ESCOLA TÈCNICA SUPERIOR D'ENGINYERIES
DEPARTAMENT DE CIÈNCIES DE LA COMPUTACIÓ

Fast traffic sign model matching and recognition on gray-scale images

Memòria del Treball Experimental presentat per en **Sergio Escalera Guerrero** i dirigit per **Petia Ivanova Radeva** dins del programa de Doctorat en Informàtica, opció **Visió per Computador**, de la Universitat Autònoma de Barcelona.

La Dra. Petia Ivanova Radeva, del Departament de Ciències de la Computació de la Universitat Autònoma de Barcelona,

CERTIFICA

que la present memòria ha estat realitzada sota la seva direcció per en **Sergio Escalera Guerrero** i constitueix el Treball Experimental del Programa de Tercer Cicle d'Informàtica, opció Visió per Computador.

Bellaterra, July, 2005

Petia Ivanova Radeva

RESUM

El reconeixement de senyals de trànsit és un treball sobre el qual s'ha treballat en els últims anys sense obtenir resultats suficientment robustos. Aquesta aplicació és molt adequada i representativa per millorar i testejar els algorismes de reconeixement d'objectes. A partir d'imatges rebudes del projecte Geomobil de l'Institut Cartogràfic de Catalunya, un procès de detecció per Adaboost ens dóna unes regions d'interès que contenen un senyal. El nostre treball consisteix en fer la normalització espacial per capturar el model y fer la consequent classificació. Estudiem diferents mètodes per fer la captura de diferents formes geomètriques de senyals i fem un ampli estudi dels mètodes de classificació. Analitzem desde els mètodes més clàssics com els k Veïns Més Propers, la Distància Tangent, l'Anàlisi dels Components Principals, l'Anàlisi Discriminant de Fisher o els Support Vector Machines, comparant-los amb diferents variants dels mètodes més actuales de Boostings, com les característiques rectangulars, el Mostreig, Naive Boosting o la Compartició de Característiques. El nostre estudi ha donat resultats suficientment robustos que permetran la creació de mapes cartogràfics

Paraules clau: Reconeixement de senyals de trànsit, Mobile Mapping, Captura del model, Classificació.

ABSTRACT

Road sign recognition is a computer vision problem on which has worked at last years without obtaining results robust enough. This application is very adapted and representative to improve and to test the object recognition algorithms. From images received from Institut Cartogràfic de Catalunya, processed by a detection procedure based on Adaboost, we receive regions of interest that contain a sign. Our work consists on spatial normalization to fit the sign and do the consequent classification. We study different methods to fit the sign for different geometric forms of the models and to analyse a complete set of classification techniques. We compare the classic methods as k-Nearest Neighbor, Tangent distance, Principal Components Analysis, Fisher Linear Discriminant Analysis or Support Vector Machines, comparing them to different variants of the most novel Boosting techniques, as Rectangular Features, Sampling, Naive Boosting or Share Features. Our work obtains robust results that will be used to create cartographic maps.

Keywords: Road sign recognition, Mobile Mapping, Model Matching, Classification.

Contents

List of figures

1.Introduction

In this document we have elaborated a pattern recognition system to a complex real method: the recognition of traffic sign images acquired by a Mobile Mapping System. The analysis of traffic sign images is a difficult problem given the low quality of the images: lack of visibility during image acquisition, different point of view of the camera, variable illumination conditions, etc. To cope with this difficult problem we need elaborate and study different spatial normalization and pattern classification methods.

1.1 Road sign recognition

Road sign recognition research has been around since the mid 1980's. High variance of sign appearance has made the detection and recognition of road signs a computer vision problem over which many studies have lately been performed. Road signs use particular colors and geometric shapes to attract drivers' attention. However, the difficulty in recognising road signs is largely due to the following reasons: (1) Colors may fade after long exposure to the sun. Moreover, paint may even flake or peel off, and signs may get damaged. (2) Air pollution and weather conditions may decrease the visibility of road signs. (3) Outdoor lighting conditions vary from day to night and may affect the apparent colors of road signs. (4) Obstacles, such as trees, poles, buildings, and even vehicles and pedestrians, may occlude or partially occlude road signs. (5) Video images of road signs often suffer from blurring in view that the camcorder is mounted on a moving vehicle.

Most of the related studies on traffic sign detection are applied to autonomous driving or assisted driving due to when drivers get tired, they may not always notice road signs. A stable road sign detection and recognition system is thus desirable to alert the driver to presence of signs. Two potential problems with an automatic road sign detection system are that if it analyses and reports a critical situation too slowly or if it makes errors, then the system would be of little use. Unfortunately, the above difficulties keep bothering researchers. We may appeal to the human visual system for a solution. In our case, the recognition of road signs system will be used to analyse urban images and create cartographic maps.

Pacheco, Batlle, and Cufi proposed adding special color barcodes under road signs to help road sign identification for vision-based systems. However, much time and resources would be expended to replace road signs, making this solution uneconomical.

There are two main approaches in this field, the color-based [1][2][3][4] and the grayscale-based sign recognition [5][6]. First approach allows to eliminate false positives whereas grayscale methods allow to consider the geometry of the model form.

The color-based studies are based on segmentation by thresholding in color space. Colour segmentation is the most common method for the initial detection of signs. Typically, this is based on the assumption that the wavelength arriving at the camera from a traffic sign is invariant to the intensity of incident light. This assumption usually manifests in the statement that HSV (or HSI) space is invariant to lighting conditions. However, the camera image is *not* invariant to changes in the chromaticity of the incident light. Further, as signs fade over time the colour of the signs is not invariant.

Another approach to detection is *a priori* assumptions about image formation. For instance, assuming the road is approximately straight allows large portions of the image to be ignored when looking for signs. Combined with colour segmentation, Hsu and Huang [7] look for signs in only a restricted part of the image. However, such assumptions can break down on curved roads, or with bumps such as speed humps. A more sophisticated approach is to use some form of detection to facilitate scene understanding, and thus eliminate a large region of the image. For example, Piccoli [8] suggests large uniform regions of the image correspond to road and sky, and thus only look alongside the road and below the sky where signs are likely to appear. However, this is inadequate in cluttered road scenes, such as tree-lined streets. He also suggests ignoring one side of the image as relevant signs will only appear on one side. This is not the case, however, on dual carriageways where signs typically appear on both sides of the road.

