Comparison of Genetic-based Feature Extraction Methods for Facial Recognition

#Joseph Shelton[1], #Gerry Dozier[2], #Kelvin Bryant[3], #Lasanio Smalls[4], #Joshua Adams[5],

#Khary Popplewell[6], #Tamirat Abegaz[7], ^Damon L. Woodard[8], ~Karl Ricanek[9]

#Computer Science Department, North Carolina Agricultural and Technical State University
1601 East Market St. Greensboro, NC, 27411, United States of America

^314 McAdams Hall, Clemson University, Clemson, S.C. 29634-0974, United States of America

~CIS Building, Room 2010, 601 South College Road, Wilmington NC, 28403

jashelt1@ncat.edu[1], gvdozier@ncat.edu[2], ksbryant@ncat.edu[3], lrsmall@ncat.edu[4], jcadams2@ncat.edu[5],
ktpopple@ncat.edu[6], tamirat@programmer.net[7], woodard@clemson.edu[8], ricanekk@uncw.edu[9]

Abstract

In previous research, Shelton et al. presented a genetic-based method for evolving feature extractors for facial recognition. The technique presented evolved feature extractors that consisted of non-uniform, overlapping patches and did not cover the entire image. In this paper, we compare the use of non-uniform, overlapping patches with uniform, overlapping patches. Our results show a statistically significant performance improvement over the technique presented in Shelton's previous paper.

Introduction

Biometric recognition is the science of identifying an individual or group of individuals based on physical/behavioral characteristics or traits (Ross, 2007). One of the most popular biometric modalities is the face (Li and Jain, 2005; Ahonen, Hadid and Pietikinen 2006; Matas et al., 2002) and perhaps one of the more widely used techniques for extracting features from facial images for the purpose of biometric recognition is the Local Binary Pattern (LBP) method (Ojala and Pietikinen, 2002).

Shelton et al. introduces a genetic-based method, GEFE (Genetic & Evolutionary Feature Extraction), for evolving LBP feature extractors that consisted of non-uniformed, unevenly distributed patches that do not cover the entire image. The proposed method proved superior to the traditional LBP which uses uniform, evenly distributed, non-overlapping patches, that cover the entire image. In this paper, we introduce an alternative GEFE approach which is similar to the original GEFE approach with the exception that the unevenly distributed, overlapping patches are of uniform size.

The original GEFE method was theorized to have a bias due to the selection process of the coordinates for a patch. The coordinates represented the left corner of a patch, which increased the possibility of the patch dimensions exceeding the boundaries of a facial image. Any patch that exceeded the bounds was shifted till the whole patch was within the image space.

Because the possible coordinates could be anywhere on an image, the probability of selecting a patch that would just end up in the lower right hand corner was greater than any other location on the image. The method used in this research seeks to eliminate any potential bias by representing the coordinates of a patch as the center of it. This creates a greater probability of a patch being placed on all corners of an image.

The remainder of this paper is as follows: we will introduce the concept of GEFE for evolving LBP Feature Extractors composed non-uniform and uniform patches, as well as describing the genetic algorithm used in this research. We will discuss our experimental setup, we will show our results and finally we will present our conclusions and future work.

GEFE using Non-Uniform and Uniform Patch Sizes

LBP is a texture operator that can be used to extract texture information in the form of image features. The images used in this work are gray-scale, and are all facial images. For the standard LBP technique (Ojala and Pietikinen, 2002), a number of uniform, non-overlapping, and evenly distributed patches are used to cover an image. Texture features are then extracted from each patch area of the image.

Shelton et al. developed a genetic-based method for evolving LBP feature extractors that were composed of patches that were non-uniform, overlapping, unevenly distributed, and did not cover the entire image. Figure 1 provides an example of the patch layout of the standard LBP method (Figure 1a) and the original GEFE method (Figure 1b).

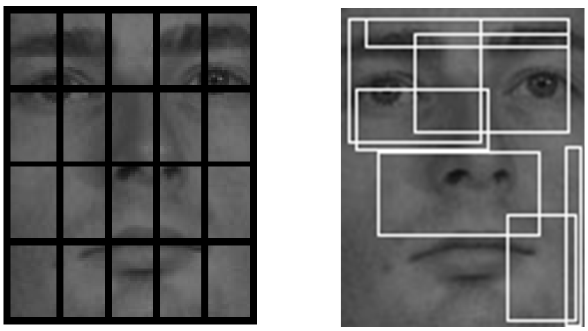

Figure 1a: Standard LBP **Figure 1b: GEFE**

Figure 1: Gray Scale Images fitted with patches

Given a layout of patches for an LBP feature extractor, the LBP method is applied to each interior pixel within the patch. Each pixel has a value between 0 and 255 that represents the intensity of its gray level. When LBP is applied to the pixels of a patch, a histogram is created that represents the unique texture pattern for that particular patch. The histograms of every patch on the image are then concatenated to form a unique set of features that represents the image.

