Introducing fuzzy decision stumps in boosting through the notion of neighborhood

Pedro C. Ribeiro Plinio Moreno
José Santos-Victor
Instituto Superior Técnico
Instituto de Sistemas e Robótica
Lisboa, Portugal
{pribeiro, plinio, jasv}@isr.ist.utl.pt

February 21, 2012

Abstract

This paper presents a novel approach to the weak classifier selection based on the GentleBoost framework. We include explicitly the notion of neighborhood in one of the most common weak learner in boosting, the decision stumps. The availability of neighboring points adds a new parameter to the decision stump, the feature set (i.e. neighborhood), and turns the single branch selection of the decision stump into a fuzzy decision that weights the contribution of each branch using a neighborhood-based confidence measure. The confidence measure of the fuzzy stumps use neighboring samples to increase the robustness to local data perturbations.

The appropriate definition of the neighborhood in the dataset allows the application of the fuzzy stumps framework in a wide range of problems. In this paper we address two types of scenarios to show their advantages: i) time-based neighborhoods and ii) space-based neighborhoods. In both scenarios we evaluate experimentally the properties of the fuzzy stumps, considering computer generated datasets and real classification problems, such as human activity recognition and object detection.

1 Introduction

Boosting algorithms combine efficiency and robustness in a very simple and successful strategy for classification problems. The advantages of this strategy have led several works to improve the performance of boosting on different problems by proposing modifications to the key elements of the original AdaBoost algorithm [1]: (i) the procedure to compute the data weights [2, 3, 4, 5], (ii) the selection of the base classifier (e.g. [6, 7, 8]) and (iii) the loss function it optimizes [9]. In this paper we address the selection of the base classifier to bring improvements to classification for computer vision problems.

Several works have shown experimental performance improvements by grouping subsets of data samples into the base classifier. On one hand, when the groups are defined previously such as the analysis of genomic data, each data sample is divided into groups of data points in order to build a base regressor for each group. Then the most relevant group regressors are selected to construct a new base regressor, which improves the prediction error [6]. On the other hand, when the objective is to search for groups in the data set, the data samples are gathered to build augmented base classifiers. This approach has been followed by the TemporalBoost [8], the SpatialBoost [7] and the scale-space based weak regressors [10]. In order to detect human events in videos, the TemporalBoost algorithm, proposed by [8], incorporates previous classifier responses into the current decision by averaging their responses. The average response of the weak classifier is selected when such a temporal support decreases the misclassification error at the current frame. Since the temporal support is not a parameter of the weak classifier selection, the parameters of the weak learner are not optimally chosen for the time-based criterion. In order to segment objects in images, the SpatialBoost algorithm, introduced by [7], computes the response of two weak classifiers: the single pixel and the neighborhood-based. The weak classifier with minimum error selects the appropriate spatial support to improve the segmentation result for each pixel. Since the SpatialBoost is designed for interactive image segmentation, the search for the weak classifier is limited to the individual features (pixels) and one type of neighborhood.

In addition, the features of the neighborhood are the pixel value of the center and the class labels of its neighbors. The scale-space regressors, proposed by [10], use spatial support as well, but the scale of the weak regressor is increased by an octave if the regression error decreases. Thus, the resolution of the data along the iterations is a monotonically non-decreasing function with faster convergence. The scale-space criterion fits very well in the regression problem but in the more general classification problem, the error could have several local minima over the scales, so a more exhaustive search over the scales is needed.

We present a new approach to weak classifier selection for boosting, which augments the search space by using neighboring data samples. Although the general objective of our work is similar to the TemporalBoost [8] and SpatialBoost [7] approaches, we include the notion of neighborhood in the weak classifier in a comprehensive manner, which allows to apply the same algorithm on different problems where the neighborhood notion is explicitly defined. SpatialBoost builds the feature neighborhoods by adding the pixel value of the center and the class label of the center's neighborhood. We propose a more general approach by using the pixel (feature) values of the neighborhood instead of the class labels. We introduce a feature set selection procedure to the weak learner, which must choose the neighborhood that minimizes the classification error at each round. Using decision stumps as weak learners, our proposal is based on the average computation of the weak learner response in the vicinity of a data sample. This new type of weak learner selects jointly the parameters of the decision stump and its neighborhood, a procedure that turns the single branch selection of the decision stump into a linear combination of the branches. Such a combination of the stump branches is commonly referred to as a fuzzy decision on [11], [12], [13]. Moreover, [14] shows empirically that the fuzzy tree decreases the variance and consequently improves the classification output. The extension of our fuzzy stumps proposal to fuzzy decision trees has a similar behavior but relies on the feature set (neighborhood) to build the decision function that combines the response of both branches for each base learner.

We explain how to exploit the advantages of the fuzzy decision stumps in two sce-

narios: time-based neighborhood and space-based neighborhood. In both scenarios we evaluate experimentally the properties of the fuzzy stumps in computer generated datasets, which allow us to control the level of noise and complexity of the problem. The synthetic tests are followed by the application of the fuzzy stumps on real classification problems: (i) Human activity recognition on videos, which is based on the temporal fuzzy stumps and (ii) car and face detection on images, which is based on the spatial fuzzy stumps. These experimental results show a better performance of the fuzzy stumps when compared to the common stumps.

2 Fuzzy decision stumps in GentleBoost

In this section we explain how to add fuzziness to the weak learner response, using neighboring data points to perform the classification.

2.1 GentleBoost with decision stumps

The GentleBoost algorithm builds a final strong classifier, whose output is the log-odd of the class given a feature point $x_i \in \mathbb{R}^d$. The problem is, at each round, to find the optimal weak classifier h_m that minimizes the classification error. This is done using adaptive Newton steps, resulting in minimizing, at each round, a weighted squared error

$$J = \sum_{i=1}^{N} w_i (y_i - h_m(x_i))^2, \tag{1}$$

where $w_i = e^{-y_i h_{m-1}(x_i)}$ are the weights and N the number of training samples.

The optimal weak classifier is then added to the strong classifier, followed by the adaptation of the data weights. Table 1 shows the GentleBoost algorithm.

The choice of the weak classifier h_m depends on the application, but a common choice is based on efficient and interpretable models such as the decision stumps. They have the form

$$h_m(x_i) = a\delta\left[x_i^f > \theta\right] + b\delta\left[x_i^f \leq \theta\right], \tag{2}$$

where $f \in \{1, \ldots, d\}$ is the feature number and δ is an indicator function (i.e.

