

Feature Selection for Face Recognition Using a Genetic Algorithm

Derya Ozkan
Bilkent University, Department of Computer Engineering
Turkey, Ankara

Face recognition has been one of the challenging problems of computer vision. In response to this problem, SIFT features [1] have been used in [2]. However, since SIFT features were addressed to object recognition; we need to select the features that best fits in the face recognition problem. So, in this paper, we are using a genetic algorithm to select the most important features for face recognition.

Keywords: Genetic Algorithm, Feature Selection, Face Recognition, SIFT Features.

1. INTRODUCTION

In this paper, we aim to select the most useful features for face recognition. For this purpose, we use a genetic algorithm to learn which features of SIFT features [1], used in object recognition, can describe an interest point of the face.

A face recognition approach using the SIFT features has been proposed in [2]. We believe that finding the subset of those features, which are more useful for face recognition, will lead to better results for the face recognition problem. It will reduce the computation time since we remove the unnecessary features.

We first give information about SIFT features and the problem of face recognition in part a; and the genetic algorithm approach in part b. In section 2, we show how we can use genetic algorithm to select the best features for face recognition. Section 3 gives the experimental results; and in conclusion after a summary, we give future works that can be done to extend the algorithm.

a. SIFT FEATURES AND FACE RECOGNITION

Face recognition is a long standing and well studied problem in computer vision. In their recent work [2] has proposed a recognition strategy using interest points extracted from the detected faces. The features of an interest point used in this strategy are Lowe's SIFT features [1]. The faces are represented using a set of keypoints; and then a matching algorithm is applied to find the similar faces in the test data using a few training faces.

The matching criterion is based on the Euclidean distances between the keypoints on the test face and the keypoints on training faces. If a single keypoint on the test face is considered, its distance to all the keypoints of all the training faces is computed. For each training face, a single keypoint with the minimum distance to the keypoint of the test face is selected. These keypoints are called as *the nearest keypoints*. For the example shown in Figure 1, there are five nearest keypoints corresponding to the five training faces. Next, among the nearest keypoints the ones that give the minimum and the maximum values are chosen. If the distance between those two keypoints differ more than a threshold, called as

minmax threshold it is concluded that the keypoint, whose distance is minimum, is a match for that keypoint of the test image. Those keypoints matching with a keypoint on the training images are selected as the ones satisfying the matching criteria.

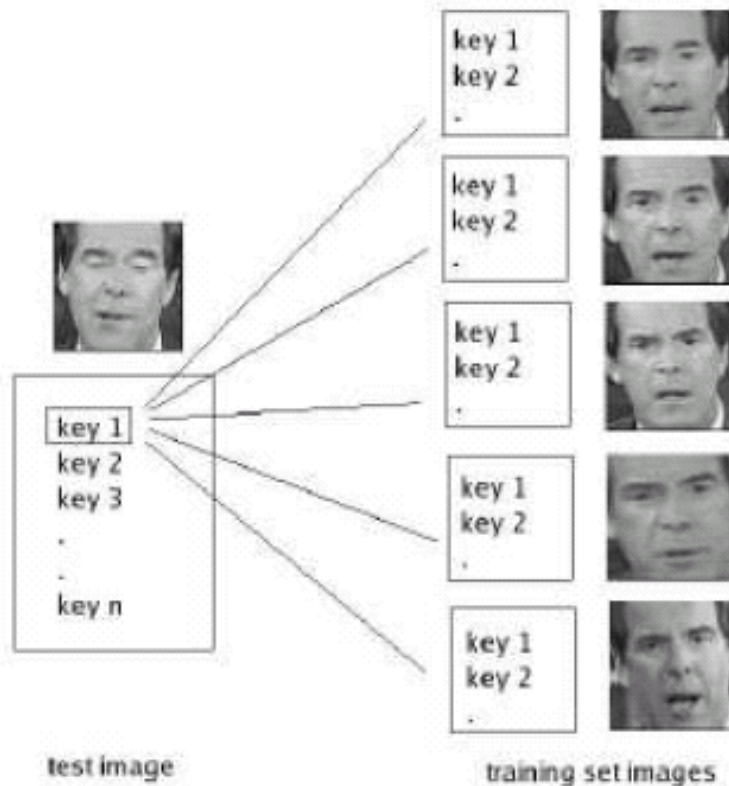


Figure 1 Example matching between a training face and a test face.

For each test image in the data set, the number of keypoints satisfying the matching criteria is found. If this number is higher than a matching threshold it is concluded that the test image and the faces in the training set belongs to the same person.

