String operation
Output: Matrix answer

```
1.  parallel for i from 0 to M
2.    j = 0
3.    while j < N
4.      switch operation do
5.        case "Subtract"
6.          answer[i,j]=left[i,j]-right[i]
7.        case "Multiply"
8.          answer[i,j]=left[i,j]×right[i]
9.        case "Divide"
10.         answer[i,j]=left[i,j]÷right[i]
11.       case "Sum each"
12.         answer[i]+=left[i,j]
13.       case "Power"
14.         answer[i,j]=(left[i,j])^2
15.       case "Root"
16.         answer[i,j]=√left[i,j]
17.       j++
18.   endwhile
19. endfor
20. return answer
```

1. Subtraction, multiplication, and division (each of the $M \times N$ elements with a one-dimensional matrix): the actual computation will be performed based on Equations (6), (7), and (8), respectively. Here, because each matrix element of the output matrix is independent, each element can be calculated in parallel as shown in lines 5-10 of Algorithm 3.

$$c[i,j]=a[i,j]-b[i] \tag{6}$$

$$c[i,j]=a[i,j]\times b[i] \tag{7}$$

$$c[i,j]=a[i,j]\div b[i] \tag{8}$$

2. *Sum each row*: the summation of each row in the $M \times N$ matrix can be calculated using Equation (9), in which the parallelism approach can be applied because their calculations are independent, as shown in lines 11-12 of Algorithm 3.

$$c[i]=\sum_{J=0}^{cols} a[i,j] \tag{9}$$

3. *Power and square root*: in every $M \times N$ element, the power or square root operator, which is primarily used during OPC, can be performed in parallel because the calculation is independent, as shown in lines 13-16 of Algorithm 3. Here, the final result of the power and square root operations can be computed using Equations (10) and (11), respectively.

$$c[i,j]=(a[i,j])^2 \tag{10}$$

$$c[i,j]=\sqrt{a[i,j]} \tag{11}$$

4.4. Facial Classification

With feature extraction, the system will determine *r* and the OIF of the selected training images. These values are then used for image identification. Several research studies [7, 38] have applied the concept of voting to search for the answer. Our facial classification scheme also applies the voting technique to select a proper image as the answer. The details of our facial classification approach are as follows:

1. *Person counting*: after determining the maximum *r*'s training image from each channel, the system will then check the person name of each image. An extra step is used to count the frequency of each person by checking if that person is already on the list. If the person is on the list, then the system will increase this person's frequency by one; otherwise, the system will add the name to the list and reset the frequency to 1.

This process will be iteratively continued until all four images are completed, as shown in Algorithm 4 (Lines 1 to 10). Here, *imgMaxR* is the image matrix with the maximum *r* of four channels that correspond to their identification *imgMaxR.Name*. *countList* is the list that is used to collect the name (*countList.name*), frequency (*countList.freq*), and OIF (*countList.OIF*) for person identification purposes. Line 3 is the process for checking whether the person is already on the list; if the person is on the list, Line 4 increments the frequency. Lines 6 to 7 are used to add a new person to the list with the increased frequency.

2. *Person identification*: after counting the frequency of each person in all four selected images, the system will consider the person with the maximum frequency as the final identification, i.e., majority voting. When one person with the maximum frequency is identified, this person will be the answer; if two or more people with the maximum frequency are identified, the system will compare the OIFs of these people, and instead, the person with the minimum OIF will be selected.

The identification process is shown in lines 11-16 of Algorithm 4, in which the maximum frequency checking is stated in lines 11-12. Here, *countMaxFreq* is a function to determine the person with the highest frequency, storing the value into *maxFreqCount*. *answer* is the identified person. *countList[maxFreqPos].name* and *countList[minOifPos].name* contain the person with the maximum frequency and minimum OIF, respectively.