1. Given: $(x_1, y_1), \ldots, (x_N, y_N)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$, set $H(x_i) := 0$, initialize the observation weights $w_i = 1/N$, $i = 1, 2, \ldots, N$
2. Repeat for $m = 1, \ldots, M$
   (a) Find the optimal weak classifier h_m over (x_i, y_i, w_i).
   (b) Update weights for examples $i = 1, 2, \ldots, N$, $w_i := w_i e^{-y_i h^*_m(x_i)}$
3. Compute the strong classifier as $H(x_i) = \sum_m^M h^*_m(x_i)$ and classify the sample x_i according to $\operatorname{sgn} H(x_i)$

Table 1: GentleBoost algorithm.

δ[*condition*] is one if *condition* is *true* and zero otherwise). The optimal value of the stump parameters $\{a, b, f, \theta\}$ are found by the minimization of J w.r.t. h_m. [15] presents a closed form for a and b and the values of f and θ are found using an exhaustive search. In addition, we consider in our experiments the natural extension of the decision stumps, the decision trees.

2.2 Fuzzy weak learners

We extend the use of one feature to a set of features, defined in a group of available neighbors for each feature. This is done by modifying the indicator functions of Eq. (2) in order to collect the values of the indicator functions within the neighborhood of x_i^f. As the neighborhood (i.e. feature set) is an additional parameter of the indicator function, our algorithm must find the feature set that achieves minimum error, in the same way as the usual decision stump finds the optimal feature f.

The neighborhood of x_i^f is defined by a set of feature indexes $\mathcal{I} = \{(i, f), (i_1, f_1), \ldots, (i, f_p), \ldots, (i_P, f_P)\}$ that selects the neighboring points $\mathcal{N} = \{x_i^f, x_{i_1}^{f_1}, \ldots, x_{i_p}^{f_p}, \ldots, x_{i_P}^{f_P}\}$. The index set $\mathcal{I}$ is obtained from a neighboring map, which needs to be defined for each type of neighborhood. For instance, if we compute the feature x_i on each frame of a video sequence, the features computed on the previous T frames $\{x_{i-1}, \ldots, x_{i-T}\}$ belong to the neighborhood of x_i. Thus, the temporal neighboring map of the video-based feature index (i, f) is $(i - t, f) \quad t = 1, \ldots, T$, so the neighborhood $\mathcal{N}$ has three parameters: the data point index i, the feature dimension f and the temporal extent T. Let us denote $\mathcal{P}$ as the parameters of the index set $\mathcal{I}(\mathcal{P})$ and its correspondent neighborhood $\mathcal{N}(\mathcal{P})$. Then, the set of indexes of the time-based

neighborhood is

$$\mathcal{I}(\mathcal{P}) = \mathcal{I}(i, f, T) = \{(i, f), (i-1, f), \ldots, (i-T, f)\} \tag{3}$$

and its correspondent neighborhood is

$$\mathcal{N}(\mathcal{P}) = \mathcal{N}(i, f, T) = \{x_i^f, x_{i-1}^f, x_{i-2}^f, \ldots, x_{i-T}^f\}. \tag{4}$$

The example of Eq. (3) and Eq. (4) consider all windows sizes from 1, only one feature, to $T+1$ that is the maximum length of the window.

In addition to the time-based data, we address spatially constructed data samples such as the features computed on a digital image. Let us consider the data point x_i as the feature values computed on an image of size $W \times H$, where W is the width and H is the height. The feature point x_i^f is related to the image point (η, ξ) as follows: $f = \eta \cdot W + \xi$. The neighboring map is based on a spatial mask, which selects the data points in the neighborhood of x_i^f. For instance, the elliptical mask centered at the image point (η, ξ) and rotated by α selects all the points in the set

$$e(f = \eta \cdot W + \xi, A, B, \alpha) = \left\{ (u^{'}, v^{'}) : \frac{(u^{'} - \eta)^2}{A^2} + \frac{(v^{'} - \xi)^2}{B^2} <= 1 \right\} \tag{5}$$

$$u^{'} = u\cos\alpha - v\sin\alpha$$

$$v^{'} = v\sin\alpha + v\cos\alpha.$$

Thus, the index set of the elliptical image neighborhood of the feature point x_i^f is

$$\begin{aligned}\mathcal{I}(\mathcal{P}) &= \mathcal{I}(i, f = u_i^{'}W + v_i^{'}, A, B, \alpha) \\ &= \{(i,f), (i, f = u_1^{'}W + v_1^{'}), \ldots, (i, f = u_{P-1}^{'}W + v_{P-1}^{'})\}, \quad (u_i^{'}, v_i^{'}) \in e(f, A, B, \alpha)\end{aligned} \tag{6}$$

and its correspondent neighborhood is

$$\begin{aligned}\mathcal{N}(\mathcal{P}) &= \mathcal{N}(i, f = u_i^{'}W + v_i^{'}, A, B, \alpha) \\ &= \{x_i^f, x_i^{u_1^{'}+Wv_1^{'}}, \ldots, x_i^{u_{P-1}^{'}+Wv_{P-1}^{'}}\}, \quad (u_i^{'}, v_i^{'}) \in e(f, A, B, \alpha)\end{aligned} \tag{7}$$

The example of Eq. (6) and Eq. (7) consider the ellipses with parameters $\mathcal{P} = i, f = \eta W + \xi, A, B, \alpha$. Let us include the neighborhood $\mathcal{N}(\mathcal{P})$ in the decision stump by computing the average response of the base learner h_m,

$$g_m(x_i) = \frac{1}{|\mathcal{N}(\mathcal{P})|} \sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} h_m(x_i^f) \tag{8}$$

$$= \frac{1}{|\mathcal{N}(\mathcal{P})|} \sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} \left(a\delta\left[x_i^f > \theta\right] + b\delta\left[x_i^f \leq \theta\right]\right), \tag{9}$$

where $|\mathcal{I}(\mathcal{P})|$ is the cardinality of the neighborhood $\mathcal{N}(\mathcal{P})$. This expression can be rearranged in order to put a and b in evidence,

$$\begin{aligned} g_m(x_i) \quad &= \quad a\left(\tfrac{1}{|\mathcal{N}(\mathcal{P})|} \textstyle\sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} \delta\left[x_i^f > \theta\right]\right) + \\ &+ \quad b\left(\tfrac{1}{|\mathcal{N}(\mathcal{P})|} \textstyle\sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} \delta\left[x_i^f \leq \theta\right]\right). \end{aligned} \tag{10}$$

