Adaboost Based Object Detector Optimization With Genetic Algorithm

Ameni YANGUI JAMMOUSSI, Sameh FAKHFAKH GHRIBI, Dorra SELLAMI MASMOUDI
Computers Imaging and Electronic Systems Group
CEMlab Research Laboratory ENIS
University of Sfax,
Sfax Engineering School,
POB, 3038 Sfax
Tunisia
yenguiameni@gmail.com, fakhfakhsameh@yahoo.fr

ABSTRACT: *Object detection has gained a great deal of attention due to the requirements in many real world applications. Adaboost based object detector have shown good results in terms of accuracy and speed. It was not until this method that object detection became widely used in real world applications. The proposed Adaboost based detectors works well, though challenges and difficulties still remain, most of which are mainly related to a large number of examples of the training sets and long training process. A large number of features are required to be selected for Adaboost which slows down training process. In this work, we make headway toward reducing the number of features. The Genetic Adaboost method is proposed to select the most relevant features and discard redundant features. The number of features for a cascade structure was reduced to 57% compared to a standard Adaboost face detector.*

Keywords: Face Detection, Adaboost, Genetic Adaboost

Received: 11 November 2013, Revised 20 December 2013, Accepted 27 December 2013

1. Introduction

One of the most challenging applications in computer vision is detecting objects efficiently in an image or video sequence. Object detection provides interesting challenges due to the requirements for pattern classification, learning and recognition techniques. The interest of developing these techniques comes from many real world applications especially in security, video surveillance and human machine interactions. In a surveillance system for example, the main goal is to detect and classify objects such as faces, pedestrians, animals, vehicles etc.

In general, object detection is a non-trivial problem. Several machine learning approaches have been proposed to this problem, showing significant improvements in detection in terms of accuracy and speed. However, though progress has been accomplished, the best proposed algorithms are far from reaching the speed and ease with which human beings achieve the same task.

Numerous techniques have been proposed for object detection. The most effective proposed methods are learning based ones. They include neuronal network, support vector machines and AdaBoost training algorithms [1]. Viola and Jones algorithm introduced in 2001 [2] falls within object detection based Adaboost approach.

The Adaboost can be seen as a linear combination of weak classifiers in order to obtain a strong one. These weak classifiers are simple features. For example, Haar-like features for face detection, Gradient Orientations Histograms for vehicle detection [10], etc.

The most popular sub-problem within the object detection domain which many researchers focus on is face detection .Since the Viola and Jones framework, the state of-the-art face detection algorithms are the Adaboost based training method and its improvement. To improve the efficiency of this latter, many related researches are proposed according to either alternative boosting algorithms or alternative feature classifiers [3].

In this paper, taking into account some limitations of the conventional Adaboost, we keep on using classical Haar-like features and focus on the optimization of the features selection in the Adaboost training process by the Genetic Algorithm (GA).

The remaining of this paper is organized as follows. Section II presents an overview of Viola and Jones framework. In section III, the conventional Adaboost is described. Section IV briefly represents some limitations of the conventional Adaboost and section V presents our proposed Genetic Adaboost method to overcome these limitations. Section VI presents experimental results and discussion. Finally, section VII summarizes our work and draws some conclusions.

2. Adaboost based Object Detector: Viola and Jones Framework

The Viola and Jones framework is considered as the de-facto standard of face detection in real world applications. They achieved very good results in terms of speed and accuracy.

This method contains three main contributions that make it possible to build an efficient face detector: the integral image, learning classifiers with Adaboost and the cascade structure.

Firstly, they use a cascade structure in order to improve computational efficiency and reduce the false positive rate. The second contribution is the boosted classifiers as well as linear combination of weak classifiers in order to achieve high accuracy. These weak classifiers which are represented by a single feature are selected from a huge amount of computed features. The last contribution is related to the integral image for fast calculation of Haar-like features.

To be more precise, the technique mainly relies on the generation of large set of Haar-like features computed by the integral image representation. The most effective generated features are then selected by the Adaboost algorithm and boosted to form strong classifiers by a linear combination for a cascade structure (Figure 1).

