Face Detection Using Particle Swarm Optimization and Support Vector Machines*

Ermioni Marami and Anastasios Tefas

Aristotle University of Thessaloniki, Department of Informatics
Box 451, 54124 Thessaloniki, Greece
emarami@csd.auth.gr, tefas@aiia.csd.auth.gr

Abstract. In this paper, a face detection algorithm that uses Particle Swarm Optimization (PSO) for searching the image is proposed. The algorithm uses a linear Support Vector Machine (SVM) as a fast and accurate classifier in order to search for a face in the two-dimension solution space. Using PSO, the exhaustive search in all possible combinations of the 2D coordinates can be avoided, saving time and decreasing the computational complexity. Moreover, linear SVMs have proven their efficiency in classification problems, especially in demanding applications. Experimental results based on real recording conditions from the BioID database are very promising and support the potential use of the proposed approach to real applications.

1 Introduction

Face detection deals with finding and localizing faces in images and videos [1]. It is by far the most active specialization in object detection, since it is an essential step in most face-analysis applications, such as facial expression analysis for human computer interfaces, face recognition for access control and surveillance, as well as multimedia retrieval.

Face detection is a rather difficult task due to the variability of the object of interest itself and the environment. For face detection, the following issues need to be considered:

- *Size*: A face detector should be able to detect faces in different sizes. This is usually achieved by either scaling the input image or the object model.
- *Position*: A face detector should be able to detect faces at different positions within the image. This is usually achieved by sliding a window over the image and applying the detection step at each image position.
- *Orientation*: Faces can appear in different orientations within the image plane depending on the angle of the camera and the face.
- *Illumination*: Varying illumination can be a big problem for face detection since it changes the color and the appearance of the face depending on the color and the direction of the light.

* This work has been funded by the Collaborative European Project MOBISERV FP7-248434 (http://www.mobiserv.eu), An Integrated Intelligent Home Environment for the Provision of Health, Nutrition and Mobility Services to the Elderly.

S. Konstantopoulos et al. (Eds.): SETN 2010, LNAI 6040, pp. 369–374, 2010.

In the present paper we propose a novel face detection algorithm that is able to locate frontal faces in a given image, in the two-dimension search space. This is the case in most applications that use face detection. For example, most of the state-of-the-art face recognition and facial expression recognition methods consider that the face has been correctly and precisely located in the image and that it is in frontal view. The most competitive face detection algorithms are searching exhaustively in the test image for localizing the face. To avoid the exhaustive search of all possible locations in the image, we propose a face detection algorithm based on swarm intelligence and more specifically the particle swarm optimization (PSO) method. Each particle is equipped with a very fast and accurate classifier and cooperates with the other particles in order to form an intelligent swarm that is able to detect faces. Applying a nature-inspired intelligent method not only has the advantage of searching the 2D solution space in an efficient way but also an exhaustive search in all possible combinations on the 2D coordinates can be avoided. Indeed, approximately only 5% of the possible image positions had to be examined.

In order to check whether each image sub-window is a face or not, a very fast and efficient classifier is used, which is a linear support vector machine (SVM) that transforms the detection problem to an inner vector product problem. It is worth noting that the proposed method can be combined with any exhaustive face detection technique so as to speed up the detection process.

2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique originally proposed by James Kennedy and Russell C. Eberhart in 1995 [2]. PSO is a search algorithm based on the simulation of the behavior of birds within a flock.

Definitions of several technical terms commonly used in PSO can be found in [3]. The *swarm* is a population of particles. Each *particle* represents a potential solution to the problem being solved. The *personal best* (pbest) of a given particle is the position of the particle that has provided the greatest success (i.e. the maximum value given by the classification method used). The *local best* (lbest) is the position of the best particle member of the neighborhood of a given particle. The *global best* (gbest) is the position of the best particle of the entire swarm. The *leader* is the particle that is used to guide another particle towards better regions of the search space. The *velocity* is the vector that determines the direction in which a particle needs to "fly" (move), in order to improve its current position. The *inertia weight*, denoted by W, is employed to control the impact of the previous history of velocities on the current velocity of a given particle. The *learning factor* represents the attraction that a particle has toward either its own success (C_1 - *cognitive* learning factor) or that of its neighbors (C_2 - *social* learning factor). Both, C_1 and C_2, are usually defined as constants. Finally, the *neighborhood topology* determines the set of particles that contribute to the calculation of the *lbest* value of a given particle.