From the equation 10 it is easy to note that the performed output averaging only modifies the indicator function. The new indicator functions are:

$$\Delta_+(x_i, \theta, \mathcal{N}(\mathcal{P})) = \frac{1}{|\mathcal{N}(\mathcal{P})|} \sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} \delta\left[x_i^f > \theta\right], \tag{11}$$

which computes the percentage of features in the set $\mathcal{N}(\mathcal{P})$ that are grater than the threshold θ and

$$\Delta_-(x_i, \theta, \mathcal{N}(\mathcal{P})) = \frac{1}{|\mathcal{N}(\mathcal{P})|} \sum_{\forall x_i^f \in \mathcal{N}(\mathcal{P})} \delta\left[x_i^f \leq \theta\right], \tag{12}$$

which computes the percentage of features in the set $\mathcal{N}(\mathcal{P})$ that are less than the threshold. The functions Δ_+ and $\Delta_- = 1 - \Delta_+$ of Eq. (11) and Eq. (12) sample the interval $[0\,1]$ according to the number of features in the set $\mathcal{N}(\mathcal{P})$. For example, if $|\mathcal{N}(\mathcal{P})| = 2$ this yields to $\Delta \in \{0,\ 1/2,\ 1\}$, if $|\mathcal{N}(\mathcal{P})|$, $\Delta \in \{0,\ 1/3,\ 2/3,\ 1\}$ and so on. The new weak learners, the fuzzy decision stumps, are expressed as follows:

$$g_m(x_i) = a\Delta_+(x_i, \theta, \mathcal{N}(\mathcal{P})) + b\Delta_-(x_i, \theta, \mathcal{N}(\mathcal{P})). \tag{13}$$

At each round, the fuzzy stumps of Eq. (13) add to the strong classifier a weighted sum of both branches, with confidence weights Δ_+ and Δ_- for the decisions a and b respectively.

Substituting the fuzzy weak stump of Eq. (13) into the error function

$$J = \sum_{i=1}^{N} w_i (y_i - g_m(x_i))^2, \tag{14}$$

the optimal decision parameters a and b are obtained by the minimization of the error of Eq. (14),

$$\frac{\partial J}{\partial a} = 0 \quad \frac{\partial J}{\partial b} = 0. \tag{15}$$

The solution of Eq. (15) yields

$$a^* = \frac{\bar{\nu}_+ \bar{\omega}_- - \bar{\nu}_- \bar{\omega}_\pm}{\bar{\omega}_+ \bar{\omega}_- - (\bar{\omega}_\pm)^2}, \tag{16}$$

$$b^* = \frac{\bar{\nu}_- \bar{\omega}_+ - \bar{\nu}_+ \bar{\omega}_\pm}{\bar{\omega}_+ \bar{\omega}_- - (\bar{\omega}_\pm)^2}, \tag{17}$$

with $\bar{\nu}_+ = \sum_i^N w_i y_i \Delta_+, \quad \bar{\nu}_- = \sum_i^N w_i y_i \Delta_-,$
$\bar{\omega}_+ = \sum_i^N w_i \Delta_+, \qquad \bar{\omega}_- = \sum_i^N w_i \Delta_-,$

and $\bar{\omega}_\pm = \sum_i^N w_i \Delta_- \Delta_+.$

Note that variables of Eq. (16) and Eq. (17) are functions of $\{x_i, f, \theta, \mathcal{N}(\mathcal{P})\}$ that we drooped for notation simplicity. Also note that if at round m, the optimal neighborhood $\mathcal{N}(\mathcal{P})$ of the fuzzy stumps contains only a feature point, the resulting weak learner becomes the usual decision stump.

There is no closed form to compute the optimal f, θ and $\mathcal{N}(\mathcal{P})$, thus exhaustive search is usually performed. Finding the optimal θ and f is a tractable problem, but finding the neighborhood $\mathcal{N}(\mathcal{P})$ could be a very hard problem by testing all possible parameters $\mathcal{P}$, thus it is essential to use a priori knowledge of the problem to reduce the search space.

2.2.1 Neighborhood selection

The optimization of the neighborhood is a very hard problem because exhaustive search could take a prohibitive long time.

The assumption of neighboring data points allow us to bound the search of the neighborhood parameters according to the local structure of the data sets, reducing the search space from a very large number of neighborhoods to a subset promising candidates. The search of the time-based feature sets is reduced to a set of predefined temporal windows bounded to a few seconds, such that the confidence measures Δ_+ and Δ_- provide robustness to local perturbations. For instance, if we bound the temporal search to 2 sec. on a 25fps camera, this yields T=50 for Eq. (3) and Eq. (4), so the search space is reduced to 50 different neighborhoods.

The search of the space-based neighborhoods is reduced to a set of predefined 2D functions. These functions are inspired by the image filters, which define a kernel around a pixel to compute the filter response. Thus, we define the elliptical masks as the natural spatial support of a feature point, selecting the points inside the ellipse as part of the neighborhood of the feature point.

Figure 1(a) shows 13 examples of binary masks generated by the procedure just described and Figure 1(b) their implementation in digital images, with the masks plotted at the center pixel of each image. Note that the first mask represents a feature point (pixel).

3 Assessments of fuzzy learners in synthetic data

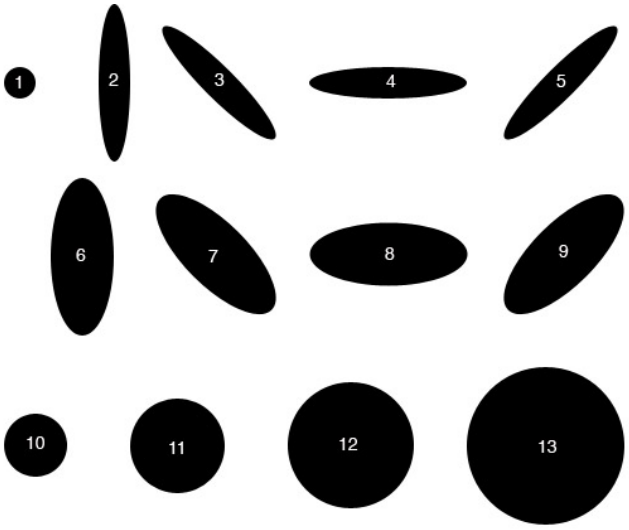
We aim to evaluate the properties of the fuzzy learners when exposed to controlled distortions of the data. These synthetic databases allow us to compare the performance and robustness of our fuzzy learners in noisy and transformed versions of the datasets. In addition, we consider the neighborhood selection problem in two scenarios: (i) space-based neighborhood selection for image classification and (ii) time-based neighborhood for 2D trajectory classification (previously presented [16]).

