Probabilistic Darwin Machines for Object Detection

Xavier Baró and Jordi Vitrià
Computer Vision Center, Universitat Autònoma de Barcelona, Campus UAB, 08193 Bellaterra, Spain.
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08007 Barcelona, Spain.
{xbaro,jordi}@cvc.uab.cat

Abstract— *Since the apparition of the first object detection systems in the late 1960s, the complexity of faced scenarios has been continuously growing. From a small set of simple objects in a homogeneous background, where simple alignment strategies or the use of geometric primitives were enough to recognize them, to nowadays, where we are dealing with sets of thousands of complex objects in cluttered scenes, where more sophisticated methods to describe and recognize objects are necessary. In this paper, we present an approach to the object detection problem based on Probabilistic Darwin Machines (PDM), where one of the most used object detection systems has been redefined in order to allow the use large feature sets, which are able to better describe the objects. Two different PDM are used in combination with a large set of visual features in order to learn an object detection system for five different real world visual object classes.*

Keywords: Object Detection, Computer Vision, Evolutionary computation, Boosting

1. Introduction

The detection and classification of objects in images that have been acquired in unconstrained environments is a challenging problem because objects can occur under different poses, lighting conditions, backgrounds and clutter. This variation in the object appearance makes unfeasible the design of handcrafted methods for object detection. Although this problem has been the subject of research from the early beginning of the computer vision field, it has not been until the recent past years that researchers have developed generic object recognition systems for a broad class of real world objects. The key point for this achievement has been the use of a machine learning framework that makes use of very large sets of sample images to learn robust models: Given a training set of n pairs $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the ith image and y_i is the category of the object present in $\mathbf{x}_i$, we would like to learn a model, $f(\mathbf{x}_i) = y_i$ that maps images to object categories.

State-of-the-art methods for visual recognition involve two steps. First, a set of visual features are extracted from the image and the object of interest is represented using these features. Feature selection plays a crucial role in recognition: it facilitates the identification of aspects that are shared by objects in the same class, despite their variability in appearance, and they support discrimination between objects and between classes that can be highly similar. In the second step a classification rule is learned from the chosen feature representation in order to recognize different instances of the object. Depending on the extracted features, different classification methodologies have been proposed in the literature.

Regarding the first step, there are two main approaches to deal with the feature extraction problem:

- *Holistic methods* use the whole object image, that corresponds to a window of the image where the object has to be detected, to define and extract a set of features that will represent a global view of the object. These systems are typically based on defining a template matching strategy by comparing image windows to different m "templates" and recording in a vector the similarity measures. Templates can be learned from data (f.e. using Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Non Negative Matrix Factorization (NDA) or some form of artificial neural net) or can be defined *a priori* (f.e. using a fixed wavelet dictionary or Gabor filter responses). Thus an image $\mathbf{x}_i$ can be considered to be a vector $(x_{i,1}, \ldots, x_{i,m})$ of m scalar values corresponding to m similarity measures.
- *Local methods* model an object as a collection of local visual features or "patches". Thus an image $\mathbf{x}_i$ can be considered to be a vector $(\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,m})$ of m patches. Each patch $\mathbf{x}_{i,j}$ has a feature-vector representation $F(\mathbf{x}_{i,j}) \in \Re^d$; this vector might represent various features of the appearance of a patch, as well as features of its relative location and scale. We can choose from a wide variety of features, such as the fragments-based representation approach of Ullman [1], the gradient orientation-based SIFT [2], or some forms of geometric invariant descriptors.

Regarding the second step, there are two main approaches for defining the classification rule for the object representation $\mathbf{x}_i$: to use a powerful m-dimensional classifier $f(\mathbf{x}_i)$ to learn a rule for assigning the object category, or to use a classification rule $F(\mathbf{x}_i)$ based on the combination of the classification results of several "simple" classifiers $f_n(\mathbf{x}_i)$. In the first case we can use Nearest Neighbor classifier, Support/Relevance Vector Machines, neural networks, etc. In the second case, the most successful approaches have been based on different versions of the AdaBoost algorithm [3],

which are based on a weighted combination of "weak" classifers $F(\mathbf{x}_i) = \sum_t \alpha_t f_t(\mathbf{x}_i)$.