In figures 2 and 3, match results for one test image and the training images are given. Along with correct matches (last picture in figure 1, an interest point on the eyebrow of the test face matches to an interest point on the eyebrow of one of the training images), there are several matches, which corresponds to different parts of the face (the first picture in figure 1, an interest point on the eye matches to an interest point on the forehead). As seen in figure 3, a test image can be classified wrongly due to such matches (wrong matches), since we apply a threshold on the number of total matches without knowing if it is true or wrong.



Figure 2 Matching results for an example test face (top) with a set of training faces (bottom).

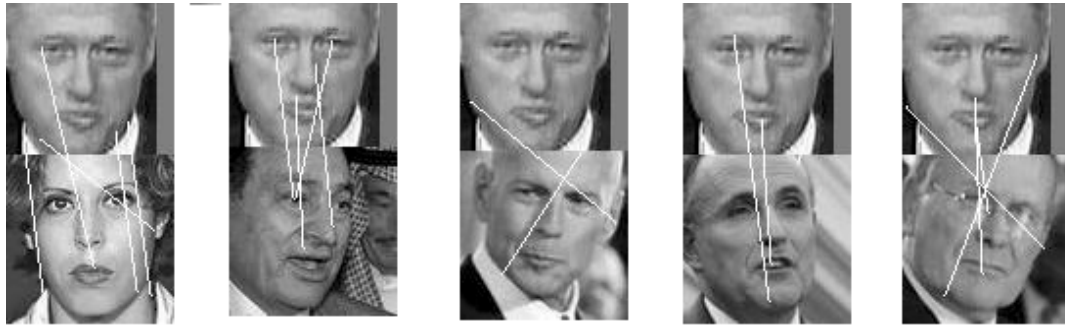


Figure 3 Matching results for an example test face (top) with a set of training faces (bottom).

In Table 1, we see some test results for different number of training images (4, 6, 8, and 10) in the training set of the four anchorpersons shown in Figure 4. In the corresponding data sets, there are 1515, 1343, 454 and 163 correct faces respectively in total. Even though true positive rates can be considered as high (between %66 and %84), we also see that there are considerably high number of false positives.

	4	6	8	10
ID	tp/fp	tp/fp	tp/fp	tp/fp
1	951 / 351	1152 / 540	1217 / 792	1274 / 2149
2	610 / 35	846 / 97	896 / 695	975 / 2195
3	343 / 204	436 / 366	378 / 1882	389 / 2743
4	126 / 2200	118 / 1189	122 / 5062	122 / 4931

Table 1- The number of true positives and number of false positives (tp/fp) are shown for the four anchorpersons in Trec vid [4].



Figure 4- Example training faces for four anchorpersons used in the experiments of Table 1.

These false positives are mostly due to wrong matches between any two interest points of the test and one of the training images. That is because the Euclidian distance between those two points are relatively low considered to interest points of other training images. However, we would expect that distance to be higher, since they correspond to different regions of the faces. Our assumption is that such unexpected results occur, since there are some non-useful features for face recognition. Thus in this paper, we focus on selecting some subset of SIFT features, that are required for describing a point on the face, than other features. With such a selection, we aim to reduce the number false positives, caused by wrong matches, and the running time of the algorithm.

b. THE GENETIC ALGORITHM

Genetic algorithm was developed by John Holland- University of Michigan (1970's)- to provide efficient techniques for optimization and machine learning applications through application of the principles of evolutionary biology to computer science. It uses a directed search algorithms based on the mechanics of biological evolution such as inheritance, mutation, natural selection, and recombination (or crossover). It is a heuristic method that uses the idea of 'survival of the fittest'.

In the genetic algorithm, the problem to be solved is represented by a list of parameters which can be used to drive an evaluation procedure, called chromosomes or genomes. Chromosomes are typically represented as simple strings of data and instructions. In the first step of the algorithm, such chromosomes are generated randomly or heuristically to form an initial pool of possible solutions called *first generation pool*.

In each generation, each organism (or individual) is evaluated, and a value of *goodness* or *fitness* is returned by a fitness function. In the next step, a second generation pool of organisms is generated, by using any or all of the genetic operators: selection, crossover (or recombination), and mutation. A pair of organisms are selected for to survive based on elements of the initial generation which have better fitness. In other words, the organisms that have relatively higher fitness than other organisms in the generation are selected to survive. Some of the well-defined organism selection methods are roulette wheel selection and tournament selection.