The image classification is a binary class problem, where the objective is to distinguish a specific mask against the remaining ones. The masks represent geometric shapes and the task is to provide the correct label of each image. In this case we evaluate the fuzzy learners using a space-based neighborhood. The second scenario is the classification of 2D trajectories for both binary and multi-class problems. The objective is to provide the correct label of each 2D point and in this case we evaluate the time-based neighborhood for the fuzzy stumps.

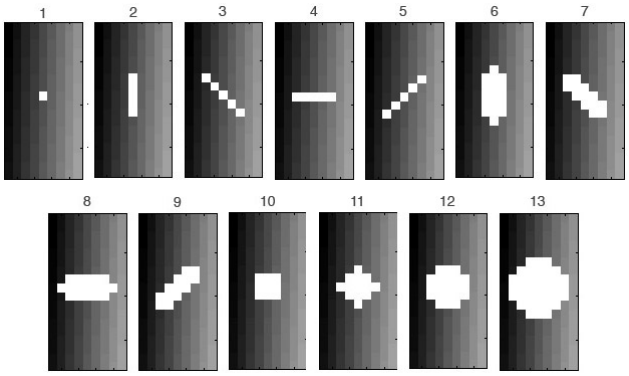
3.1 Spatial support for fuzzy learners

The experiments of this section compare the capabilities of our fuzzy stumps and trees with spatial support against the common decision stumps and trees. Figure 2(a) shows the masks of the positive class (square) and the four shapes of the negative class: i) circle, ii) triangle, iv) rhombus and v) hexagon. In order to evaluate the robustness of the methods to different noise levels and data transformations, we generate modified versions of the initial masks. The modifications include: i) pixel noise, ii) shape rotation and iii) shape translation.

The space-based neighborhoods are two dimensional structures that bring high-



(a)



(b)

Figure 1: The 13 masks used to search for the best space-based neighborhood, in the case of Section 3. The plot 1(a) on top shows the continuous masks and the bottom plot 1(b) their respective discrete implementation. The mask number 1 represents only one pixel and the remaining ones define different neighbor sets centered at that pixel. Masks from numbers 2 to 5 have parameters $(A = 1, B = 0.1)$ and masks from 6 to 9 have parameters (A=1,B=0.4) and in both cases $\alpha = 0, 45, 90, 135$ degrees. The last four masks are circles and the parameters are (A=B=0.4), (A=B=0.6), (A=B=0.8) and (A=B=1).

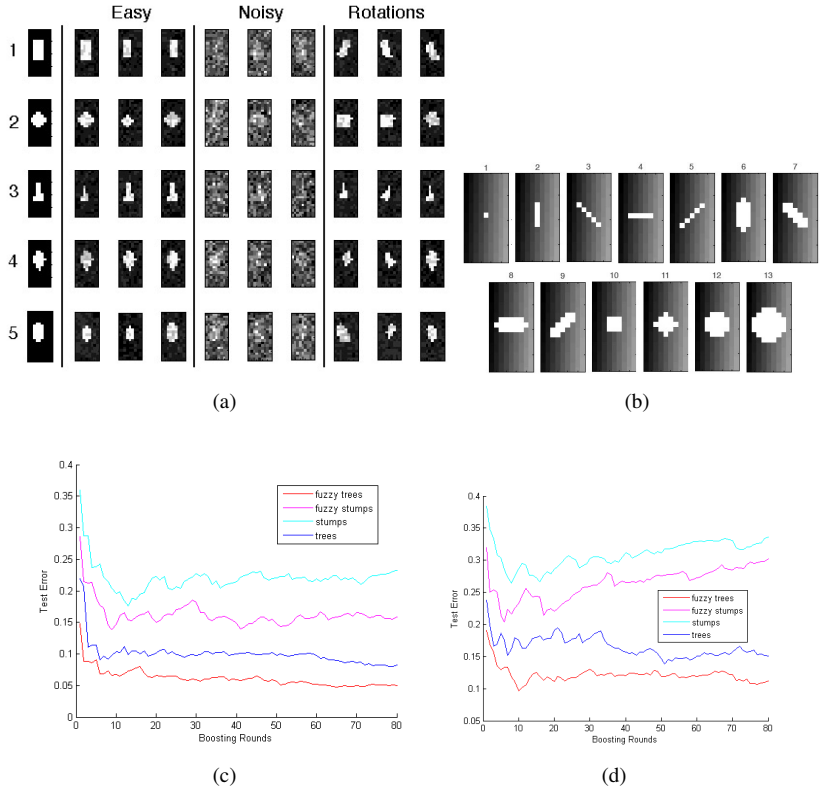


Figure 2: The original image set is presented in the first column of Image 2(a). The second group of columns of Figure 2(a) with the title "Easy", plots examples of the distortions that correspond to a pixel noise with 5% variance and rotations of 2 degrees. In the same figure, the group of columns labeled as "Noisy" corresponds to the 67.5% noise case and 18 degrees of variance in rotations. The last group of columns of Figure 2(a) aims to illustrate the influence of the 18 degrees rotations and 5% variance of the added noise. Figure 2(b) shows the subset of neighborhoods $\mathcal{N}(\mathcal{P})$ of Eq. (7) for searching the best space-based fuzzy weak learner. Figures (c) and (d) show the evolution of the error in the testing data set over the boosting rounds that corresponds to the last column of Figure 2(a), labeled as Rotations.

order interactions among features, so in this scenario we also include the decision trees with four splits, for both the decision stumps and the fuzzy stumps. Thus, we have four types of weak classifiers for the GentleBoost algorithm: (i) decision stumps, (ii) decision trees with four nodes, (iii) fuzzy stumps and (iv) fuzzy trees with four nodes. The comparison between the four learning procedures relies on the maximum of the recognition rate over the GentleBoost rounds.

