

Genetic & Evolutionary Biometrics: Feature Extraction from a Machine Learning Perspective

Joseph Shelton, Aniesha Alford, Lasanio Small, Derrick
Leflore, Jared Williams,
Joshua Adams, Gerry Dozier, Kelvin Bryant
Center for Advanced Studies in Identity Sciences
NC A&T
Greensboro, USA
jashelt1@ncat.edu, aalford@ncat.edu,
lsmall@ncat.edu, daleflor@ncat.edu,
jcwill24@gmail.com, jcadams2@ncat.edu,
gvdozier@ncat.edu, ksbyrant@ncat.edu

Tamirat Abegaz
Computer Science
Clemson University
Clemson, SC, USA
tamirat@programmer.net

Karl Ricanek
Computer Science Department
UNC-W
Wilmington, USA
ricanekk@uncw.edu

Abstract—Genetic & Evolutionary Biometrics (GEB) is a newly emerging area of study devoted to the design, analysis, and application of genetic and evolutionary computing to the field of biometrics. In this paper, we present a GEB application called GEFE_{ML} (Genetic and Evolutionary Feature Extraction – Machine Learning). GEFE_{ML} incorporates a machine learning technique, referred to as cross validation, in an effort to evolve a population of local binary pattern feature extractors (FEs) that generalize well to unseen subjects. GEFE_{ML} was trained on a dataset taken from the FRGC database and generalized well on two test sets of unseen subjects taken from the FRGC and MORPH databases. GEFE_{ML} evolved FEs that used fewer patches, had comparable accuracy, and were 54% less expensive in terms of computational complexity.

Keywords– *Biometrics, Cross Validation, Estimation of Distribution Algorithm, Feature Extraction, Genetic & Evolutionary Computation, Local Binary Pattern*

I. INTRODUCTION

Genetic & Evolutionary Biometrics (GEB) [8,21,22,23] is an emerging sub-area of Genetic & Evolutionary Computation (GEC) [10,12,13,16,19,20,24,27,28,29,30] that is devoted to the design, development, and application of Darwinian-style methods (those based on natural selection [25]) for solving problems within the area of biometrics. These methods (also referred to as GECs) discover optimal or near optimal solutions to problems as follows. Initially, a population of randomly generated candidate solutions (CSs) is created. Each CS is then evaluated and assigned a fitness based on a user-specified evaluation function, which is used to determine its relative ‘goodness’. Next, parents are chosen from the population based on their fitness and are allowed to create offspring CSs.

The offspring are each assigned a fitness and typically replace weaker members of the previous population. This evolutionary process of selecting parents, allowing them to procreate, and replacing weaker members of the population with the newly formed offspring is repeated until a user-specified stopping condition is reached. Such stopping conditions would be a certain number of evaluations executing, populations converging onto a solution, or the GEC finding an optimal solution. Figure 1 provides a flowchart of a typical GEC.

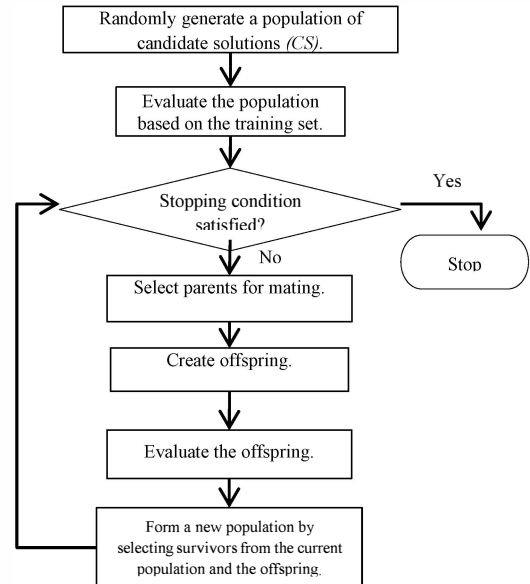


Figure 1. Flowchart of a Typical GEC

In [4], Shelton et al. developed a GEC in the form of a genetic & evolutionary feature extractor (GEFE). GEFE was used to evolve a population of Local Binary Pattern [1,2,3,5,6] (LBP)

based feature extractors (FEs). These FEs were of two types: (a) those that consisted of patches that were of non-uniform size and (b) those that consisted of patches that were of uniform size. The evolved FEs used a small number of patches (approximately 8) that covered only a small portion of an image (approximately 25%) while outperforming the standard LBP approach.

