We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Genetic & Evolutionary Biometrics

Aniesha Alford, Joseph Shelton, Joshua Adams,
Derrick LeFlore, Michael Payne, Jonathan Turner,
Vincent McLean, Robert Benson, Gerry Dozier,
Kelvin Bryant and John Kelly

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/51386

1. Introduction

Genetic & Evolutionary Computation (GEC) is the field of study devoted to the design, development, and analysis of problem solvers based on natural selection [1-4] and has been successfully applied to a wide range of complex, real world optimization problems in the areas of robotics [5], scheduling [6], music generation [7], aircraft design [1], and cyber security [8-11], just to name a few. Genetic and Evolutionary Computations (referred to as GECs) differ from most traditional problems solvers in that they are stochastic methods that evolve a population of candidate solutions (CSs) rather than just operating on a single CS. Due to the evolutionary nature of GECs, they are able to discover a wide variety of novel solutions to a particular problem at hand – solutions that radically differ from those developed by traditional problem solvers [3,12,13].

GECs are general-purpose problem solvers [1,2,4]. Because of this fact and their ability to hybridize well with traditional problem solvers [1], a number of new subfields have emerged. In the field of Evolutionary Robotics [5,14], GECs are used in path planning [15], robot behavior design [16], and robot gait design [17]. In the field of Evolutionary Design [1,18], GECs are being used to evolve lunar habitats [19], emoticons [20], and music [7,21]. GECs have also been used successfully in a wide variety of scheduling applications [22,23] – which in turn has spawn a subfield known as Evolutionary Scheduling [6,24].

Currently we are seeing the emergence of a new and exciting field of study devoted towards the design, development, analysis, and application of GECs to problems within the area of biometrics [25-29]. We refer to this new subfield of study as Genetic and Evolutionary Biometrics (GEB) [25-27,31]. In this chapter, we will provide a brief history of GEB as well as

introduce a number of GEB applications (which we refer to as GEBAs). The GEBAs presented in this chapter are actually hybridized forms of traditional methods used within the biometrics research community. These GEBAs typically evolve solutions that are radically different from and, in many instances, more efficient than solutions developed by traditional methods currently used within the biometrics research community.

The remainder of this chapter is as follows. In Section 2, we discuss GECs and provide a brief history of the field of GEB. We also present an overview of the Local Binary Patterns (LBP) [32-34] method and in Section 3 we present the GEBAs used in this work. In Section 4, we describe the experiments performed and we present our results. In Section 5, we provide an additional discussion of our results, and in Section 6, we provide a summary of our work as well as directions for further research.

2. Background

2.1. Genetic & Evolutionary Computation (GEC)

GECs typically work in the following fashion [1-4,12,13]. Initially, a population of CSs is randomly generated. Each CS is then evaluated and assigned a fitness based on a user-defined evaluation function. Next, parents are selected from the population based on their fitness and are allowed to produce offspring CSs. The offspring are then assigned a fitness and usually replace the worst performing CSs within the current population. This evolutionary process is then repeated until a user-specified stopping condition is satisfied. Figure 1 provides a flowchart of the typical GEC process.
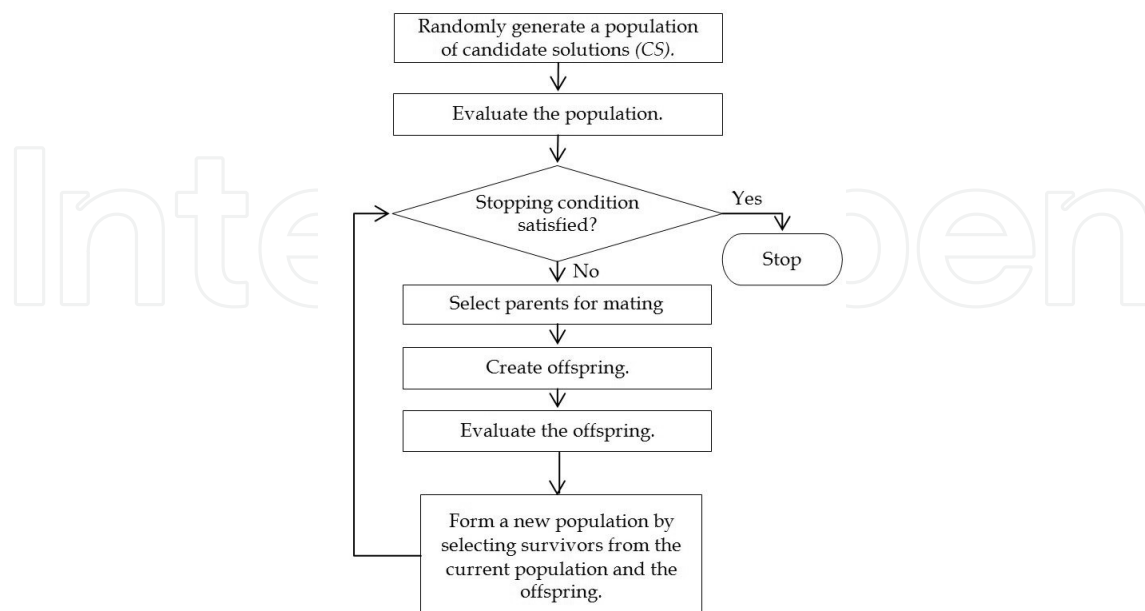
Figure 1. Flowchart of a typical GEC.

2.2. A Brief History of GEB

Within the field of biometrics, GEB applications (GEBAs) have been successfully applied to the areas of feature selection [25,30,31,33,35,36], feature weighting [25,30,31,37,38], feature extraction [26-29], and biometric-based security [9-11,39].

2.2.1. GEBAs for Feature Selection and Weighting

Concerning GEBAs for feature selection, Galbally et al. [35] developed Genetic Algorithms (GAs) [1] for feature selection and applied them in an effort to solve the signature verification problem. The GAs were applied to two training sets which were formed using the signatures of 330 subjects from the MCYT Signature database [40]. The first training set consisted of five signatures of each subject and the second training set consisted of 20 signatures of each subject. A test set was formed using the remaining signatures. Their results showed that the GAs, when compared to the baseline method that used all of the features, were able to reduce the number of features used while improving the recognition accuracy of the system.

Ramadan and Abdel-Kader [36] applied two GECs, Particle Swarm Optimization (PSO) [13] and a GA, for feature selection for a facial recognition problem. Images from the Cambridge ORL database [41] were used to form a training set, which consisted of four images of each of the 40 subjects, and a test set, which consisted of six images of each of the 40 subjects. Two baseline methods, the Discrete Cosine Transform (DCT) [42] and Discrete Wavelet Transform (DWT) [43] methods, were then used to extract the original set of features. Ramadan and Abdel-Kader demonstrated that their GECs could increase the recognition accuracy over the baseline methods while using fewer features. In addition, the PSO used fewer features than the GA.

Kumar et al. [44] also applied two GECs, a Memetic Algorithm (MA) [45] and a GA, for feature selection for a face recognition system. The GECs were tested on two facial databases: the Cambridge ORL database [41] and a subset of 20 subjects from the YaleB database [46]. The following feature extraction techniques were used to create the original feature templates: Principal Component Analysis (PCA) [47], Linear Discriminant Analysis [48], and Kernel PCA [49]. The MA and GA were applied in an effort to reduce the dimensionality of the feature templates as well as to increase the recognition rate. Their results showed the GECs outperformed the baseline methods, which used all of the extracted features, in terms of feature usage and recognition rate. Their results also showed that the MA outperformed the GA.

In [37], Abegaz et al. applied two GEBAs for feature selection and feature weighting for facial recognition. The first GEBA, referred to as Genetic & Evolutionary Feature Selection (GEFeS), evolved subsets of biometric features also in an effort to increase the recognition rate while reducing the number of features used. The second GEBA, referred to as Genetic & Evolutionary Feature Weighting (GEFeW), evolved weights for the features. These GEBAs, which were instances of a Steady-State Genetic Algorithm (SSGA) [1-4] and Estimation of Distribution Algorithms (EDAs) [50], were applied to Eigenface[51] and LBP facial feature templates. Their results showed that GEFeS and GEFeW outperformed the baseline meth-

ods, achieving higher recognition rates while using significantly fewer features. Their results also showed that the EDA instance of GEFeW was the best performing GEBA. In addition, the LBP instances outperformed the Eigenface instances.