Figure 2 provides an example of the LBP method being applied to a particular pixel value for the center pixel with an intensity value of 120. The center pixel is surrounded by 8 neighboring pixels, shown in the 1st sample pattern in Figure 2. The differences are calculated and shown in the 2nd pattern.

Upon inspection of the 3rd pattern, one can see a series of zeros and ones. This pattern is created by taking the difference between each neighbor pixel and the center pixel. If the difference is negative, then the conversion value will be zero. If the difference is zero of greater, then the conversion value for that neighbor will be one. The third pattern is then 'unwrapped' to form a binary string and the string is converted to an integer number, which is the LBP value for that center pixel. For the center value in Figure 2, the LBP is 14 due to the sequence: 00001110. Where the binary string starts it's unwrapping depends on the user, but this research starts the unwrapping process at the leftmost corner.

The number of possible binary patterns using 8 neighbors is 256. Each binary pattern is classified as either uniform or non-uniform. A uniform pattern is a bit string that has two or less bit changes (including the wrap-around from the last bit to the first bit). A non-uniform pattern is a bit string that has more than two bit changes (once again, including the wrap-around).

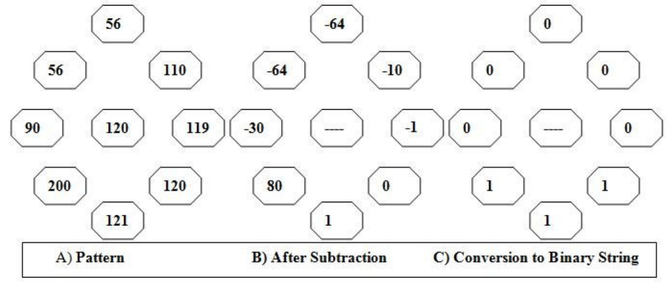

Figure 2: The LBP Method

The number of possible binary patterns using 8 neighbors is 256. Each binary pattern is classified as either uniform or non-uniform. A uniform pattern is a bit string that has two or less bit changes (including the wrap-around from the last bit to the first bit). A non-uniform pattern is a bit string that has more than two bit changes (once again, including the wrap-around). As shown in Figure 3, the uniform pattern has one change between the fourth and fifth bits, and one change between the eighth and first bits. Since the string wraps around, the last and first bits in the string must be compared. The non-uniform pattern has changes between the second and third bits, the third and fourth bits, the fifth and sixth bits, and the seventh and eighth bits. Out of the total 256 possible patterns, 58 of those patterns are uniform. Two of the 58 patterns are 00000000 and 11111111.

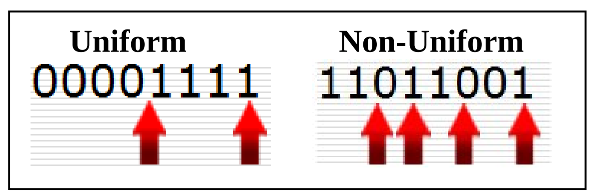

Figure 3: Uniform VS Non-Uniform

A subhistogram is created for every patch of an image, and is composed of 59 bins. Bins 1 to 58 correspond to the 58 possible uniform patterns using 8 neighbors. The 59th is a bin that holds the count of all non-uniform patterns found in the patch. The work of Ojala and Matti Pietikainen suggests that the most discriminating features of a facial image contain predominantly uniform patterns. The subhistograms associated with each patch are then concatenated to form a histogram representing the features extracted by the LBP method.

As in Shelton et al.'s previous research, this paper uses a steady-state GA (SSGA) to evolve a population of feature extractors (FE) (Davis, 1991; Fogel, 2000). In previous research, Shelton et al. evolved FEs consisting of non-uniform patches (not to be confused with the uniformity/non-uniformity of LBP patterns presented earlier).