Algorithm 4: Proposed Image Identification

```
Input: Matrix imgMaxR[4]
Output: String answer
1.  i = 0
2.  while i < 3
3.    if (imgMaxR[i].Name in countList)
4.      countList[imgMaxR[i].Name].freq++
5.    else
6.      countList.Add(imgMaxR[i].Name)
7.      countList[imgMaxR[i].Name].freq++
8.    endif
9.    i++
10. endwhile
11. maxFreqCount=countMaxFreq(countList)
12. if (maxFreqCount==1)
13.   answer=countList[maxFreqPos].name
14. else
15.   answer=countList[minOifPos].name
16. endif
17. return answer
```

5. Performance Evaluation

To evaluate the feasibility of PO-PCC, we performed a comparative performance evaluation using two different main scenarios: O-PCC and PO-PCC.

5.1. Experimental Configuration

Our testbed is a standard configuration on a personal computer Windows 7 Ultimate operating systems (64 bits): CPU Intel(R) Core(TM) i-3770K 8-Cores 3.50 GHz (8 MB L3 Cache), 500 GB 5400 RPM Disk, and 8192×2 MB DDR3-SDAM.

NET C# [22] programming was used to implement the algorithm to evaluate the face recognition system. Thus, the actual computational time can be measured in a multi-core architecture on a single machine in addition to the recognition precision. These two metrics are the main metrics, including their standard deviations for this research. A standard 5-fold cross-validation was applied in each setup [13, 14].

Two well-known makeup image datasets were used: YouTube Makeup (YMU) and Virtual Makeup (VMU) [5, 9], which are generally used for comparative analysis in most related works [28, 29, 31]. The images in both databases are 24-bit RGB in PNG 130×150 pixel formats in both makeup and non-makeup facial images. There are 257 images for VMU divided into 4 groups: no makeup, full makeup, and only makeup on the lips or eyes. Of the 257 images, 51×4 are used for each class, and the remaining images (53) are only the use of full makeup. In YMU (604 in total), each subject (person) has 2 images, one for makeup and the other for non-makeup (151 people in total) [30].

To evaluate the diversity of nationalities in further detail, another dataset was created from Google search with "non-makeup" as a keyword in the facial images, which we called Google Makeup (GMU) [32]. Here, the nationality selection was based on the classification provided by Sanjeev and Harpreet [18]: Mongoloid, Caucasoid, and Negroid. There are also several pre-processing steps, i.e., the images were resized into 201×269 pixels, and all of the eye positions were aligned. Similarly to VMU, the GMU database was processed further to generate their corresponding makeup images using the "taaz" [36].

The GMU database contains 1,768 images in four groups (442 subjects): no makeup, full makeup, only makeup on the lips, and only makeup around the eyes; as a result, a total of 442×4 images are obtained. Due to the limitation of computer memory usage during the computations, only 150×4 images were employed.

The evaluation process was divided into three scenarios. First, O-PCC was evaluated against the existing facial recognition methods, i.e., PC, LBP, and PCA, including our Previous Proposal (PCC).

Second, the parallelism of each approach was investigated, i.e., PO-PCC, P-PC, P-LBP, P-PCA [6], and P-PCC. With P-PCA, the database was divided into sub-databases depending on the number of CPU cores in parallel. All CPU cores (8) were applied to achieve the highest performance.

With the constraint of the size limitation of these three datasets, these two scenarios varied the number of trained images as follows: the range of 50 to 204 (by a factor of 2); the range of 100, 200, 400, and 604; and the range of 100, 200, 400, and 600 for VMU, YMU, and GMU, respectively. In addition, another measurement was also performed to state the degree of parallelization by altering the number of CPU cores (from 1 to 2, 4, and 8 cores) with our parallel optimization model (PO-PCC) using the maximum number of images in all datasets.