Alford et al. [38] compared the performances of GEFeS, GEFeW, and a hybrid feature selection and weighting GEBA referred to as Genetic & Evolutionary Feature Weighting/Selection (GEFeWS), for face-only, periocular-only, and face + periocular recognition. The GEBAs were implemented using a SSGA and an EDA, and the original feature sets were formed using the Eigenface and LBP methods. Their results showed that the GEFeS, GEFeW, and GEFeWS instances significantly outperformed the baseline methods in terms of recognition accuracy and feature usage. In addition, the EDA instance of GEFeWS outperformed the other GEBAs. Their results also showed that the performances of the GEBAs using multiple biometric modalities were better than those using only a single modality. In addition, the LBP instances performed better than the Eigenface instances.

Alford et al. [25] extended their work by developing a hybrid GEBA for feature selection and weighting that dramatically reduced the number of features necessary for recognition, increased the recognition rate, and also evolved feature masks (FMs) that generalized well to unseen subjects. The GEBA, known as GEFeWS$_{ML}$ (GEFeWS – Machine Learning), was applied to face-only, periocular-only, and face + periocular LBP templates formed for subjects within the Face Recognition Grand Challenge (FRGC) database [52]. Their results showed that GEFeWS$_{ML}$ achieved higher recognition rates than the baseline methods, while using less than 50% of the extracted features. In addition, FMs evolved via the validation set performed better than those evolved via the training set.

In [31], Alford et al. evaluated the performance of GEFeWS$_{ML}$ on a subset of images extracted from the Craniofacial Longitudinal Morphological Face (MORPH) [53] database. They also tested the cross-generalization ability of the resulting FMs by applying the FMs evolved for the FRGC datasets in [25] to the MORPH test set and applying the FMs evolved for the MORPH to the FRGC test set. Their results showed that the FMs evolved by GEFeWS$_{ML}$ could also generalize well to unseen images from a different database.

2.2.2. GEBAs for Feature Extraction

Concerning GEBAs for feature extraction, Shelton et al. [29] developed a GEBA referred to as Genetic & Evolutionary Feature Extraction (GEFE), which evolved LBP-based feature extractors (FEs) for facial recognition. Unlike the standard LBP method (SLBPM), GEFE evolved FEs that were allowed to overlap and only extracted from a subset of the image. Shelton et al. tested the performance of GEFE using two GECs, a SSGA and an EDA. They also evolved two types of FEs: (a) those that consisted of patches that were of non-uniform size and (b) those that consisted of patches that were of uniform size. Their results showed that the GEFE instances outperformed the SLBPM in terms of accuracy, feature usage, and computational complexity. In addition, the GEFE instances that evolved FEs composed of uniform sized patches outperformed the GEFE instances that evolved non-uniformed sized patches. Their results also showed that the EDA instances of GEFE outperformed the SSGA instances.

Shelton et al. [26] then extended their work, by incorporating the machine learning technique known as cross validation [54], in an effort to evolve FEs that achieve high recognition rates, extract from a small region of the image, and generalize well to unseen subjects. The GEBA, known as GEFE$_{ML}$ (GEFE-Machine Learning), was trained on a subset of images taken from the FRGC database. The evolved FEs were then tested on the images of unseen subjects taken from the FRGC database and Craniofacial Longitudinal Morphological Face (MORPH) database [53]. Their results showed that the GEFE$_{ML}$-evolved FEs, when compared to the SLBPM, used fewer patches, achieved comparable recognition rates on both datasets, and were significantly less expensive in terms of computational complexity. In addition, the resulting FEs generalized well to the unseen subjects from both databases.

In [27], a two-stage process known as Darwinian Feature Extraction (DFE) was developed. This technique used FEs that were evolved by GEFE$_{ML}$ to create Darwinian Feature Extractors (dFEs). The first stage of DFE superimposes a set of FEs to create a hyper FE. This hyper FE is then used to create a probability distribution function (PDF) of the occurrence of extraction per pixel. In the second stage, the PDF is sampled to determine which pixels will be processed via the LBP method. The selected pixels are then grouped using a clustering process. Next, a number of cluster centers are randomly selected within an image, and the selected pixels are then assigned to their closest cluster centers. The SLBPM is then applied to the resulting clusters. Their results showed that dFEs, when compared to the FEs evolved by GEFE$_{ML}$, achieved a higher recognition rate at a reduced computational complexity.

Adams et al. [28] extended the work of [27] by developing three types of Darwinian Feature Extraction – Clustering (DFE$_c$). Unlike DFE, which uses random cluster centers, DFE$_C$ uses K-Means clustering [55] (DFE$_{KM}$), Kohonen clustering [56] (DFE$_K$) and a combination of the two (DFE$_{K+KM}$ and DFE$_{KM+K}$) to create a user-specified number of clusters. These GEBAs were applied to datasets formed from subjects taken from the FRGC database. Their results showed that DFE$_{KM}$ and DFE$_{K+KM}$ performed as well as DFE in terms of accuracy, while DFE$_K$ and DFE$_{KM+K}$ performed worse than DFE. In addition, their results showed that DFE$_{K+KM}$ outperformed the other DFE$_C$ methods.

2.2.3. Genetic & Evolutionary Biometric Security

There has also been an increase in the use of GEBAs for biometric-based security, giving rise to a new field of study known as Genetic & Evolutionary Biometric Security (GEBS). Shelton et al. [9,10] introduced a number biometric-based access control protocols that used disposable FEs and their associated feature vectors (FVs) to mitigate replay attacks on a facial recognition systems. In [9], the researchers showed that the FEs evolved by GEFE$_{ML}$ and their associated FVs were unique from oneanother and achieved high recognition rate. As a result, either the FEs or their associated FVs could be used to mitigate replay attacks by disposing of a particular FE or FV after usage. In [10], Shelton et al. extended their work by demonstrating that permuting the order of feature extraction resulted in the development of FVs with greater distinction from one another. The net effect of this being a further reduction in the possibility of the occurrence of a successful biometric-based replay attack.In [11],

Shelton et al.showed that the dFEscreated by the DFE method could also be used to mitigate replay attacksas well.

Adams et al. [39]developed a method for mitigating image reconstruction attacks from a compromised biometric gallery consisting of biometric templates extracted using the SLBPM. First, they demonstrated how a General Regression Neural Network [57] and a SSGA could be used to reconstruct facial images from the biometric templates. They then developed a neurogenetic method to distort the biometric templates, so that if reconstructed, they would not match with their original biometric templates.

2.3. Local Binary Patterns Method (LBP)

The Local Binary Patterns Method (LBP), which was developed by Ojala et al. [32], is a popular texture descriptor that has been used in a variety of applications [25-28,30, 31, 34, 58-61]. Within the biometrics community, the LBP method has become a very popular feature extraction technique [25-28, 30, 31, 58] due to its computational simplicity and its invariance against monotonic gray level changes [34].

The LBP works as follows. First, an image is segmented into a grid of evenly sized regions, which are referred to as patches. Within each patch, the LBP operator is then used to label each interior pixel by subtracting its intensity value,i_c, from the intensity value of each of its P neighboring pixels at a radius R,i_p, where $p = 0,..., P\text{-}1$. A texture, which is essentially a bit pattern, is then formed, as shown in Equation 1, where if the difference between the intensity values is negative, a zero is added to the bit string, otherwise a one.

$$T = \{s(i_0 - i_c),\ s(i_1 - i_c),\ ...,\ s(i_{P-1} - i_c)\} \tag{1}$$

$$s\ (i_p - i_c)\ = \begin{cases} 0, & \text{if } i_p < i_c \\ 1, & \text{if } i_p \geq i_c \end{cases} \tag{2}$$

Each interior pixel can then be represented by a unique LBP code by assigning a binomial weight to the elements in the texture as follows:

$$LBP = \sum_{p=0}^{P-1} s\ (i_p - i_c)\ 2^p \tag{3}$$

Associated with each patch is a histogram where each bin represents the frequency of a particular LBP code. Using a neighborhood size of P, there are 2^P unique LBP codes. However, Ojala et al. [32] showed that only a subset of the possible patterns, known as uniform patterns, are necessary to describe the texture of an image. For our research, uniform patterns are those texture patterns that have at most two one-to-zero or zero-to-one bit transitions when the texture is traversed circularly. These uniform patterns account for a high percentage of the resulting texture patterns and also contain the most texture information

[33]. Therefore, instead of having a histogram consisting of 2^P bins, each histogram would now consist of only *P(P-1)+3* bins, where *P(P-1)* bins account for the uniform patterns with exactly two bit transitions, one bin represents the all zero bit pattern, another bin represents the all ones bit pattern, and an additional bin is used to store the frequency of the non-uniform patterns.