After selection, the *crossover* (or *recombination*) operation is performed on the selected chromosomes, with some probability of crossover (P_c)- typically between 0.6 and 1.0. Crossover results in two new child chromosomes, which are added to the second generation pool. The crossover operation is done by simply swapping a portion of the underlying data structure of the chromosomes of the parents. This process is repeated with different parent

organisms until there are an appropriate number of candidate solutions in the second generation pool.

In the mutation step, the new child organism's chromosome is randomly mutated by randomly altering bits in the chromosome data structure, with some *probability of mutation* (P_m) that is about 0.01 or less.

The aim of these is to produce a second generation pool of chromosomes that is both different from the initial generation and hence have better fitness, since only the best organisms from the first generation are selected for surviving. The same process is applied for the second, third,... generations until an organism is produced which gives a solution that is "good enough".

The overall processes of the algorithm is summarized in Figure 5. Also the flow chart of the genetic algorithm is given in Figure 6. The components of the genetic algorithm explained above can also be summarized as below:

- Encoding technique.....(*gene, chromosome*)
- Initialization procedure....(*creation*)
- Evaluation function.....(*environment*)
- Selection of parents.....(*reproduction*)
- Genetic operators.....(*mutation, recombination*)
- Parameter settings.....(*practice and art*)

```
1. {
2.  initialize population;
3.  evaluate population;
4.  while TerminationCriteriaNotSatisfied
5.    {
6.      select parents for reproduction;
7.      perform recombination and mutation;
8.      evaluate population;
9.    }
10. }
```

Figure 5- Pseudo-code for genetic algorithm.