5.2. Experimental Results and Discussion

The first scenario (non-parallelism) with the VMU database is shown in Figures 3 and 4. Our proposed method-O-PCC-outperforms the other methods, i.e., it requires only 6.63 s, which is faster than the other approaches (PCA, PC, and PCC) by factors of 4, 5, and 0.4, respectively (Figures 3 and 4 shows the recognition accuracy of O-PCC, which is also distinctive, i.e., 100% compared with the other approaches, i.e., 96.10%, 100%, and 100% for PCA, PC, and PCC, respectively.

In addition, these two figures also show the results in the second scenario. Our parallelism optimization model-PO-PCC-yields the best performance. For example, the computational time and precision of PO-PCC, P-PCA, P-PC, and P-PCC are 7.39, 7.81, 34, and 2.34 s, respectively, and 100%, 70.59%, 100%, and 100%, respectively. LBP and its parallelism version can be recognized only with 100 trained images due to the intensive computational time required.

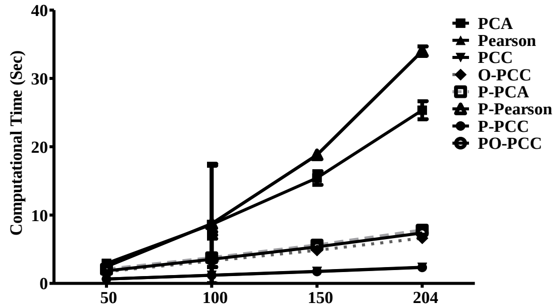


Figure 3. Computational time versus the number of trained images of the VMU database.

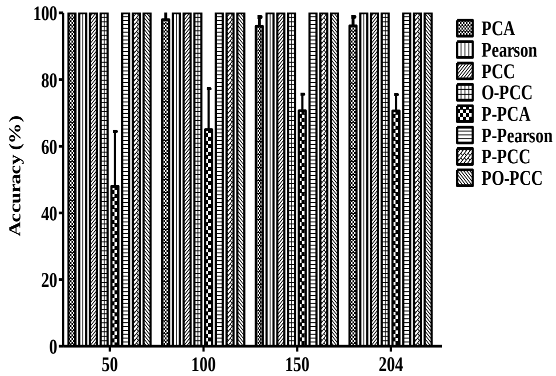


Figure 4. Percentage accuracy versus the number of trained images of the VMU database.

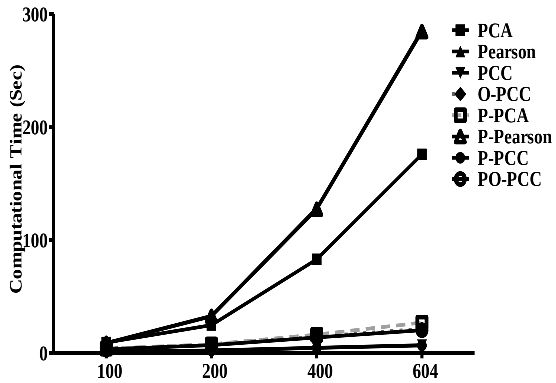


Figure 5. Computational time versus the number of trained images of the YMU database.

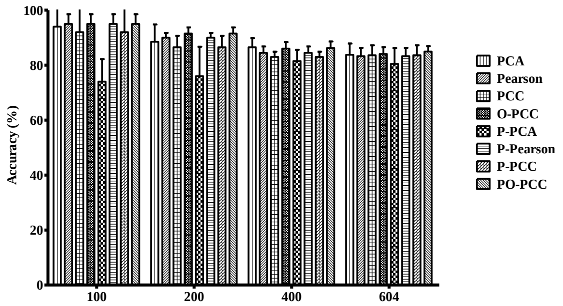


Figure 6. Percentage of accuracy versus the number of trained images of the YMU database.

For example, although their accuracies can reach 100%, the computational times are higher than 8,301.61 and 841.03 seconds, which are excessively high.