The histograms for each patch are then concatenated to form a feature template for the image, which consists of the number of bins, *P(P-1)+3*, times the number of patches used, *N*, features.

3. The GEBAs

3.1. Genetic & Evolutionary Feature Weighting/Selection—Machine Learning (GEFeWS$_{ML}$)

Genetic & Evolutionary Feature Weighting/Selection—Machine Learning (GEFeWS$_{ML}$) was developed by Alford et al. [25] for hybrid feature weighting and selection for facial and periocular recognition. GEFeWS$_{ML}$ evolves feature masks (FMs) that use a low percentage of features, achieve high recognition rates, and that generalize well to unseen instances. The technique, which is an instance of an Estimation of Distribution Algorithm (EDA) [50], works as follows.

First, the set of available instances is split into a training set, a validation set, and a test set. The EDA then randomly generates an initial population of Q real-valued candidate FMs. Each candidate FM, fm_i, can be represented by a tuple$\langle M_i,\ fit_i \rangle$where $M_i = \langle \mu_{i,0},\ \mu_{i,1},\ ...,\ \mu_{i,n-1} \rangle$ represents the set of mask values and where fit_i represents the fitness value. The mask values are initially within the range [0..1]. Any mask value that is lower than 0.5 is set to 0 and the corresponding biometric feature would not be used during comparisons. Otherwise, the mask value is used to weight the corresponding biometric feature.

Next, each candidate FM is applied to the probe and gallery templates within both the training and validation sets. The resulting probe templates are then compared to each gallery template using the Weighted Manhattan Distance (wMD) formula shown in Equation 4, where h_j and h_l are two templates that are being compared, n is the original number of features in the biometric templates, $\mu_{i,k}$ is a FM value, k represents thek^{th} feature, and the function f_{WS} represents the process of feature weighting/selection as performed by GEFeWS$_{ML}$. The subject associated with the template within the gallery set with the smallest weighted Manhattan distance when compared to the probe is considered the match.

$$wMD_{WS}(h_j,\ h_l,\ fm_i) = \sum_{k=0}^{n-1} |h_{j,k} - h_{l,k}|\ f_{WS}(u_{i,k}) \qquad (4)$$

$$f_{WS}(u_{i,k}) = \begin{cases} 0 & \text{,if } u_{i,k} < 0.5 \\ u_{i,k} & \text{,if } u_{i,k} \geq 0.5 \end{cases} \tag{5}$$

The fitness function shown in Equation 6 is then used to evaluate each candidate FM based on its performance on the training and validation sets, where ε is the number of recognition errors that occurred when the candidate FM was applied to the probe and gallery templates, m is the number of features used by the candidate FM, and where n is the original number of features in the biometric templates.

$$fit_i = 10\varepsilon + \frac{m}{n} \tag{6}$$

The best performing candidate FM on the validation set is then stored and will be referred to as FM^*. Next, the top 50% best performing candidate FMs, based on their performance on the training set, are used to create a probability density function (PDF). The PDF is then sampled to create $(1-\alpha)Q$ offspring, where α is the user-defined percentage (in this case, 25%) of FMs that are allowed to survive into the next generation (known as the elites). The offspring are then evaluated based on their performance on the training and validation sets. Any offspring whose performance on the validation set is better than that of FM^* will become the new FM^*. A new population is then formed using the $(1-\alpha)Q$ offspring and the αQ elites. This evolutionary process is then continued until a user-specified stopping condition has been satisfied. Once the condition has been met, the best performing FM from the population based on its performance on the training set, which will be referred to as FM^{ts}, as well as FM^* are returned. These FMs are then applied to the test set in order to determine how well they generalize to unseen instances.

3.2. Reverse Engineered Feature Extraction (REFE)

Reverse Engineered Feature Extraction (REFE) is a technique that creates feature extractors (FEs) that obtain higher recognition rates than the SLBPM, while using a smaller number of patches, as compared to the SLBPM. This technique works by first analyzing statistically the FMs evolved by GEFeWS$_{ML}$ [25] to determine the percentage of features selected for use within each patch. For each FM, apatch percentage vector (PPV) is calculated by dividing the number of features used in a patch by the number of total features in a patch. For the results presented in this chapter, each image is segmented into 24 patches and each patch consisted of 59 bins. Therefore, the total number of features in a patch is 1416 (24 patches × 59 bins).

Each PPV is then evaluated at each patch position to create a corresponding FM. Because each image is segmented into 24 equally size patches, the PPV is evaluated based on the top 24 patches to the top patch, in terms of their percentage. In addition to the PPVs for the 30 best performing FMs, an average PPV (PPV$_{avg}$) is also formed. PPV$_{avg}$ is the average percentage of features used at each patch position for the 30 PPVs. Therefore, 31 total PPVs are cre-

ated. The PPVs were then used to create a set of FEs. Each FE is created by selecting the top user-specified number ofpatches based on the patches PPV value, starting with all 24 patches as a baseline working down to the top performing patch. Ties were broken based on the patch position. A candidate FM is then created from each FE in the following manner. If a patch is selected to be present in the FE, then all of the features extracted by that patch will be used for matching. Otherwise, no features will be extracted from the patch.

3.3.Genetic & Evolutionary Feature Extraction-Machine Learning (GEFE$_{ML}$)

Genetic &Evolutionary Feature Extraction—Machine Learning (GEFE$_{ML}$), which was developed by Shelton et al. [26],evolves LBP-based FEs that result in high recognition accuracies, use a low percentage of features, and generalizes well to unseen images. Unlike the SLBPM, which uses non-overlapping uniform patches that extract from the entire image, GEFE$_{ML}$ evolves FEs with overlapping uniform patches that only extract from a subset of an image. The GEFE$_{ML}$ technique, which is also an instance of an EDA [50], works as follows.

As done with GEFeWS$_{ML,}$ first the set of available instances is split into a training set, a validation set, and a test set. GEFE$_{ML}$ then randomly generates an initial population of Q candidate FEs, where each FE, fe_i, can be represented as a six-tuple,$\langle X_i, Y_i, W_i, H_i, M_i, f_i \rangle$. The first two elements in the tuple represent the x and y coordinates of the centers of the N possible patches, where $X_i = \langle x_{i,0}, x_{i,1}, ..., x_{i,N-1} \rangle$ and $Y_i = \langle y_{i,0}, y_{i,1}, ..., y_{i,N-1} \rangle$. The next two elements represent the widths and heights of the N patches respectively, where $W_i = \langle w_{i,0}, w_{i,1}, ..., w_{i,N-1} \rangle$ and$H_i = \langle h_{i,0}, h_{i,1}, ..., h_{i,N-1} \rangle$. It is important to note that the size of the patches were uniform due to the results presented in [29], therefore, the weights and heights of each patch were equivalent. The final two elements in the six-tuple represent the masking values of each patch, where $M_i = \langle m_{i,0}, m_{i,1}, ..., m_{i,N-1} \rangle$, and wheref_i is the fitness for fe_i. The masking values determine whether a patch is activated or deactivated.

Next, the fitness function shown in Equation 7 is used to evaluate each candidate FE based on its performance on the training and validation sets, where γ represents the percentage of image space (measured in pixels) covered by fe_i.

$$f_i = 10\varepsilon + \gamma \qquad (7)$$

To evaluate the candidate FEs, eachfe_i is applied to the probe and gallery sets in both the training and validation set to create feature vectors (FVs) for the images. The resulting FVs for the probe and gallery sets are then compared using the Manhattan Distance measure. The subject associated with the template within the gallery set with the smallest Manhattan distance when compared to the probe is considered the match.

The best performing FE on the validation set is then stored and will be referred to as FE^*. As with GEFeWS$_{ML}$, the top 50% best performing candidate FMs, based on their performance on the training set, are used to create a PDF. The PDF is then sampled to create $(1-\alpha)Q$ offspring, where $\alpha = 5\%$. The offspring are then evaluated based on their performance on the

training and validation sets. Any offspring FE that outperforms FE^* becomes the new FE^*. Finally, a new population is formed using the $(1-\alpha)Q$ offspring and the αQ elites. This evolutionary process is then continued until a user-specified stopping condition has been satisfied, after which two FEs are returned: the best performing FE on the training set, FE^{ts}, and FE^*. The returned FEs are then applied to the test set to evaluate how well they generalize to unseen instances in comparison to the SLBPM.