In their paper, Shelton et al. worked with GEFEs that were instances of two well-known GECs: a steady-state genetic algorithm (SSGA) [15] and an estimation of distribution algorithm (EDA) [14]. Their results showed that the GEFE instances that evolved FEs composed of uniform sized patches were superior to those instances that evolved non-uniform sized patches. Their results also showed that those GEFE instances that were EDAs outperformed their SSGA counterparts. All GEFE instances outperformed the standard LBP method in terms of recognition accuracy while using approximately 1/3 the number of features.

Shelton et al.'s rationale for reducing the number features used for recognition was motivated by a two-stage hierarchical recognition system proposed by Gentile et al. [17]. This system reduced the overall number of feature checks needed for recognition by first comparing a probe, p , using a short-length biometric template (of the n subjects within a dataset) consisting of a subset of k features to obtain a set of r subjects of the dataset that matched p the closest. Next, p is compared with the r subjects using the full biometric template of m features. The proposed system in [8] had a total computational complexity, with respect to feature checks, of $nk + rm$ which resulted in fewer feature checks of a conventional biometric recognition system which has a computational complexity of nm .

Since Shelton et al. were primarily focused on evolving short-lengthed biometric templates for a 'Gentile-style' recognition system, their GEFE instances trained on a subset of the images of *all* of the n subjects of the dataset. In this paper, we extend the work of Shelton et al. to investigate how well the FEs evolved by GEFE generalize to unseen subjects by incorporating the machine learning concept of cross validation [31,32,33,34,35]. This GEB application will be referred to as Genetic and Evolutionary Feature Extraction – Machine Learning (GEFE_{ML}).

The remainder of this paper is as follows: Section 2 provides an overview of LBP and an EDA. Section 3 provides a description of GEFE_{ML}, specifically, the representation of a FE, the function used to evaluate FEs via a training set, and evolving FEs using cross validation. Sections 4 and 5 present our experiments and our results. Section 6 provides a brief discussion. Our conclusions and future work are presented in Section 7.

II. BACKGROUND

A. LBP

LBP is a feature extraction technique that uses the textures of an image to create a feature vector for it. The common way to perform LBP is by first separating an entire image into even sized patches [6] as shown in Figure 2.

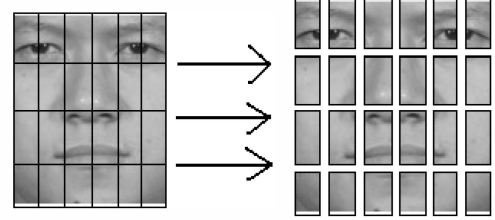


Figure 2. Standard LBP

Within each patch, a texture pattern can be extracted for each pixel, C_p in the patch that has t neighboring pixels surrounding it. The value of t and the distance between neighboring pixels and C_p are user specified. The pixel intensity values between C_p and its neighboring pixels are compared and a texture pattern is produced. The pattern is represented as a binary string and is produced by taking the difference of each neighboring pixel and C_p . If the difference is negative, the texture pattern will have a zero bit in the position of the string relative to the position of the neighboring pixel with C_p ; else the bit will have a one bit, meaning the length of the string will be equal to the number of neighboring pixels, t .

Each patch has a histogram associated with it, and the histogram keeps track of the frequencies of patterns within that patch. For a neighborhood size of t , there are 2^t possible patterns that can be extracted. To reduce the number of frequencies in histograms, the patterns are classified as uniform or non-uniform. A uniform pattern is a pattern where the bit changes two or less times when traversing it circularly and a non-uniform pattern would have more than two bit changes. This reduces the size of a histogram from 2^t to $(t * (t-1)) + 2$; there are $t * (t-1)$ patterns with exactly two changes and there are two patterns with no changes, specifically the patterns consisting of all zeros and all ones. The definition of uniform and non-uniform patterns is given more detail in [6].