1.1 Haar features

One of the most successfully used *a priori* image features, at least for a broad class of visual objects, are known as Haar features. These features, which are related to the wavelet decomposition, were originally proposed in the framework of object detection by Viola and Jones [4] in their face detection algorithm.

The two-dimensional Haar decomposition of a square image with n^2 pixels consists of n^2 wavelet coefficients, each of which corresponds to a distinct Haar wavelet. The first such wavelet is the mean pixel intensity value of the whole image; the rest of the wavelets are computed as the difference in mean intensity values of horizontally, vertically, or diagonally adjacent squares. Haar features extend this definition to all possible adjacent rectangles in the window (see Figure 1).

Viola and Jones [4] used the Adaboost algorithm [3] to learn a real time face detector with a very high classification performance. In their proposal, weak classifiers are threshold-based classification rules on the values of several Haar features of the image. In order to select the best features, they performed an exhaustive search on the whole set of Haar features for a window. Finally, they built a cascade of strong classifiers to achieve a very low level of false positives.

It is important to note that Haar features constitute an overcomplete dictionary of the image and that there are more than 2^{18} different features for a small image window of 576 pixels (24x24 pixels). This fact imposes a high computational cost on the learning step of the Adaboost algorithm, which involves several rounds of exhaustive searches. From a practical point of view, the development of a high performance object detector represents, when using conventional hardware, a learning time of the order of several hundred hours.

The work of Viola and Jones was extended by Lienhart and Maydt [5], who showed that the use of a larger feature set may improve the convergence and the performance of the final classifier. The extension of the feature set was done by adding rotated versions of original Haar-like features, and thus adding a factor to the exponential relation between the size of the feature set and the training time.

1.2 Dissociated Dipoles

Another natural extension of the Haar features is the dissociated dipoles proposed by Balas and Sinha in [6] in the context of computational neuroscience. As in the case of Haar-like features, they are region based comparisons, but in this case we always have only two regions, and these regions do not have to be adjacent regions. The main reason to consider these features is because while Haar-like features are local descriptors, the dissociated dipoles maintain this ability but also includes non-local descriptors of the images. In addition, they Sinha demonstrated that human visual system can perform that type of non-local comparisons, and that these comparisons benefits recognition systems.

The proposal of Balas and Sinha, in spite of the fact that can be easily adapted to Adaboost-based detectors, has not been used in any real world object detector due to computational limitations: there are more than 2^{28} different dissociated dipoles in standard image window of 24×24 pixels, what makes the use of the Adaboost algorithm unfeasible.

1.3 Learning object detectors with a huge number of visual features.

In order to address this fundamental limitation, in this paper we propose to cast the search-based weak learner of the classical approach as an optimization problem, where the goal is to find the parameters of the weak classifier that minimize the error function without exploring all the solution space.

There are many different approaches to deal with optimization problems, most of them based on gradient descent, as line search methods, normalized steepest methods or the Newton steps method. All these methods require a differentiable function, and uses the gradient direction to move from a certain solution to a better one. Given the nature of our classification function, we propose the use of evolutionary algorithms are a good choice.

Evolutionary computation refers to a wide family of methods that are inspired on the Darwin's evolution theory. Natural evolution is a population-based optimization process. The simulation of this process using computers results in stochastic optimization techniques which often outperform classical methods of optimization when applied to difficult real-world problems. Darwin hypothesized that living beings adapted and differentiated to varying conditions or niches in their environment through a process of evolution. Although in Darwin times genetics was an unknown field, most of the works which are based on evolution assumes the actual knowledge about genetics and define algorithms based on a chromosome-based encoding, and the processes observed in natural evolution of species. However, the Darwinian processes can be defined in a more general manner, with no assumptions about the implementation of these processes. This more general definition has been widely developed in memetics [7], [8], the field that attempts to cope with evolutionary models of information transmission. Calvin [8] defined the essential processes in any evolutionary model, in order to ensure a quality improvement along generations as:

1) There must be a pattern involved.
2) The pattern must be copied somehow.
3) Variant patterns must sometimes be produced by chance.
4) The pattern and its variant must compete with one another for occupation of a limited work space.

5) The competition is biased by a multifaceted environment. That's Darwin's natural selection.
6) New variants always preferentially occur around the more successful of the current patterns. This is what Darwin later called an inheritance principle.