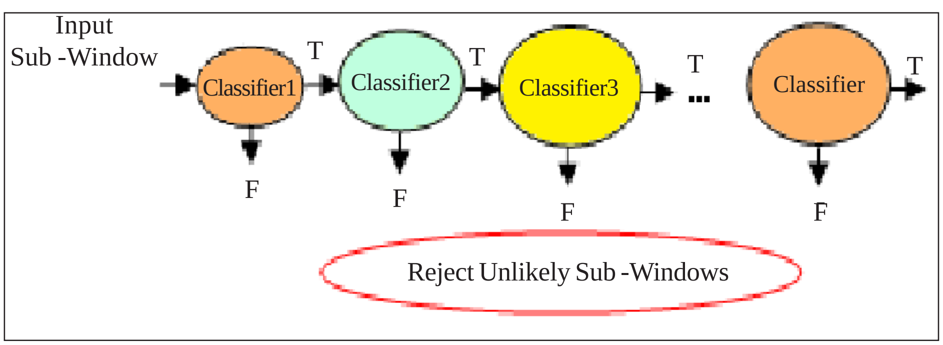

Figure 1. The cascade structure

The input sub-windows are processed by a sequence of classifiers obtained by the boosting process. In the earlier stages, the classifiers which are simples reject a large number of negative sub-windows. Going through the cascade, each classifier slightly more complex than the others (the number of combined weak classifiers increases)

3. Basic Adaboost

Proposed by Yoav Freund and Robert Schapire (1995) [3], the Adaboost technique (Adaptive Boosting) initially present a popular machine learning technique for selecting a set of weak classifiers from a pool of over complete weak classifiers. In the training stage, a very large set of labeled samples is used to identify the best weak classifiers, and a strong classifier is constructed by a weighted linear combination of these weak classifiers.

Each hypothesis in the training algorithm is constructed using a single feature. The algorithm is described as follows.

• Given example images ($x1, y1$), ..., (xn, yn) where $yi = 0, 1$ for negative and positive examples respectively.

• Initialize weights $W_{1,i} = \frac{1}{2m}, \frac{1}{2l}$, for $yi = 0, 1$ respectively, where m and l are the number of negatives and positives examples.

• For $t = 1...T$:

1. Normalize the weights

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

So that w_t is a probability distribution.

2. 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,

$$\in_j = \sum_i W_i \,|\, h_j(x_i) - y_i) \,|$$

3. Choose the classifier, ht, with the lowest error $\in_t$.

4. Update the weights:

$$W_{t+1,i} = W_{t,i} \beta_t^{1-\in_1}$$

Where $ei = 0$ if example xi is classified correctly, $ei = 1$ otherwise, and $\beta_t = \frac{\in_t}{1-\in_t}$

• The final strong classifier is:

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

4. Limitation in Conventional Adaboost Training Process

In our work, some limitations that are related to the learning process are faced. For each iteration in Adaboost learning process, a new weak classifier is selected according to the error criterion. In some cases, we notice that no improvement made according to the detection rate or false positive rate. In general, some features greatly enhance the performances but others do not contribute and even end up with a performance drop. Consequently, often some selected features although leading to lower errors, are irrelevant or redundant which increase the training time and memory resources. A key question here how to adding selected features without degrading performances?

This limitation motivated us to find out a search technique of weak classifiers that outperforms the solution based on lowering the classification error.

In our case, selected features by Adaboost within a single stage are dependant on each others and there is no analytic relation between the number of features and corresponding detection performances, the optimization task is non linear, hard and then seems to be suitable to be treated by genetic algorithms (GA). In the literature review, there are some attempts to apply the GA in Adaboost based face detection [9].

In Treptow and Zell [5], an evolutionary search was employed within Adaboost framework to train a single stage by selecting accurate features from large features pool in reasonable time [4]. As an amelioration of Treptow and Zell work, Zalhan M. et al had used the genetic algorithm with the same manner for cascade of classifiers in less training time[6] [7]. In their proposed technique referred as GABoost, the GA carried out an evolutionary search in the feature space which was enriched with more types of features.