Figures 5 and 6 show the recognition performance for the YMU database, which is similar to the recognition performance for VMU. The non-parallel version of O-PCC maintains the best performance, i.e., only 21.32 s, which is faster than PCA, PC, and PCC by factors of 8, 13, and 0.3, respectively. Additionally, O-PCC is more accurate than the other approaches, i.e., 84.10% versus 83.78%, 83.28%, and 83.61% for PCA, PC, and PCC, respectively. With the parallelism, the proposed method is still outstanding in both computational time and accuracy, i.e., 20.27, 26.89, 284.53, and 6.78 s and 84.93%, 80.47%, 83.28%, and 83.61%, for the parallel version of O-PCC, PCA, PC, and PCC, respectively.

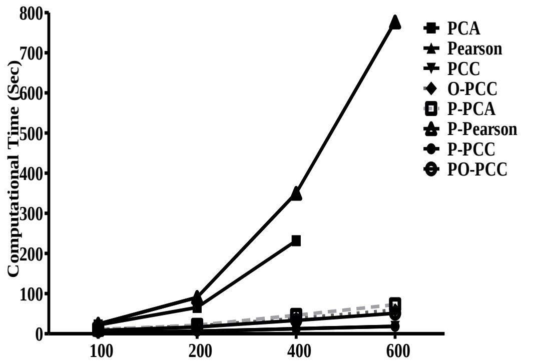


Figure 7. Computational time versus the number of trained images of the GMU database.

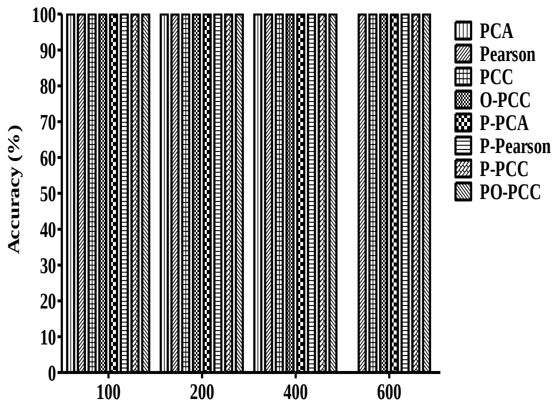


Figure 8. Percentage of accuracy versus the number of trained images of the GMU database.

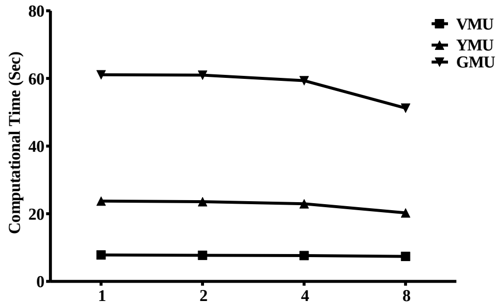


Figure 9. Computational time versus the number of cores of all three databases.

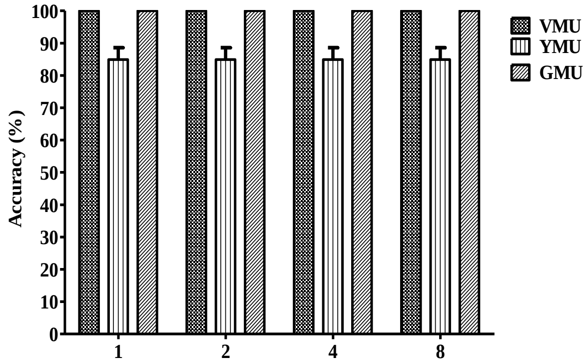


Figure 10. Percentage of accuracy versus the number of cores of all three databases.

Note that again, with 100 trained images, the computational time and accuracy of LBP and its parallelism version are as follows: 1,667.71 and 1,621.68 s and 91% and 91%, respectively.

Similar to VMU and YMU, Figures 7 and 8 show the recognition performance for the GMU database; the non-parallel version of O-PCC maintains excellent performance, i.e., 59.14 s, which is faster than PCA, PC, and PCC by factors of 6, 13, and 0.3, respectively. All algorithms achieve 100% in accuracy. Although 600 trained images are used, the PCA cannot continue the recognition due to the memory exception.