3.4. Darwinian-Based Feature Extraction (DFE) using Random Clustering

Darwinian-based Feature Extraction (DFE) [27] is a two-stage process for developing FEs to be used for facial recognition. In the first stage, a set of FEs evolved by $GEFE_{ML}$ are superimposed onto each other to create a hyper FE. From this hyper FE, a probability distribution function (PDF) is created which is essentially a two-dimensional array that represents the number of times a certain pixel is selected for use by the set of FEs. In the second stage of the process, a Darwinian-based FE (dFE) is created by sampling the PDF via k-tournament selection [62]. This dFE can be represented as a 3-tuple, $\langle c, \mu, \rho \rangle$, where c is the number of clusters in the dFE, μ is the selection pressure of tournament selection, and ρ is the resolution of the clusters. To create a dFE, first a number of pixels, $c \times \beta \times \rho$, where β represents the user-specified number of pixels per cluster, are selected from the PDF for use in the clustering process. Pixels are selected for use via k-tournament selection in which k = $\mu^* \sigma$ pixels, where σ represents the total number of positions in the PDF that have been processed at least once, compete for selection. The pixel with the greatest consistency (i.e. the pixel that was used the most by the set of FEs) will be considered the winner of the tournament and will not be allowed to win again. Next, *c* locations within an image are randomly selected to serve as cluster centers. The pixels that were selected for use are then assigned to their closest cluster center. The SLBPM is then applied to the clustered pixels in a manner similar to how the SLBPM is traditionally applied to pixels within a patch. Afterwards, the histograms associated with each cluster are concatenated and are used as the feature vectors for the given images.

3.5. Darwinian-Based Feature Extraction using Kohonen and K-Means Clustering (DFE_{K+KM})

Darwinian Feature Extraction-Clustering [28] using K-means[55] and Kohonen[56] clustering (DFE_{K+KM}) works as follows. First, Kohonen clustering is applied in the following manner. The algorithm iterates though each of the pixels selected for use, moving the nearest cluster center towards the given pixel. The distance that the center is moved is based on a user specified learning rate. For example, given a learning rate of 0.25, the magnitude that the center is moved would be 25% of the distance between the current pixel and the center. After iterating through all of the pixels, the distance between each center's starting and ending position is calculated. If the locations of the centers have not moved or a user-specified number of iterations have occurred, the Kohonen clustering process is halted. K-Means clustering is then applied to the resulting clusters in the following manner. First, each pixel selected for use via tournament selection is assigned to one of the *K* clusters centers determined by

the Kohonen clustering process. Once each pixel has been assigned, the positions of the K centers are recalculated based on the average of the pixels assigned to that center. Once each of the centers has been repositioned, the distance between their old and new position is measured. When the positions of the centers remain constant, the K-Means clustering process is considered complete. The SLBPM is then applied to the clustered pixels and the histograms associated with each cluster are concatenated and are used as the feature vectors for the given images.

4. Experimental Results

4.1. Databases

For our experiments, images were selected from two diverse databases: the Face Recognition Grand Challenge (FRGC) database [52] and the Craniofacial Longitudinal Morphological Face (MORPH) database [53]. The images within the two facial databases were acquired in different manners. The images within the FRGC database were collected in a controlled setting (i.e. controlled lighting, frontal pose, and neutral facial expression) and were acquired in a single sitting. In contrast, the images in the MORPH database were collected in an uncontrolled setting, were acquired over a period of time, and were of diverse ethnicities.

From the FRGC database, 489 subjects were selected. From the MORPH database, 300 subjects were selected. From the subjects selected from each database, three datasets were formed: a training set, a validation set, and a test set. The FRGC datasets were as follows: 163 of the subjects were used to form the training set, which will be referred to as FRGC-163$_{trn}$; An additional 163 subjects were used to form the validation set, which will be referred to as FRGC-163$_{val}$; The remaining 163 subjects were used to form the test set, which will be referred to as FRGC-163$_{tst}$. The MORPH datasets were as follows: 100 subjects were used to form the training set, which will be referred to as MORPH-100$_{trn}$; 100 subjects were used to form the validation set, which will be referred to as MORPH-100$_{val}$; the remaining 100 subjects were used to form the test set, which will be referred to as MORPH-100$_{tst}$.

For each of the facial datasets, three frontal facial images of each subject were selected and used to form the probe and gallery sets. The probe sets consisted of one image per subject, and the gallery sets consisted of the remaining two images per subject. The SLBPM was then used to extract 1416 (24 patches × 59 bins) facial features from each image and served as the baseline for our experiments.

Using these datasets, we performed two experiments. For the first experiment, each GEBA was used to evolve FMs or FEs for the FRGC and MORPH facial templates within the training sets. This will be referred to as '*Opt*' because we are attempting to maximize the recognition accuracy while minimizing the percentage of features used. The resulting FMs/FEs were then applied to the test sets in order to evaluate how well they generalized to unseen subjects within the respective test sets. The application of the FM/FE that performed best on the training set (i.e. FM^{ts} and FE^{ts}) will be referred to as '*Opt-Gen*' because we are evaluating the

generalization ability of these FMs/FEs. Similarly, the application of the best performing FM or FE on the validation set (i.e. FM^{ts} and FE^{ts})will be referred to as '*Val-Gen*' because we are evaluating how well these FMs and FEs generalize.

For the second experiment, we evaluated the cross-generalization ability of the resulting FMs/FEs for each GEBA. To do so, the FMs/FEs returned for the FRGC templates were applied to the MORPH test set and the FMs/FEs returned for the MORPH templates were applied to the FRGC test set.

4.2. Results

The results of our experiments were generated as follows. The SLBPM was applied to the training and test sets for the variousdatabases. Its performance served as the baseline for our experiments.

GEFeWS$_{ML}$ was run 30 times on the FRGC, MORPH, and DLFW datasets. The EDA evolved a population of 20 FMs and always retained 5 (α=25%) elites. A maximum of 1000 function evaluations were allowed for each run, and at the end of each run, the best performing FM on the training set, *FM* ^{ts}, and the best performing FM on the validation set, *FM* *, were returned. These FMs were then applied to their respective test set in order to evaluate their generalization performances.

REFE analyzed the FMs evolved by GEFeWS$_{ML}$for the three datasets. The GEFeWS$_{ML}$ evolved FMs were used to create a number of FEs. For each dataset, 62 PPVs were created (31 corresponding to the FM^{ts}s and 31 corresponding to the FM*s). These PPVs were used to create FMs based on patch position and percentage of features used.The resulting FMs were then applied to their respective test set in order to evaluate their generalization performances.

GEFE$_{ML}$ was also run 30 times on the datasets, evolved a population size of 20 FEs, always retained one elite, and allowed a maximum of 1000 function evaluations per run. At the end of 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**). The resulting FEs were then applied to the test sets in order to evaluate how well they generalized to the test sets.

To construct a hyper FE, DFE used the 30 *FE* *s evolved by GEFE$_{ML}$ for the FRGC, MORPH, and DLFW datasets. To test DFE on FRGC, dFEs were created with c values of 16, 12 and 8. These values were chosen based on the average number of patches activated in the 30 *FE* *s validated on FRGC-163$_{val}$, which was 16; c values of 12 and 8 came from 75% and 50% of 16 clusters. For each c,a ρ of 100% to 10% was used, using every tenth percentage in between. A selection pressure from 100% to 0% was also used, using every tenth percentage.

To test DFE on MORPH, dFEs were created with c values of 13, 10, and 7. These values were chosen based on the average number of patches activated in the *30 FE* *s validated on MORPH-100$_{val}$, which was 13; c values of 10 and 7 came from 75% and 50% of 16 clusters. For each c,a ρ of 100% to 10% was used, using every tenth percentage in between. A selection pressure from 100% to 0% was also used, using every tenth percentage.

To test the performance of DFE_{K+KM}, the best hyper FEs from both FRGC (using $c = 16$) and MORPH (using $c = 13$) were used in the Kohonen clustering process. The results from Kohonen clustering were then used for K-Means clustering. This process was performed for each of the 30 DFE runs. The resulting feature extractors were then applied to the FRGC and MORPH test sets.

The performances of these methods were separated into equivalence classes using ANOVA and t-tests based on their average recognition accuracy, the percentage of features used, and their computational complexity, which is the number of pixels processed or extracted by each method. Those methods that achieved higher recognition rates, used a lower percentage of features, and had a reduced computational complexity, were considered to be best.