Each 18×9 image mask represents one data sample $x_i \in \mathbb{R}^{162}$ (Eqs. 2,13) (feature points, $f = 1, \ldots, 162$). The selection of the best neighborhood of the fuzzy stumps follows the method described in section 2.2.1, using the masks shown in Figure 2(b), so the neighborhood search is reduced to 13 sets. Remind that at each round the fuzzy stumps selects the set $\mathcal{N}(i, f = \eta W + \xi, A, B, \alpha)$ of Eq. (7) that minimizes the weighted error of Eq. (14).

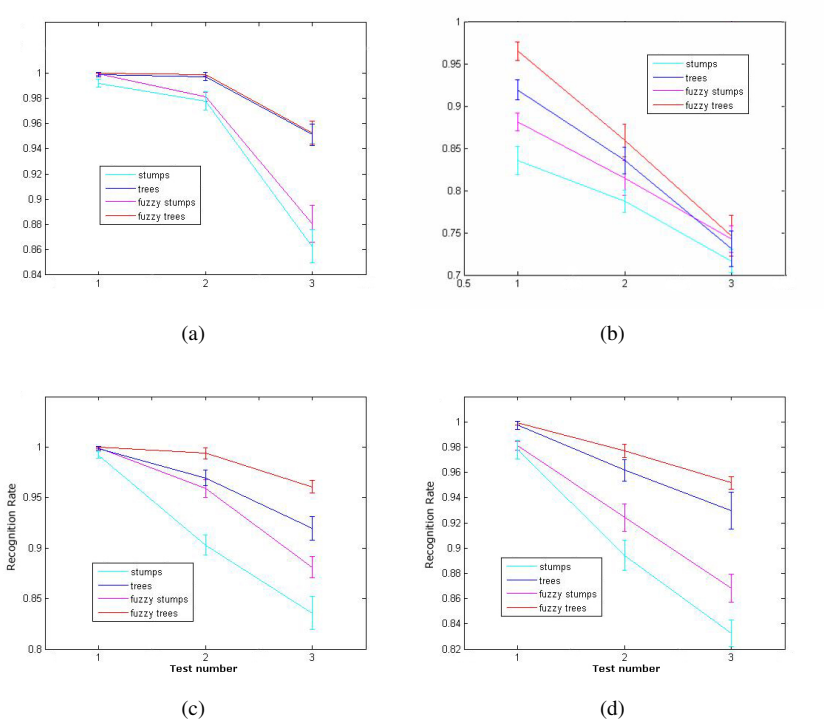
Figure 2(a) shows examples of the generated data samples. The distortions of the initial image masks have one or more of the following components:

- Image noise, which is simulated by the addition of noise to every pixel independently. The noise model is a Gaussian with zero mean and variance ranging from 5% to 67.5% of the maximum pixel value.

- In plane 2D rotations, which are determined by Gaussian functions with zero mean and variance varying from 2 deg to 18 deg.

- Small image translations, which are applied to the entire dataset and their range is from one to two pixels in each direction. For every image, the probability of being translated follows a uniform distribution.

For each experiment, the generation of the training and testing data comprises 11 different groups of 200 samples per class are generated. One of the groups corresponds to the training set of every classifier and the remaining 10 groups correspond to the testing set. Figures 2(c) and 2(d) show two examples of the evolution of the error in the testing set, which present the general trend of the results: the fuzzy learning strategy brings improvements for both the decision stumps and the decision trees with four splits.



(a) (b)

(c) (d)

Figure 3: Maximum recognition rate achieved for each method while varying the noise parameters for the train and test sequences. 3(a) presents the results varying the gaussian noise from 5% to 45% with the rotation variation fixed to 2 degrees and 3(b) presents the results varying the gaussian noise from 5% to 67.5% with the rotation variation fixed to 18 degrees. In 3(c) and 3(d) the pixel noise variance is fixed to 5% and 25% respectively while the rotation variance changes from 2 to 18 degrees in both cases.

In the first group of experiments the level of noise is the same for the training and testing data sets. Figures 3 show the maximum recognition rate over rounds achieved by GentleBoost with: (i) decision stumps, (ii) fuzzy stumps, (iii) decision trees with four splits and (iv) fuzzy decision trees with four splits. It is important to remark that the tree version of each weak learner is better than its single-stump counterpart. In addition, the fuzzy version of each learner attain better recognition rate than the simple one. Figures 3(a) and 3(b) correspond to variations in the pixel noise parameter wile the rotation is fixed to 2 and 18 degrees respectively. In the first case, with less rotations, the improvement obtained with the fuzzy version is less visible, but in Figure 3(b), with 18 degrees variance for the rotation parameter, the fuzzy procedure brings

clearer improvements. It is also noticeable, from Figure 3(b), that with more pixel noise present in the images the difference between the learners is narrowed.

When varying the variance of the rotations parameter maintaining the amount of pixel noise fixed, 5% in Figure 3(c) and 25% in Figure 3(d), we see that the fuzzy learners are clearly more robust and this difference tends to increase when augmenting the rotations.

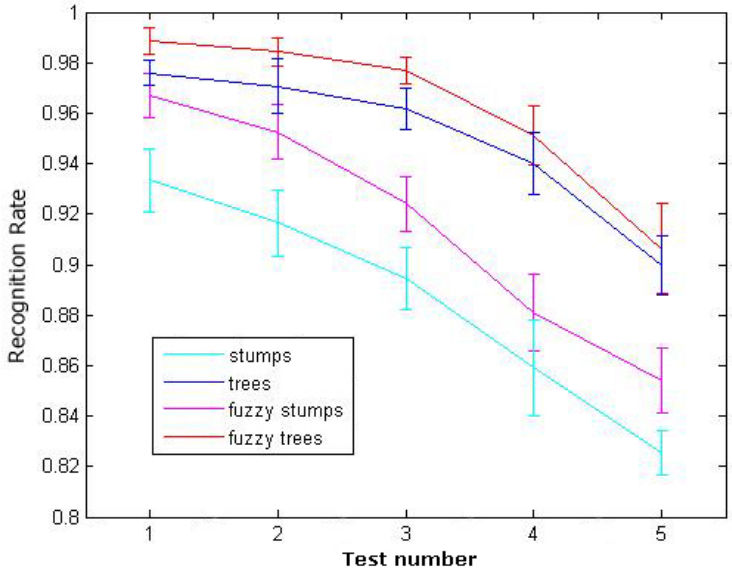
In the second group of experiments, the levels of noise of the training and testing sets are different. Figure 4 shows the recognition rates of this setup where the methods were trained with the parameters that correspond to test number 3. In Figure 4(a) the training was performed with a pixel noise of 25% and a rotation variance of 18 degrees, varying then the pixel noise from 5% to 45% (test number 1 to 5). Figure 4(b) corresponds to the methods trained with pixel noise variance of 5% and rotation variance of 10 degrees. The results confirm that the fuzzy stumps perform better than the decision stumps in all situations and the behavior is very similar to the obtained in the previous setup. It is important to remark the very similar performance of the fuzzy stumps and decision trees on Figure 4(b) and even in the most difficult case the fuzzy stumps perform better than the decision trees. This suggests that the neighboring interactions between features are as important as other high-order interactions, when the data set allows to define such a neighborhood notion.