The frequencies from the histograms of each patch are concatenated to form a feature vector for an image. The feature vector can be compared to other feature vectors of images using some distance measure, and the resulting measure will represent the similarity of vectors and, consequently, images.

B. EDA

In [4], an EDA instance of GEFE outperformed the SSGA instance, which is the rationale for only using an EDA in this research. An EDA initially creates a population of candidate FEs, and these FEs are evaluated based on some fitness

evaluation. The top 50% of FEs from the population are selected and a probability density function (PDF) is built. Offspring are created from sampling the PDF, and they are then evaluated and assigned fitness. The offspring and an arbitrary number of best performing FEs, the elites, from the previous population make up a new population. The newer population undergoes the process of creating offspring and replacement until some terminating condition has been met.

III. GEFEML

A. Representation of Candidate FEs

A definition of a FE, fe_i , can be found in [4]. The definition states that fe_i can be represented as a six-tuple, $\langle X_i, Y_i, W_i, H_i, M_i, f_i \rangle$. $X_i = \{x_{i,0}, x_{i,1}, \dots, x_{i,n-1}\}$ and $Y_i = \{y_{i,0}, y_{i,1}, \dots, y_{i,n-1}\}$ represents the x-coordinates and the y-coordinates of the center of the n possible patches. The widths and heights of the n patches are represented by $W_i = \{w_{i,0}, w_{i,1}, \dots, w_{i,n-1}\}$ and $H_i = \{h_{i,0}, h_{i,1}, \dots, h_{i,n-1}\}$. Because the patches are uniform, $W_k = \{w_{k,0}, w_{k,1}, \dots, w_{k,n-1}\}$ is equivalent to, $w_{k,0} = w_{k,1}, \dots, w_{k,n-2} = w_{k,n-1}$, and $H_k = \{h_{k,0}, h_{k,1}, \dots, h_{k,n-1}\}$ is equivalent to, $h_{k,0} = h_{k,1}, \dots, h_{k,n-2} = h_{k,n-1}$. We use uniform sized patches due to results in [4]. $M_i = \{m_{i,0}, m_{i,1}, \dots, m_{i,n-1}\}$ represents the masking values for each patch and f_i represents the fitness of fe_i . The masking value determines whether a patch is activated or deactivated.

B. Evaluation function of a FE

To evaluate the fitness of a FE, we apply it to a dataset of subjects, D . Each subject has a series of snapshots, and one snapshot is assigned to a probe set, while the other snapshots are applied to a gallery set. A FE, fe_i , is applied to the probe set and gallery set to create FVs for all images. FVs from the probe set are matched to all FVs in the gallery set, and whichever two images are considered the closest and do not belong to the same subject is considered an error. The fitness, f_i , associated with a candidate FE, fe_i , is equal to the number of incorrect matches, $\varepsilon(D)$, multiplied by 10 plus the percentage of image space (measured in pixels) covered by fe_i , $\gamma(fe_i)$. Equation 1 provides an example of the evaluation function used by GEFEML to assign fitnesses to candidate FEs.

$$f_i = 10\varepsilon(D) + \gamma(fe_i) \quad (1)$$

C. Cross Validation in GEFEML

We demonstrate how cross validation recorded the best generalizing FE during the evolutionary process in Figure 3. Initially, all candidate FEs are evaluated on the training set and the validation set. The best fit FE on the validation set is labelled FE*. After offspring FEs have been created, they are evaluated on the training set and validation set. Any offspring FE that outperforms FE* becomes the new FE*. After a stopping condition is satisfied, two FEs are returned: the best performing FE on the training set, FE^{ts}, and FE*.

IV. EXPERIMENTS

In our two experiments, four datasets were taken from the Facial Recognition Grand Challenge (FRGC) [7] and the Craniofacial Longitudinal Morphological Face (MORPH) [26] databases. From the FRGC database, a total of 314 subjects were used to create three datasets: FRGC-100a, consisting of 100 subjects, FRGC-109, consisting of 109 subjects, and FRGC-100b consisting of 100 subjects different from those in FRGC-100a. MORPH-500, was formed by using 500 subjects taken from the MORPH database.