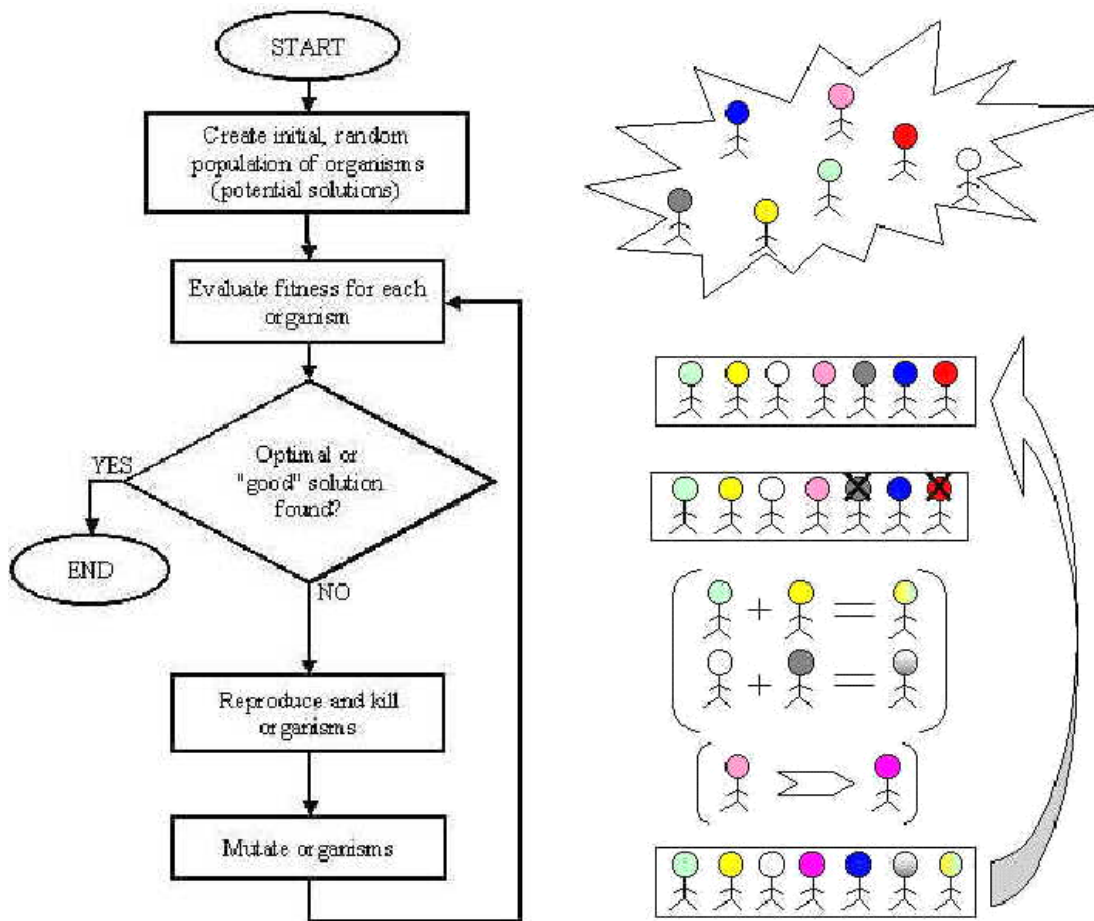


Figure 6- The Flow chart of the genetic algorithm

2. FEATURE REDUCTION WITH GENETIC ALGORITHM

In our genetic algorithm, each chromosome is a string of binary numbers, either 0 or 1 of size 128, since we have 128 SIFT features describing an interest point. Let c_{ij} denote the j 'th component of the component i . $c_{ij} = 0$ indicates that we should not use the j 'th feature, whereas $c_{ij} = 1$ indicates that we should use it.

Initially, we 10 interest points and desired distances between each pair of them. Let d_{pr} denote the desired distance between any two interest points d and r . These desired distances are assumed to be 0 for any two very similar interest points (like one coming from the eye of a person and another one coming from the eye of another image of the same person), and 1 for any two non-similar interest points. (The distances are Euclidian distances that are normalized to 1). Then the difference between the desired distance of these two points and the distance calculated by only using the selected features of a chromosome should reach to zero in the ideal case. So our fitness function becomes minus (desired distance of two points - the distance calculated by only using the selected features of a chromosome).

To we give the mathematical analysis of the algorithm, let c_i denote the i 'th chromosome where $c_i = \langle 1 \ 0 \ 0 \ 0 \ 1 \ 0 \dots 0 \ 1 \rangle$ (a vector of size 128 and each element is either 0 or 1). If we denote E_{i_pr} as the Euclidian distance between the two interest points, i and j ; then E_{i_pr} is given as:

$$E_{i_pr} = \left| \frac{\sum_{j=1}^{128} ((p_j - r_j)^2 * c_{ij})}{(\sum_{j=1}^{128} c_{ij}) * 255} \right|$$

The denominator in the equation is used to normalize the distance. It is multiplied by 255, since each feature value is in the range 0 and 255.

From the above notations, our fitness function for the i 'th chromosome becomes:

$$f_i = \text{Avg}_{P_k, r_k \text{ pairs}} (-1 * |(d_{pr}) - (E_{i_pr})|)$$

The multiplication of the difference of distances with -1 is provided, since we are looking for higher values for best fits. Hence, the difference itself will reach to 0, for the ideal case and it will reach to 1 for the worst case.

3. EXPERIMENTS

Currently, the no experiments have been conducted to show that the selected features are the best for face recognition. In the next version of the paper, we aim to put here the experimental results.

4. CONCLUSION AND FUTURE WORK

In the paper, after giving the definitions for face recognition and genetic algorithm approach; a genetic algorithm has been suggested to select the most useful features of the face. Recently, SIFT features and their distances have been used in [2] for the face recognition problem. However, the tests showed that some test images (faces) are wrongly classified due to some wrong matches between the interest points of the test images and training images. The underlying reason for such wrong matches is that two interest points have relatively small distances, even though we would expect them to have higher distances. Our assumption is that if we use some subset of the SIFT features, which are more useful for describing an interest point of the face, we can achieve better results. Using such subset of the

features will also reduce the run time. So, we propose a genetic algorithm for feature selection scheme.

For our future work, we are planning to apply the genetic algorithm on a number of interest points of some faces and determine the best features for face. Then using only these selected features, same tests as in [2] will be done for performance and accuracy analysis.

5. REFERENCES

- [1] David G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110
- [2] D. Ozkan, G. Akcay, P. Duygulu, Interesting Faces in the News, submitted to *International Conference on Computer Vision (ICCV) 2005*
- [3] Trec video retrieval evaluation <http://www-nlpir.nist.gov/projects/trecvid/>.
- [4] Jeniffer Pittman, Genetic Algorithm for Variable Selection, ISDS, Duke University, <http://www.niss.org/affiliates/proteomics200303/presentations20030306/6>
- [5] Wikipedia, the free encyclopedia, *Genetic algorithm*, http://en.wikipedia.org/wiki/Genetic_algorithm
- [6] Wendy Williams, Genetic Algorithms: A Tutorial, web.umn.edu/~ercal/387/slides/GATutorial.ppt