3.2 Discussion

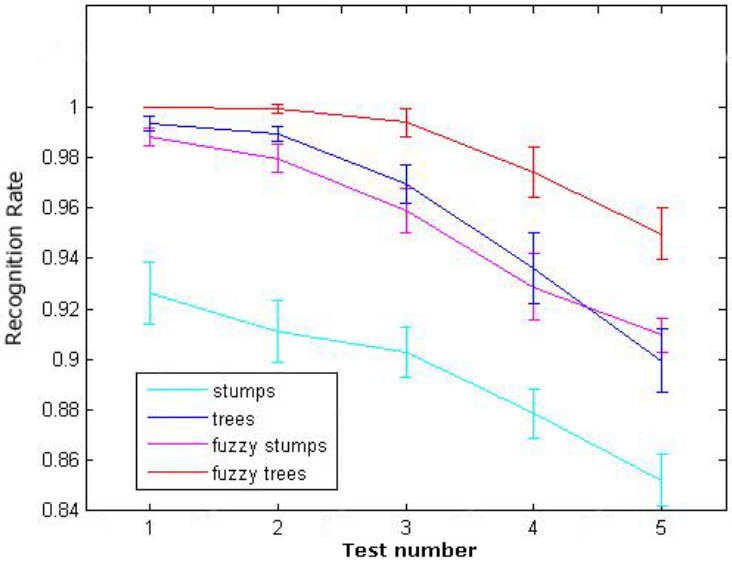
We evaluate the performance of our fuzzy weak learners, in two problems:

- Binary image classification using the spatial support for the fuzzy learners.
- 2D trajectory classification using the temporal support for the fuzzy stumps (previously presented [16]).

Although the very different nature of the problems, we were able to apply successfully the fuzzy learners in both setups. The requirement for this behavior is the definition of an explicit and adequate notion of neighborhood for each problem. In addition, the better performance of the fuzzy stumps when compared to the decision



(a)



(b)

Figure 4: Maximum recognition rate achieved for each method while varying the noise parameters only in the test. In each case the classifiers where trained with a specific set of noise parameters and tested then varying the amount of one of those parameters. In Figure 4(a) the training was performed with a pixel noise of 25% and a rotation variation of 18 (that corresponds to test number 3), then the test results where obtained varying the pixel noise from 5% to 45%. In 4(b) the train was done with a pixel noise variance of 5%, and with a rotation variance of 10 degrees (that corresponds to test number 3). Then the test was performed varying the rotation parameter from 2 to 18 degrees

stumps, shows experimentally the advantages of the fuzzy indicator function over the common indicator function and suggest to apply the FuzzyBoost directly in similar problems.

The fuzzy stumps (Eq. (13)) have two main advantages over the sharp decision of the common stumps (Eq. (2)):

- Prevents the addition of wrong decisions due to noisy feature points and
- brings additional robustness to data transformations.

4 Fuzzy learners on real problems

In this section we address real classification problems following the procedure of the previous section, which evaluates the fuzzy learners in two different scenarios: (i) time-based fuzzy stumps for human activity recognition (previously presented [16]) and (ii) space-based fuzzy learners for object detection.

4.1 Space-based fuzzy learners for object detection

The objective of these tests is to classify an image as the object (positive class) or the background (negative class) in two different problems: (i) car detection and (ii) face detection. The spatial neighborhood is applied to the pixels of the Sobel filter response of each image. We follow the comparison standard of Section 3.1, so we have four types of weak classifiers for the GentleBoost algorithm: (i) decision stumps, (ii) decision trees, (iii) fuzzy stumps and (iv) fuzzy trees.

The car detection data set (UIUC Image Database - DataSet3) contains images of side views of cars and background patterns. The face detection data set (CBCL MIT face data - DataSet4) contains frontal facial and non-facial images.

4.1.1 Car detection

The UIUC Image Database [17] is composed of 1050 training examples, 550 car and 500 non-car cropped images, all of size 40x100 pixels. In addition, the database pro-

vides the original images and the location of the cars in the test images. To generate the positive and negative data we use a sliding window method that extracted 200 cars and 1705 non-cars. Figure 5 shows data set images of the computed features, the negative and positive classes and the masks selected. The data samples for classification are provided by the magnitude of the image gradient, which is computed from the Sobel filter response on horizontal and vertical directions.

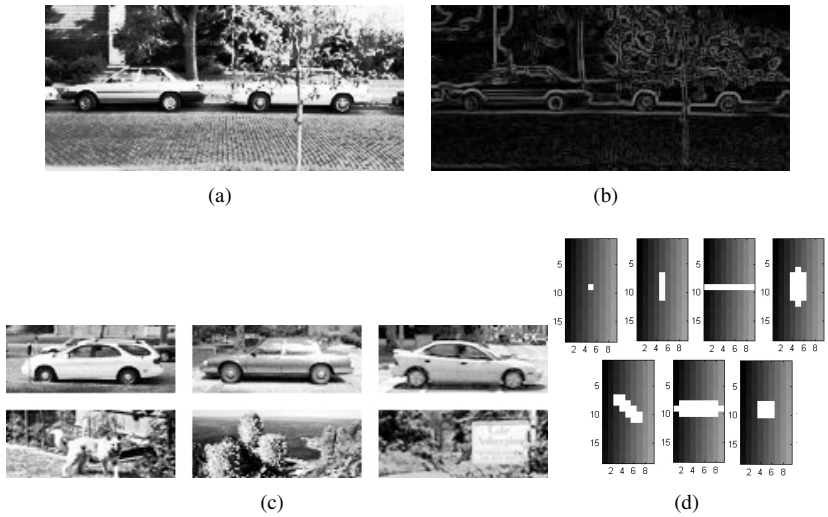


Figure 5: Figure 5(a) shows one example of the original image and Figure 5(b) its corresponding data sample computed by the Sobel filter response. 5(c) shows positive and negative examples in the UIUC Image Database for Car Detection. 5(d) shows the masks selected for building the neighborhoods

The selection of the best spatial neighborhood of the fuzzy learners follows the procedure of Section 2.2.1, using the masks of Figure 5(d). The a priori information considered to reduce the number of masks for searching is the spatial support of the gradient response. In the case of car images, the line-based shape captures the neighborhood information of the data samples. Thus, we consider a group of 7 masks, which are comprised mainly of horizontal and vertical bars obtained from an elliptical shape, as described in 2.2.1.