4.2.1. Experiment I

The results of our first experiment are shown in Tables 1 and 2. Table 1 shows the performances of the GEBAs on the FRGC dataset. Table 2 shows the performances of the GEBAs on the MORPH dataset. Within each table, the first column represents the methods, where the asterisk denotes the performance of the GEBAs on the training set. The second column represents the average recognition accuracy for each method. The third column represents the average computational complexity of each method, and the final column represents the average percentage of features used by each method. Note that for the SLBPM, the accuracy was deterministic. In addition, the results for REFE were the performances of the best performing FMs.

With respect to $GEFeWS_{ML}$, the FMs evolved for the training set used an average of 45.02% of the features to achieve a higher recognition rate than the SLBPM. The FRGC Opt-Gen and Val-Gen performances show that the resulting FMs generalized well to the test set. When the generalization performances were compared, the Val-Gen performance was better in terms of accuracy, while the Opt-Gen performance used a lower percentage of features. This may be due to the need for more features for adequate generalization.

REFE performed well on $FRGC\text{-}163_{tst}$. The best performing FM created by analyzing the FM^{ts}s evolved by $GEFeWS_{ML}$ achieved a 93.84% recognition rate while using only 29.16% (7 patches) of the features. The best performing FM created by analyzing the FM^*s achieved a higher recognition rate of 94.46% using the same percentage of features. Although REFE did not outperform the SLBPM in terms of recognition accuracy, it's important to note that it significantly reduced the computational complexity of feature extraction by reducing the number of patches from 24 to 7.

In terms of recognition accuracy and feature usage, $GEFE_{ML}$ outperformed the SLBPM on $FRGC\text{-}163_{trn}$ and $FRGC\text{-}163_{tst}$. However, the average computational complexity of the FEs evolved by $GEFE_{ML}$ was higher than the computational complexity of the SLBPM. This is due to the fact that the patches within the evolved FEs had a large amount of overlap. This overlap resulted in an increased number of pixels to be processed and, consequently, an increase in computational complexity. Though the computational complexities for FEs

evolved by GEFE$_{ML}$ were larger, they were selected due to their superior recognition accuracy when compared to the SLBPM. When the performances of the FE^{ts}s were compared to the FE*s in terms of recognition accuracy, the FE*s (Val-Gen) performed statistically better than the FE^{ts}s (Opt-Gen).

For FRGC-163$_{tst}$, DFE outperformed the SLBPM in terms of recognition accuracy as well as computational complexity. The recognition accuracy for DFE on FRGC-163$_{tst}$ was 97.24% and the computational complexity was approximately 10% less than the SLBPM.

DFE$_{K+KM}$ also outperformed SLBPM in terms of recognition accuracy and computational complexity. This method achieved a recognition accuracy of 97.20% on FRGC-163$_{tst}$. Because the hyper FE from DFE was used, it had the same computational complexity as DFE.

When the performances of the GEBAs on the test set were compared in terms of accuracy, DFE and DFE$_{K+KM}$outperformed the other GEBAs. However, in terms of feature usage and computational complexity, REFE performed best.

Method	Accuracy	Comp. Complexity	% of Features
SLBPM (Training)	92.02%	10379	100.00%
SLBPM (Test)	95.09%	10379	100.00%
GEFeWS$_{ML}$ (Opt)*	97.63%	10379	45.02%
GEFeWS$_{ML}$(Opt-Gen)	94.42%	10379	45.02%
GEFeWS$_{ML}$ (Val-Gen)	94.58%	10379	48.45%
REFE (Opt-Gen)	93.84%	3027	29.16%
REFE (Val-Gen)	94.46%	3027	29.16%
GEFE$_{ML}$ (Opt)*	96.85%	11776	66.67%
GEFE$_{ML}$ (Opt-Gen)	96.64%	11776	66.67%
GEFE$_{ML}$ (Val-Gen)	96.87%	11776	66.67%
DFE (HFE$_{val}$on Test Set)	97.24%	9011	66.67%
DFE$_{K+KM}$	97.20%	9011	66.67%

Table 1. FRGC Results.

GEFeWS$_{ML}$ outperformed the SLBPM on MORPH-100$_{trn}$, using significantly fewer features while significantly increasing the recognition rate over the SLBPM. In addition, the resulting FMs generalized well to the test set. In terms of accuracy, the Val-Gen performances performed better statistically than the Opt-Gen performance. However, the Opt-Gen performances used a lower percentage of features in comparison to the Val-Gen performances.

The REFE created FMs performed well on MORPH-100$_{tst}$. The best performing FM created with respect to the FM^{ts}s outperformed the SLBPM, achieving a 70.95% recognition rate

while using only 41.66% of the features and significantly reducing the computational complexity by using only 10 patches. The best performing FM created with respect to the FM*s also performed well on the test set, achieving a recognition accuracy that was slightly lower than the SLBPM while using only 29.16% (7 patches) of the features.

$GEFE_{ML}$ outperformed the SLBPM on MORPH-100$_{trn}$ and MORPH-100$_{tst}$. The FE^{ts}s had an average 39.66% recognition accuracy when applied on MORPH-100$_{trn}$, which was significantly better than the recognition accuracy achieved by the SLBPM. Like the FEs trained on FRGC, the average computational complexities of the FEs were also higher than for the SLBPM. The computational complexity of evolved FEs on the MORPH set were much higher due to the overlap as well as the large dimensions of the patches.However, FEs were still chosen primarily because of the recognition accuracy. In addition, when the performances of the FE^{ts}s were compared to the FE*s in terms of recognition accuracy, the FE*s performed better.

DFE outperformed the SLBPM in terms of recognition accuracy, computational complexity, and feature usage. DFE achieved a recognition rate of 59.73% while having a computational complexity close to 50% less than the SLBPM.

DFE_{K+KM} also outperformed the SLBPM in terms of recognition accuracy, computational complexity, and feature usage. This method achieved a recognition rate of 62.07% while having a computational complexity reduction of nearly 50% with respect to SLBPM.

Comparing the generalization performances of the GEBAs, REFE outperformed the other GEBAs in terms of recognition accuracy, computational complexity, and feature usage.

Method	Accuracy	Comp. Complexity	% of Features
SLBPM (Training)	27.00%	10379	100.00%
SLBPM (Test)	70.00%	10379	100.00%
GEFeWS$_{ML}$ (Opt)*	39.53%	10379	45.99%
GEFeWS$_{ML}$ (Opt-Gen)	61.67%	10379	45.99%
GEFeWS$_{ML}$ (Val-Gen)	64.30%	10379	48.61%
REFE (Opt-Gen)	70.95%	4324	41.66%
REFE (Val-Gen)	67.98%	3027	29.16%
GEFE$_{ML}$ (Opt)*	39.66%	20034	58.33%
GEFE$_{ML}$ (Opt-Gen)	44.37%	20034	58.33%
GEFE$_{ML}$ (Val-Gen)	57.67%	11050	54.17%
DFE (HFE$_{val}$on Test Set)	59.73%	5525	54.17%
DFE$_{K+KM}$ (HFE$_{val}$on Test Set)	62.07%	5525	54.17%

Table 2. MORPH Results.

4.2.2. Experiment II

Tables 3 and 4 show the results of our second experiment. Table 3 presents the cross-generalization performances of the MORPH FMs/FEs on the FRGC test set, while Table 4 presents the cross-generalization performances of the FRGC FMs/FEs on the MORPH test set. The first column of each table represents the methods, the second column represents the average recognition accuracy, the third column represents the average computational complexity of each method, and the final column represents the average percentage of features used by each method. As for Experiment I, for the SLBPM, the accuracy was deterministic and the results shown for REFE were the performances of the best performing FMs.

With respect to GEFeWS$_{ML}$, the FM^{ts}s and FM*s evolved for the MORPH dataset generalized well to the FRGC test set. Although the average accuracy waslower than the SLBPM and lower than the results presented in Table 1, it is important to note that the FMs were optimized for the MORPH dataset, while the SLBPM and FMs used for Experiment I were applied directly to and trained on the FRGC test set. In addition, the FMs used less than 50% of the features in contrast to the SLBPM which used 100% of the features. When the Opt-Gen and Val-Gen performances were compared, in terms of accuracy, Val-Gen performed than Opt-Gen. However, Opt-Gen used fewer features.

Method	Accuracy	Comp. Complexity	% of Features
SLBPM (Test)	95.09%	10379	100.00%
GEFeWS$_{ML}$(Opt-Gen)	92.70%	10379	45.99%
GEFeWS$_{ML}$ (Val-Gen)	93.72%	10379	48.61%
REFE (Opt-Gen)	72.97%	4324	41.66%
REFE (Val-Gen)	68.97%	3027	29.16%
GEFE$_{ML}$ (Opt-Gen)	88.41%	20034	58.33%
GEFE$_{ML}$ (Val-Gen)	93.37%	11050	54.17%
DFE (HFE$_{val}$on Test set)	94.66%	5525	54.17%
DFE$_{K+KM}$	93.82%	5525	54.17%

Table 3. MORPH to FRGC Cross-Generalization Results.