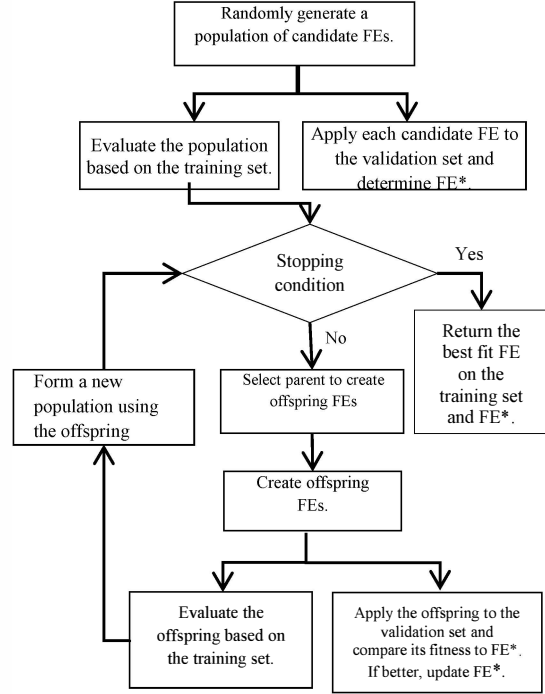


Figure 3. Flowchart of the Evolutionary Process of GEFEML using Cross validation

In Experiment I, FRGC-100a was used as the training set, FRGC-109 as the validation set, and FRGC-100b was used as a test set. In Experiment II, MORPH-500 was used as a second test set. The datasets consisted of 3 images for each subject and were further divided into a probe set consisting of one image and a gallery set consisting of the other two images. Figure 4 provides an example of a subject from FRGC and MORPH. All images were pre-processed, aligned in a similar manner, and had the dimensions of 100 x 127 pixels.

In both experiments, the objective was twofold: to evolve FEs that generalize well to unseen subjects and to evolve FEs that are more efficient in terms of the amount of processing (associated computational complexity) needed to extract features from an image. In terms of computing the computational complexity of a FE, the fundamental unit of work used was the number of pixels that needed to be processed. The total amount of work, w_i , done by a candidate

FE, fe_i , can be expressed as follows where x_i and y_i represent the dimensions of each patch (in pixels) within fe_i and where n_i represents the number of patches within fe_i : $w_i(n_i, x_i, y_i) = n_i(x_i - 2)(y_i - 2)$.

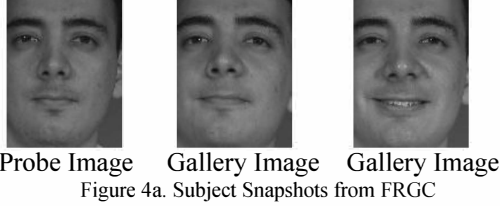


Figure 4a. Subject Snapshots from FRGC



Figure 4b. Subject Snapshots from MORPH

V. RESULTS

The results for both experiments were generated as follows. GEFE_{ML} was run 30 times with maximum function evaluation limits of 1000, 2000, 3000 and 4000 (a total of 120 runs in all). On each run, GEFE_{ML} returned the best performing FE on the training set (FE^{ts}), and the best performing FE with respect to the validation set (FE^*). In addition, on each run, GEFE_{ML} evolved a population of 20 candidate FEs and always allowed the best individual to remain in the population.

In both experiments, the standard LBP method was used as the baseline. This baseline method divided an image into a 6 by 4 set of nearly uniform sized patches. These patches were as follows: 15 patches were of 17x32 pixels, 5 patches were of 17x31, 3 patches were of 15x32 pixels, and 1 patch was 15x31 pixels. No overlap occurred between patches. GEFE_{ML} was allowed to evolve FEs consisting of a number of uniform sized patches ranging from [1..24] and the patches were allowed to overlap.

A. Results: Experiment I

Table I presents the results of Experiment I. The first column denotes the method used and the number of FE evaluations that were performed. The second column represents the results of applying the corresponding method to FRGC-100a. The third column represents the results of how well the 30 FE^{ts} s generalized on FRGC-100b, and the final column represents the results of how well the 30 FE^* s generalized on FRGC-100b. For these columns, the first number represents the average recognition accuracy and the second number represents the average computational complexity.

