Genetic algorithms-based dominant feature selection for face detection application

Ameni Yangui Jammoussi*,
Sameh Fakhfakh Ghribi and
Dorra Sellami Masmoudi

Department of Electrical Engineering,
Sfax University,
Sfax Engineering School, Tunisia
Email: yenguiameni@gmail.com
Email: fakhfakhsameh@yahoo.fr
Email: dorrasellamimasmoudi@yahoo.com
*Corresponding author

Abstract: A key challenge in computer vision applications is detecting objects in an image which is a non-trivial problem. One of the better performing proposed algorithms falls within the Viola and Jones framework. They make use of Adaboost for training a cascade of classifiers. The challenges of Adaboost-based face detector include the selection of the most relevant features which are considered as weak classifiers. However, selection of features based on lowering classification error leads to high computation complexity. To overcome this limitation, a novel genetic Adaboost is proposed in our work. In the same context of optimisation, a selection method based on Pareto concept of the most relevant features referred to as dominant features is proposed. This optimisation allows to reduce the initial feature space by 28%. Moreover, we notice that dominant features with genetic Adaboost further improve the performance of genetic Adaboost, reducing the total number of features by 20%.

Keywords: Adaboost; genetic Adaboost; dominant features; face detection; computational vision.

Reference to this paper should be made as follows: Jammoussi, A.Y., Ghribi, S.F. and Masmoudi, D.S. (2016) 'Genetic algorithms-based dominant feature selection for face detection application', *Int. J. Computational Vision and Robotics*, Vol. 6, Nos. 1/2, pp.77–93.

Biographical notes: Ameni Yengui Jammoussi received her Ing degree from Sfax National Engineering School in 2007. Subsequently she received her Master degree in 2009 and her PhD in 2013 in Computer Systems Engineering at Sfax National Engineering School. She is currently an Assistant Professor at University of Gabes.

Sameh Fakhfakh Ghribi received her Ing degree from Sfax National Engineering School in 2003. Subsequently she received her Master degree in 2004 and her PhD in 2011 in Computer Systems Engineering at Sfax National Engineering School. She is currently an Assistant Professor at University of Sfax. Her researches include wavelet transforms, evolutionary and genetic algorithms, and the design of devices for biomedical applications.

Dorra Sellami Masmoudi received her Ing degree from Sfax National Engineering School in 1994 and has been awarded by the President of the Republic of Tunisia. Subsequently she joined IMS. Lab at Bordeaux and received her PhD in 1998 in Electronics System Design and she has been Assistant Professor at Sfax National Engineering School from 1999. She obtained her HdR degree in 2006. She is currently a Full Professor. Her researches include neural and fuzzy based systems, evolutionary and genetic algorithms, pattern recognition, image and video processing, feature extraction, classification data fusion, it covers a large spectrum of applications: neural, fuzzy and neuro-fuzzy-based image processing, biometry-based personal identification and medical image processing for breast cancer automatic diagnosis.

This paper is a revised and expanded version of a paper entitled 'Adaboost based object detector optimization with genetic algorithm' presented at ICSPT, Sousse-Tunisie, March 2013.

1 Introduction

The human being is more and more interested in reproducing intelligence which is one of the most impressive features of the nature. Researchers are trying to build intelligent machines that have various capabilities. Building machines or robot is probably one of the most challenging problems which humans are trying to solve. Recently, many projects have started with the purpose of learning machine to detect objects. Many classes of objects can be efficiently detected by the way of machine learning techniques such as faces, cars, pedestrians, etc. Faces, pedestrians and cars, have been detected with minimal error rates by learning their appearance using extensive training sets. The most popular sub-problem within the object detection domain which many researchers focus on is face detection.

The interest of research and development in the field of face detection concern many real world applications. It holds the key to many high level applications such as face recognition, human-machine interactions, security and tracking faces among others.

At a first glance, the face detection may not seem very hard namely when considering how the task is easily solved by human. But it is far beyond in reality, it is very difficult to make this task solved by a computer. For this purpose, a number of promising face detection techniques has been developed.

The various techniques proposed in the literature are classified by Yang and Kriegman (2002) into four main categories: knowledge-based, feature invariant, template matching and appearance-based methods.

1.1 Knowledge-based methods

These methods make use of the human consent based on the content of face images using rule-based methods. They are characterised by their easiness to come up with simple coded rules. Such rules present features of a face and works well under uncluttered background. These methods are based on a set of simple rules, which typically extract relationships between different facial features.

1.2 Feature invariant methods

As their name suggests, they are methods regrouping all feature-based ones. In contrast to knowledge-based approaches, researchers have been trying to find invariant features. Based on the extracted features, they build statistical models describing their spatial relationships related to the existence of a face. Then, the extracted features should be relevant, invariant to pose, lighting conditions, orientation change, etc. Specific facial features either are concerned with geometric features such as eyes, nose, etc. Or they include appearance related characteristics such as colour, texture, edge, shape, intensity, etc. However, one problem with these feature-based algorithms is that invariance can be ensured in open environment context.