Ghica study was based almost exclusively on neural networks [9], which are used for image filtering and sign recognition while others as Aoyagi and Asakura used genetic algorithms [10] to detect road signs from gray-level video imagery. Unfortunately, due to the discrete nature of crossover and mutation operators, optimal solutions are not guaranteed. Lalonde and Li [11] reported a color indexing approach to identify road signs, but the computation time will increase greatly in complex traffic scenes. In addition, many other studies on detecting and recognizing road signs by morphological methods and fuzzy reasoning have been reported.

The studies on gray-scale images are based on geometric reasoning, and most of them on the Hough transforms, and usually color is used as a complementary technique to eliminate false positives results of the classification method.

1.2 Mobile Mapping

An emerging solution to the problems faced in modern data collection campaigns is the integration of various navigation and remote sensing technologies together on a common moving platform. These Mobile Mapping Systems (MMS) [12] are capable of providing fast, efficient, cost-effective, and complete data collection for compiling cartographic information from a mobile vehicle. We use the information from a Mobile Mapping process obtained from the GeoMobil project in order to extract cartographic information. The *Institut Cartogràfic de Catalunya*'s GeoMobil project vehicle [13] includes an image capture subsystem based on a pair of digital cameras of 1024 x 1024 pixels (Fig. 1). The captured information is previously filtered by a heterogeneous cascade classifier system formed by a rectangle features cascade in the detection step.

For the problem of recognition of road signs, we have compared different classification methods. Recently, boosting algorithms as part of multiclassifier methods have achieved a lot of popularity showing their strong advantage compared to the classical pattern classification algorithms. Boosting is a general method for improving the accuracy of any given learning algorithm. The boosting algorithm calls this “weak” or “base” learning algorithm repeatedly, each time feeding it a different subset of the training examples (or, to be more precise, a different distribution or weighting over the training examples). Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that, hopefully, will be much more accurate than any one of the weak rules. We applied extensive comparison of different popular boosting methods to the classical classification methods as Nearest Neighbor, Principal Component Analysis, Fisher Linear Discriminant Analysis, Tangent Distance or Support Vector Machines.

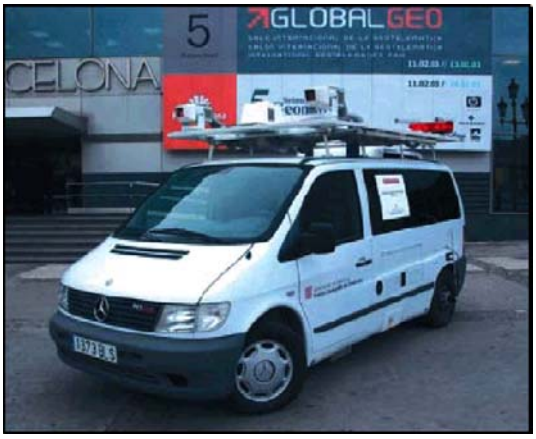


Figure 1. GeoMobil

With the purpose to perform a better approach to the recognition, we divide the problem of sign recognition in three phases: detection, model matching and recognition.

1.3 Detection by Adaboost

The Adaboost algorithm presents a general framework to combine classifiers in order to solve the supervised pattern recognition problem. This approach consists of a) choosing a (weak) classifier, b) modifying example weights in order to give priority to examples where the previous classifiers fail, and c) combining classifiers in a multiple classifier. As a result, each state of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. The input images of our procedure are provided by the weak classifiers cascade detection process as road signs [14].

1.4 Model matching

Given an Adaboost image, it determines a region of interest (ROI) that contains a sign. However, besides the ROI we miss information about scale and position, so before applying recognition we need to apply a spatial normalization. Concerned with the correlation of sign distortion, we look for affine transformations that can perform the spatial normalization to improve final recognition.

In this document, we show different spatial normalization techniques on greyscale images depending on the different received types of road signs to fit.

1.5 Classification

With the purpose of obtaining a good classification performance for the road sign recognition problem, we study different classification techniques, from Similarity Maximization, to Probabilistic and Geometric Methods, including k-Nearest Neighbour, Tangent Distance, Principal Components Analysis, Fisher Linear Discriminant Analysis, Support Vector Machines, and different Boosting variants as Boosting Sampling, Boosting with Rectangular Features, Boosting by Sharing Features (Joint Boosting) or Naive Boosting, comparing the results of the classical and the present and novel methods of classification.

Our main target is to recognize, using the received regions of interest from the detector, the different types of signs. For this task, we will use different methods for the spatial normalization on grayscale images, to allow the capture of the signs, and we will analyse different classification methods, from the classic ones to the most novel ones, with the purpose of obtaining a robust system to recognise road signs. This system will help on the task to create cartographic maps.

2-Methodology to traffic sign recognition

For the problem of road sign recognition, we divide the problem in two parts: model matching and classification. The model matching will be solved by geometry reasoning, studying algorithms to capture the model focusing on the geometry of the different types of signs. For the problem of classification we study different classification algorithms, from the classics to the novel ones, to compare the reliability of the methods in a particular object recognition problem, the traffic sign recognition.

2.1 Pattern recognition techniques used in the road sign recognition system

Pattern recognition is the problem of identifying small, meaningful patterns in a big pattern. Pattern classification is interested in labeling these small patterns, and is an important part of the pattern recognition scheme. A learning process is done by processing the images, extracting important features of the image according to the algorithm in implementation at the time. There has been several techniques suggested and used for machine learning. Although it has been long debated whether these are really learning or not in the philosophical sense, the computer scientists agreed to use this terminology [15].

