Genetic-Based Type II Feature Extraction for Periocular Biometric Recognition: Less is More

Joshua Adams Damon L. Woodard Gerry Dozier Philip Miller Kelvin Bryant George Glenn

***Abstract*— Given an image from a biometric sensor, it is important for the feature extraction module to extract an original set of features that can be used for identity recognition. This form of feature extraction has been referred to as Type I feature extraction. For some biometric systems, Type I feature extraction is used exclusively. However, a second form of feature extraction does exist and is concerned with optimizing/minimizing the original feature set given by a Type I feature extraction method. This second form of feature extraction has been referred to as Type II feature extraction (feature selection). In this paper, we present a genetic-based Type II feature extraction system, referred to as GEFE (Genetic & Evolutionary Feature Extraction), for optimizing the feature sets returned by Loocal Binary Pattern Type I feature extraction for periocular biometric recognition. Our results show that not only does GEFE dramatically reduce the number of features needed but the evolved features sets also have higher recognition rates.**

I. Introduction

Genetic & Evolutionary Computing (GEC) is the field of study devoted to the design, development, and analysis of problem solvers based on artificial/simulated evolution [1], [2], [3]. Most GECs evolve a population of individuals (also known as chromosomes or candidate solutions (CSs)) in order to discover an optimal or near-optimal solution. A typical GEC works as follows. Initially a randomly generated population of CSs is created. Next, a fitness is assigned to each CS of the initial population based on a user-defined evaluation function. The fitness assigned to a CS by the evaluation function provides a measure of the 'goodness' of that solution in terms of solving the problem at hand.

After each CS of the initial population has been evaluated, two parents are selected[1] based on their fitness and are allowed to create offspring. Offspring are then evaluated and typically replace the worst members of the current population. The evolutionary process of selection, procreation, and replacement is continued for a user-specified number of function evaluations. Figure 1 provides a pseudo-code version of a GEC.

GECs have been used to solve a wide variety of problems [1], [2], [3]. GECs have been successfully used for feature extraction [4], [5]. In [6], Adams et al. describe two forms of feature extraction: Type I and Type II (feature selection).

Joshua Adams, Gerry Dozier, Kelvin Bryant, and George Glenn are with CASIS at the Department of Computer Science, North Carolina A&T State University, Greensboro, NC 27411, USA (phone: 336-334-7245; email: {jcadams2 | | gvdozier | ksbryant | gtglen}@ncat.edu

Philip Miller and Damon L. Woodard are with CASIS at Clemson University, Clemson, SC 29634, USA email: {pemille |woodard}@clemson.edu

[1]Two parents are recombined in the case of genetic computations and one parent is mutated in the case of evolutionary Computations.

```
Procedure GEC{
    t = 0;
    Initialize Pop(t); /* Initial Population */
    Evaluate Pop(t);
    while (While Not Done){
      dad = Select_Parent(Pop(t)); /* Dad */
      mom = Select_Parent(Pop(t)); /* Mom */
      offspring = Create_Offspring(mom,dad):
      Evaluate(Offspring);
      Pop(t+1) = Replace(worst,Offspring);
      t = t + 1;
    }
}
```

Fig. 1. Pseudo-Code Example of a GEC

Type I feature extraction is used by most biometric systems for extracting an original feature set [7], [8], [9], [10], [11]. Type II feature extraction on the other hand can be viewed as feature set optimization/minimization. Given a original feature set the objective is to either reduce the dimensionality without affecting recognition accuracy and/or discover a subset of features that increase the recognition rate beyond that of the original feature set.

In this paper, we present a genetic-based Type II feature extraction method for periocular biometric recognition. Our results show that this method minimizes the number of features needed in addition to optimizing the recognition accuracy on datasets taken from two well known databases within the biometrics research community [12], [13], [14]. The remainder of this paper is as follows. Section 2 provides a brief overview of periocular feature extraction & recognition [10], [15]. In Section 3, GEFE [6] (Genetic & Evolutionary Feature Extraction) is introduced, in Section 4 we describe our experiments, and in Section 5 we present our results. In Section 6, we present our conclusions and future work.