TABLE I: TRAINING AND GENERALIZING ON THE FRGC DATASET

Method	FRGC-100a Optimization Acc (Comp)	FRGC-100b Opt-Gen Acc (Comp)	FRGC-100b Val-Gen Acc (Comp)
Baseline	0.980 (10379.0)	N/A	0.980 (10379.0)
GEFE _{ML} (1000e)	1.000 (5897.8)	0.983 (5897.8)	0.991 (5362.7)
GEFE _{ML} (2000e)	1.000 (5371.95)	0.976 (5371.95)	0.985 (5217.9)
GEFE _{ML} (3000e)	1.000 (5681.5)	0.978 (5681.5)	0.987 (6347.2)
GEFE _{ML} (4000e)	1.000 (5526.9)	0.983 (5526.9)	0.988 (5870.7)

In terms of recognition accuracy and computational complexity, GEFE_{ML} outperformed the baseline method for both FRGC-100a optimization and FRGC-100b Val-Gen. On FRGC-100a optimization, GEFE_{ML} evolved FE^{ts} s that had 100% recognition accuracy. The average computational complexities of the FE^{ts} s were approximately 45% less than the baseline computational complexity. When the GEFE_{ML} Opt-Gen performances were compared in terms of accuracy, there was no significant difference. Likewise, when the GEFE_{ML} Val-Gen performances were compared in terms of accuracy, there was no significant difference. However, when the GEFE_{ML} Opt-Gen performances were compared to the GEFE_{ML} Val-Gen performances in terms of recognition accuracy, the Val-Gen performances were statistically better. These results were confirmed through the use of an ANOVA and t-tests.

For Experiment I, GEFE_{ML} with 1000 evaluations achieved the highest recognition accuracy for Opt-Gen and Val-Gen. For this method, the baseline, FE^{ts} and FE^* are shown in Figure 5.

When comparing the images, the areas of the patches within FE^{ts} and FE^* are much smaller than the area of the baseline method. Additionally, one can observe that the patches within FE^{ts} and FE^* are clustered predominantly within the periocular region. This result supports the research of Miller et al. [5] that suggests that the periocular region can be a highly discriminating modality for recognition. Also, notice that FE^* appears to extract more features from the periocular region than FE^{ts} . This observation suggests that in order to adequately generalize to unseen subjects a slight increase in the number of features may be required.

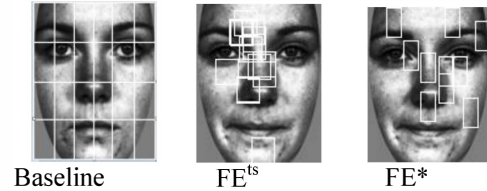


Figure 5. Baseline, FE^{ts} and FE^* on FRGC-100

B. Results: Experiment II

Table II presents the results of Experiment II. As in Table I, the first column denotes the method used and the number of evaluations used. The second column represents the results of

how well the 30 FE^{ts} s generalized on MORPH-500 (the second test set), and the final column represents the results of how well the 30 FE^* s generalized on MORPH-500.

TABLE II: GENERALIZING ON MORPH DATASET

Method	MORPH-500 Opt-Gen <i>Acc (Comp)</i>	MORPH-500 Val-Gen <i>Acc (Comp)</i>
Baseline	N/A	0.432 (10379.0)
GEFE _{ML} (1000e)	0.268 (5897.8)	0.313 (5362.7)
GEFE _{ML} (2000e)	0.251 (5371.95)	0.304 (5217.9)
GEFE _{ML} (3000e)	0.256 (5681.5)	0.310 (6347.2)
GEFE _{ML} (4000e)	0.256 (5526.9)	0.319 (5870.7)

For Experiment II, GEFE_{ML} achieved lower average recognition accuracies than the baseline method; however, the computational costs of the GEFEML methods were significantly lower than the baseline due to the reduction of the size and the number of patches. When the accuracies of the GEFE_{ML} methods were compared to each other, for Opt-Gen and Val-GEN, the method with 2000 evaluations performed poorer than the other methods. When comparing the Opt-Gen accuracy to the Val-Gen accuracy, the performances of the FE^* s (for 1000, 2000, 3000, and 4000 evaluations) outperformed their Opt-Gen counterparts. These results were, once again, confirmed using an ANOVA and a t-test. The GEFE_{ML} that was allowed to use 4000 evaluations achieved the highest average recognition accuracy. Figure 6 shows the baseline, FE^{ts} , and FE^* on MORPH-500.