Apart from the evolutionary algorithms that simulate the natural behavior, recently, a new paradigm in the evolutionary computation field is taking huge relevance. This paradigm implements the Darwinian processes by means of probability models. The patterns are encoded by means of a set of random variables. These random variables are estimated by means of a probability model, which is sampled in order to obtain copies of the pattern with random variations. The fittest samples are used in order to estimate a new probability model (competition), which better represents the promising regions of the search space. Therefore, in this new paradigm, instead of working with a search space which points represent the patterns, in this case each point of the search space is a probability distribution, and the goal of the evolutionary process consist of finding the probability model which better represent the promising patterns of the original search space.

On of the most extended implementation for the Probabilistic Darwin Machines, is commonly known as *Evolutive Algorithms Based on Probabilistic Models* (EAPM), a new paradigm which starts with the work of Baluja and Caruana [9], where the traditional operators of the genetic algorithms (mutation and crossover) had been replaced with the estimation and sampling of a probabilistic model. The new algorithm was named *Population Based Incremental Learning* (PBIL), and consists of a simple univariate model, where all the variables are assumed to be independent. The best individuals of each generation are used to update these variables, and finally the model is sampled to obtain a new generation. In spite of its simplicity, these algorithms demonstrated to converge to good solutions for several problems, and introduced an interesting view on evolutionary computation: the extraction of a statistical description of the promising solutions, in terms of a probability distribution is the base of PDMs and the new systematic way to solve hard search and optimization problems that they represent. Working on this novel approach we want to assess the real value of using large visual feature sets in real world detection tasks and to this end we develop a set of detectors for very general visual classes: faces, pedestrians, cars, text and traffic signs.

The paper is organized as follows: section 2 presents the evolutive object detection framework. Section 3 is an introduction to the Probability Darwin Machines framework, where the algorithms used in the comparatives are explained. Finally, in section 4 the results of the proposed detection schema are presented and analyzed, to conclude the paper in section 5.

2. Object Detection

The object detection used in this paper is based on the Adaboost algorithm and Haar-like features of Viola and Jones in [4], but in our case we do not use combinations of features and threshold to obtain the simple classifiers, only the sign is used instead. This type of features are called ordinal features, which have demonstrated to be robust against various intra-class variations and illumination changes [10]. In addition, we experiment with two larger feature sets than Haar-like features, which cannot be used with the classical Adaboost approach, but can be used with the Adaboost based on PDMs. These feature sets are the *Dissociated Dipoles* introduced by Sinha in [6] and the *Weighted Dissociated Dipoles*, an extension of the dissociated dipoles that allows to represent most part of the Haar-like features (see Figure1).

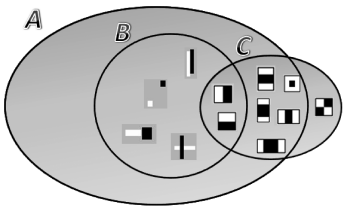


Fig. 1: Graphical comparison of feature sets: A) Weighted dissociated dipoles, B) Dissociated dipoles and C) Haar-like

The main idea of Adaboost is to associate to each sample a measure of how many times each sample has been missclassified by the previous added weak learners, and therefore, allowing to add new simple classifiers that concentrate on those samples which are systematically miss classified. At the end, the final classifier is a weighted combination of several simple classifiers. In Adaboost jargon, the simple classifier is named *Weak Classifier* and the final classifier is named *Strong Classifier*. The process of learning the best weak classifier using the weights distribution is named *Weak Learner*. Now, in the Adaboost algorithm two main parts can be differentiated:

- *The weights distribution maintenance*: The way weights are updated corresponds to a gradient descend in the error. We use the Gentle Adaboost [11], which uses Newton stepping rather than exact optimization at each step (see Alg. 1).
- *The Weak Learner*: The weak learner depends on which weak classifiers we use, and corresponds to a search over all the possible configurations of the parameters that define the weak classifier to deal with the one that minimizes the weighted error using the weights distribution. When the search space is sufficiently small and the weak classifiers are simple enough, this step is done by an exhaustive search, which is the case of Viola and Jones [4], Lienhart [5] and many other works that use Adaboost. When complex features are used, or the search space is too large, this traditional approach

becomes unfeasible. We propose the use of evolutionary computation to face this problem.