In the following section, we try to overcome these problems by the way of the GA. Instead of selecting features one by one, the proposed method selects simultaneously a number of features to construct a strong classifier.

5. Genetic Adaboost

In our work, we make use of the GA to further improve Adaboost performances by choosing a set of features without redundancy. In fact, redundant features create unnecessary computation and usage of large memory space. Two constrained objectives can be considered. They are the detection rate which has to be maximized and the false positive rate which has to be minimized. The variable that we intend to optimize taking into account the false positive rate as a constraint is the detection rate. In what follows, we describe different steps for our proposed method.

5.1 Representation

Each individual in the population corresponds to a set of weak classifiers. The number of weak classifiers by individual (denoted by T) is variable during the optimization process.

5.2 Construction of the initial population

For more efficient detector, decision is preferably made from earlier stages with minimum computing complexity. Thus, fewer number of classifiers are recommended, so that only hard samples are kept to subsequent stages. When we go forward in the cascade, better performances are required from stage to another suggesting more classifiers to be used. In our approach, a maximal dimension of strong classifiers (denoted by T_{max}) is initialized in Genetic Algorithm and depends on the index of the current stage. In genetic based Adaboost, 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 $I_{i,j}$, $j \in \{1, 2, ..., T_i\}$. T_i and $I_{i,j}$ are generated randomly.

5.3 Fitness computing

The chromosome solutions is evaluated by this function. It depends on criteria which should be maximized or minimized. In our system, the method ε - constraint [8] is adopted.

5.4 Reproduction

For this step the elitist method is used, which is intended to keep on the best individuals. Thus, these latters are reinserted in the future population and the remainder of the future population is constructed based on the wheel selection method.

5.5 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.

5.6 Mutation

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

5.7 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.

To validate our method, we applied it to face database using the classical Haar-like features.

6. Experimental Results

6.1 Choice of the Parameters

We had to choose the cascade parameters and the training examples which determines the number of stages and so the number of features in each stage. The training data set consists of 130 images with 507 labeled frontal faces. The system was trained using 500 faces and 1000 non faces. For the validation set, we have used 100 faces and 300 non faces. The faces were cropped to images of size 19 *19 pixels.

For each stage classifier, the minimal detection rate is 0.98 and the maximal false positive rate is 0.5 on the validation data. We

Positive Train Examples	500
Negative Train Examples	1000
Positive Validation Examples	100
Negative Validation Examples	300
Minimum Detection rate	0.98
Maximum False Positive rate	0.5
Population size	100
Crossover rate	0.8

Table 1. Simulation Parameters

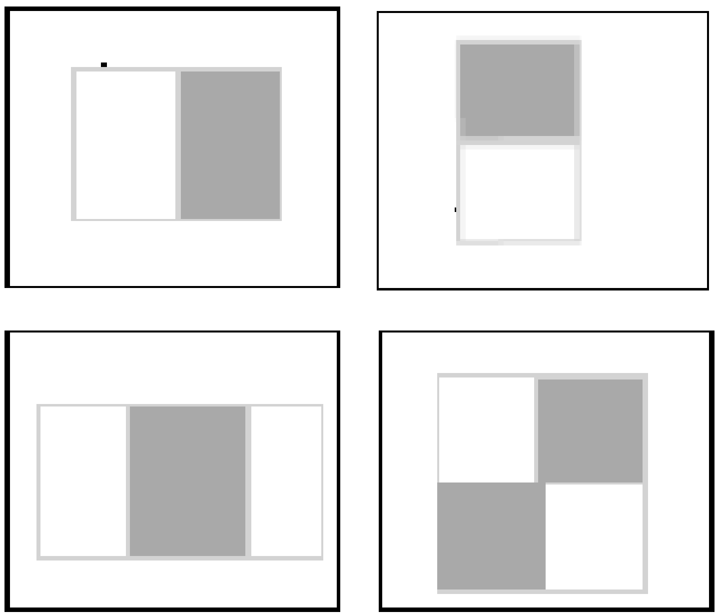

Figure 2. Four types of Haar-like features

Stage	Adaboost	Genetic Adaboost
1	4	4
2	15	15
3	54	26
4	120	30
5	135	34
6	120	16
7	85	61
8	97	78
9	105	48
Total	735	312

Table 2. Number of Features of Each Stage

used 100 as population size, which was initially an arbitrary number but was confirmed by good results. Concerning the crossover rate, since the crossover step should be frequently done, the crossover rate must be high and is generally set to 0.8.