The best performing FMs created by REFE for the MORPH dataset did not generalize well to FRGC-163$_{tst}$. Neither FMs outperformed the SLBPM in terms of recognition accuracy. However, the best performing FM with respect to the FM^{ts}s achieved a 72.97% recognition rate while using only 41.66% of the features and while having a more than 50% lower computational complexity. The best performing FM with respect to the FM*s achieved a 68.97% recognition rate while using only 29.16% of the features and also having a 50% lower computational complexity. This reduction in features and processing time (in terms of computational complexity) is important to highlight, especially since the SLBPM was applied directly to the test set.

With respect to $GEFE_{ML}$, the recognition accuracyof the SLBPM was superior to the average accuracy of both the FE^{ts}s as well as the FE^*s. However, the percentage of features for $GEFE_{ML}$was 50% less than the SLBPM. The Val-Gen recognition accuracy was also statistically greater than the Opt-Gen accuracy, and the computational complexity was less for Val-Gen than for Opt-Gen.

For DFE, the recognition accuracy was slightly lower than the accuracy of the SLBPM. However, the percentage of features used by DFE, as well the computational complexity of DFE, was far less than for the SLBPM.

DFE_{K+KM} performed well on FRGC-163$_{tst}$, achieving a recognition rate slightly lower than the SLBPM while having a nearly 50% lower computational complexity and feature usage.

Comparing the cross-generalization performances of the GEBAs, DFE cross-generalized best to the FRGC dataset. The dFEs achieved recognition rates comparable to the SLBPM, while using significantly fewer features and requiring less processing time.

When the FMs evolved by $GEFeWS_{ML}$for FRGC were applied to the MORPH test set, they did not outperform the SLBPM, which was applied directly to the test set. However, the performances of the FM^{ts}s evolved for FRGC were better statistically than the Opt-Gen results presented in Table 2. In addition, the Val-Gen performances for both experiments were statistically equivalent.

The best performing FMs created by REFE for FRGC generalized well to the MORPH test set. The FMs outperformed the SLBPM in terms of accuracy, computational complexity, and feature usage.

With respect to $GEFE_{ML}$, the recognition accuracyof the SLBPM was superior to the average accuracy of both the FE^{ts}sand FE^*s. However, there is still a reduction in the percentage of features used by $GEFE_{ML}$. The Val-Gen recognition accuracy was still statistically greater than the Opt-Gen accuracy, though the computational complexity remained the same. The recognition accuracy resulting from cross-generalizing FEs on the MORPH dataset is greater than the accuracy of generalizing FEs in Experiment I.

The recognition accuracy of DFE was less than the recognition accuracy of the SLBPM. However, the percentage of features used by DFE, as well as the computational complexity, was far less than the percentage of feature for the SLBPM.

DFE_{K+KM} outperformed the SLBPM in terms of accuracy, computational complexity, and feature usage. The GEBA achieved a 71.12% recognition rate, while using only 66.67% of the features and processing fewer pixels.

Comparing the performances of the SLBPM and the GEBAs, REFE was the best performing technique, generalizing well to the test set, while processing less than 50% of the available pixels and using less than 30% of the features.

Method	Accuracy	Comp. Complexity	% of Features
SLBPM (Test)	70.00%	10379	100.00%
GEFeWS$_{ML}$ (Opt-Gen)	65.19%	10379	45.02%
GEFeWS$_{ML}$ (Val-Gen)	65.60%	10379	48.45%
REFE (Opt-Gen)	72.98%	3027	29.16%
REFE (Val-Gen)	71.14%	3027	29.16%
GEFE$_{ML}$ (Opt-Gen)	64.76%	11776	66.67%
GEFE$_{ML}$ (Val-Gen)	65.63%	11776	66.67%
DFE (HFE$_{val}$on Test set)	62.57%	9011	66.67%
DFE$_{K+KM}$	71.12%	9011	66.67%

Table 4. FRGC to MORPH Cross-Generalization Results.

5. Discussion

In Experiment I, we showed that the GEBAs could evolve FEs and FMs that achieve accuracies comparable to the performance of the SLBPM on the test sets, while using significantly fewer features and a using a lower computational complexity. To further analyze the performances of the GEBAs, the Receiver Operator Characteristic (ROC) curve for the performance of the best FMs/FEs on the test sets were plotted in Figures 2 and 3. The ROC curves for GEFeWS$_{ML}$ were created using the normalized Weighted Manhattan Distance(NwMD) formula shown in Equation 8, and the ROC curves for GEFE$_{ML}$ were created using the normalized Manhattan Distance(NMD) formula shown in Equation 9. For these formulas, $h_i$and $h_j$are the feature templates being compared, $fm_l$is the GEFeWS$_{ML}$evolved FM, z is the zth featurewithin the templates, n is the total number of extracted features, andthe functionf_{WS}, asshown in Equation 5, represents the process of feature weighting/selection as performed by GEFeWS$_{ML}$. The ROC charts plot the true accept rate (y-axis) to the false accept rate (x-axis). For a given threshold from 0.0 to 1.0, all images in the probe dataset were compared to all images in the gallery dataset. If the NMD of the probe image and gallery image was below the threshold, it was considered a match. The true accept rate was increased if the two matching images were of the same subject, while the false accept rate was increased if the images were not of the same subject.

$$NwMD_{WS}(h_i, h_j, fm_l)=\sum_{z=0}^{n-1} \frac{|h_{i,z}-h_{j,z}|f_{WS}(u_{i,z})}{\max(h_{i,z}, h_{j,z})f_{WS}(u_{i,z})} \tag{8}$$

$$NMD(h_i, h_j)=\sum_{z=0}^{n-1} \frac{|h_{i,z}-h_{j,z}|}{\max(h_{i,z}-h_{j,z})} \tag{9}$$

Figures 2 and 3 show the ROC curves of the SLBPM, and the best performing FM^{ts}, FM^*, FE^{ts}, and FE^* on FRGC-163$_{tst}$ and MORPH-100$_{tst}$ respectively. The evolved FMs and FEs seem to perform comparable to the SLBPM while using significantly fewer features. It also appears that the FEs perform better than the SLBPM and the FMs on both datasets.