Algorithm 1 The Gentle Adaboost [11]

Input:

Sequence of M labeled examples $(x_1, y_1), ..., (x_M, y_M)$ with $y_i \in \{-1, +1\}$

Initialize $W_1(i) = 1/M$

for $t = 1, .., T$ **do**

Fit the regression function $h_t(x)$ which minimizes

$$\epsilon_t = Pr_{i \sim W_t}[h_t(x_i) \neq y_i]$$

Get weak hypothesis $h_t(x) \mapsto \{-1, +1\}$ with error ϵ_t

Update:

$$W_{t+1}(i) \leftarrow W_t(i) \times exp(-y_i \times h_t(x_i))$$

Normalize W_{t+1} so

$$\sum_{i=1}^{M} W_{t+1}(i) = 1$$

end for

Output: the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} h_t(x)\right)$$

This paper is focused on the Weak Learner, and on how we can formulate it as an optimization problem, where any type of evolutionary algorithm can be applied, allowing the use of huge feature sets with a reasonable training time. The first step is to formalize the problem:

Given a training set $\langle(\mathbf{x}_1, y_1), ..., (\mathbf{x}_M, y_M)\rangle$, where $y_i \in \{-1, +1\}$ is the target value for sample $\mathbf{x}_i$, the goal of an object detection learning algorithm is to deal with the strong classifier $H(\mathbf{x}_i) = y_i$. In the boosting framework, we define a distribution $W = \{w_1, ..., w_M\}$ over the training set, where each w_i is the weight associated to the sample $\mathbf{x}_i$, and $H(\mathbf{x})$ corresponds to an additive model $H(\mathbf{x}) = \sum_t \alpha_t h_t(\mathbf{x})$ where the final decision is a combination of the decisions of several *weak classifiers* $h(\mathbf{x}) \in \{-1, +1\}$. In contrast to the strong classifier $H(\mathbf{x})$ where we expect a good performance for any sample $\mathbf{x}_i$ in the training set, in the case of weak classifier we only expect they are better than a random decision.

Given $\mathcal{H}$ the set of all possible weak classifiers, $h^{\mathbf{s}} \in \mathcal{H}$ a certain weak classifier defined by parameters $\mathbf{s}$, W the weights distribution of the Adaboost and $\epsilon(h^{\mathbf{s}}) = Pr_{i \sim W}[h^{\mathbf{s}}(\mathbf{x}_i) \neq y_i]$ the error function, the regression step consists on finding $\mathbf{s}^*$ that $\epsilon(h^{\mathbf{s}^*}) \leq \epsilon(h^{\mathbf{s}}) | \forall h^{\mathbf{s}^*}, h^{\mathbf{s}} \in \mathcal{H}$, where the complexity of finding $\mathbf{s}^*$ depends on the size of $\mathcal{H}$. In the following, we analyze the definition of the problem using different types of features.

Using ordinal measures, Haar-like features can be parameterized by the upper-left position of one of the regions (X, Y) and their size (W, H), because the size and position of all the regions in Haar-like features is predefined for each type T. Therefore, a Weak Classifier with ordinal Haar-like [5] features can be defined as:

$$h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}$$
$$\text{where} \quad \mathbf{s} = (X, Y, W, H, T) \tag{1}$$

To evaluate this feature, the mean intensity value of negative region (black one) is subtracted from mean value of positive region (white one). The sign of this subtraction is used as the final class value. In the case of dissociated dipoles defined by Shina [6], we have two regions with no restriction on position and sizes, therefore, we need to parameterize both regions independently:

$$h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}$$
$$\text{where} \quad \mathbf{s} = (X_e, Y_e, W_e, H_e, X_i, Y_i, W_i, H_i) \tag{2}$$

Although this type of features allows the representation of the image using non-local comparisons, and can simulate some of the Haar-like features, we lose some interesting configurations of Haar-like features, as the line detectors. We extend the dissociated dipoles to the *Weighted Dissociated Dipoles*, where we assign to each dipole a weight $RW \in \{1, 2\}$, obtaining a new feature set that allows to simulate the main part of Haar-like features (see Figure1). The new weak classifier can be written as:

$$h^{\mathbf{s}}(\mathbf{x}) \mapsto \{-1, +1\}$$
$$\text{where} \quad \mathbf{s} = (X_e, Y_e, W_e, H_e, RW_e, X_i, Y_i, W_i, H_i, RW_i) \tag{3}$$

In addition to the mandatory parameters defined above, we can add extra parameters to improve the results. For instance, we add a polarity parameter which inverts the classification value. This parameter allows a fast step on the search process, inverting the regions.