II. Periocular Feature Extraction and Recognition

Biometric recognition systems function by acquiring a biometric sample, extracting a set of features from that sample and then encoding those features into a convenient form for comparison. In the context of this research, features are extracted from an image sample of the region immediately surrounding the eyes using a technique introduced by Ojala et al. [15] termed Local Binary Patterns (LBP). Figure 2 shows a sample image of the periocular region.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.59

LPB extracts features by subdividing the image sample into smaller regions, called patches, and then creating a histogram for each patch by analyzing pixel-to-pixel intensity changes within each patch. In essence, LBP facilitates the extraction of the unique textural features in the periocular region of the face. More specifically, LBP defines the function, $LBP_{P,R}$, which is a measure of the intensity change of P pixels surrounding a center pixel with a radius, R, within a patch. This research uses a P value of 8 and an R value of 1 so that all interior patch pixels serve as a center pixel for the $LBP_{P,R}$ function. The actual intensity change between the P pixels on the radius is represented as a set T of the sign of the result of subtracting the intensity value, I, of the center pixel from each of the P pixel intensity values on the radius. Positive and differences of zero are represented by a value of one while negative differences are represented by zero:

$$T = \{s(I_0 - I_c), \ldots, s(I_{P-1} - I_c)\}.$$

LBP gives a unique value representing the intensity texture by weighting the values in T:

$$LBP_{P,R} = \sum_{p=0}^{P-1} T_p * 2^p$$

In addition to the intensity differences between the center pixel and the pixels on the radius, we also consider the intensity value, $s(I_p - I_c)$, changes between adjacent pixels on the radius. The number of intensity value changes of these neighboring pixels is termed the degree of uniformity. Our implementation of the LBP algorithm gives significance to uniformity degrees of two. If we consider our $P = 8$ bits to be arranged in a circle, there can be $P-1$ bit patterns where a sign change occurs exactly two times (e.g. 00111110). If we take any of these patterns with a degree of uniformity of two and shift them around in modulo fashion we can create P different variations of each of the P-1 bit patterns. Thus we have $P(P-1)$ or 56 possible textures/features. There is also a pattern with all zeros, a pattern with all ones and patterns with a degree of uniformity greater than two (we group all the patterns with degree of uniformity ¿ 2 into a single texture group). So, there are a total of 56 + 3 = 59 total possible textures. Our histogram has locations for each of these patterns and records the number of times each of the 59 patterns appear in a particular patch [10].

The periocular images used for this research were divided into 24 patches where each patch contained a matrix of 25 x 25 pixels (See Figure 3). The corresponding histogram for Figure 2 is shown in Figure 4. As described earlier, the features of each patch is described with a 59 bin histogram and since we have a total of 24 patches, there are 24 x 59 = 1416 total bins for the total combined histogram for the image. The combined histogram of all 24 patches can be viewed as a vector (template [8]), h_i, of 1416 integers where i represents the index of a particular individual. Using the features of the histogram (i.e. the number of patterns in each bin), recognition is performed by comparing a captured probe template with all the vectors in a gallery set $H = \{h_0, h_1, ..., h_{m-1}\}$. Template comparison is achieved by computing the Manhattan distance metric (City Block), $d(h_i, h_j)$ between two templates. The individual from the gallery set with the shortest distance to the probe template is considered the match for the probe.

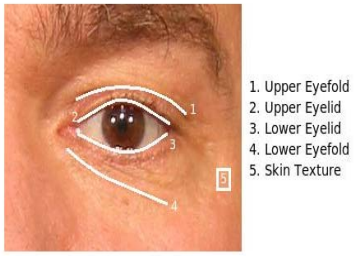

Fig. 2. An Example of a Periocular Region of an Eye

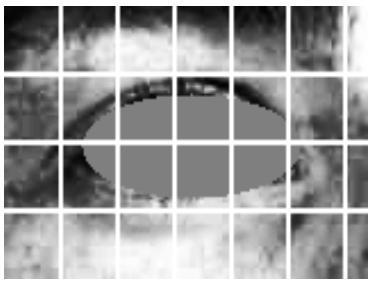

Fig. 3. The 24 different 25 x 25 pixel patches of the periocular region. $P = 8$ for a patch results in a histogram consisting of 59 features. With 24 patches, the total number of features for the image is 1416.