The position of each particle is changed according to its own experience (*pbest*) and that of its neighbors (*lbest* and *gbest*). Let $\mathbf{z}_i(t)$ denote the position of particle p_i, at time step t. The position of p_i is then changed by adding a velocity $\mathbf{u}_i(t)$ to the current position, i.e.:

$$\mathbf{z}_i(t) = \mathbf{z}_i(t-1) + \mathbf{u}_i(t) \tag{1}$$

The velocity vector reflects the socially exchanged information and, in general, is defined in the following way:

$$\mathbf{u}_i(t) = W\mathbf{u}_i(t-1) + r_1C_1(\mathbf{z}_{pbest_i} - \mathbf{z}_i(t)) + r_2C_2(\mathbf{z}_{leader} - \mathbf{z}_i(t)) \tag{2}$$

where $r_1, r_2 \in [0, 1]$ are random values.

The presented face detector uses the **fully connected graph** as neighborhood topology, where all members of the swarm connect to one another and *leader* = *gbest* in Eq. 2. In the *fully connected* topology the swarm tends to converge faster than in other topologies.

3 Combining PSO and SVMs for Face Detection

We trained SVMs using linear and polynomial kernels [4]. The comparison between them shows that, in our case, the computational complexity for nonlinear SVMs is 1000 times more intensive than linear SVMs. The linear SVMs give a slightly lower success rate ($\approx 1\%$) but they are much faster. Nonlinear SVMs are more intensive because we have to compute inner products for all the support vectors which in our case are more than 1000. Training data for the linear SVM consisted of 2901 grayscale face images and 28121 grayscale non-face images of size 19×19. These images are taken from the CBCL Face Database, which is available at http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html. After fine-tuning the training parameters using cross-validation, we trained the SVMs to the whole training and test set and the success rate for this classifier was 98.74%.

In order to eliminate or minimize the effect of different lighting conditions, images used for training SVMs were histogram equalized and normalized so that all pixel values are between 0 and 1. These preprocessing methods were applied on the entire training images. Normalization and histogram equalization is therefore necessary during detection as well, and they were applied separately in each sub-window investigated. Conclusively, histogram equalization is required but computationally intensive process. Thus, a process of reduced computational burden (that avoids the exhaustive search) is of outmost importance for real-time detection. Such a method that utilizes PSO is discussed next.

Algorithm 1 describes the face detection process using a linear SVM as classifier and the PSO method to decrease computation time. The general idea as described previously is that each particle has its own intelligence using the SVM classifier in order to evaluate the current position on whether it contains a face or not. The particles communicate and inform one another for the most possible face position at each iteration.

Algorithm 1. *Face Detector using PSO*

```
Initialize parameters: inertia, correction factor, maxVelocity, minVelocity
Load trained linear SVM and Read input image
Initialize particles to random positions (x, y)
iteration = 0, repeat = 0
while repeat < R do
  for each particle do
    Update position
    Process this square of image with histogram equalization
    Evaluate SVM's result for this position
    Update pbest
  end for
  Update gbest's index
  Update velocities for each particle
  iteration + +
  if gbest's position has not changed and iteration > K then
    repeat + +
  end if
end while
gbest's position = left upper corner of detected face
```

After K iterations we inspect whether the particles converge or not. If for $R+1$ successive iterations *gbest* is at the same position, the swarm seems to converge and we assume that this location probably contains a face. We check whether the value of SVM output at this point is above a predetermined threshold. If the value of *gbest* is larger than this threshold, we terminate the detection procedure. If *gbest*'s value is below the threshold, we initialize the particles again at random positions and we repeat the above procedure. After fine-tuning the algorithm, we selected for our experiments the values 5 and 3 for parameters K and R respectively.

4 Experimental Results

Frame Detection Accuracy (FDA) is an evaluation metric used to measure any object detection algorithm's performance. This measure calculates the spatial overlap between the ground-truth and the algorithm's output [5]. If the face detection algorithm aims to detect multiple faces, the sum of all the overlaps is normalized over the average number of ground-truth and detected objects. If N_G is the number of ground-truth objects and N_D the number of detected objects, FDA is defined as:

$$FDA = \frac{Overlap\_Ratio}{\left[\frac{N_G+N_D}{2}\right]} \tag{3}$$

where

$$Overlap\_Ratio = \sum_{i=1}^{N_{mapped}} \frac{\left|G_i \cap D_i\right|}{\left|G_i \cup D_i\right|}, \tag{4}$$

N_{mapped} is the number of mapped object pairs in the image, G_i is the i-th ground-truth object image region and D_i is the i-th detected object image region.