Once the different feature sets have been described, we summarize the problem as finding the best instance of a set of random variables $\mathbf{s} = \{X_1, ..., X_K\}$, to minimize $\epsilon(h^{\mathbf{s}}) = Pr_{i \sim W}[h^{\mathbf{s}}(\mathbf{x}_i) \neq y_i]$, where the only difference between each feature set is how to evaluate $h^{\mathbf{s}}$ and the dimension of $\mathbf{s}$.

If we analyze the random variables, it is easy to discover that there are multiple dependencies between them, for instance, regions with positions near the right or bottom sides of the training window cannot have large sizes. The rest of the paper is concentrated on defining an evolutionary approach based on the use of Probability Darwin Machines, which allows to solve this optimization problem taking into account these dependencies.

3. Probabilistic Darwin Machines

In the literature we can find a wide variety of Probabilistic Darwin Machines (PDMs), in which the most important difference is the used probability model. Taking into account the considered interactions between variables, we can classify the

models within three main types: Univariate models where no interactions are considered, bivariate models with only pair-wise interactions and finally the models that allow multiple interactions. Once the most convenient probability model is selected, different estimation and sample strategies can be used, thus, we can find different algorithms that share the same type of model. The most known and used algorithms are the UMDA [12], PBIL [9] and cGA [13] for univariate models, MIMIC [14], COMIT [15] and BMDA [16] in the case of bivariate models and finally, considering models with multiple interactions the FDA [17], EcGA [18], BOA [19] and EBNA [20].

The use of a complex model allows to better represent the features space, but it adds complexity to the estimation and sampling stages. As a general rule, more complex models are more reliable but at the expense of bigger execution times. In Fig. 2, a graphical comparison of the stages involved in the evolutionary progress for general Darwin Machines, Genetic-based Darwin Machines, and Probabilistic Darwin Machines is shown.

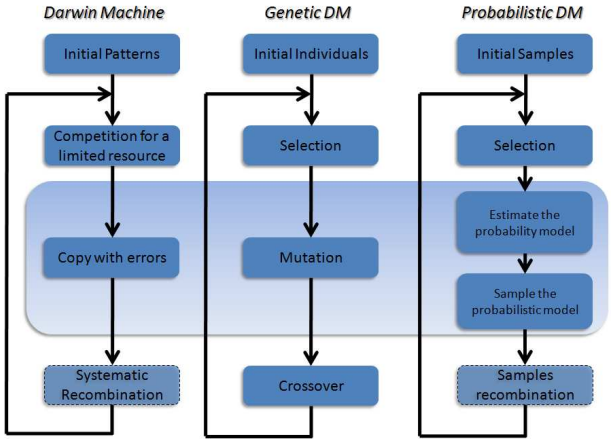


Fig. 2: Graphical comparison between Darwin Machines, Genetic based Darwin Machines (GA), and Probabilistic Darwin Machines

In section 2, we show that the optimization formulation derived from the object detection problem has multiple dependencies between the variables, and thus, we need to use a model that allows to represent all these interactions. In this section two PDMs are introduced: The first one is a standard algorithm referred as Extended Compact Genetic Algorithm (EcGA), and the second one is the PDBNBE, a novel PDM based on Naïve Bayes Models Estimation.

1) EcGA: The *Extended compact Genetic Algorithm* was introduced by Harik in [18]. The underlying idea is to use a factorization of the join probability model as a variable length product of marginal distributions. They use a rapid method in order to find groups of related variables, where each group is considered independent from the rest of the groups. The length of each product is related to the number of variables in the same group. Using this method, the join probability distribution of the n variables is calculated as:

$$P(\mathbf{x}) = \prod_{c \in C} P(\mathbf{x}_c) \tag{4}$$

where C is the set of groups, and $P(\mathbf{x}_c)$ is the marginal distribution of the variables on group c. Since this algorithm builds disjoint groups of variables, for all $c, k \in C$ we can assume:

$$\bigcup_{c \in C_l} \mathbf{X}_c = \{X_1, \ldots, X_n\}, \mathbf{X}_c \cap \mathbf{X}_k = \emptyset \tag{5}$$

In order to create the groups of variables, Harik initially builds an initial partition of n groups of one variable. From this starting point, this algorithm begins an iterative process that fuse pairs of groups. In order to select the groups to be fused, the author defines the *combined complexity*, a measure based in a combination of the sum of the marginal distributions entropies and a complexity penalization based on the *minimum description length principle*. At each iteration, those two groups that being fussed obtains the higher reduction of this measure are selected in order to be fused, creating a larger group with the variables contained in both of them.