With parallelism, PO-PCC has the lowest time computation, i.e., 51.24, 72.68, 776.90, and 18.45 s for the parallel version of O-PCC, PCA, PC, and PCC, respectively. However, all algorithms with parallelism can achieve 100% accuracy, except P-PCA, which can only achieve an accuracy of 99.83%. Again, the computational time of LBP and its parallelism version with 100 trained images are 258,482.23 and 13,074.75 s, respectively, and the accuracy is 100% in both cases.

Regarding the third scenario while diversifying the number of cores of our proposal (PO-PCC), Figures 9 and 10 show that increasing the number of cores can lower the time computation, i.e., from 7.85 to 7.39 s, from 23.76 to 20.27 s, and from 61.12 to 51.24 s for the VMU database, YMU database, and GMU database, respectively, which has no effect on the accuracy (regarding an increase in the number of cores).

6. Conclusions and Future Work

This research proposed a novel FRS called PO-PCC for use as a robust system for makeup change or cosmetic facial images. Different approaches, including PC, channel selection, HE, and the OIF, were investigated to improve the recognition precision and reduce the computational time, especially in the feature extraction stage. To further reduce time complexity with consideration of large-scale datasets and real-time applications, another investigation was also performed to consider the degree of parallelism in a single machine with a multi-core structure, particularly in the HE and PC stages.

In addition to two well-known cosmetic datasets, VMU and YMU, GMU was also constructed as a more practical dataset (from Google searches) for our intensive evaluation; from these databases, PO-PCC was found to outperform the other recognition approaches (no parallelism), i.e., PCA, LBP, PC, PCC, and O-PCC, by factors of 6.98, 31194.02, 15.20, 0.37, and 1.15, respectively. With their corresponding parallel schemes, the speed of PO-PCC is still superior to that of the other approaches by factors of 1.42, 1577.88, 2.91, and 0.39 for PO-PCC, P-PCA, P-PC, and P-PCC, respectively.

With regard to the recognition rate, all algorithms can maintain a value of 100%, except parallel PCA (99.83%). In addition, the precision is not affected by variations in the number of cores (still 100%). LBP is infeasible in terms of computational time complexity (more than 3 h). Furthermore, although PO-PCC can achieve a significant increase in computational speed while maintaining high accuracy, further investigations and analyses should be performed, e.g., a makeup face with multi-views and heterogeneously including further parallelism optimizations in each recognition stage; these are topics for future work.

References

[1] Ahmed F., Bari H., and Hossain E., “Person-Independent Facial Expression Recognition Based on Compound Local Binary Pattern (CLBP),” *The International Arab Journal of Information Technology*, vol. 11, no. 2, pp. 195-203, 2014.

[2] Bansal A. and Chawla P., “Performance Evaluation of Face Recognition Using PCA and N-PCA,” *International Journal of Computer Applications*, vol. 76, no. 8, pp. 14-20, 2013.

[3] Chan L., Salleh S., Ting C., and Ariff A., “Face Identification and Verification Using PCA and LDA,” *in Proceedings of International Symposium on Information Technology*, Kuala Lumpur, pp. 1-6, 2008.

[4] Chavez P., Berlin G., and Sowers L., “Statistical Methods for Selecting LandSat MSS Ratios,” *Journal of Applied Photographic Engineerin*, vol. 8, no. 1, pp. 23-30, 1982.

[5] Chen C., Dantcheva A., and Ross A., “Automatic Facial Makeup Detection with Application in Face Recognition,” *in Proceedings of International Conference on Biometrics*, Madrid, pp. 1-8, 2013.

[6] Chunhong J., Guangda S., and Xiaodong L., “A Distributed Parallel System for Face Recognition,” *in Proceedings of International Conference on Parallel and Distributed Computing, Applications, and Technologies*, Chengdu, pp. 797-800, 2003.