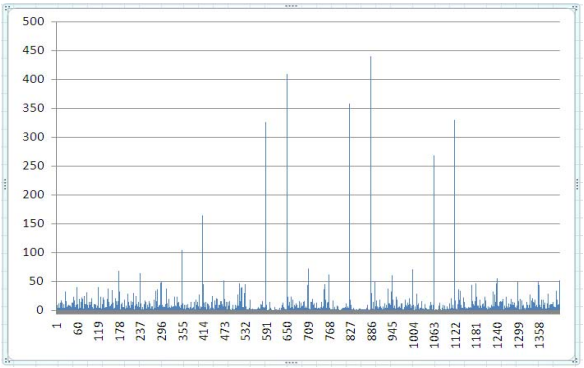

Fig. 4. An Example Histogram of an individual consisting of 1416 features corresponding to the 24 different 25 x 25 patches of the periocular region shown in Figure 2.

III. GEFE: Periocular Feature Set Optimization using X-TOOLSS

GEFE is based on The eXploratory Toolset for the Optimization Of Launch and Space Systems (X-TOOLSS) [16] which is suite of 11 GECs which interface with evaluation functions expressed as executables of any programming language or scripts. The GEC used to generate the results presented in this paper is an instance of the X-TOOLSS

Steady-State Genetic Algorithm (SSGA) which we will refer to as GEFE$_{ssga}$. In GEFE$_{ssga}$, each CS was represented as a real-valued feature mask (FM) of numbers ranging within [0.0 .. 1.0]. A feature was 'turned-off' if its associated value in the FM was less than 0.5. The fitness returned by the evaluation function for a FM was simply the number of errors associated with the FM multiplied by 10 plus the percentage of original feature set that was used. The objective of the evaluation function was to minimize the number of recognition errors as well as the number of features needed for recognition. Binary tournament selection [1], [2], [3] was used to select parents. Parents created offspring via uniform crossover [1] and Gaussian mutation [2].

IV. Experiments

Six experiments were used to test GEFE$_{ssga}$. The first three experiments used a dataset that was taken from the FRGC database [14] and is composed of 1,230 images taken from 410 subjects. These images were divided into two sets: a probe set of 410 images (one of each subject) and a gallery set of 820 images (an additional two images of each of the 410 subjects). Based on the probe and gallery sets three experiments were constructed to test the recognition rate of GEFE$_{ssga}$. In the first experiment (FRGC$_r$), the recognition rate associated with the periocular region of the left eye only was tested. In the second experiment (FRGC$_l$), recognition rate of the periocular region associated with the right eye was tested. For the third experiment (FRGC$_{both}$), the recognition rate using both eyes was tested.

The last three experiments used a dataset that was taken from the FERET database [12], [13] and is composed of 162 images taken from 54 subjects. As with the previous three experiments, the images of the dataset were divided into three probe sets (one image of each subject) and three gallery sets of 108 images (an additional two images of each of the 54 subjects). These three experiments were: FERET$_l$ recognition test for the left eye only, FERET$_r$ recognition test using the right eye only, and FERET$_{both}$ recognition test using both eyes.

For the four single eye experiments, the original feature set consisted 1416 features. For FRCG$_{both}$ and FERET$_{both}$, the original feature set consisted of 2832 features (1416 features for each eye). For each of the six experiments, the objective of GEFE$_{ssga}$ was to evolve FMs that: (a) minimize the number of features needed for periocular biometric recognition and (b) improve recognition accuracy.

V. Results

For all six experiments, GEFE$_{ssga}$ evolved a population 20 FMs. The standard deviation of the Gaussian mutation operator was set to 0.2. For each experiment, the GEFE$_{ssga}$ was run a total of 30 times and was allowed a total of 1000 function evaluations on each run. Table 1 shows the results of three experiments. The first column in Table 1 denotes the experiment, the second column denotes the type of feature set (Type I or GEFE$_{ssga}$ optimized), and the third column represents the number of features used on average along with the size of the original feature set. The final column represents the recognition accuracy (Rank 1 recognition rate [10]).

In Table 1, we include the results of Miller et al. [10] (Type I results) using the same FRGC and FERET datasets as well as the results of Adams et al. [6] (Type II results) which uses the same FRGC dataset. This was done to show the performance difference on two datasets that are based on two databases that are well known within the biometrics research community.

In Table 1 for the three experiments the GEFE$_{ssga}$ evolved features sets are smaller and more accurate. For FRGC$_r$, the evolved feature sets, on average, used only 50.5% of the original features extracted by the Type I LBP feature extractor. The results for FRGC$_l$ were similar. The performance of GEFE$_{ssga}$ on FRGC$_{both}$ was also similar to the results on FRGC$_l$ and FRGC$_r$. The evolved feature set used approximately 49% of the original features and had a higher recognition rate.