Statistical pattern recognition systems possess the structure given in Fig 2. This figure illustrates the ”modes” of he recognition system: training and testing. The preprocessing unit gets the input data, segments it into meaningful parts, and eliminates noise. When training, the feature extracting unit gathers suitable features, which will help the classifier to partition the feature space the best way possible. The learning module is fed with the output of this unit in order to train the classifier unit. The training may take several iterations to optimize the selected features. When testing, upon receiving the measured features, the classifier identifies the pattern classes.

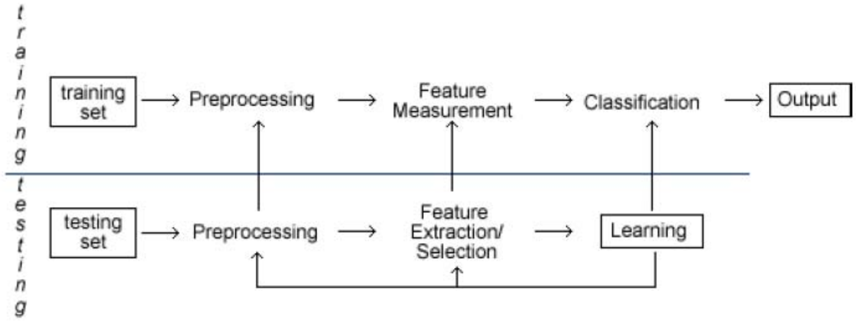


Figure 2: The components of a statistical pattern recognition system.

The classification component here is of special interest, because its success, i.e. the ability to correctly classify the test instances, depends mostly on the output of the feature extraction algorithm. Currently, there are no feature extraction algorithms that perform well globally. That is, the quality of the extracted features depends on the input data. Classification aims to group objects in a category, based on the variance in their feature values. Objects in the same category inhibit similar feature values; while for the ones in distinct categories, the similarity decreases. The classifiers are designed to handle several discrepancies in feature sets, yet each has its pros and cons when dealing with specific feature sets.

2.2 Pattern Classification

There are mainly three classes of classifiers. Each has its advantages, though the outcoming result depends mostly on the training set, and the feature selection algorithm. These classes consist of classifiers that depend on similarity maximization methods, probabilistic methods, and geometric methods, respectively.

2.2.1 Similarity Maximization Methods

The first class of classifiers uses similarity metrics and assigns class labels for maximising the similarity between patterns, deciding a good classification. The question is how to define similarity. The nearest mean classifiers define the features of a class as a vector and represent the class with the mean of the elements of this vector. Thus, any unlabeled vector of features will be classified as the class with nearest mean value.

Template matching uses a template for defining class labels, and tries to find the most similar template for classification. However, there are problems with this approach. For instance, if we want to classify face images, we need to supply a template for each face label. Also, scaling affects the matching, with improper data, the algorithm might fail to produce good results.

Another important classifier of this type uses the Nearest Neighbor (NN) algorithm. The data is represented as points in space, and classification is done based on a some kind of distance (ie. The Euclidean one) For the k-NN, the classifier checks the k nearest points and decides in favor of the majority.

2.2.2 Probabilistic Methods

Probabilistic methods depend on the prior probabilities of classes and class-conditional densities of the instances. Bayesian and logistic classifiers belong to this type of classifiers. The logistic classifiers deal with unknown parameters based on the maximum-likelihood.

The most well known of probabilistic methods makes use of Bayesian Decision Theory. The decision rule assigns class labels to that having the maximum posterior probability. The posterior can be calculated by the well-known Bayes rule:

$$posterior = \frac{likelihood \times prior}{evidence} \qquad (1)$$

Let's define the variables. w = w_i is the state of the nature, i.e instance belongs to class i. Hence P(w_i) is the prior probability that the instance belongs to class i. p(x|w_i) is the class-conditional probability density function: the density for x given that the instance is of class i. Finally, p(x) is defined as Σp(x|W_j)*P(w_j) over all classes.
Given the definitions, equation (1) is equivalent to

$$P(w_j \mid x) = \frac{P(x \mid w_j) \times P(w_j)}{P(x)} \qquad (2)$$

The classification is done in favor of the jth class, if P(w_j |x) > P(w_i|x) ∀ $w_i \in$ C, $w_i \neq w_j$, where C is the set of classes ($w_j \in$ C).
Naive Bayes, a.k.a Idiot Bayes, uses Bayesian Decision Theory, while presuming a conditional independence, rather "naively". For instance, with the naive model, we can calculate the full joint distribution of causes and effects as

$$P(c, e_1, e_2, ..., e_n) = P(c)\prod_i P(e_i \mid c) \qquad (3)$$

where c is the cause, and e_i are the conditionally independent effects (i = 1..n).
Finally, Bayesian Belief Nets represent the functional dependencies and independencies among model variables, i.e. features. Whenever some parameters take some values, the nodes of the network are affected and take a probability value, by the Bayes' rule.

2.2.3 Geometric methods

Geometric classifiers build decision boundaries by directly minimizing the error criterion, since no related experiments are supplied. An example to these classifiers is Fisher's linear discriminant, which mainly aim to reduce the size of the feature space to lower dimensions in case of a huge number of features. It minimizes the mean squared error between the class labels and the tested instance. Neural networks are also examples of geometric classifiers.
The decision trees take the instance described by its features as input, and outputs a decision, denoting the class information in our case. Each node denotes a feature, and

each iteration we go down to the lower depth, selecting a child node depending on the feature value for the particular instance.

The Pattern Classification methods we analyse in this document are the following:

2.3 Similarity Maximization Methods

2.3.1 K-Nearest Neighbor

K-Nearest Neighbour can be useful for as due to the model matching of the sign allow us the use of the correlation scheme.
One of the classic classification methods applied when no information about data distribution is available is the nearest neighbour classification in the image space. Under this scheme, an image in the test set is recognized by assigning to it the label of most of the closest points in the learning set. If all images are normalized to have zero mean and unit variance, then this procedure is equivalent to choosing the image in the learning set that best correlates with the test image. Because of the normalization process, the result is independent of light source intensity and the effects of a video camera automatic gain control.