[7] Dagher I., Hassanieh J., and Younes A., “Face Recognition Using Voting Technique for the Gabor and LDP Features,” *in Proceedings of International Joint Conference on Neural Network*, Dallas, pp. 1-6, 2013.

[8] Dandotiya D., Gupta R., Dhakad S., and Tayal Y., “A Survey Paper on Biometric Based Face Detection Techniques,” *International Journal of Software and Web Sciences*, vol. 4, no. 2, pp. 67-76, 2013.

[9] Dantcheva A., Chen C., and Ross A., “Can Facial Cosmetics Affect the Matching Accuracy of Face Recognition Systems?” *in Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Syst*ems, Arlington, pp. 391-398, 2012.

[10] Dantcheva A., Ross A., and Chen C., “Makeup Challenges Automated Face Recognition Systems,” *SPIE-The International Society of Optics and Photonics*, pp. 1-4, 2013.

[11] Givens G., Beveridge J., Draper B., Grother P., and Phillips P., “How Features of the Human Face Affect Recognition: A Statistical Comparison of Three Face Recognition Algorithm,” *in Proceedings of IEEE Comput Society Conference Computer Visual Pattern Recognition*, Washington, pp. 381-388, 2004.

[12] Gumus E., Kilic N., Sertbas A., and Ucan O., “Evaluation of Face Recognition Techniques Using PCA, Wavelets and SVM,” *Expert Systems with Applications*, vol. 37, no. 9, pp. 6404-6408, 2010.

[13] Guo G., Wen L., and Yan S., “Face Authentication with Makeup Changes,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 814-825, 2014.

[14] Hu J., Ge Y., Lu J., and Feng X., “Makeup-Robust Face Verification,” *in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, pp. 2342-2346, 2013.

[15] Huang D., Shan C., Ardabilian M., Wang Y., and Chen L., “Local Binary Patterns and Its Application to Facial Image Analysis: A Survey,” *IEEE Transactions Systems Man,*

Cybern, C Applications and Reviews, vol. 41, no. 6, pp. 765-781, 2011.

[16] Jafri R. and Arabnia H., "Survey of Face Recognition Techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41-68, 2009.

[17] Kaur A., Kaur L., and Gupta S., "Image Recognition Using Coefficient of Correlation and Structural Similarity Index in Uncontrolled Environment," *International Journal of Computer Applications*, vol. 59, no. 5, pp. 32-39, 2012.

[18] Kumar S. and Kaur H., "Face Recognition Techniques: Classification and Comparisons," *International Journal of Information Technology and Knowledge Management*, vol. 5, no. 2, pp. 361-363, 2012.

[19] Lu X., "Image Analysis for Face Recognition," Personal Notes, Available online at http://www.face-rec.org/interesting-papers/General/ImAna4FacRcg_lu.pdf, 36 pp., Last Visited, 2003.

[20] Luo Y., WU C., and Zhang Y., "Facial Expression Feature Extraction Using Hybrid PCA and LBP," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 2, pp. 120-124, 2013.

[21] Meng K., Su G., Li C., Fu B., and Zhou J., "A High Performance Face Recognition Sys-Tem Based on a Huge Face Database," *in Proceedings of International Conference on Machine Learning and Cybernetics*, Guangzhou, pp. 5159-5164, 2005.

[22] Microsoft, "MSDN", "Parallel.For Method," "NET Framework 4.5. 2014", Available online at http://msdn.microsoft.com/enus/library/dd783539%28v=vs.110%29.aspx, Last Visited, 2014.

[23] Min L., Bo L., and Bin W., "Comparison of Face Recognition Based on PCA and 2DPCA," *Advances in Information Sciences and Service Sciences*, vol. 5, no. 6, pp. 545-553, 2013.