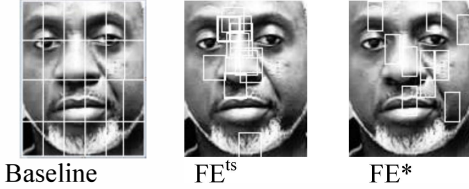


Figure 6. Baseline, FE^{ts} and FE^* on MORPH-500

VI. DISCUSSION

In Experiment I, GEFE_{ML} performed better than the baseline method in terms of average recognition accuracy and average computational complexity. In Experiment II, the average performance of GEFE_{ML} did not outperform the baseline method in regards to accuracy; however GEFE_{ML} had approximately a 40% reduction in computational complexity.

The Cumulative Match Characteristic (CMC) curves and the Receiver Operator Characteristic (ROC) curves provide further analysis of the performance of GEFE_{ML}. The ROC curves were created using the normalized Manhattan Distance formula, NMD [8], shown in Equation 2, where h_i and h_j represent the two feature vectors being compared, l represents the length of the feature vectors, and z represents the z^{th} feature in a feature vector.

$$NMD(h_i, h_j) = \sum_{z=0}^{l-1} \frac{|h_{i,z} - h_{j,z}|}{\max(h_{i,z} - h_{j,z})} \quad (2)$$

The generalizing performance of the best FE^* and the best FE^{ts} were measured on FRGC-100 and MORPH-500 using CMC and ROC curves. We chose FE^* and FE^{ts} from the GEFE_{ML} that used a maximum of 1000 FE evaluations because it had the best average performance on FRGC-100 with respect to the GEFE_{ML} instances.

Figure 7a shows the CMC curves of the baseline method, FE^* and FE^{ts} (which appears as ' FE^{ts} ') based on their performances on the first test set, FRGC-100. One can see that at Rank 1 and Rank 2, FE^* and FE^{ts} outperform the baseline method. At Rank 3, all three methods perform equally; however, the performances of FE^* and FE^{ts} are computationally less expensive.

In Figure 7b, one can see that the ROC curves of the three methods are similar; however, the curve of FE^* has a slight separation from the other two methods. Of the three methods, with respect to Figure 7b, FE^* has the best performance.

Figure 8a provides the CMC curves of the performances of the three methods with respect to the second test set, MORPH-500. The baseline method outperforms FE^* and FE^{ts} at each rank with a maximum accuracy of 64% at Rank 10. The Rank 10 accuracies for FE^* and FE^{ts} were 58.2% and 55.8%. These results are encouraging when one considers that FE^* and FE^{ts} were trained on only 100 subjects from a different database. This is also the case with the ROC curves shown in Figure 8b. The performances of the three methods are somewhat similar; however, the performances of FE^* and FE^{ts} come at a fraction of the cost of the baseline method.