2.3.2 Tangent Distance

Tangent Distance could be useful in case to have a classification in which we have some slight rigid/affine transformations (as rotation) and we need an invariant measure of the recognition problem.

Tangent distance is a invariant distance measure [16]. Reasonably small transformations of certain image objects do not affect class-membership. When an image $x \in \Re^D$ (seen as a one-dimensional vector here) is transformed (e.g. scaled and rotated) by a transformation $t\{x,\alpha\}$ which depends on *L* parameters $\alpha \in \Re^L$ (e.g. the scaling factor and rotation angle), the set of all transformed patterns

$$M_x = \{t(x,\alpha) : \alpha \in \Re^L \subset \Re^D\} \quad (4)$$

is a manifold of at most dimension *L* in pattern space. The distance between two patterns can now be defined as the minimum distance between their respective manifolds, being truly invariant with respect to the *L* regarded transformations. A manifold can be aproximated by a tangent vector as is shown in fig 3. In fig 3 Top: Representation of the effect of the rotation in pixel space. Middle: Small rotations of an original digitized image of the digit “2", for different angle values of α. Bottom: Images obtained by moving along the tangent to the transformation curve for the same original digitized image P by adding various amounts (α) of the tangent vector T.

The distance between two manifolds approximated by its tangent vectors is shown in fig 4. The fig 4 shows the Euclidean distance and the tangent distance between points P and E. The curves S_p and S_e represent the sets of points obtained by applying the chosen transformations (for example translations and rotations) to P and E. The lines going through P and E represent the tangent to these curves. Assuming that working space has more dimensions than the number of chosen transformations (on the diagram, assume 3D) the tangent spaces do not intersect and the tangent distance is uniquely defined.

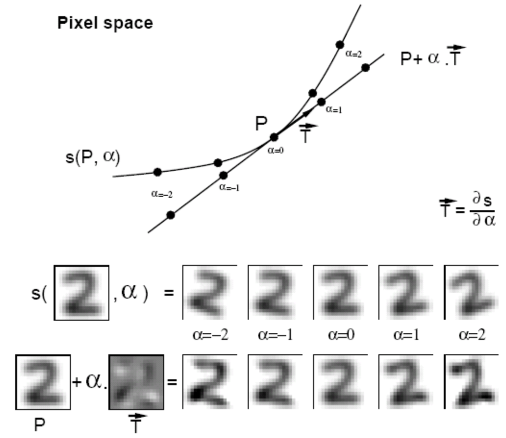


Figure 3. Representation of the effect of the rotation in pixel space

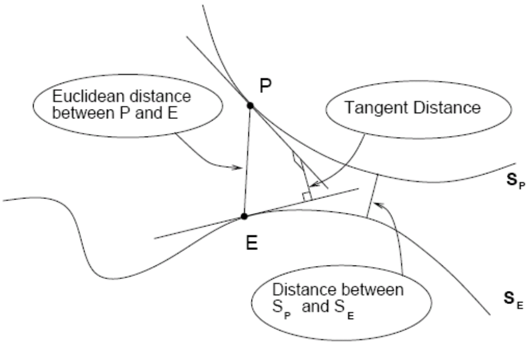

Euclidean distance between P and E
P
Tangent Distance
S_P
E
Distance between S_P and S_E
S_E

Figure 4. Illustration of the Euclidean distance and the tangent distance between P and E.

Some approximations of tangent vectors for different transformations are the following:

X-translation: This transformation is useful when the classification function is known to be invariant with respect to the input transformation:

$$t_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x + \alpha \\ y \end{pmatrix}$$

The Lie operator is defined by:

$$L_X = \frac{\partial}{\partial x}$$

Y-translation:

$$t_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x \\ y + \alpha \end{pmatrix}$$

The Lie operator is defined by:

$$L_Y = \frac{\partial}{\partial y}$$

Rotation:

$$t_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x \cos\alpha - y \sin\alpha \\ x \sin\alpha + y \cos\alpha \end{pmatrix}$$

The Lie operator is defined by:

$$L_R = y\frac{\partial}{\partial x} + (-x)\frac{\partial}{\partial y}$$

Scaling:

$$t_\alpha : \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x + \alpha x \\ y + \alpha y \end{pmatrix}$$

The Lie operator is defined by:

$$L_S = x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}$$

Now, to compute the tangent distance, let the function s transform an image P to s(P, α) according to the parameter α. We require s to be differentiable with respect to α and P, and require s(P, 0) = P. If P is a 2 dimensional image for instance, s(P, α) could be a rotation of P by the angle α. If we are interested in all transformations of images which conserve distances (isometry), s(P, α) would be a rotation by α_θ followed by a translation by α_x, α_y of the image P. In this case $\alpha = (\alpha_1, ..., \alpha_m)$ is a vector of parameters of dimension 3. In general, $\alpha = (\alpha_1, ..., \alpha_m)$ is of dimension m. Since s is differentiable, the set S_P = { x | ∃α for which x = s(P, α)} is a differentiable manifold which can be approximated to the first order by a hyperplane T_P. This hyperplane is tangent to S_P at P and is generated by the columns of matrix

$$L_P = \left.\frac{\partial s(P,\alpha)}{\partial \alpha}\right|_{\alpha=\mathbf{0}} = \left[\frac{\partial s(P,\alpha)}{\partial \alpha_1}, \ldots, \frac{\partial s(P,\alpha)}{\partial \alpha_m}\right]_{\alpha=\mathbf{0}}$$

which are vectors tangent to the manifold. If E and P are two patterns to be compared, the respective tangent planes T_E and T_P can be used to define a new distance D between these two patterns. The tangent distance D(E,P) between E and P is defined by:

$$D(E,P) = \min_{x \in T_E, y \in T_P} ||x - y||^2$$

The equation of the tangent planes T_E and T_P is given by:

$$\begin{gathered} E'(\alpha_E) = E + L_E\alpha_E \quad (5) \\ P'(\alpha_P) = P + L_P\alpha_P \end{gathered}$$

where L_E and L_P are the matrices containing the tangent vectors and the vectors α_E and α_P are the co-ordinates of E′ and P′ (using bases L_E and L_P) in the corresponding tangent planes. Note that E′, E, L_E and α_E denote vectors and matrices in linear equations. For example, if the pixel space was of dimension 5, and there were two tangent vectors, we could rewrite equation (5) as

$$\begin{bmatrix} E'_1 \\ E'_2 \\ E'_3 \\ E'_4 \\ E'_5 \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \end{bmatrix} + \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \\ L_{31} & L_{32} \\ L_{41} & L_{42} \\ L_{51} & L_{52} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