[24] Moriguchi J., Igarashi T., Nakao K., and Chen Y., "Dual-Subspaces Based Quantitative Analysis of Facial Appearance," *in Proceedings of International Conference on Software Engineering and Data Mining*, Chengdu, pp. 652-656, 2010.

[25] Pali V., Goswami S., and Bhaiya L., "An Extensive Survey on Feature Extraction Techniques for Facial Image Processing," *in Proceedings of International Conference on Computer Intelligence and Communication* Networks, Bhopal, pp. 142-148, 2014.

[26] Pamudurthy S., Guan E., Mueller K., and Rafailovich M., "Dynamic Approach for Face Recognition Using Digital Image Skin Correlation. Audio- and Video-Based Biometric Person Authentication," *in Proceedings of International Conference on Audio-and Video-Based Biometric Person Authentication*, Hilton Rye Town, pp. 1010-1018, 2005.

[27] Rahim A., Hossain N., Wahid T., and Azam S., "Face Recognition Using Local Binary Patterns," *Global Journal of Computer Science and Technology Graphics and Vision*, vol. 13, no. 4, pp. 1-7, 2013.

[28] Rujirakul K., So-In C., Arnonkijpanich B., Sunat K., and Poolsanguan S., "PFP-PCA: Parallel Fixed Point PCA Face Recognition," *in Proceedings of International Conference on Intelligent Systems, Modelling and Simulation*, Bangkok, pp. 409-414, 2013.

[29] Rujirakul K., So-In C., and Arnonkijpanich B., "PEM-PCA: A Parallel Expectation-Maximization PCA Face Recognition Architecture," *The Scientific World Journal*, vol. 2014, pp. 1-16, 2014.

[30] Rujirakul K., So-In C., and Arnonkijpanich B., *Information Science and Applications*, Springer, 2014.

[31] Rujirakul K., So-In C., and Arnonkijpanich B., "Weighted Histogram Equalized PEM-PCA Face Recognition," *in Proceedings of International Computer Science and Engineering Conference*, Khon Kaen, pp. 144-150, 2014.

[32] Rujirakul K. and So-In C., "Makeup (GMU)". Available online at http://web.kku.ac.thchakso/faceDBweb/face_dataset.html, Last Visited, 2014.

[33] Shah D., Shah J., and Shah T., "The Exploration of Face Recognition Techniques," *International Journal of Application or Innovation in Engineering and Management*, vol. 3, no. 2, pp. 238-246, 2014.

[34] Shah J., Sharif M., Raza M., and Azeem A., "A Survey: Linear and Nonlinear PCA Based Face Recognition Techniques," *The International Arab Journal of Information Technology*, vol. 10, no. 6, pp. 536-545, 2013.

[35] So-In C. and Rujirakul K., "wPFP-PCA: Weighted Parallel Fixed Point PCA Face Recognition," *The International Arab Journal of Information Technology*, vol. 13, no. 1, pp. 59-69, 2016.

[36] TAAZ Virtual Makeover & Hairstyles, Available online at http://www.taaz.com, Last Visited, 2015.

[37] Ueda S. and Koyama T., "Influence of Make-Up on Facial Recognition," *Perception*, vol. 39, no. 2, pp. 260-264, 2010.

[38] Zhao W., Chellappa R., Phillips P., and Rosenfeld A., "Face Recognition: a Literature Survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003.

[39] Zhou H., Mian A., Wei L., Creighton D., Hossny M., and Nahavandi S., "Recent Advances on Singlemodal and Multimodal Face Recognition: A Survey," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 6, pp. 701-716, 2014.

Kanokmon Rujirakul received an MSc. from NIDA in 2008 and is currently a Ph.D. candidate in the Department of Computer Science, Khon Kaen University. Her research areas include image processing, mobile computing, soft computing, and distributed systems.

Chakchai So-In received a Ph.D. from WUSTL in 2010 and is currently an associate professor in the Department of Computer Science, Khon Kaen University. His research interests include mobile computing, signal processing, network systems, and parallel and distributed Systems.