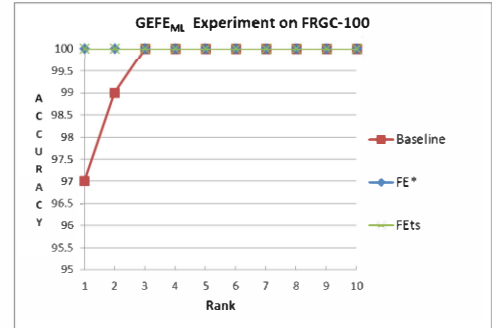


Figure 7a. CMC curve of Baseline, FE^* and FE^{ts} on FRGC-100

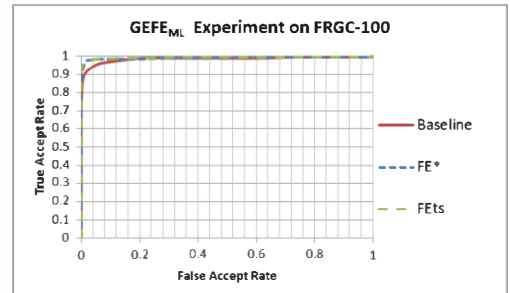


Figure 7b. ROC curve of Baseline, FE^* and FE^{ts} on FRGC-100

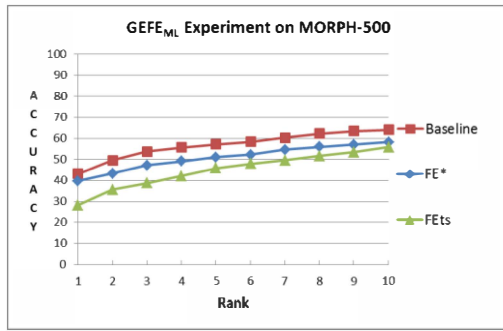


Figure 8a. CMC curve of Baseline, FE* and FE^{ts} on MORPH-500

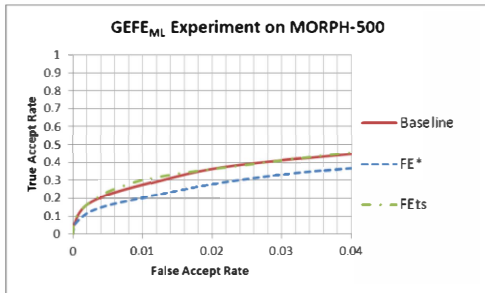


Figure 8b. ROC curve of Baseline, FE* and FE^{ts} on MORPH-500

VII. CONCLUSION AND FUTURE WORK

In this paper, a GEB application, GEFE_{ML}, was presented which was used to evolve FEs that were able to generalize well to unseen subjects taken from two different databases. The evolved FEs were not only comparable in terms of accuracy with the baseline LBP method but used approximately 45% less processing time (in terms of computational complexity). Our future work will include the development of training, validation, and test sets composed of images from a variety of databases.

ACKNOWLEDGMENT

This research was funded by the Office of the Director of National Intelligence (ODNI), Center for Academic Excellence (CAE) for the multi-university Center for Advanced Studies in Identity Sciences (CASIS), NSF SSTEM, Lockheed Martin and the National Science Foundation (NSF) Science & Technology Center: Bio/computational Evolution in Action CONSortium (BEACON). The authors would like to thank the ODNI, the NSF, and Lockheed Martin for their support of this research.