The quantities L_E and L_P are attributes of the patterns so in many cases they can be precomputed and stored. Computing the tangent distance

$$D(E,P) = \min_{\alpha_E,\alpha_P} ||E'(\alpha_E) - P'(\alpha_P)||^2$$

amounts to solving a linear least squares problem. The optimality condition is that the partial derivatives of D(E,P) with respect to α_P and α_E should be zero:

$$\frac{\partial D(E,P)}{\partial \alpha_E} = 2(E'(\alpha_E) - P'(\alpha_P))^\top L_E = 0$$
$$\frac{\partial D(E,P)}{\partial \alpha_P} = 2(P'(\alpha_P) - E'(\alpha_E))^\top L_P = 0$$

Substituting E´ and P´ by their expressions yields to the following linear system of equations, which we must solve for α_P and α_E:

$$L_P^\top(E - P - L_P\alpha_P + L_E\alpha_E) = 0$$
$$L_E^\top(E - P - L_P\alpha_P + L_E\alpha_E) = 0$$

The solution of the system is

$$(L_{PE}L_{EE}^{-1}L_E^\top - L_P^\top)(E-P) = (L_{PE}L_{EE}^{-1}L_{EP} - L_{PP})\alpha_P$$
$$(L_{EP}L_{PP}^{-1}L_P^\top - L_E^\top)(E-P) = (L_{EE} - L_{EP}L_{PP}^{-1}L_{PE})\alpha_E$$

where $L_{EE} = L_E^T L_E$, $L_{PE}=L_P^T L_E$,$L_{EP}=L_E^T L_P$ and $L_{PP}=L_P^T L_P$. The tangent distance is obtained by computing $||E'(\alpha_E)-P'(\alpha_P)||$ using the value of α_P and α_E in equations (X) and (Y).

2.4 Probabilistic and Ensemble Methods

2.4.1 Boosting Rectangular Features

In this case, the set of features are subtractions between regions. In our experiments we use the features shown in fig. 5, that could be extended with their rotated variants. To estimate a feature, given that white part has value 1 and black part has value –1, convolution is applied, resulting in the value of that feature for a given image. To accelerate this operation we use the Integral Image to represent the image. In the Integral Image, each pixel is the accumulative sum of the region from the origin of the image to the estimated pixel (fig. 6a). By this way, to calculate the area in a region D (fig. 6b), the result is 4+1-(2+3) (fig. 6b), doing this operation invariant about the global Brightness [17][18].

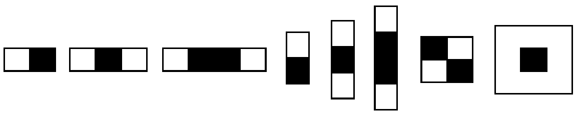
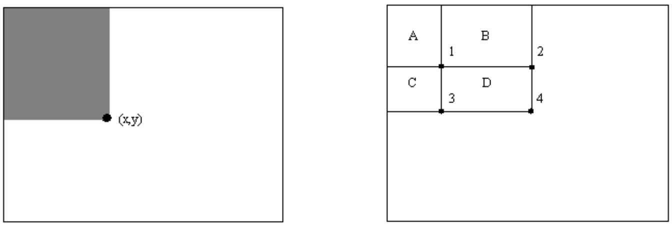


Figure 5. Set of rectangular features.



(a) (b)

Figure 6. (a) Integral Image at point (x,y). (b) Area estimation for region D : 4 + 1 – (2 + 3)

Applying the set of features for different resolutions of the image we have the values to estimate the hypothesis at weak learner for the boosting process.
The variant of Adaboost we use in our experiments for rectangular features is the Real Adaboost. It consists of the algorithm shown in fig. 7. The algorithm is based on selecting at each iteration the best weak classifier that minimizes the error, by calculating the accumulated error of the missclassified examples by calculating a threshold of the results of the rectangular features. After that, we update weights decreasing for the well classified, and repeat the procedure until we have an ensemble classifier to classify well all samples or since we finish all iterations.

```
Given two sets of samples for two classes,
Initialize weights $w_1 = 1/n$, where n is the number of samples.
For t=1,…,T
    o  Select the best weak classifier that minimizes the error
    o  Define the classifier $h_t = \{\alpha, t, f\}$, where $\alpha = \log\left(\frac{1-err_t}{err_t}\right)$
    o  Update the weights $w_{t+1} = w_t B_t^{1-e_i}$, where $e_i$=0 if example $x_i$ is classified correctly or $e_i$=1 otherwise, and
       $B_t = err_t/(1-err_t)$
    o  Normalize the weights $w_{t+1,i} = w_{t,i} \Big/ \sum_{j=1}^{n} w_{t,j}$
The final strong classifier is $C(x) = \begin{cases} 1 & if \ \sum_{t=1}^{T} \alpha_t h_t \geq 0 \\ 0 & if \ \sum_{t=1}^{T} \alpha_t h_t < 0 \end{cases}$
```

Figure 7. Real Adaboost. h_t is the classifier at stage $_t$. *t* is the threshold and *f* is the selected feature.

2.4.2 Joint Boosting

Joint boosting is a fast multi-class boosting procedure, by finding common features that can be shared across the classes (and/or views). The detectors for each class are trained jointly, rather than independently. The features selected jointly are closer to edges and generic features typical of many natural structures instead of finding specific object parts (fig. 8) [19].