1.3 Template matching methods

In these methods, the detection is achieved by computing the correlation between standard stored templates of a face and an input image. They are characterised by their simplicity and rapidity. However, they had shown a limitation in terms of robustness because it is difficult to enumerate templates for different possible cases.

1.4 Appearance-based methods

In contrast to template matching where templates are defined by experts, these methods use models learned from training sets to represent the variability of facial appearance. Appearance-based methods, which consist in extracting features from pixel intensity of face and non-face images, are based on techniques from statistical analysis to machine-learning. Accordingly, a huge dataset covering all kind of variability in the image, regarding background complexity, illumination conditions, pose and scale, will convey robustness to these kinds of approaches. Moreover, learning makes these techniques potentially applying on different kinds of objects. It is just matter of updating the different feature contributions in the considered objects. The majority of these methods are based on a learning algorithm using a large set of face images in the training process.

One of the better performing algorithms which falls within appearance-based methods is proposed by Viola and Jones (2001, 2004). It uses adaptive boosting (Adaboost), a popular machine learning technique for selecting a set of better performing weak classifiers from a pool of over complete weak classifiers (Freund and Shapire, 1995). Several machine learning approaches have been proposed to this aim, showing significant improvements in detection in terms of accuracy and speed.

The compromise between fast training and effective features remains a challenging problem in Adaboost-based face detection. Variability in face detection problem related on the one hand to the face representation and on the second hand to background and acquisition condition has guided us to choose the appearance-based approaches based on machine learning.

In these approaches, the main key questions are: what is the most relevant feature to extract for efficient classification? And what is the convenient feature to keep from an over complete pool of the extracted features in the subsequent stages of the algorithm?

Approaches followed in the literature to tackle these problems are based on proposing more relevant features, or focusing on improving the boosting algorithm, or a combination of both approaches.

1.5 Limitation in conventional Adaboost training process

In our work, some limitations which are related to the learning process are faced. The weak classifier functionality is to find a weak rule H_t: $X \rightarrow Y$ which is appropriate for a distribution D_t. But what we mean by appropriate? Here, the quality of a weak classifier depends on its error, according to the weights:

$$\epsilon_t = \Sigma_{i \backslash h_t(x_i) \neq y_i} D_t(i) \quad (1)$$

As noted in the mentioned equation, the error is measured with respect to the distribution D_t, on which the weak learner is trained. In practice, the weak learner is an algorithm weighting (by D_t) the training samples. For each iteration in Adaboost algorithm, a new weak classifier is selected according to the error criterion.

In some cases, no improvement is made with respect to the detection rate or false positive one.

Some weak classifiers greatly enhance the performances but others do not contribute and even end up with a performance drop. This is can be illustrated by an example of training a single stage of the cascade, adding features through the Adaboost training process, the evolution of the detection rate and false positive rate is given in Table 1. If we examine the results below, we notice that the first five selected features improve the results in term of the DR and increase the FPR. However, the increase of the FPR can be treated in the subsequent stages in the cascade structure. The 6th selected feature do not contribute to the training process. Also the 12th feature slightly degrade the performance on the validation set in term of FPR.

Table 1 Evolution of the DR and FPR rates (in percentage) on the training set and validation set in the Adaboost training process

Features	*Training set*		*Validation set*	
	DR	*FPR*	*DR*	*FPR*
1	87.7	21.6	72.58	20.2
2	96.56	39.52	88.98	24.2
3	98.1	48.96	92.37	34.6
4	99.4	66.2	96.82	62.4
5	99.8	80.52	98.51	79.4
6	99.8	80.52	98.51	79.4
7	99.92	81.36	98.51	80.2
8	99.96	82.72	98.72	82.6
9	99.96	80	98.1	77.4
10	100	78	98.3	79.6
11	99.96	70.44	98.1	73.4
12	100	71.64	98.1	74.8

This can be attributed to the fact that, often some of the selected features although leading to lower errors, are irrelevant, which increases the training time and memory resources.

A key question here is how to add relevant features without decreasing the training performances?

In the following sections, we try to overcome these limitations by optimisation with genetic algorithms (GAs).

This limitation motivated us to find a search technique of weak classifiers that outperforms the solution based on lowering the classification error. A face detection task is considered as a classifier training problem, searching the parameters for a best modelling of a given training data. In the standard model, we need to specify many parameters and then estimate their values from training data. When these standard models are simple, it is possible to find their optimal parameters by solving equations explicitly. However, when the task becomes more complex, it is very difficult to find the optimal parameters.

In this work, we focus on the optimisation of the feature selection process of the most relevant ones by GAs. In fact, selected features by Adaboost, within a single stage are dependent on each other's and there is no analytic relation between the number of features and the corresponding detection performance. Thus, the optimisation task is nonlinear, hard and then seems to be suitable to be treated by GAs.