The *combined complexity* measure can be defined as the sum of two complexities, the population complexity

$$\begin{aligned} J_p &= N \sum_{c \in C} h(\mathbf{X}_c = \mathbf{x}_c) \\ &= -N \sum_{c \in C} \sum_{\mathbf{x}_c} P(\mathbf{X}_c = \mathbf{x}_c) \log P(\mathbf{X}_c = \mathbf{x}_c) \end{aligned} \tag{6}$$

and the model complexity

$$J_m = \log N \sum_{c \in C} dim \mathbf{X}_c \tag{7}$$

where $dim\mathbf{X}_c$ is the number of required parameters in order to describe the marginal distribution $\mathbf{X}_c$.

2) PDMNBE: Although the probability model used in EcGA is able to create clusters of variables with dependencies, allowing to describe multiple dependencies among some problem variables, this structure is limited to these learned clusters. The use of more powerful probability model can allow the use of more complex dependencies, obtaining a model with a better adaptation to each problem. In the other hand, more complex models can require complex estimation algorithms which needs larger amount of data and increase the learning time. From the point of description power, the Bayesian Network (or a belief network) have demonstrated to be the better solution, and is widely used in the literature.

Given a set of random variables $\{X_1, X_2, ..., X_N\}$, representing a Bayesian network using probabilistic graphical models is done using an acyclic directed graph, where each node corresponds to a variable (measured parameter, latent variable or hypothesis) and whose arcs encode the dependence between variables. Denoting as π_i the set of parents of X_i (nodes with an arc pointing to X_i), and assuming that each node is conditionally independent of its non-descendants given its parents, the joint probability distribution can be

conveniently written as the product of the local distributions of each node and its parents:

$$Pr(X_i, ..., X_N) = \prod_{i=1}^{N} Pr(X_i|\pi_i) \quad (8)$$

Learning a Bayesian network from data is a two-fold problem: Structure learning and parameter estimation. Although there exist good methods to estimate the structure and parameters of a Bayesian network, because exact inference is $\#P$-complete and thus the existent methods are often too costly, approximate methods like Markov Chain Monte Carlo and loopy belief propagation must be used. The applicability of the Bayesian networks is limited by the fact that these methods have an unpredictable inference time and its convergence is difficult to diagnose.

In [21] Lowd & Domingos demonstrated from an empirical point of view that the representation power of a Bayesian network can be approximated using Naïve Bayes models, and presented an optimal estimation algorithm so-called *Naïve Bayes models Estimation* (NBE) by their authors. The NBE algorithm consists of an Expectation Maximization (EM) wrapped in an outer loop that progressively adds and prunes mixture components. The input data is split into a training set and a hold-out set. It begins with a single component consisting of each variable's marginal distribution. At each cycle, k new components are added, using a random training example to initialize each component, and removing these k seed examples from the data to avoid overfitting. The number of components k is doubled at each cycle. If there are m components before the cycle starts and n new ones are added, the weight $Pr(c)$ of each pre-existing component is rescaled by $m/(m+n)$, and each new component receives an initial weight of $1/(m+n)$.

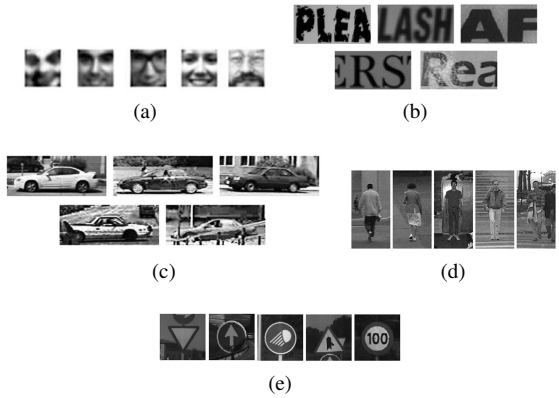
Within each cycle, until the log likelihood of the hold-out data fails to increase by at least a fraction δ_{EM}, the expanded set of components is fitted using EM. At each iteration, the current model is saved if it yields the best holdout log likelihood so far. Since each step of EM takes time linear in the number of components, every five EM steps and after it ends, the low-weight components are pruned out in order to speed up the learning process. When an entire refinement step passes with little (less than δ_{Add}) or no improvement on the hold-out set, two final steps of EM on the best model are done with all the data.