We compare the performance of the four types of weak classifiers by computing the Receiver Operating Characteristic (ROC) curve. We choose the ROC curve because it is a good evaluation criterion on binary problems where the number of samples is

highly biased to one of the classes, like the image databases utilized on this work. Figure 6(a) shows the curves of the four classifiers, where the fuzzy decision stumps perform better than the decision stumps, and the fuzzy decision trees better than the decision trees.

4.1.2 Faces detection

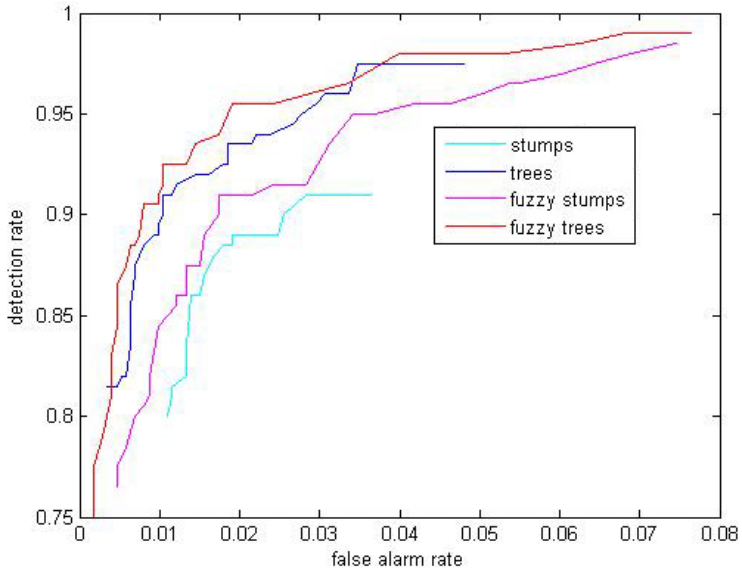
The CBCL MIT face data set [18] is composed of a total of 24,045 test images, 472 faces and 23,573 non-faces. The training data has 2,429 faces and 4,548 non-faces and all the images have 19x19 pixel and are grayscale. In Figure 7(a) are plotted 8 positive examples in the top two rows and 8 negative examples at the bottom rows.

The selection of the best spatial neighborhood of the fuzzy learners follows the procedure of Section 2.2.1, using the masks of Figure 7. In the case of the faces, rectangular and line shaped masks capture the facial landmarks such as eyes, eyebrows, nostrils and mouth corners. Figure 8(a) shows the ROC curves of the four classifiers considered, which have a behaviour similar to the previous section, the car detection problem. Thus, the fuzzy learners bring better performance for both decision trees and decision stumps.

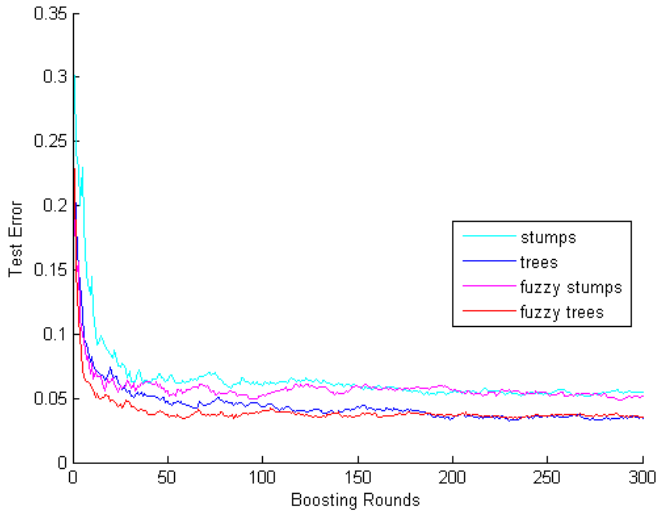
5 Conclusion

We present a new boosting method that introduces a fuzzy function in decision stumps, which can be applied to several classification problems. The properties brought by the fuzzy stump are based on the notion of neighborhood. The feature set that represents the neighborhood is included explicitly in the expression of the weak classifier. This characteristic allows us to apply the fuzzy procedure to several problems, given that is possible to define explicitly neighborhood sets in the data. Thus, our base classifier relies on the feature set (neighborhood) to build a confidence measure that combines the response of both branches for each stump, improving the robustness of the decision stump to local perturbations of the data.

We explained how to exploit the advantages of the fuzzy decision stumps in two

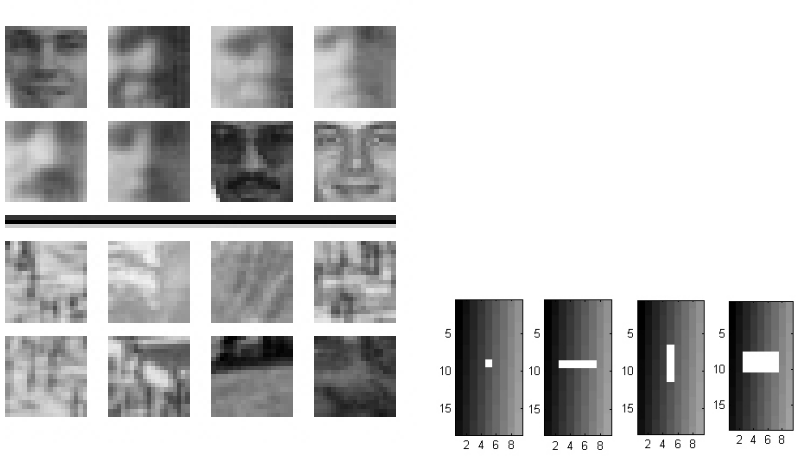


(a)



(b)

Figure 6: Comparison between both decision trees and stumps and its fuzzy versions in the cars database. 6(a) shows the true positives vs. false positives and 6(b) the classification error over the rounds.