Use of GAs takes profit of two advantages. Firstly, GA perform as a searching mechanism to select the most effective features overall the feature space. Secondly, a new method inspired from the Pareto frontier is proposed to construct a frontier composed of the most relevant features referred to as dominant features. The use of GAs comes to improve the selection process by reducing the feature set, keeping on the most relevant and discarding the redundant weak classifiers.

The remaining of this paper is organised as follows: Section 2 describes our methodology for Adaboost optimisation. Section 3 presents experimental results and performances for our face detection system. Finally, Section 4 summarises our work and draws some conclusions.

2 Methodology for Adaboost optimisation

2.1 Basic Adaboost

A boosting algorithm is able to construct a strong classifier by a linear combination of weak classifiers chosen from a huge amount of set. The single strong classifier obtained is much more reliable than the weaker ones. The set of weak classifiers that contributes to the final response can be simple in some cases. However, a scheme of training them has been devised in order to have a small error in the final classification. In face detection, each classifier is constructed by a single feature. The time spent in the training process is very important, which can be attributed to the exhaustive search of the features in the whole set. Taking into account this conventional Adaboost, we can wonder whether the selection of weak classifiers based on lowest errors leads to the optimum solution.

Each hypothesis in the training algorithm is constructed using a single feature. The algorithm is described in the following:

- Given a series of samples $(x1, y1)$, ..., (x_n, y_n) where $y_i = 0, 1$ for a binary outcome.
- Initialise weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negative and positive samples.
- For $t = 1...T$:
  1. Normalise the weights

     $$w_{t,i} = \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

     so that $w_{t,i}$ is a probability distribution and i is an image index.
  2. For each feature j, train a classifier h_j (each classifier corresponds to a single feature). The error is evaluated with respect to $w_{t,i}$,

     $$\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|.$$
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:

     $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

     where $e_i = 0$ if sample x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t; \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha_t = log(1/\beta_t)$.

2.2 Genetic Adaboost

2.2.1 Basic principle

In this paper, we review the previous researches which investigate the GAs to ameliorate the face detection algorithm (Zhang and Zhang, 2010; Yang et al., 2010). Then, we will present our contribution.

In Chen et al. (2004), the GAs was exploited to construct a large database by re-sampling existing faces. In fact, they have expanded the face samples by applying crossover and mutation. In Treptow and Zell (2004), an evolutionary search was employed within Adaboost framework in single stage classifiers to select accurate features from large features pool in reasonable time. As an amelioration of the Treptow and Zell work, Zalhan et al. (2007a, 2007b) had used the GAs inside the Adaboost framework to select features, which provides better results in the obtained cascade of classifiers in less training time. In their proposed technique referred as GABoost, the GAs carried out an evolutionary search in the feature space which was enriched with more types of features. In our work, we make use of the GAs to further improve the feature selection process in Adaboost. Two constrained objectives can be considered. They are the detection rate which has to be maximised and the false positive rate which has to be minimised. Thus, a multi-objective GA for the optimisation process should be adopted.

2.2.2 GAs formulation

In mono-objective GAs, individuals can be compared easily. However, in multi-objective GAs, this task becomes more complex. There are several methods that are proposed to solve the multi-objective problem.

In the literature, the multi-objective problem has been treated, based on two different approaches. The first one is to bring back a multi-objective formulation to a mono-objective one, referred to as Aggregate method. The second approach, referred to as non-aggregate method, is to try to afford answers taking into consideration the whole of the criteria (Ghribi, 2011; Coello, 1996).

The entire GAs procedure is summarised as follows (cf., Figure 3).

1 *Representation:* Each individual in the population corresponds to an association of weak classifiers. The number of weak classifiers by individual (denoted by T) is variable and can be changed in the optimisation process.

2 *Construction of the initial population:* For more efficient detection, decision is preferably made from earlier stages with minimum computing complexity. Thus, less number of classifiers is recommended, so that only hard samples are kept to subsequent stages. When we go forward in the optimisation process, better performances are required from a stage to another. In fact, classification task is getting harder when going forward in the cascade, suggesting more classifiers to be used for better efficiency. In our approach, a maximal dimension of strong classifiers (denoted by T_{max}) is initialised in GAs and depends on the rank of the current stage

   In our proposed method, initial individuals are of variable length. We denote by T_i the number of genes in individual I_i, with $T_i < T_{max}$. Each gene of I_i is denoted by $I_{i,j}$, $j \in \langle 1, 2, ..., T_i \rangle$. T_i and $I_{i,j}$ are generated randomly. Besides, it is well known that the better the initial population the more efficient are optimisation results. For this purpose, we propose a new method for ameliorating the initial population by selecting the more relevant features referred to as dominant features. This method will be explained in details in a separate subsection.