The results of the last three experiments are even more dramatic in terms of feature set optimization/minimization. The evolved features sets, on average, used at least 52% fewer features and had an increased recognition rate of at least 10 percentage points. For both FRGC and FERET, the experiments using two eyes resulted in better recognition rates than using a single eye. This increase in performance also comes with the use of fewer features than those used for the left and right eyes combined. The results show that GEFE$_{ssga}$ is an effective Type II feature extraction method.

Experiment	Type	Features	Accuracy
FRGC$_l$	Type I	1416.0	84.38%
	GEFE$_{ssga}$	**715.5**	**86.85%**
FRGC$_r$	Type I	1416.0	83.90%
	GEFE$_{ssga}$	**715.4**	**86.26%**
FRGC$_{both}$	Type I	2832.0	89.76%
	GEFE$_{ssga}$	**1399.4**	**92.16%**
FERET$_l$	Type I	1416.0	72.22%
	GEFE$_{ssga}$	**674.8**	**80.25%**
FERET$_r$	Type I	1416.0	70.37%
	GEFE$_{ssga}$	**670.2**	**80.80%**
FERET$_{both}$	Type I	2832.0	74.07%
	GEFE$_{ssga}$	**1346.1**	**85.06%**

TABLE I
Comparison Between Type I [10] and GEFE$_{ssga}$ Optimized Feature Sets

VI. Conclusion & Future Work

In this paper, GEFE$_{ssga}$ was introduced as a method for Type II biometric feature extraction (feature set optimization/minimization). Our results show that GEFE$_{ssga}$ was able to effectively reduce the number of features needed for periocular biometric recognition while increasing the recognition accuracy. The results are based on two datasets

taken from two databases that are well known within the biometric research community.

Our future work will be devoted towards: discovering the best GECs for periocular Type II feature bextraction, investigating alternative distance metrics for Type I LBP feature extraction, and developing deterministic Type II feature extraction methods for periocular biometric recognition.

ACKNOWLEDGMENT

This research was funded by the Office of the Director of National Intelligence (ODNI) Center of Academic Excellence (CAE) for the multi-university Center for Advanced Studies in Identity Sciences (CASIS). The authors would like to thank the ODNI for their support.

REFERENCES

[1] L. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[2] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, 2000.

[3] J. Kennedy and R. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[4] M. Pei, E. Goodman, and W. F. Punch, "Feature extraction using genetic algorithms," in *Proceeding of International Symposium on Intelligent Data Engineering and Learning98 (IDEAL98), Hong Kong* , 1998.

[5] M. Raymer, W. Punch, E. Goodman, L. Kuhn, and A. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164–171, July 2000.

[6] J. Adams, D. Woodard, G. Dozier, P. Miller, G. Glenn, and K. Bryant, "GEFE: Genetic & evolutionary feature extraction for periocular-based biometric recognition," in *Submission to: The ACM Southeastern Conference 2010*. New York, NY, USA: ACM, 20.

[7] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[8] A. K. Jain and A. Ross, "Introduction to biometrics," *Handbook of Biometrics (EDs: Jain, Flynn, and Ross)*, pp. 1–22, 2008.

[9] D. Maltoni and R. Cappelli, "Fingerprint recognition," *Handbook of Biometrics (EDs: Jain, Flynn, and Ross)*, pp. 23–42, 2008.

[10] P. Miller, A. Rawls, S. Pundlik, and D. Woodard, "Personal identification using periocular skin texture," in *SAC '10: Proceedings of the 2010 ACM symposium on Applied Computing*. New York, NY, USA: ACM, 2010.

[11] M. Savvides, J. Heo, and S. W. Park, "Face recognition," *Handbook of Biometrics (EDs: Jain, Flynn, and Ross)*, pp. 43–70, 2008.

[12] P. j. Phillips, H. Moon, and S. A. Rizvi, "The feret evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* , vol. 22, pp. 1090–1104, 2000.

[13] P. j. Phillips, H. Wechler, J. Huang, and P. Rauss, "The feret database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[14] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoff, J. Marques, J. Min, and W. Worek, "Overview of face recognition grand challenge," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[15] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 971–987, 2002.

[16] M. L. Tinker, G. Dozier, and A. Garrett, "The exploratory toolset for the optimization of launch and space systems (x-toolss)," *http://xtoolss.msfc.nasa.gov/*, 2010.