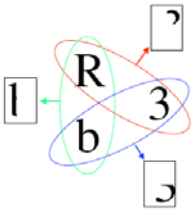


Figure 8. Objects may share features. In this example, we can see how each pair of objects shares a part.

The basic idea of the algorithm is that for each possible subset of classes, we find a feature that is most useful for distinguishing that subset from the rest. We then pick the best such feature/subset, and repeat, until the overall error (across all classes) stops decreasing, or until we reach a limit on the number of features we can afford to compute (to bound the run-time cost). The result is that many fewer features are needed to achieve a desired level of performance than if we were to train the classifiers independently.

For this task, we share weak-learners across classes. For example, if we have 3 classes, we might define the following classifiers:

$$H(v, 1) = G^{1,2,3}(v) + G^{1,2}(v) + G^{1,3}(v) + G^{1}(v)$$
$$H(v, 2) = G^{1,2,3}(v) + G^{1,2}(v) + G^{2,3}(v) + G^{2}(v)$$
$$H(v, 3) = G^{1,2,3}(v) + G^{1,3}(v) + G^{2,3}(v) + G^{3}(v)$$

where each $G^{S(n)}(v)$ is itself an additive model of the form $G^{S(n)}(v)=\sum_{m=1}^{M_n} h_m^n(v)$, where m is the stage and *n* refers to a node in the "sharing graph" (Fig. 9), which specifies which functions can be shared between classifiers. S(n) is the subset of classes that share the node n.

The idea of the algorithm (fig. 10) is that at each boosting round, we examine various subsets of classes, $S \subseteq C$, and considering fitting a weak classifier to distinguish that subset from the rest. We pick the subset that maximally reduces the error on the weighted training set for all the classes. The best weak learner h(v, c) is then added to the strong learners H(v, c) for all the classes $c \in S$, and their weight distributions are updated. v_i^f is the f'th feature of the i'th training example, $z_i^f \in \{-1,+1\}$ are the labels for class c, and w_i^f are the unnormalized example weights. N is the number of training examples, M is the number of rounds of boosting, $\delta(x)$ is the indicator function for regression stump, and n denotes a node in the "sharing graph".

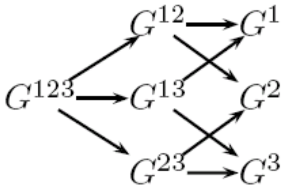


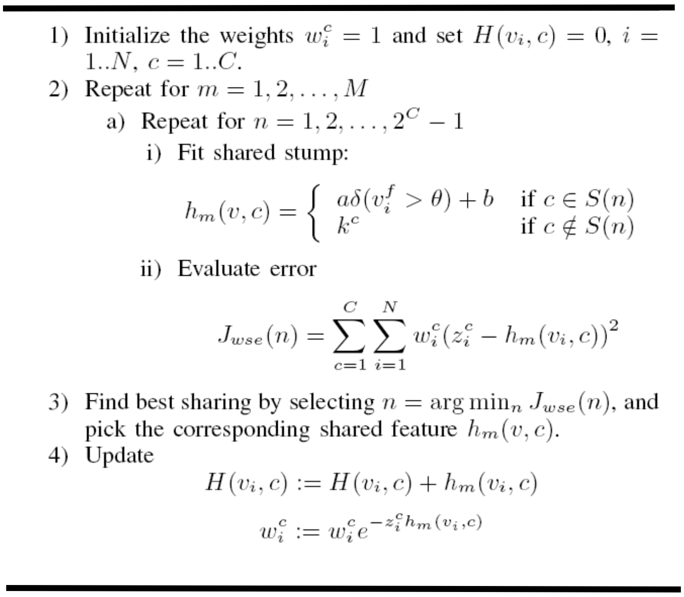
Figure 9.All possible ways to share features amongst 3 classifiers. In this representation, each classifier h(features, class label) is constructed by adding, only once, all the nodes that connect to each of the leaves. The leaves correspond to single classes. Each node corresponds to a different grouping of classes.

The parameters for the regression stumps, shown in the algorithm, calculated by Weighted Least Squares given as follows:

$$b = \frac{\sum_{c \in S(n)} \sum_i w_i^c z_i^c \delta(v_i^f \leq \theta)}{\sum_{c \in S(n)} \sum_i w_i^c \delta(v_i^f \leq \theta)}$$

$$a + b = \frac{\sum_{c \in S(n)} \sum_i w_i^c z_i^c \delta(v_i^f > \theta)}{\sum_{c \in S(n)} \sum_i w_i^c \delta(v_i^f > \theta)} \qquad (6)$$

$$k^c = \frac{\sum_i w_i^c z_i^c}{\sum_i w_i^c} \quad c \notin S(n)$$



```
1) Initialize the weights $w_i^c = 1$ and set $H(v_i, c) = 0$, $i = 1..N$, $c = 1..C$.
2) Repeat for $m = 1, 2, \ldots, M$
    a) Repeat for $n = 1, 2, \ldots, 2^C - 1$
        i) Fit shared stump:
            $h_m(v, c) = \begin{cases} a\delta(v_i^f > \theta) + b & \text{if } c \in S(n) \\ k^c & \text{if } c \notin S(n) \end{cases}$
        ii) Evaluate error
            $J_{wse}(n) = \sum_{c=1}^{C} \sum_{i=1}^{N} w_i^c (z_i^c - h_m(v_i, c))^2$
3) Find best sharing by selecting $n = \arg\min_n J_{wse}(n)$, and
   pick the corresponding shared feature $h_m(v, c)$.
4) Update
        $H(v_i, c) := H(v_i, c) + h_m(v_i, c)$
        $w_i^c := w_i^c e^{-z_i^c h_m(v_i, c)}$
```

Figure 10. Joint boosting with regression stumps.