3 *Fitness computing:* This function makes it possible to evaluate the effectiveness of the chromosomes solutions. It depends on criteria which should be maximised or minimised. The objectives of the used GAs are defined based on the method ϵ-constraint (Coello, 1996). This method is based on a minimisation of an objective f_i by considering that the other objectives f_j with $j \neq i$ must be lower than a value ε_j. In general, the selected objective is the one that the decision maker wishes to optimise in priority:

   $$\min f_i(x) \text{ with } f_j(x) \le \varepsilon_j, \forall j \neq i \tag{3}$$

   In our system, the objective is the maximisation of the detection rate and the false positive rate is considered as a constraint.

4 *Reproduction:* For this step the elitist method is used, which is intended to prevent the lost of the best individuals. Thus, technically, best individuals are reinserted in the future population and the remainder of the future population is constructed based on the wheel selection method.

5 *Crossover:* The crossover is an exchange per blocks of elements between two chains to generate one or two others of them. A site of crossover is randomly selected over the length of each parent chromosome and a cut of the chromosome is done. This cut produces two pieces which can be permuted. The resulting children chains contain each a piece inherited from each parent.

6 *Mutation:* In binary population, some bits of population are chosen to sudden mutation, according to mutation's probability. Their values are then reversed.

7 *Population sorting:* In this step, we perform the union of populations before and after genetic operations (crossover and mutation), then we sort them according to the detection rate, the best half of the resulted population are chosen to participate in the future generation by the elitism mechanism.

The algorithm is described in the following:

- Given sample images (x_1, y_1), ..., (x_n, y_n) where $y_i = 0, 1$ for negative and positive examples respectively. An individual I_i from population at certain generation where $I_{i,j}$ is the element (gene) of the individual. j is the feature index.
- Initialise weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negative and positive samples.
- Initialise T is variable length with $T \leq T_{dim}$. The size of T is chosen randomly. A T_{dim} vector includes a string of T integer values and the remaining values are set to zero.
- For $t = 1...T$:

  1 Normalise the

  $$w_{t,i} = \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$

  so that w_t is a probability distribution.

  For each feature j, train a classifier h_j is restricted to using a single feature.

  The error is evaluated with respect to w_t,

  $$\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|.$$

  Consider ϵ_t *the corresponding error to the gene* $I_{i,t}$.

  2 Update the weights:

  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

  where $e_i = 0$ if example x_i is classified correctly,

  $e_i = 1$ otherwise,

and

$$\beta_t = \begin{cases} \dfrac{\epsilon_t}{1-\epsilon_t}, \text{ if } \epsilon_t < 0.5; \\ \dfrac{1-\epsilon_t}{\epsilon_t}, \text{ otherwise} \end{cases}$$

3 The final strong classifier is:

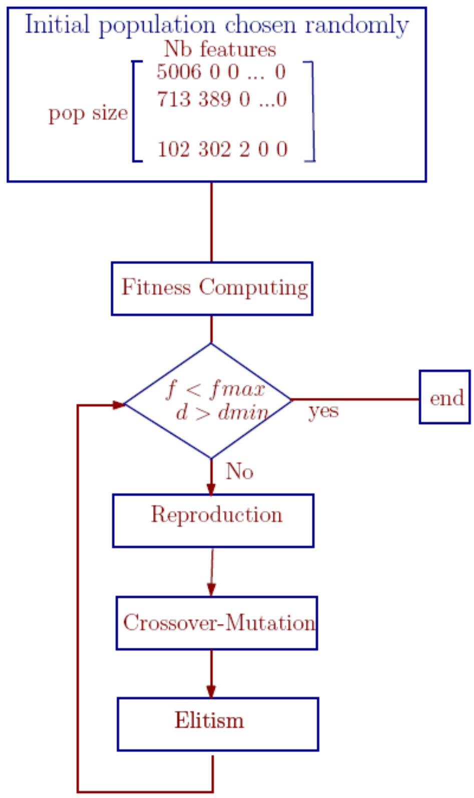
$$h(x) = \begin{cases} 1, \sum_{t=1}^{T} \alpha_t h_t(x) \geq \dfrac{1}{2} \sum_{t=1}^{T} \alpha_t; \\ 0, \text{ otherwise} \end{cases}$$

where $\alpha_t = log(1/\beta_t)$.

4 Evaluate a function with respect to the final strong classifier.

fitness = DR

Figure 1 Different steps of a generic GA (see online version for colours)