Non Uniform GEFE

A candidate FE, fe_i, is a 6-tuple, $<X_i,Y_i,W_i,H_i,M_i,f_i>$, where $X_i = \{x_{i,0}, x_{i,1},\ldots, x_{i,n-1}\}$ represents the x-coordinates of the center pixel of the *n* possible patches , $Y_i = \{y_{i,0}, y_{i,1}, \ldots , y_{i,n-1}\}$ represents the y-coordinates of the center pixel of the possible patches, $W_i = \{w_{i,0}, w_{i,1}, \ldots , w_{i,n-1}\}$ represents the widths of the *n* possible patches, $H_i = \{h_{i,0}, h_{i,1},\ldots, h_{i,n-1}\}$ represents the heights, $M_i = \{m_{i,0}, m_{i,1},\ldots, m_{i,n-1}\}$ represents the mask for each patch, and fit_i represents the fitness of fe_i. The mask is a vector that determines which patches are used when building the feature vector for an image. The purpose of masking out patches is to reduce the number of features that need to be compared when measuring similarity between images. The FE can create patches with non-uniform sizes, meaning that the widths and heights for each *n* patch can be unique. Given a probe set and a gallery set the fitness is the number of errors made when comparing each probe to the gallery multiplied by 10 plus the fraction of the n patches from which features were extracted (1).

$$fit_i = NumErrors*10+((\sum_{k=0}^{n-1} m_{i,k})/n+1) \qquad (1)$$

Uniform GEFE

Candidate FEs consisting of patches with uniform patch size are similar with the exception that for any FE, fe_k , $W_k = \{w_{k,0}, w_{k,1}, \ldots , w_{k,n-1}\}$ is of the form, $w_{k,0} = w_{k,1},\ldots, w_{k,n-2} = w_{k,n-1}$, meaning that the widths of every patch is the same. Similarly, $H_k = \{h_{k,0}, h_{k,1}, \ldots , h_{k,n-1}\}$ is of the form, $h_{k,0} = h_{k,1},\ldots, h_{k,n-2} = h_{k,n-1}$, meaning that the height of every patch is the same.

Steady State Genetic Algorithm

The SSGA used to evolve candidate FEs works as follows. First a population of candidate FEs is randomly generated. Each candidate FE is then evaluated and assigned a fitness. After the initial population has been created, two parents are selected via binary tournament selection (Fogel, 2000, Abraham, Nedjah and Mourelle, 2006) and are used to create one offspring via uniform crossover and Gaussian mutation (Davis, 1991; Fogel, 2000; Kennedy and Eberhart 2001; Abraham, Nedjah and Mourelle, 2006). The offspring is then evaluated, assigned a fitness, and replaces the worst fit candidate FE in the population. The evolutionary process of selecting parents, creating a offspring, and replacing the worst fit FE in the population is repeated a user specified number of times. Figure 4 provides a pseudocode version of an SSGA.

```
compute SSGA{
t = 0;
initialize pop(t)
evaluate pop(t)
While(Not done){
   Parent1 = Select_From_Pop(t)
   Parent2 = Select_From_Pop(t)
   Child = Procreate(Parent1, Parent2)
   Evaluate(Child)
   Replace(Worst(Pop(t+1), Child)
   t = t+1;
   }
}
```

Figure 4: Pseudo-code for the GEFE SSGA

Experiment

We performed our experiment on a subset of 105 subjects taken from the Facial Recognition Grand Challenge (FRGC) dataset (Phillips et al., 2005). Each subject in the FRGC dataset has three slightly different images associated with it, as seen in Figure 5. Our dataset of 105 subjects consisted of a probe set (one image per subject), and a gallery set (two images per subject).

The probe set contains one of the images of each subject, and the gallery set contains the other two images for each of the subjects. Since our dataset contained 105 subjects, a total of 105 images were in the probe set and 210 images were in the gallery set. The dimensions of our images were 100 by 127 pixels.

For this experiment, we compared the Standard LBP method (SLBP), GEFE with non-uniform sized patches ($GEFE_n$), and GEFE with uniform sized patches ($GEFE_u$).

Probe Image Gallery Image Gallery Image

Figure 5: Subject 27's Snapshots

Results

For our results, an SSGA was used to evolve a population of 20 candidate feature extractors. The SSGA used uniform crossover and Gaussian mutation, (where the Gaussian mu σ = 0.1). The SSGA was run 30 times for $GEFE_u$ and $GEFE_n$. For each run, a total of 1000 function evaluations were allowed.

In Table I, the average performance of the three methods is shown. The SLBPM needed to be run only once since the patch characteristics were deterministic. $GEFE_n$ used an average of 36.90% of patches, with an average accuracy of 99.84% while $GEFE_u$ used an average of 35.82% of patches, with an average accuracy of 100%. Both $GEFE_u$

and $GEFE_n$ outperformed SLBPM in terms of accuracy while using a fewer number of features.

A t-test was used to confirm the observation that $GEFE_u$ had a statistically significant better performance (in terms of accuracy) that $GEFE_n$.