In addition, Figure 4 shows the graph of the accuracy of the best performing REFE created FMs on FRGC-163$_{tst}$ and MORPH-100$_{tst}$. By using only the patches that correspond to the highest feature usages, one can achieve recognition accuracies better than using all 24 patches as done by the SLBPM.
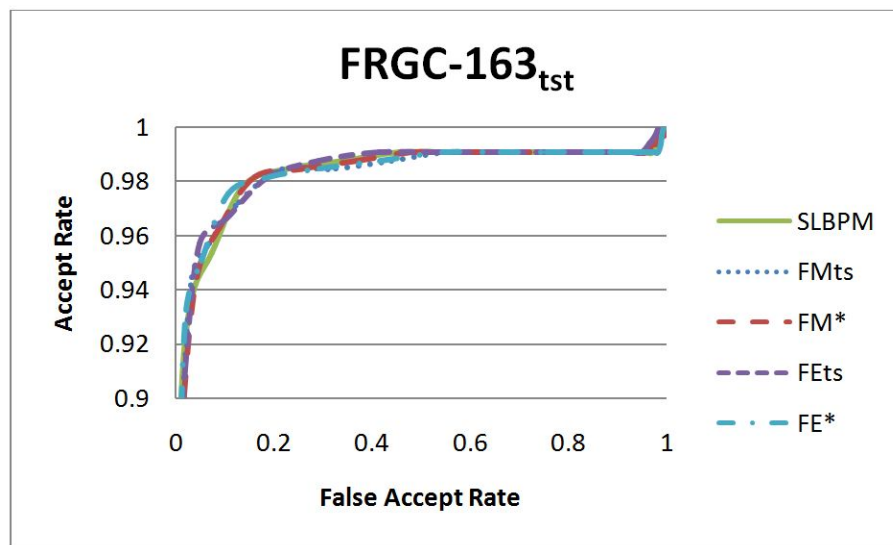
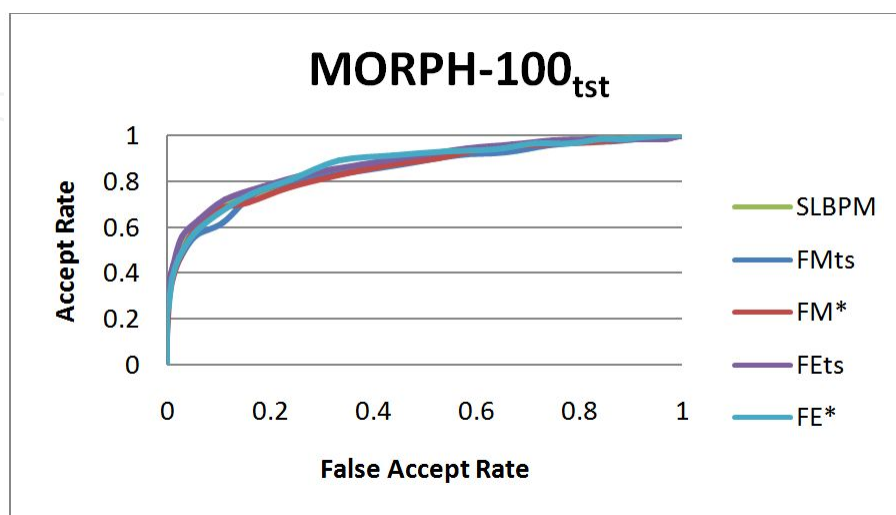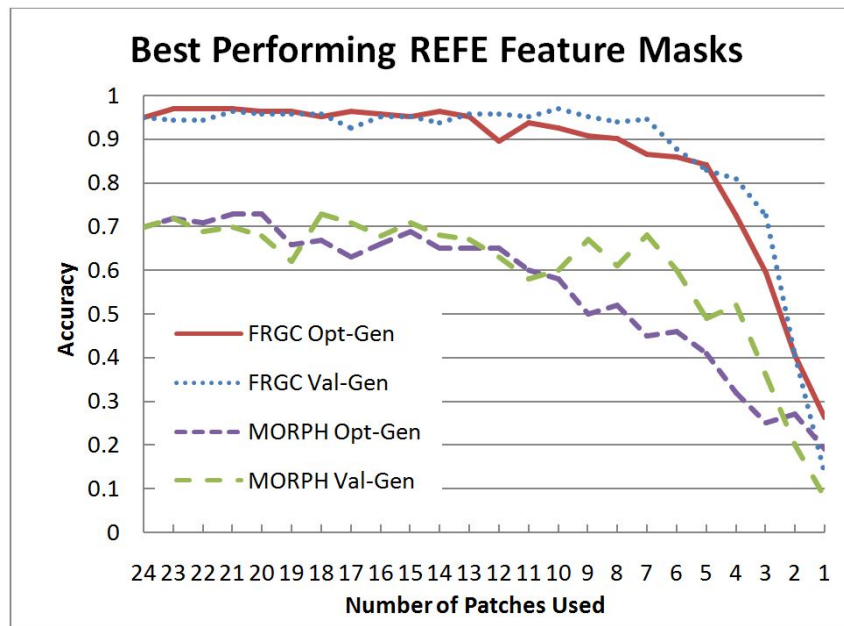**Figure 2.** ROC curves for the SLBPM and the best performing FMs and FEs on FRGC-163$_{tst}$.

Figure 3. ROC curves for the SLBPM and the best performing FMs and FEs on MORPH-163$_{tst}$.

Figure 4. Performance of the best REFE feature masks for FRGC and MORPH test sets.

6. Conclusions and Further Work

In conclusion, we presented several GEBAs for feature selection/weighting and for feature extraction. The GEBAs were applied to images of subjects taken from two facial. The resulting FMs/FEs developed for each database were then applied to instances of each unrelated database to test the notion of cross-generalization. Our results showed that the GEBAs achieved recognition rates comparable to the SLBPM while using significantly fewer features and a considerably lower computational complexity.

Further work will be devoted towards the use of these GEBAs to develop more secure biometric systems.

Acknowledgements

Author details

Aniesha Alford[1], Joseph Shelton[2*], Joshua Adams[2], Derrick LeFlore[2], Michael Payne[2], Jonathan Turner[2], Vincent McLean[2], Robert Benson[2], Gerry Dozier[2], Kelvin Bryant[2] and John Kelly[1]

*Address all correspondence to: jashelt1@ncat.edu

1 Department of Electrical and Computer Engineering, North Carolina A&T State University, USA

2 Department of Computer Science, North Carolina A&T State University, USA

References

[1] Davis, L. (1991). Handbook of genetic algorithms. Van Nostrand Reinhold, New York.

[2] Dozier, G., Homaifar, A., Tunstel, E., & Battle, D. (2001). An Introduction to Evolutionary Computation (Chapter 17). Intelligent Control Systems Using Soft Computing Methodologies, A. Zilouchian& M. Jamshidi (Eds.), CRC press , 365-380.

[3] Engelbrecht, A. P. (2007). Computational Intelligence: An Introduction,. John Wiley & Sons, Ltd.

[4] Fogel, D. (1995). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence,. IEEE Press.

[5] Nolfi, S., & Floreano, D. (2000). Evolutionary robotics- the biology, intelligence, and technology of self-organizing machines. In: Intelligent Robotics and Automation Agents. MIT Press Cambridge, MA.

[6] Dahal, K. P., Tan, K. C., & Cowling, P. I. (2007). Evolutionary Scheduling. Springer Verlag 978-3-54048-582-7 pages., 628.

[7] Miranda, E. R., & Biles, J. A. (2007). Evolutionary Computer Music. Springer-Verlag, New York, Inc., Secaucus, NJ, USA.

[8] Dasgupta, D. (1999). An overview of artificial immune systems and their applications. In Dasgupta, D., editor, Artificial Immune System and Their Applications Springer-Verlag , 3-23.

[9] Shelton, J., Bryant, K., Abrams, S., Small, L., Adams, J., Leflore, D., Alford, A., Ricanek, K., & Dozier, G. (2012). Genetic & Evolutionary Biometric Security: Disposable Feature Extractors for Mitigating Biometric Replay Attacks. *Proceedings of the 10th Annual Conference on Systems Engineering Research (CSER).*

[10] Shelton, J., Dozier, G., Adams, J., & Alford, A. (2012). Permutation-Based Biometric Authentication Protocols for Mitigating Replay Attacks. Paper presented at Proceedings of the 2012 IEEE World Congress on Computational Intelligence (WCCI).

[11] Shelton, J., Adams, J., Alford, A., Venable, M., Neal, S., Dozier, G., & Bryant, K. (2012). Mitigating Replay Attacks Using Darwinian-Based Feature Extraction. *IEEE Symposium on Computational Intelligence for Security and Defense Applications.*

[12] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

[13] Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm Intelligence*, Morgan Kaufmann.

[14] Nolfi, S., & Floreano, D. (2000). Evolutionary Robotics:The Biology, Intelligence and Technology of Self-Organizing Machines. MIT Press.

[15] Vadakkepat, P., Chen, Tan. K., & Ming-Liang, W. (2000). Evolutionary artificial potential fields and their application in real time robot path planning. *Proceedings of the 2000 Congress on Evolutionary Computation*, 256-263, 1.

[16] Hoffmann, F., & Pfister, G. (1997, November). Evolutionary design of a fuzzy knowledge base for a mobile robot. *International Journal of Approximate Reasoning*, 17(4), 447-469.

[17] Harvey, I., Husbands, P., Cliff, D.., Thompson, A., & Jakobi, N. (1997, June). Evolutionary robotics: the Sussex approach. *Robotics and Autonomous Systems*, 20(2-4), 205-224.

[18] Bentley, P. J. (1999). Evolutionary design by computers. San Francisco, Morgan Kaufmann.

[19] Gargiulo, A. A., Schnell, A. R., Tinker, M. L., Dozier, G., & Garrett, A. (2000). Use of Evolutionary Computation for Three-Dimensional Structural Design of a Lunar Habitat. http://nxt.ncat.edu/pdf/Use%20of%20EC_for_Prelim_Struct_Design_Lunar_Habitat.pdf.

[20] Dozier, G., Carnahan, B., Seals, C., Kuntz, L. A., & Ser-Geon, Fu. (2005). An interactive distributed evolutionary algorithm (IDEA) for design. 2005 IEEE International Conference on Systems, Man and Cybernetics, 10-12 Oct , 1, 418-422.

[21] Miranda, E. R. (2004). At the Crossroads of Evolutionary Computation and Music: Self-Programming Synthesizers, Swarm Orchestras and the Origins of Melody. *Evolutionary Computation*, 12(2), 137-158.