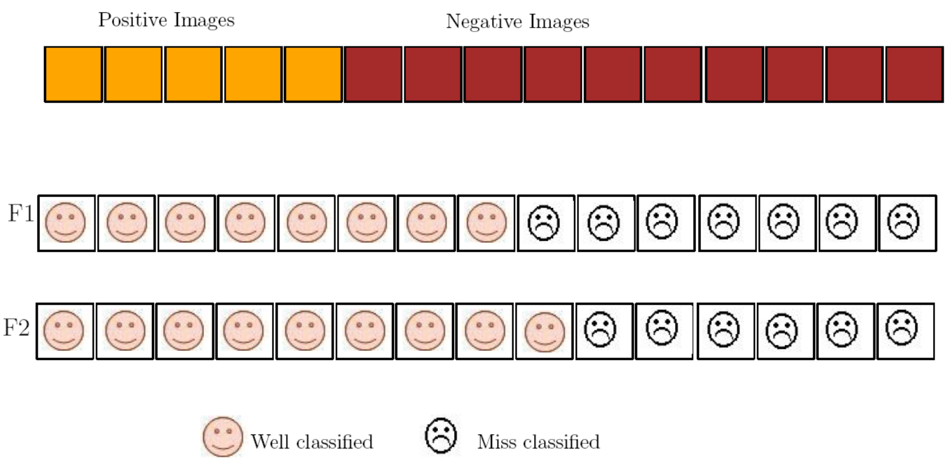


All GAs steps fall within the conventional ones, except the construction of the initial population step, which is based on dominance concept. Thus, we add some details on the way we select dominant features in the following section.

2.2.3 Dominant feature selection

The concept of dominant features was used in some researches for feature extraction process. For instance in Bamini and Kavitha (2010), the dominant features which are extracted from the LBP image, called dominant local binary pattern (DLBP), consider the most frequently occurred patterns in a texture image. In their paper, authors demonstrate that a minimum set of 80% patterns can efficiently illustrate the image textural information.

Figure 2 The principle of dominance: feature F2 dominates feature F1 (see online version for colours)



In order to ameliorate the performance of the Adaboost in the feature selection process and reduce the training time, the idea is to search the dominant features which are more relevant to classify positive and negative images.

Dominant features

The idea behind the dominant features is based on the best classification rate of an input image I by a single feature f_i. A positive/negative image contributes to the classification rate. Starting from the idea that we should not loss any feature that well classifies new image especially negative ones which are harder to be classified. For example, given feature 1, that well classifies two negative images, feature 2 classifies correctly the same two images and another negative image. In this case, feature 2 dominates feature 1. The idea can be illustrated by a simple example (cf., Figure 2). In this example, we take five positive images and ten negative ones.

Pareto concept

The concept of dominance is inspired from Pareto (1896) methods, that falls within multi-objective GAs. Pareto methods are based on the concept introduced by Pareto which privileges one research satisfying all the objectives as well as possible. The definition is as follows.

Definition 1: Pareto's concept of dominance in case of minimisation problem

One point $x \in E$ *dominates* $x' \in E$ if $\forall k,\ f_k(x) \le f_k(x')$ with at least one k such as

$$f_k(x) < f_k(x')$$

In our case, f_k is the right classification rate of image I in the dataset. The Pareto's concept is illustrated by Figure 3. The algorithm of determining dominant features is described as follows.

- Compute the classification result C_i of each single feature.
- Compare C_i to labels (C_i == *labels*)
- Compute the Pareto frontier
- initialise *frontier*1 to all features

  for each feature i in the total set

    for each feature j in *frontier*1

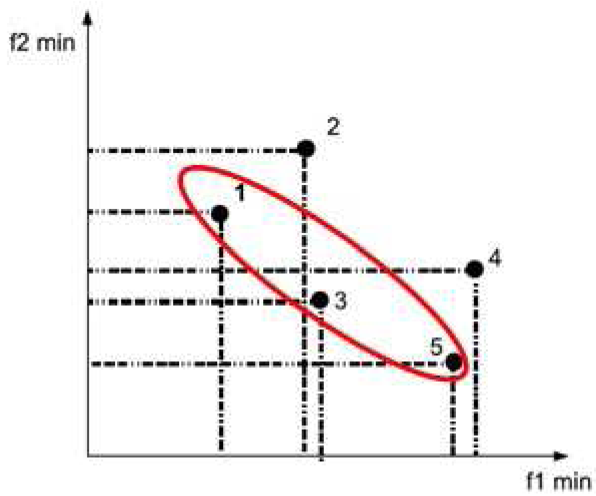
    search non-dominant features

if C_i *or* C_j == C_i feature j is dominated by feature i eliminate feature j from *frontier*1

end

end

Figure 3 Pareto frontier example (see online version for colours)



3 Experimental results

3.1 Training methodology

We had to choose the cascade parameters and the training samples which determine the number of stages and so the number of features in each stage. The training dataset consists of 130 images with 507 labelled frontal faces. The faces were cropped to images of size 19 ∗ 19 pixels. We had to choose the cascade parameters and the training samples which determine the number of stages and so the number of features in each stage. First, we devote our experiments to a single stage exploration.

3.1.1 Training a single stage

The system was trained using 500 faces and 1,000 non-faces. For the test set, we have used 500 faces and 500 non-faces. Using the Haar-like features, we start by training a single stage to compare the performance of the genetic Adaboost to the conventional Adaboost. According to Table 2, we notice that the obtained results are better especially in term of the DR. Results are obtained with only ten generations of the GAs. The proposed optimisation process seems to be promising and we should validate it by training a cascade structure.