2.4.3 Naive Boosting

In some cases, the similarity between classes is too high, and some classification methods accumulate errors classifying incorrectly. One possible solution is to train by boosting, to select only the best features that discriminate classes. In addition, small displacements and different sizes of the image can affect us the classification result. One possible solution is to train the boosting using the features extracted from Naive Bayes probability matrix. We use this process for binary speed signals.

Given a set of examples, the Naive Bayes probability matrix contains at each pixel, the appearance probability of this pixel from the set of samples of the class. To obtain a reliable approach, we have to additionally calculate another matrix with the probability of not containing that pixel. In fig. 11 we can see the probability matrix of containing salient pixels of the speed signs.

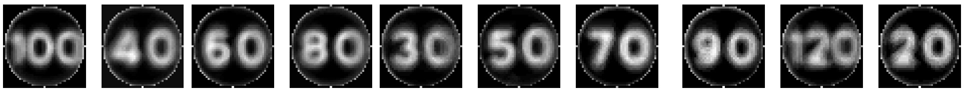


Figure 11. Probabilies Matrix of appearance for speed signs.

Given an input image and the probability functions for two classes, in order to classify the image we estimate (7), where x_d is the value of pixel d and p_d-q_d are the probability to appear and not to appear of that pixel respectively. We label for the class with higher probability

$$\sum_{d=1}^{D}(x_d \log p_d + (1-x_d)\log q_d) > \sum_{d=1}^{D}(x_d \log \tilde{p}_d + (1-x_d)\log \tilde{q}_d) \qquad (7)$$

To perform the classification of a given input image to classify, we use the chamfer distance to fit the image to the models probability image. First, we binarize the input image and we apply different transformations of scalling, translating and rotating, to find the transformation that best fits with the distance map of a given probabilities matrix (fig. 12).

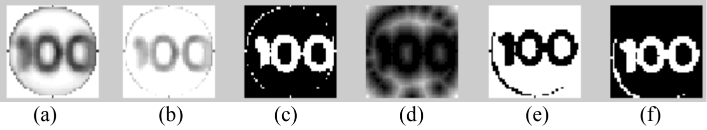


Figure 12. (a) Probabilities image. (b) Brightness correction. (c) Otsu binarization. (d) Distance Map. (e). Image to adjust. (f). Image fit to model.

Applying the previous method directly is not robust enough. We can boost the process using so many features as pixels the image has. By this way we select the features best discriminant between classes.

2.4.4 Boosting Sampling

This technique selects for each iteration of the boosting algorithm a subset of samples related to the worst classified images of the training set at previous iterations. We have used a weak learner Fisher Linear Discriminant Analysis with previously applying Principal Components Analysis.

2.5 Geometric methods

2.5.1 Principal Components Analysis

Principal Component Analysis, from a statistical perspective, is a method for transforming correlated variables into uncorrelated variables, finding linear combinations of the original variables with relatively large or small variability, and reducing data [20].

Given the set of N column vectors { $\vec{x}_i$ } of dimension D, the mean of the data is:

$$\vec{\mu}_x = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i$$

The scatering total matrix is defined as:

$$S_T = \frac{1}{N}\sum_{i=1}^{N}(\vec{x}_i - \vec{\mu})(\vec{x}_i - \vec{\mu})^T$$

We choose the eigenvectors of S_T that correnpond to the X % largest eigenvalues of S_T to compute Wpca, obtaining a transformation $X^n \rightarrow Y^m$, reducing data, of the form:

$$Y = Wpca(X - \overline{X})$$

Where X is the sample to project and $\overline{X}$ is the mean.

2.5.2 Fisher Linear Discriminant Analysis

Given the binary classification problem, Fisher projects at one dimension each pair of classes (reducing to C-1 where C is the number of classes), multiplying each sample by its projection matrix, which minimize the distance between samples of the same class, and maximizes the distance between the two classes. The result is shown in fig 13, where the blue and red points belong to the samples of the two projected classes, and the green line indicates the theshold that best separates them [21].

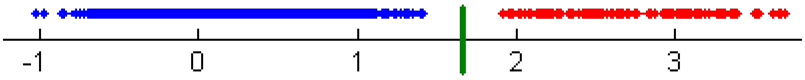


Figure 13: Fisher projection for two classes and threshold value.

The algorithm is:

Given the set of N column vectors $\{\vec{x}_i\}$ of dimension D, we calcule the mean of the data. For K classes $\{C_1,C_2,\ldots,C_k\}$, the mean of the class k that contains N_k elements is:

$$\vec{\mu}_{xk} = \frac{1}{N_k}\sum_{\vec{x}_i \in C_k} \vec{x}_i$$

The separability maximization between classes will be defined as the quotient between the scatter matrix between-class:

$$S_b = \sum_{k=1}^{K} N_k(\vec{\mu}_{xk} - \vec{\mu}_x)(\vec{\mu}_{xk} - \vec{\mu}_x)^T$$

and the scatter matrix intra-class:

$$S_w = \sum_{k=1}^{K}\sum_{\vec{x}_i \in C_k}(\vec{x}_i - \vec{\mu}_{xk})(\vec{x}_i - \vec{\mu}_{xk})^T$$

obtaining a projection that define an optimal discriminant features.

The projection matrix W maximizes:

$$\frac{\det(W^T S_B W)}{\det(W^T S_w W)} \qquad (8)$$

Let $\{\vec{w}_1, \vec{w}_s, ..., \vec{w}_D\}$ be the generalized eigenvectors of S_B and S_W. Then, selecting the d<D that corresponds to the highest eigenvalue, we have the projection matrix W=$\left[\vec{W}_1, \vec{W}_2, ..., \vec{W}_d\right]$, project the samples to the new space by using:

$$\vec{y} = W_d^T \vec{x}$$

The generalized eigenvectors of (8) are the eigenvectors of $S_B S_W^{-1}$.

2.5.3 Support Vector Machines

The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given the attributes [22].