REFERENCES

- [1] Timo Ojala, Matti Pietikainen, "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns", IEEE Trans. Pattern Analysis and Machine Intelligence; 971-987; 2002
- [2] Alice Porebski, Nicolas Vandenbroucke Ludovic Macaire, "Haralick feature extraction from LBP images for color texture classification", IEEE Trans. Image Processing Theory, Tools and Applications; 1-8; 2008
- [3] Zhenan Sun, Tieniu Tan, Xianchao Qiu, "Graph matching iris image blocks with Local Binary Pattern", National Laboratory of Pattern Recognition, 2006
- [4] Joseph Shelton et al., "Genetic and evolutionary feature extraction via X-TOOLSS in The 8th annual International Conference on Genetic and Evolutionary Methods (GEM), 2011.
- [5] 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.
- [6] Timo Ahonen, Abdenour Hadid, Matti Pietikinen, "Face description with Local Binary Patterns: application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006, doi:10.1109/TPAMI.2006.244.
- [7] 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.
- [8] A. Alford et al., "Genetic & evolutionary methods for biometric feature reduction", (invited paper) Special Issue on: "Computational Intelligence in Biometrics: Theory, Methods and Applications", Guest Editor: Dr. Qinghan Xiao, International Journal of Biometrics, 2011.
- [9] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.15, No.11, pp.1148-1161, 1993.
- [10] Veeramachaneni, K., Osadciw, L. A., & Varshney, P. K. (2003). "Adaptive multimodal biometric fusion algorithm using particle swarm." (B. V. Dasarthy, Ed.) Electrical Engineering, 5099, 211-221.
- [11] G. Dozier et al., "Minimizing the number of bits needed for iris recognition via bit inconsistency and GRIT," Proceedings of the 2009 IEEE Workshop on Computational Intelligence in Biometrics Theory, Algorithms, and Applications, Nashville, March 30 April 2nd, 2009
- [12] J. Kennedy, R. Eberhart, "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942-1948. (1995)
- [13] D.E. Goldberg, "Genetic algorithms in search, optimization & machine learning", Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.
- [14] P. Larranaga, and J. A. Lozano, "Estimation of Distribution Algorithms: a new tool for evolutionary computation", Kluwer Academic Publishers, 2002.
- [15] Smith, J.; Fogarty, T.C.; , "Self adaptation of mutation rates in a steady state genetic algorithm," Evolutionary Computation, 1996., Proceedings of IEEE International Conference on , vol., no., pp.318-323, 20-22 May 1996
- [16] L. Davis, "Handbook of genetic algorithms". New York: Van Nostrand Reinhold, 1991.
- [17] J.E. Gentile, N. Ratha, and J. Connell, "An efficient, two-stage iris recognition system", In Proc. 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS), 2009.
- [18] Jing Xiao; Michalewicz, Z.; Lixin Zhang; Trojanowski, K.; , "Adaptive evolutionary planner/navigator for mobile robots," Evolutionary Computation, IEEE Transactions on , vol.1, no.1, pp.18-28, Apr 1997
- [19] Fonseca, C.M. and P.J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization, in evolutionary computation", Vol. 3, No. 1, pp. 1-16. (1995)
- [20] Andrew Kusiak. "Evolutionary computation and data mining", In Bhaskaran Gopalakrishnan and Angappa Gunasekaran, editors, Intelligent Systems in Design and Manufacturing III, volume SPIE-4192, pages 1-10, 2000.
- [21] R. M. Ramadan, and R. F. Abdel-Kader, "Face recognition using Particle Swarm Optimization-based selected features," In International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 2, No. 2, June 2009.
- [22] J. Galbally, J. Fierrez, M. Freire, and J. Ortega-Garcia, "Feature selection based on genetic algorithms for on-line signature verification." IEEE Workshop on Automatic Identification Advanced Technologies, 2007.

- [23] D. Kumar, S. Kumar, and C.S. Rai, "Feature selection for face recognition: a memetic algorithmic approach." *Journal of Zhejanga University Science A*, Vol. 10, no. 8, pp. 1140-1152, 2009.
- [24] G. Dozier, A. Homaifar, E. Tunstel, and D. Battle. "An introduction to evolutionary computation" (Chapter 17), *Intelligent Control Systems Using Soft Computing Methodologies*, A. Zilouchian & M. Jamshidi (Eds.), pp. 365-380, CRC press, 2001.
- [25] C. Darwin, "The origin of species", London: John Murry, 1872
- [26] Ricanek Jr., K and Tesafaye, T., "MORPH: A longitudinal image database of normal adult age-progression", *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, p.341-345, April 10-12, 2006.
- [27] Fogel, D. "Evolutionary computation: toward a new philosophy of machine intelligence", IEEE Press, New York. 1995
- [28] Back, T. "Evolutionary algorithms in theory and practice" Oxford University Press, Inc. New York. 1996
- [29] A. P. Engelbrecht, "Computational intelligence: an introduction" John Wiley & Sons, Ltd. 2002
- [30] J. H. Holland "Adaptation in natural and artificial systems" The University of Michigan Press. (1975)
- [31] T. M. Mitchell, "Machine learning". McGraw-Hill Companies, Inc. (1997)
- [32] S. Haykin, "Neural networks: a comprehensive foundation", 2nd Edition. Prentice-Hall, Inc. New Jersey. (1999)
- [33] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani "Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence", Prentice-Hall, Inc. New Jersey. (1997)
- [34] P. D. Wasserman, "Neural computing", Van Nostrand Reinhold, New York. (1989).
- [35] M. Negnevitsky. "Artificial Intelligence: a guide to intelligent systems", 2nd Edition. Addison-Wesley. (2005)