Table 2 Comparison of the DR and FPR rates for genetic Adaboost and conventional Adaboost

Number of features	*Adaboost*		*Genetic Adaboost*	
	DR	*FPR*	*DR*	*FPR*
20	83%	51.2%	88.13%	45.8%
30	85.4%	14.4%	88.13%	22.6%
50	85.4%	9.2%	88.13%	14.8%

3.1.2 Training a cascade structure

The system was trained using 500 faces and 1,000 non-faces. For the validation set, we have used 100 faces and 300 non-faces. For each stage classifier, the minimal detection rate is 0.98 and the maximal false positive rate is 0.5 on the validation data.

For the learning process, we have to start with a big number of negative samples. Then, at each stage, only the samples that are classified as positive are kept on the subsequent training set. Thus, the next stage in the process is trained to classify the samples that have been misclassified by the previous stages. Furthermore, a few number of hard samples (like faces) are left to the latest stages of the cascade. Consequently, the number of negative images to train the model and so the number of features per stage decreases and the obtained cascade seems to be not consistent. In order to overcome this problem, we train each layer of the cascade with the same number of negative samples. That is, we add new negative samples for each stage to maintain the initial number of negative samples.

In the training process, we conduct two experiments. The first one was achieved to evaluate the whole set of features using genetic optimisation. The second one was carried out using both the dominant features in the initial population construction and the genetic optimisation. For the negative samples, the number is higher than the number of faces in order to represent faces as well as the background of real images. However, it is a hard task to know a priori the negative images that are more representative of the non-faces class and the number of non-faces that are sufficient to train an efficient cascade.

3.2 GA optimisation

It should be underlined that GAs suitability for improving the Adaboost-based face detector performances comes from its stochastic generation of populations. Indeed, GAs performs well on constrained multi-objective problems and are able to explore large nonlinear design spaces. In fact, we use 100 as population size, which was initially an arbitrary number but was confirmed by good results. Since the crossover step should be frequently done, the crossover rate must be high and is generally set to 0.8. Usually in GAs, we use a low value of mutation rate. As we adopt an elitist method in our work which decreases the population diversity, we use high mutation rate to ensure diversity (Table 3).

Figure 4 Comparison of the number of selected features between Adaboost and genetic Adaboost (see online version for colours)

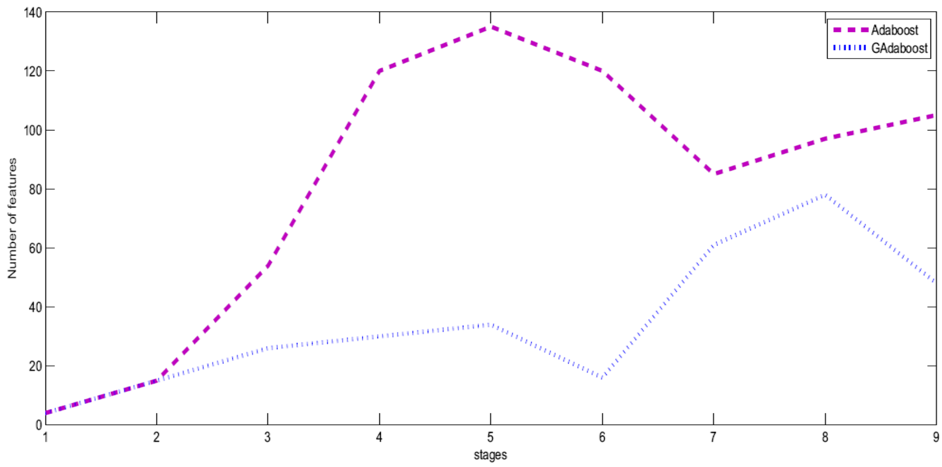


Table 3 Simulation parameters

Parameter	*Value*
Population size	100
Crossover rate, P_c	0.8
Mutation rate, P_m	0.1
Minimal detection rate, $d_{\min}$	0.98
Minimal false positive rate, $f_{\max}$	0.5

3.2.1 GAs-based selection on the whole feature set

Using four types of Haar-like features, we obtain 50,040 features for an image of size 19 × 19 pixels. With these features, we train a cascaded classifier containing ten stages. The number of features of each stage is given in Table 4. The total number for the different stages is 464. Using the GAs optimisation, the number of weak classifiers was reduced considerably (cf., Figure 4). In fact, the total number of features was reduced to 57% of that constructed based on the Adaboost method (Jammoussi et al., 2013). The proposed genetic Adaboost ensures the optimisation of the system performances given a number of features. It was achieved by selecting the most relevant features and eliminating redundancy.

3.2.2 GA-based selection using dominant features in initial population

The application of our method for selecting the dominant features has reduced the number of features by 28%, we obtain 36,150 features. To further demonstrate the powerfulness of the optimisation method, we train a cascaded classifier. The training process is achieved with ten stages and 373 features. The total number of weak classifiers for different stages using dominant features was reduced by 20%. According to the results mentioned above, the number of stages for each cascade is 10, which is relatively small. We obtain high false positives rate due to reduced number of stages in the cascade. Thus, we intend to increase the number of stages in the cascade either by starting by a big number of negative images in the training process or by using somewhat a small database and then adding new negative examples at each stage.