Using 4 types of Haar-like features (Figure 2), we obtain 50040 features for an image of size 19 *19 pixels by varying the width, height and starting position. With these features, we train a cascaded classifier containing 9 stages.

For the learning process, we have to start with a big number of negative examples. Then, at each stage, only the examples that are classified as positive are kept on the subsequent training set. Thus, the next stage in the process is trained to classify the examples that have been misclassified by the previous stages. Furthermore, a few number of hard examples (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.

6. Comparative results and discussion

The obtained results using the conventional Adaboost and Genetic Adaboost are illustrated in the table (Table 1). Using the same parameters for the two methods, the number of features for each stage and the total number of the cascade are given. The total number of features was reduced to 0.57% of that constructed from the Adaboost method.

The number of new non-faces added to each layer is not very large. That's why the cascade contains many weak classifiers to reach the goal false positive rate, and so the number of weak classifiers with classical Adaboost is high. Using the GA optimization, the number of weak classifiers was reduced considerably.

We used a small training database to validate our proposed method. However, we have to use a bigger database in order to obtain an efficient detector in terms of accuracy and speed.

7. Conclusions and future works

In this work, we have investigated the genetic algorithm in the Adaboost process to optimize the system performances given a number of features. The optimization was achieved by selecting the most relevant features and eliminating redundancy.

The reduction of the number of features achieved by the Genetic Adaboost method speeds up the final face detector and makes the obtained cascade suitable for hardware implementation. Furthermore, our method can be applied disregarding the type of features. As the optimization results tied to the effectiveness of the initial population, we intend in the future work to reduce the feature space by eliminating the irrelevant features that do not contribute to the training process.

References

[1] Yang, M., Kriegma,D., Ahuja, N. (2002). Detecting Faces in Images: Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (1) 34-58, January.

[2] Viola, P., Jones, J. (2001) Robust Real-time Object Detection, *In*: Second International Workshop on Statistical and Computational Theories of Vision-Modeling, Learning, Computing and Sampling, Vancouver, Canada.

[3] Yengui Jammoussi, A., Sellami Masmoudi, D. (2011). Feature Extraction based Joint Integral Histogram based Adaboost for Face Detection System, *In*: World Academy of Science Engineering and Technology.

[4] Freund,Y., Shapire, R. E . (1995). A decision-thoretic generalization of on-line learning and an application to boosting, *In*: European Conference on Computational Learning Theory, p. 23-27.

[5] Treptow, A., Zell, A. (2004). Combining Adaboost learning and evolutionary search to select features for real-time object detection, *In*: IEEE Congress on Evolutionary Computation (CEC), V.2, p. 2107-2213, Portland.

[6] Zalhan, M., Khalid, M., Yusof, R. (2007) Feature Selection using Genetic Algorithm for Face Detection, *In*: International Colloquium on Signal Processing and its Applications, Malaysia, March.

[7] Zalhan, M., Khalid, M.,Yusof, R. (2007). Enhanced Feature Selections of Adaboost Training for Face Detection using Genetic Algorithm (GABoost), *In*: Proceedings of the Third IASTED International Conference on Computational Intelligence, Canada, July.

[8] Coello, C. A. C. (1996). An Empirical study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design, Ph.D. thesis, Departement of Computer Science, Tulane University New Orleans.

[9] Yang, M., Crenshaw, J., Augustine, B., Mareachen, R., Wu, Y. (2010). Adaboost-based face detection for embedded systems, *Computer Vision and Image Understanding*, 114, 1116-1125.

[10] Alef, B. (2006). Embedded Vehicle Detection by Boosting, *In*: IEEE Intelligent Transportation Systems Conference, Toronto, Canada.