The speed performance of the presented face detector is directly related to various parameters, such as the swarm size, the image's initial size and the repeat cycles. The average number of positions examined using PSO, was a small percentage of all the possible combinations of the 2D coordinates ranging from 5 to 6 %. This demonstrates the significant reduction in the number of possible solutions to which we have to apply the classifier each time. That is, using PSO for searching we are able to reduce the time needed for a detection by a factor of 20 for any given face detection algorithm. Moreover, using linear SVMs instead of nonlinear gives another $\times 10^3$ boost in the detection speed.

We applied the presented algorithm to the BioID Face Database (available at http://www.bioid.com/support/downloads/software/bioid-face-database.html) consisted of 1521 gray level images with an initial resolution of 384×286 pixels. To detect faces of various sizes, prior to applying the algorithm, we scaled the initial image using different scaling factors. Emphasis has been placed on *real world* conditions and, therefore, the testset features a large variety of illumination, background and face size.

Figure 1 shows the output of our face detector on some images from the BioID Face Database along with the overlap and the detector output value. Black rectangles represent the ground-truths of every image, while green (lighter gray) windows represent the proposed algorithm's output.

Table 1 lists the detection rate for the presented algorithm in comparison with Viola-Jones' state-of-the-art algorithm [6] using as threshold for overlap the value 25.00. For initial scaling factor 0.19 and 0.24, images are too small for the Viola-Jones algorithm to be detected while our algorithm gives a very good detection rate. So, we apply the algorithms to larger images (scaling factor 0.25

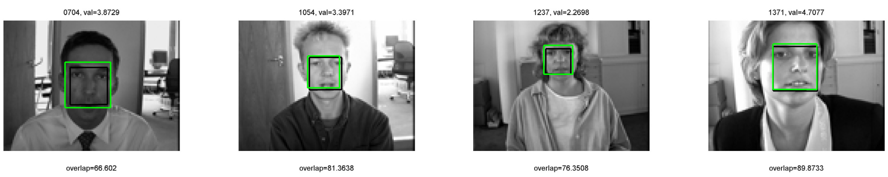

Fig. 1. Output of our face detector on a number of test images from the BioID Database

Table 1. Detection rates for the presented algorithm (SVM-PSO) in comparison with Viola-Jones' algorithm (OpenCV) for the BioID Face Database. The initial scaling factors for the tested images are given in the first row.

Detector	Scale 0.19	Scale 0.24	Scale 0.25	Scale 0.3
SVM-PSO	93.95%	93.23%	93.82%	94.08%
OpenCV	0%	0%	83.30%	94.21%

and 0.3) and the detection rates for our algorithm remain very good. The Viola-Jones algorithm gives for scaling factor 0.25 a relatively good detection rate, whilst for scaling factor 0.3 it gives a detection rate similar to our algorithm. We should, also, mention that the classifier used in Viola-Jones' algorithm is trained using many more training samples than our classifier.

5 Conclusions

We presented a fast and accurate face detection system searching for frontal faces in the image plane. To avoid exhaustive search in all possible combinations of coordinates in 2D space, we used a PSO algorithm. What is more, in order to save time and decrease the computational complexity we used as classifier a linear SVM. Experimental results demonstrated the algorithm's good performance in a dataset with images recorded under *real world* conditions and proved its efficiency. The proposed method can be combined with any face detector, e.g., the one used in OpenCV, to reduce their execution time.

References

1. Goldmann, L., Mönich, U., Sikora, T.: Components and their topology for robust face detection in the presence of partial occlusions. IEEE Transactions on Information Forensics and Security 2(3) (September 2007)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
3. Reyes-Sierra, M., Coello, C.C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research 2(3), 287–308 (2006)
4. Burges, C.: A tutorial on support vector machines for pattern recognition. Kluwer Academic Publishers, Boston (1998)
5. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 319–336 (2009)
6. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)