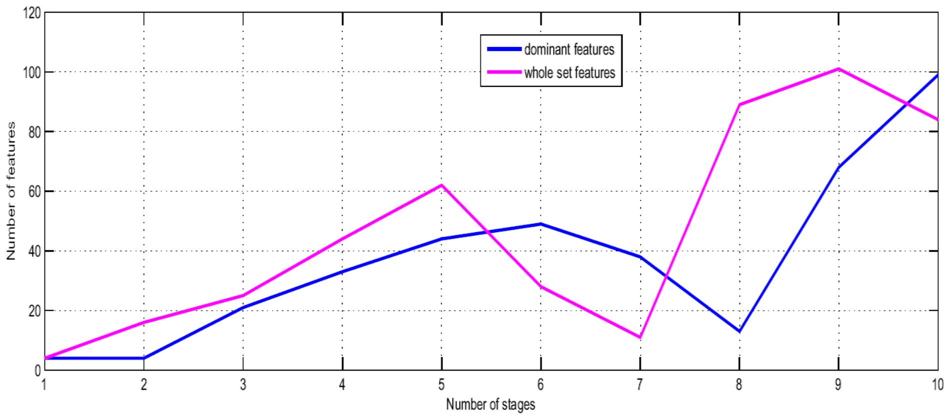
Table 4 Number of features for each stage after training process using initial population with the whole set of features and initial population with dominant features

Stage	*Whole set of features*	*Dominant features*
1	4	4
2	16	4
3	25	21
4	44	33
5	62	44
6	28	49
7	11	38
8	89	13
9	101	68
10	84	99
Total	464	373

3.3 Comparative study and discussion

Compared to standard Adaboost-based face detector, our proposed method based on GAs reduces considerably the total number of features. The number of weak classifiers is directly related to computation time, especially the first few stages are critical in terms of detection speed since most test windows are rejected by the first weak classifiers in a cascade structure. Consequently, this reduction speeds up the final face detector and makes it easy and less expensive to implement on hardware architecture. Besides, according to Table 4 and Figure 5, dominant features further improve the results. In fact, dominant features are more relevant and can contribute well on the system performances.

Figure 5 Comparison of the number of selected features with genetic Adaboost using dominant features and the whole feature set (see online version for colours)



3.4 Training improvement

For the training process, we have to start with a big number of negative samples. Then, at each stage, only samples that are classified as positive are kept on the subsequent training set. Thus, the next stage in the process is trained to classify samples that have been misclassified by the previous stages. Furthermore, a few number of hard samples (like faces) are left to the latest stages of the cascade. Consequently, the number of negative images to train the model and so the number of features per stages decreases and the obtained cascade seems to be not consistent. In order to overcome this problem, we start with a small number of training set and add to it new negative samples at each stage to maintain the initial number of negative samples. One of the main drawbacks of Adaboost face detection is that the result directly tied to the size and consistence of the training datasets.

In the results presented above, the number of new non-faces added to each layer is not very large. That is why the cascade contains many weak classifiers to reach the goal false positive rate, and so the number of weak classifiers with conventional Adaboost is high. Using the GAs optimisation, the number of weak classifiers was reduced considerably. In fact, the total number for the different stages is 464.

3.5 Influence of database size

As shown in Figures 6 and 7, the database size has an important effect on the system's performance. The obtained results with 2,500 faces and 5,000 non-faces are greater than the obtained performances with 500 faces and 1,000 non-faces.

Figure 6 Influence of the database size on the detection rate (DR) using the conventional Haar-like features (see online version for colours)