Research has been done that notes certain areas of a face to be discriminating enough to effectively distinguish between different persons (Matas et al., 2002). Figure 6 shows an approximate positioning of patches for the best feature extractors created using the $GEFE_n$ and the $GEFE_u$. For Figure 6b, the patches are meant to be the same size.

To avoid confusion, one of the patches was drawn in green.

It is interesting to see that the majority of patches are around the ocular region. Because the $GEFE_n$ and the $GEFE_u$ choose this region to focus on suggests that this area holds textures that are unique enough to differentiate individuals from one another. This result is consistent with conclusions presented in other research (Woodard et al., 2010; Miller et al., 2010).

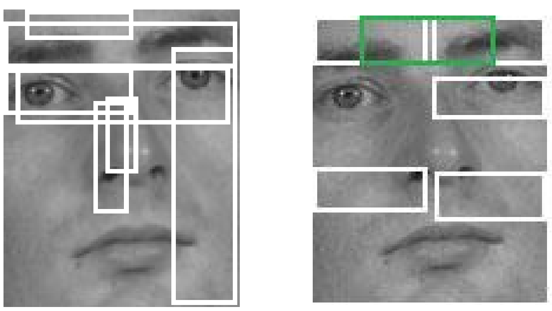

Figure 6a: SSGA Non_Uniform **Figure 6b: SSGA Uniform**

Figure 6: Best Individuals

Conclusion and future Work

In this paper, two forms of GEFE were compared (along with SLBPM). Both $GEFE_u$ and $GEFE_n$ had better performance than SLBPM. $GEFE_u$ had a better performance than $GEFE_n$. Our future work will be devoted toward the investigation and comparison of GEFE using a variety of other forms of Genetic and Evolutionary Computing. A second endeavor will be to use the smaller feature sets evolved by GEFE in an effort to develop hierarchical biometrics systems similar to the one proposed in Gentile's paper (Gentile, 2009).

Acknowledgments

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) and by the National Science Foundation (NSF), Science & Technology Center: Bio/Computational Evolution in Action Consortium (BEACON). The authors would like to thank the ODNI and the NSF for their support of this research.

TABLE I

Experimental results for LBP (even distribution) and SSGA Experiments

Methods	Patches Used	Avg. Accuracy	Best Accuracy
SLBPM	100.0%	99.04%	99.04%
$GEFE_n$	38.65%	99.68%	100.0%
$GEFE_u$	35.82%	100.0%	100.0%

References

[1] Timo Ojala, Matti Pietikainen, *Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns*, IEEE Trans. Pattern Analysis and Machine Intellegence; 971-987; 2002

[2] Arun Ross, *An Introduction To Multibiometrics,* Appeared in Proc. of the 15th European Signal Processing Conference (EUSIPCO), (Poznanm Poland), September 2007

[3] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*. Springer, 2005.

[4] Damon L. Woodard Shrinivas J. Pundlik Jamie R. Lyle Philip E. Miller, *Periocular Region Appearance Cues for Biometric Identification* In *CVPR Workshop on Biometrics, 2010*

[5] James E. Gentile, Nalini Ratha, Jonathan Connell *SLIC: Short-Length Iris Codes*, in *Submission to: Biometrics: Theory, Applications and Systems (BTAS '09), 2010*

[6] 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.

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

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

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

[10] 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.

[11] Jiming Liu, Y. Y Tang, Y. C. Cao, *An Evolutionary Autonomous Agents Approach to Image Feature Extraction IEEE Trans. Evolutionary Comput.* **1** 2 (1997), pp. 141–158.

[12] Ajith Abraham, Nadia Nedjah, Luiza de Macedo Mourelle *Evolutionary Computation: from Genetic Algorithms to Genetic Programming* “Studies in Computational Intelligence” Springer (2006), 1-20.

[13] 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.

[14] Joseph Shelton, Kelvin Bryant, Joshua Adams, Khary Popplewell, Tamirat Abegaz, Kamilah Purrington, Damon L. Woodard, Karl Ricanek, Gerry Dozier, *Genetic Based LBP Feature Extraction and Selection for Facial Recognition* in *ACM Southeast Conference (ACMSE)*, 2011.

[15] J. Matas, P. B´ılek, M. Hamouz, and J. Kittler, *Discriminative Regions for Human Face Detection* in *The 5th Asian Conference on Computer Vision, 23–25 January 2002, Melbourne, Australia*

[16] J. Gentile, N. Ratha, and J. Connell. *An efficient, two-stage iris recognition system* in *International Conference on Biometrics: Theory, Applications, and Systems (BTAS ’09), pages 1–5, 2009*