[22] Murata, T., Ishibuchi, H., & Tanaka, H. (1996, September). Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers & Industrial Engineering*, 30(4), 957-968.

[23] Hart, E., Ross, P., & Corne, D. (2005). Evolutionary Scheduling: A Review. *Genetic Programming and Evolvable Machines*, 6(2), 191-220.

[24] Dahal, K. P., Tan, K. C., & Cowling, P. I. (2007). Evolutionary Scheduling. *Springer Berlin /Heidelberg*, 317-330.

[25] Alford, A., Steed, C., Jeffrey, M., Sweet, D., Shelton, J., Small, L., Leflore, D., Dozier, G., Bryant, K., Abegaz, T., Kelly, J. C., & Ricanek, K. (2012). Genetic & Evolutionary Biometrics: Hybrid Feature Selection and Weighting for a Multi-Modal Biometric System. *IEEE SoutheastCon 2012*.

[26] Shelton, J., Alford, A., Abegaz, T., Small, L., Leflore, D., Williams, J., Adams, J., Dozier, G., Bryant, K., Ricanek, K., & Woodard, D. L. (2012). Genetic & Evolutionary Biometric Security: Feature Extraction from a Machine Learning Perspective. *IEEE SoutheastCon 2012*.

[27] Shelton, J., Venable, M., Neal, S., Adams, J., Alford, A., & Dozier, G. (2012). *Pixel Consistency, K-Tournament Selection, and Darwinian-Based Feature Extraction, Midwest Artificial Intelligence and Cognitive Science (MAICS) Conference*.

[28] Adams, J., Shelton, J., Small, L., Venable, M., Neal, S., Kim, J., & Dozier, G. (2012). Darwinian-based Feature Extraction using K-means and Kohonen Clustering. Paper presented at Midwest Artificial Intelligence and Cognitive Science (MAICS) Conference.

[29] Shelton, J., Dozier, G., Bryant, K., Smalls, L., Adams, J., Popplewell, K., Abegaz, T., Woodard, D., & Ricanek, K. (2011). Genetic and Evolutionary Feature Extraction via X-TOOLSS. in The 8th Annual International Conference on Genetic and Evolutionary Methods (GEM).

[30] Alford, A., Bryant, K., Abegaz, T., Dozier, G., Kelly, J., Shelton, J., Small, L., Williams, J., & Woodard, D. L. (2011). Genetic & Evolutionary Methods for Biometric Feature Reduction. Special Issue on: "Computational Intelligence in Biometrics: Theory, Methods and Applications", Guest Editor: Qinghan Xiao, International Journal of Biometrics.

[31] Alford, A., Adams, J., Shelton, J., Bryant, K., Kelly, J. C., & Dozier, G. (2012). Analyzing the Cross-Generalization Ability of a Hybrid Genetic & Evolutionary Application for Multibiometric Feature Weighting and Selection. *Genetic and Evolutionary Computation Conference (GECCO-2012)*.

[32] Ojala, T., Pietikainen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29, 51-59.

[33] Ojala, T., Pietikainen, M., & Mäenpää, T. (2002). MultiresolutionGray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *Proceedings of IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7), 971-987.

[34] Marcel, S., Rodriguez, Y., & Heusch, G. (2006). On The Recent Use of Local Binary Patterns for Face Authentication. *International Journal of Image and Video Processing-Special Issue on Facial Image Processing*, 1-9.

[35] Galbally, J., Fierrez, J., Freire, M., & Ortega-Garcia, J. (2007). Feature Selection Based on Genetic Algoirthms for On-Line Signature Verification. Paper presented at IEEE Workshop on Automatic Identification Advanced Technologies Alghero, Italy. 198-203.

[36] Ramadan, R. M., & Abdel-Kader, R. F. (2009, June). Face Recognition Using Particle Swarm Optimization-Based Selected Features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2(2).

[37] Abegaz, T., Dozier, G., Bryant, K., Adams, J., Mclean, V., Shelton, J., Alford, A., Ricanek, K., & Woodard, D. L. (2011). Applying GECs for Feature Selection and Weighting using X-TOOLSS". *The 8th Annual International Conference on Genetic and Evolutionary Methods(GEM)*.

[38] Alford., A., Popplewell, K., Dozier, G., Bryant, K., Kelly, J., Adams, J., Abegaz, T., Shelton, J., Woodard, D. L., & Ricanek, K. (2011). Hybrid GEMs for Multi-Biometric Recognition via X-TOOLSS. *The 8th Annual International Conference on Genetic and Evolutionary Methods (GEM)*.

[39] Adams, J., Dozier, G., Bryant, K., Shelton, J., Alford, A., Abegaz, T., & Leflore, D. (2012, 15-18 March). Neurogenetic Reconstruction of Biometric Templates: A New Security Threat? Paper presented at IEEE Southeast Con 2012, Orlando, FL.

[40] Garcia- Ortega, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., Satue, A., Hernaez, I., Igarza, J. J., Vivaracho, C., Escudero, C., & Moro, Q. I. (2003, December). MCYT baseline corpus: a bimodal biometric database. *IEEE Proc. Vis. Image Signal Process.*, 150(6), 395-401.

[41] AT&T Laboratories Cambridge. (2000). ORL Face Database. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[42] Ahmed, N., Natarajan, T., & Rao, K. R. (1974, Jan). Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1), 90-93.

[43] Shensa, M. J. (1992, Oc). The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms. *IEEE Transactions on Signal Processing*, 40(10), 2464-2482.

[44] Kumar, D., Kumar, S., & Rai, C. S. (2009). Feature selection for face recognition: a memetic algorithmic approach. *Journal of Zhejanga University Science A*, 10(8), 1140-1152.

[45] Neri, F., Cotta, C., & Moscato, P. (2012). Handbook of Memetic Algorithms. *Studies in Computational Intelligence*, Springer, 379.

[46] PAMI. (2001). The Extended Yale Face Database B. http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html.

[47] Jollife, I. T. (1986). *Principal Component Analysis.*, New York, Springer-Verlag.

[48] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Mullers, K. R. (1999, Aug). Fisher discriminant analysis with kernels," Neural Networks for Signal Processing IX, 1999.

Paper presented at Proceedings of the 1999 IEEE Signal Processing Society Workshop. 41-48.

[49] Schölkopf, B., Smola, A., & Müller, K. R. (1997). Kernel Principal Component Analysis. *ARTIFICIAL NEURAL NETWORKS- ICANN'97, Lecture Notes in Computer Science*, 1327, 583-588.

[50] Larrañaga, P., & Lozano, J. A. (2002). Estimation of Distribution Algorithms: A new tool for evolutionary computation. Springer.

[51] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 13(1), 71-86.

[52] Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., & Worek, W. (2005). Overview of face recognition grand challenge. Paper presented at Proc. IEEE Conference on Computer Vision and Pattern Recognition.

[53] Ricanek, K., & Tesafaye, T. (2006, April 10-12.) MORPH: A Longitudinal Image Database of Normal Adult Age-Progression". Paper presented at Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition. 341-345.

[54] Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill Companies, Inc.

[55] Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002, July). An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7).

[56] Kohonen, T. (1990). The self-organising map. *Proceedings IEEE*, 78(9), 1464-1480.

[57] Specht, D. F. (1991, Nov). A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6), 568-576.

[58] Miller, P. E., Rawls, A. W., Pundlik, S. J., & Woodard, D. L. (2010). Personal identification using periocular skin texture. Proceedings of the 2010 ACM Symposium on Applied Computing (SAC'10). ACM, New York, NY, USA, 1496-1500.

[59] Srensen, L., Shaker, S. B., & de Bruijne, M. (2010, Feb). Quantitative analysis of pulmonary emphysema using local binary patterns. *IEEE Transactions on Medical Imaging*, 29(2), 559-569.

[60] Unay, D., Ekin, A., Cetin, M., Jasinschi, R., & Ercil, A. (2007, August 23-26,). Robustness of Local Binary Patterns in Brain MR Image Analysis. Paper presented at Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Lyon, France,. 2098-2101.

[61] Woodard, D., Pundlik, S., Lyle, J., & Miller, P. (2010). Periocular region appearance cues for biometric identification. CVPR Workshop on Biometrics. San Francisco, CA , 162-169.

[62] Miller, B. L., & Goldberg, D. E. (1996, June). Genetic algorithms, selection schemes, and the varying effects of noise. *Evol. Comput*, 4(2), 113-131.