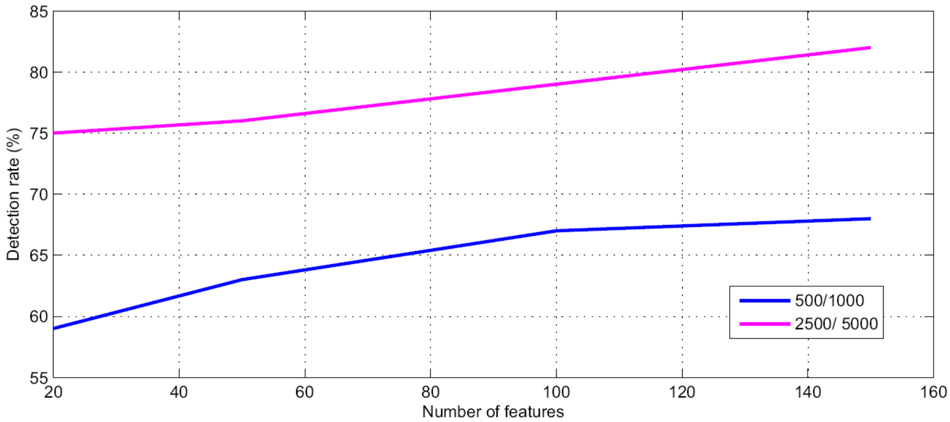
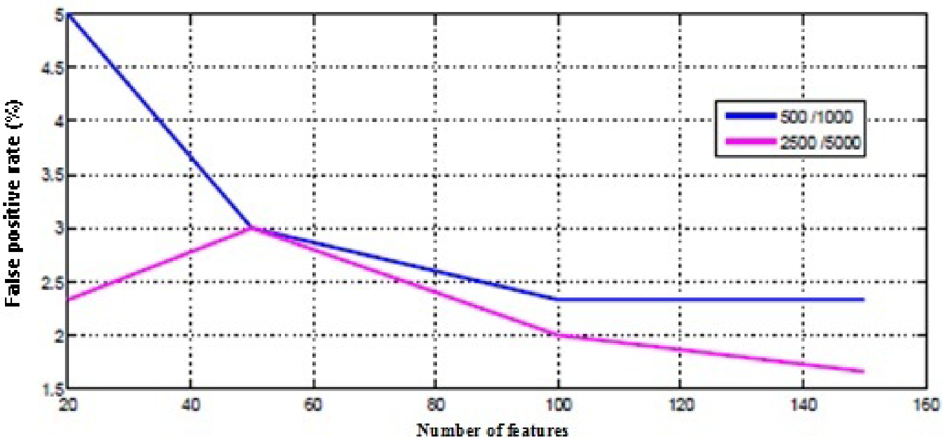


Figure 7 Influence of the database size on the false positive rate (FPR) using the conventional Haar-like features (see online version for colours)



4 Conclusions

In this paper, we have proposed the investigation of the GAs within the Adaboost training process for efficient feature selection. Instead of selecting sequentially weak classifiers with Adaboost training process, we proposed to select them at the same time to construct a strong classifier for each layer. The maximum number of features for each layer is fixed in advance and then the features are selected without redundancy. In the same context, to further improve the training process, we have proposed a new approach based on selection of dominant features determined with Pareto concept. This approach is beneficial in reducing the initial feature space. Another advantage of our proposed method, is that it allows to search the dominant features disregarding the features types.

Our proposed method based on both GAs and dominant features makes possible the reduction of the number of features for each layer. The reduction of the number of features obtained by the two proposed methods makes it suitable for hardware implementation. There is still room to further improve the system performances, so our future work consists on applying a Pareto multi-objective method.

References

Bamini, A.M.A. and Kavitha, T. (2010) 'Dominant local binary pattern based face feature selection and detection', *International Journal of Engineering and technology*, Vol. 2, No. 2, pp.77–80.

Chen, J., Chen, X. and Gao, W. (2004) 'Expand training set for face detection by genetic algorithm resampling', *Proceeding of the sixth IEEE International Conference on Automatic Face and Gesture Recognition*.

Coello, A.C. (1996) *An Empirical study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*', PhD thesis.

Freund, Y. and Shapire, R.E. (1995) 'A decision-theoretic generalization of on-line learning and an application to boosting', *European Conference on Computational Learning Theory*, pp.23–27.

Ghribi, S.F. (2011) *Genetic Algorithm based Wavelet Transform Optimization and Applications*, PhD thesis.

Jammoussi, A.Y., Ghribi, S.F. and Masmoudi, D.S. (2013) 'Adaboost based object detector optimization with genetic algorithm', *IEEE International Conference on Signal Processing and Telecommunication (ICSPT)*.

Pareto, V. (1896) 'Cours d'´economie politique', Vol. 1 and 2, p.438.

Treptow, A. and Zell, A. (2004) 'Combining Adaboost learning and evolutionary search to select features for real-time object detection', *IEEE Congress on Evolutionary Computation (CEC)*, Vol. 2, pp.2107– 213.

Viola, P. and Jones, M. (2001) 'Robust Real-time object detection', *Second International Workshop on Statistical and Computational Theories of Vision Modeling, Learning, Computing and Sampling.*

Viola, P. and Jones, M. (2004) 'Robust real-time face detection', *International Journal of Computer Vision*, Vol. 57, No. 2, pp.137–154.

Yang, M. and Kriegman, D. (2002) 'Detecting faces in images: survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, pp.34–58.

Yang, M., Crenshaw, J., Augustine, B., Mareachen, R. and Wu, Y. (2010) 'Adaboost-based face detection for embedded systems', *Computer Vision and Image Understanding*, Vol. 114, No. 11, pp.1116–1125.

Zalhan, M., Khalid, M. and Yusof, R. (2007a) 'Feature selection using genetic algorithm for face detection', *International Colloquium on Signal Processing and its Applications*.

Zalhan, M., Khalid, M. and Yusof, R. (2007b) 'Enhanced feature selections of Adaboost training for face detection using genetic algorithm (GABoost)', *Proceedings of the Third IASTED International Conference on Computational Intelligence*.

Zhang, C. and Zhang, Z. (2010) *A Survey of Recent Advances in Face Detection*, Technical report.