If this algorithm is introduced in the PDM framework, we obtain a novel evolutionary algorithm referred as *Probabilistic Darwin Machine based on Naïve Bayes models Estimation* (PDMNBE).

4. Results

Once the optimization formulation has been defined, and the methodology to perform the optimization introduced, in this section we compare the learning ability of Probabilistic Darwin Machines to learn object detection problems. To get a reference value, the results are compared with the ones obtained using a Genetic Algorithm. We have selected different real world object detection problems, using for each one a public database to facilitate the reproduction and comparison of the results. A brief description of each data set is given in the following, and some examples of positive samples for each dataset are shown in Fig. 3.

- *Faces:* We use the MIT-CBCL face database with a random selection of 1.000 face images and 3.000 non-face images.
- *Text:* We use the text location dataset from the *7th International Conference on Document Analysis and Recognition (ICDAR03)*.
- *Cars:* We use the UIUC cars database, with a total of 1.050 images containing 550 instances of lateral views of different cars in urban scenes and 500 images of background.
- *Pedestrians:* We use the INRIA Person Dataset, with 2.924 images divided into 924 pedestrian instances and 2.000 background images.
- *Traffic Signs:* In this case we use real images acquired in the context of a mobile mapping project provided by the *Institut Cartográfic de Catalunya* (ICC). The database consists of 1.000 images containing a traffic sign and 3.000 background images.



(a) (b) (c) (d) (e)

Fig. 3: Data set examples. *a)* Faces *b)* Text *c)* Cars *d)* Pedestrians *e)* Traffic signs

In order to allow a better comparison, we fix some shared parameters for all the methods, such as a population size of 100 individuals and the maximum number of iterations to 200. In the case of Genetic algorithms, a Gaussian based mutation probability (the Gaussian is centered at zero with a variance of the half of the variable range, decreasing the variance along the generations), and scattered cross-over strategy with a cross-over fraction of 0.8 are used. In Table 1, the Area under the ROC Curve (AUC) is shown for each combination of method, feature set, and dataset.

Experiment		Method		
Features	**Dataset**	**GA**	**EcGA**	**PDMNBE**
Haar-like	Cars	69.65%(3)	73.80%(1)	70.62%(2)
	Faces	55.22%(2)	52.96%(3)	55.43%(1)
	Text	45.84%(3)	47.83%(2)	47.92%(1)
	Pedestrians	54.00%(3)	55.39%(2)	55.91%(1)
	Traffic Signs	63.30%(3)	64.60%(2)	65.37%(1)
Mean Rank		2.8	2	1.2
Dis. Dip	Cars	79.27%(1)	76.19%(3)	78.59%(2)
	Faces	67.47%(3)	68.63%(2)	69.40%(1)
	Text	50.33%(3)	50.80%(1)	50.48%(2)
	Pedestrians	68.16%(3)	68.22%(2)	70.37%(1)
	Traffic Signs	61.53%(3)	63.13%(1)	63.06%(2)
Mean Rank		2.6	1.8	1.6
W.D.D.	Cars	95.21%(1)	85.99%(3)	94.64%(2)
	Faces	87.74%(2)	79.73%(3)	88.27%(1)
	Text	80.35%(2)	79.88%(3)	83.27%(1)
	Pedestrians	88.40%(2)	80.55%(3)	90.01%(1)
	Traffic Signs	87.92%(3)	89.71%(2)	89.82%(1)
Mean Rank		2	2.8	1.2
Mean Rank		2.4667	2.2	1.3333

Table 1: Area under the ROC curve for all combination of feature sets, datasets, and evolutionary methods.