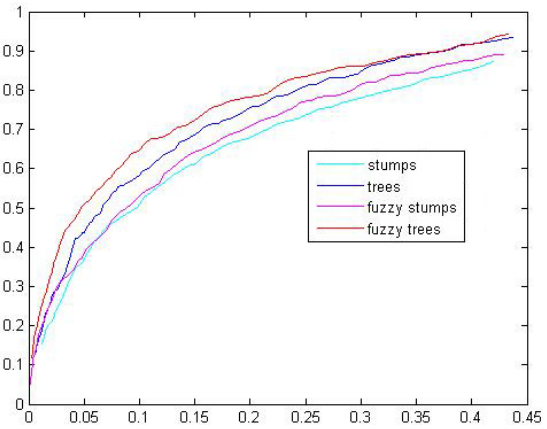
(a) (b)

Figure 7: 7(a) shows positive and negative examples in the CBCL MIT face data. 7(b) shows the masks considered for the space-based fuzzy stump in the face detection application

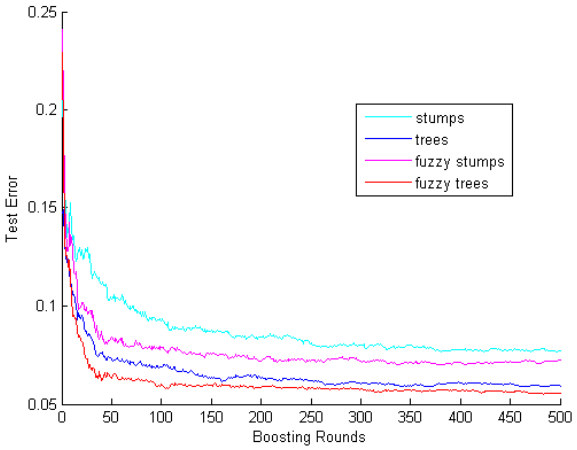
scenarios: time-based neighborhood and space-based neighborhood. For both scenarios, we assess experimentally the advantages of the fuzzy stumps in two different setups:

- Synthetic datasets. This setup allow us to simulate several types of distortions to the data.
- Real datasets in the context of computer vision. We tested the algorithm in four real databases that comprise common tasks in computer vision applications: i) Human activity recognition in video sequences ([16]) and ii) Object detection in images.

The experiments show the appealing nature of this fuzzy procedure due to its better performance and wide range of applicability. However, the question of how to search over all the possible neighborhoods is problem dependent and can surely be further improved. As future work we aim to address two issues related to the neighborhood search: i) Propose efficient algorithms that extract promising neighborhoods and ii) study the spatio-temporal structure to define neighborhoods in that space. Addressing this problems will allow the application of the fuzzy procedure to spatio-temporal data,



(a)



(b)

Figure 8: Comparison between both decision trees and stumps and its fuzzy versions in the face database. 8(a) shows the true positives vs. false positives and 8(b) the classification error over the rounds.

for example to segment objects in video sequences.

References

[1] Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences. 1997;55(1):119–139.

[2] Oza NC, Russell S. Online Bagging and Boosting. In: Jaakkola T, Richardson T, editors. In Artificial Intelligence and Statistics 2001. Morgan Kaufmann; 2001. p. 105–112.

[3] Freund Y. An adaptive version of the boost by majority algorithm. In: COLT '99: Proceedings of the twelfth annual conference on Computational learning theory. New York, NY, USA: ACM; 1999. p. 102–113.

[4] Domingo C, Watanabe O. MadaBoost: A Modification of AdaBoost. In: COLT '00: Proceedings of the Thirteenth Annual Conference on Computational Learning Theory. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2000. p. 180–189.

[5] Bradley JK, Schapire RE. FilterBoost: Regression and Classification on Large Datasets. In: Advances in Neural Information Processing Systems. vol. 20; 2007. p. 185–192.

[6] Luan Y, Li H. Group Additive Regression Models for Genomic Data Analysis. Biostatistics. 2008;9(1):100–113.

[7] Avidan S. SpatialBoost: Adding Spatial Reasoning to AdaBoost. In: In Proc. European Conf. on Computer Vision; 2006. p. 386–396.

[8] Smith P, da Vitoria Lobo N, Shah M. TemporalBoost for event recognition. In: International Conference on Computer Vision. vol. 1; 2005. p. 733– 740.

[9] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). Annals of Statistics. 2000;28(2):337–407.

[10] Park JH, Reddy CK. Scale-Space Based Weak Regressors for Boosting. In: ECML '07: Proceedings of the 18th European conference on Machine Learning. Berlin, Heidelberg: Springer-Verlag; 2007. p. 666–673.

[11] Janikow CZ. Fuzzy decision trees: issues and methods. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on. 1998 Feb;28(1):1–14.

[12] Jang JSR. Structure determination in fuzzy modeling: a fuzzy CART approach. In: Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference on; 1994. p. 480–485 vol.1.

[13] Suarez A, Lutsko JF. Globally optimal fuzzy decision trees for classification and regression. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 1999 Dec;21(12):1297–1311.

[14] Olaru C, Wehenkel L. A complete fuzzy decision tree technique. Fuzzy Sets and Systems. 2003;138(2):221–254.

[15] Torralba A, Murphy KP, Freeman WT. Sharing visual features for multiclass and multiview object detection. IEEE Transactions On Pattern Analysis and Machine Intelligence. 2007;29(5):854–869.

[16] Ribeiro PC, Moreno P, Santos-Victor J. Boosting with Temporal Consistent Learners: An Application to Human Activity Recognition. In: Proc. of 3rd International Symposium on Visual Computing; 2007. p. 464–475.

[17] Agarwal S, Awan A, Roth D. UIUC Image Database for Car Detection;. Available from: `http://l2r.cs.uiuc.edu/~cogcomp/Data/Car/`.

[18] Biological MCF, Learning C. CBCL Face Database #1;. Available from: `http://cbcl.mit.edu/projects/cbcl/software-datasets/FaceData2.html`.