Following Demšar statistical comparison methodology [22], we compute all the statistics to check for the null hypothesis case. From those statistical values, we can state that the null-hypothesis can be rejected in the case of Haar-like features, Weighted Dissociated Dipoles and in the overall case. If we proceed with the Nemenyi post-hoc test, the critical difference in the case of Haar-like features is 1.48. Since the rank differences between GA and EcGA is $2.8 - 2 = 0.8 < 1.48$, both methods are statistical equivalent. The rank difference between GA and PDMNBE is $2.8 - 1.2 = 1.6 > 1.48$, therefore we can state that for Haar-like features, PDMNBE is statistically significant better than GA. Finally, since the rank difference between EcGA and PDMNBE is $2 - 1.2 = 0.8 < 1.48$, there is no statistically significant difference between these two methods. For the Weighted Dissociated Dipoles, we obtain the same critical difference $CD = 1.48$, and following the same process, we can state than GA and EcGA are statistically equivalents, and in this case, PDMNBE is better than EcGA.

Finally, the critical difference in the global comparison is $CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 0.86$. If we analyze the overall rank differences, we can state that PDMNBE performs statistically better than GA ($2.47 - 1.2 = 1.27 > 0.86$) and EcGA ($2.2 - 1.2 = 1.0 > 0.86$), while non difference exist between GA and EcGA ($2.47 - 2.2 = 0.27 < 0.86$).

5. Conclusion and Future Work

We presented an evolutionary object detection framework based on Probabilistic Darwin Machines. Using these methods in combination with the Adaboost algorithm, we are able to use richer feature sets which improve the results obtained with the classical Haar-like features. In this scenario, the Probabilistic Darwin Machines, and in special the one based on Naïve Bayes Models Estimation outperform the results obtained with Genetic Algorithms.

Acknowledgment

This work is partially supported by MEC grant TIN2006-15308-C02-01, Ministerio de Ciencia y Tecnologia, Spain, and Consolider Ingenio CSD 2007-00018. This work has been developed in a project in collaboration with the *Institut Cartogràfic de Catalunya* under the supervision of Maria Pla.

References

[1] S. Ullman and E. Sali, "Object classification using a fragment-based representation," London, UK, 2000, pp. 73–87.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the ICCV'09, Corfu*, 1999, pp. 1150–1157.

[3] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. of the ICML'96*, 1996, pp. 148–156.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on CVPR*, vol. 1, 2001, pp. 511–518.

[5] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Procs. of ICIP'02*. Rochester, USA: IEEE, September 2002, pp. 900–903.

[6] B. Balas and P.Sinha, "Dissociated dipoles: Image representation via non-local comparisons," Annual meeting of the Vision Sciences Society, Sarasota, FL., 2003.

[7] S. Blackmore, *The Meme Machine (Popular Science)*. Oxford University Press, USA, May 2000.

[8] W. Calvin, "The six essentials? minimal requirements for the darwinian bootstrapping of quality," May 1997.

[9] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *The Int. Conf. on Machine Learning 1995*. Morgan Kaufmann Publishers, 1995, pp. 38–46.

[10] K. Thoresz and P. Sinha, "Qualitative representations for recognition," *Journal of Vision*, vol. 1, no. 3, pp. 298–298, 12 2001.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Dept. of Statistics, Stanford University Technical Report., 1998.

[12] H. Muhlenbein, "The equation for response to selection and its use for prediction," *Evolutionary Computation*, vol. 5, pp. 3:303–346, 1997.

[13] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE-EC*, vol. 3, no. 4, p. 287, November 1999.

[14] J. S. de Bonet, C. L. Isbell, Jr., and P. Viola, "MIMIC: Finding optima by estimating probability densities," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, p. 424.

[15] S. Baluja and S. Davies, "Using optimal dependency-trees for combinational optimization," in *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 30–38.

[16] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing - Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds. London: Springer-Verlag, 1999, pp. 521–535.

[17] H. Mühlenbein and T. Mahning, "The factorized distribution algorithm for additively decomposed functions," in *Second Symposium on Artificial Intelligence(CIMAF'99). La Habana*, 1999, pp. 301–313.

[18] G. Harik, "Linkage learning via probabilistic modeling in the ecga," University of Illinois at Urbana-Champaign, Tech. Rep. 99010, 1999.

[19] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Procs. of GECCO'99*, 1999, pp. I:525–532.

[20] R. Etxeberria and P. L. naga, "Global optimization with bayesian networks." in *In Second Symposium on Artifcial Intelligence(CIMAF-99), Cuba*, 1999, pp. 332–339.

[21] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," in *Procs. of ICML'05*, NY, USA, 2005, pp. 529–536.

[22] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, January 2006.