Given a training set of instance-label pairs $(x_i, y_i), i = 1,...,l$ where $x_i \in R^n$ and $y \in \{1,-1\}^l$, the support vector machines require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

Subject to

$$y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i,$$
$$\xi_i \geq 0$$

Here training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function ϕ. Then SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. We can define, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ called the kernel function. Though new kernels are being proposed by researchers, the most common four basic kernels are:

Linear: $K(x_i, x_j) = x_i^T x_j$

Polynomial: $K(x_i, x_j) = \left(\gamma x_i^T x_j + r\right)^d, \gamma > 0$

Radial basis function (RBF): $K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|^2\right), \gamma > 0$

Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Here, γ, *r*, and *d* are kernel parameters.

3. Our system

Given a region of interest from the detector which a high probability to contain a sign, we divide the problem of recognition of traffic signs in spatial normalization and classification of traffic signs.

3.1 Circular signs

The first type of signs we studied are the circular signs. We used their geometry properties in order to capture the sign. First, we considered morphological operators to clear the content of it: the first procedure is a morphological closing, followed by an isotropic Perona-Malik filter, canny contours detector, and distance map, with the purpose of detecting the center of the sign. This process is shown in Fig. 14.

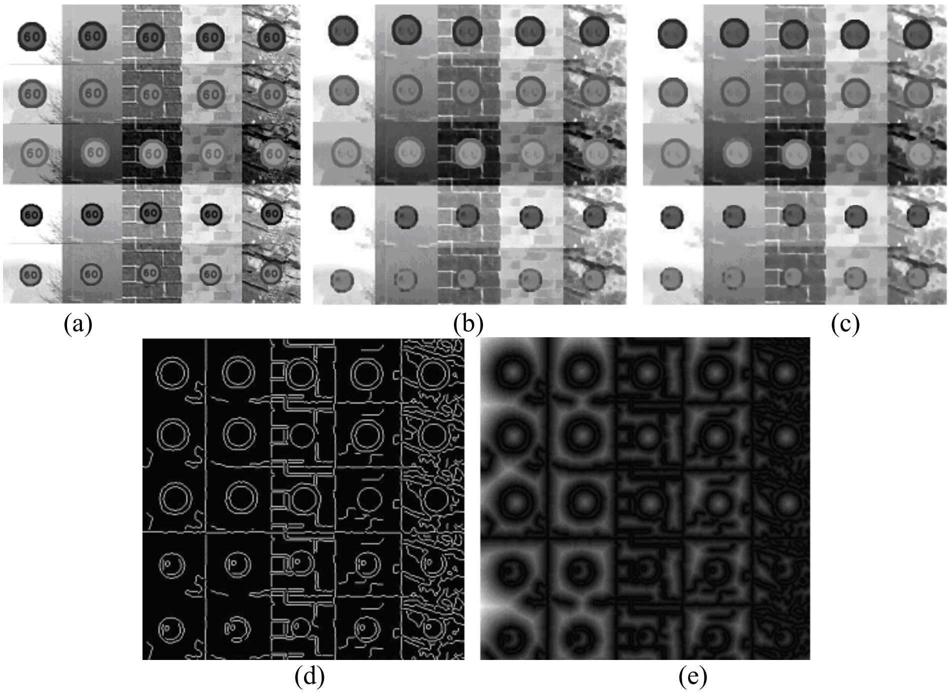


Fig 14. (a) Input image. (b) Closing. (c) Anisotropic Perona-Malik filter. (d) Canny contours. (e) Distance Map

In some cases, the previously mentioned techniques allowed to find the center of the sign, but morphological operators do not assure the elimination of its content, and in most cases we can loose part of its contents due to its size and the application of a high structure element for morphological closing.

3.1.1 False contours extraction

After that, we considered the contours of the image to discriminate them in order to perform the localization of the sign. The first method applied was 'False contours extraction'. First, Canny contours detector is applied followed by the suppression of unrelevant contours by model fitting. The main idea of the method is that in a contour map extracted from the reduced ROI where the sign is contained, the length of continuous sign contour extracted from Canny detector should be larger compared to the false contours contained. The method selects relevant contours discriminating them by their length. Fig 15 shows the result of the method application for three different preprocessed signs.

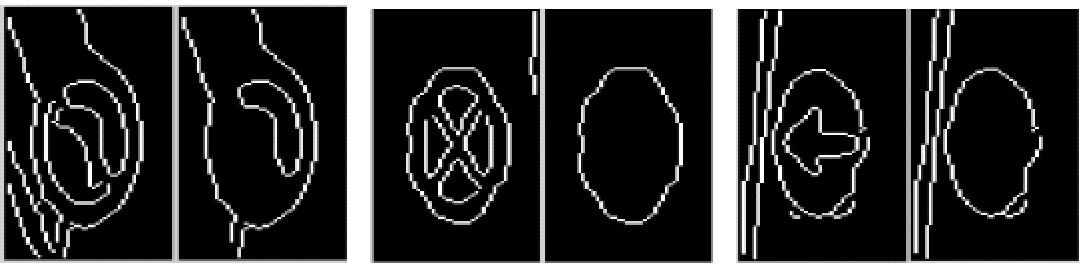
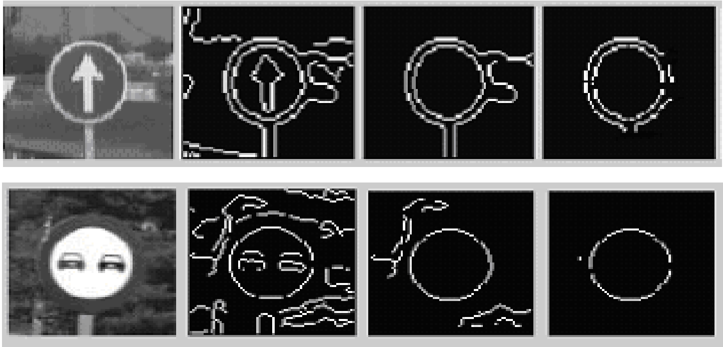


Figure 15. Three different eliminations of false contours.

3.1.2 Symmetry properties

For road sign detection problem we can use the symmetry properties of certain signs to detect points of interest, having a symmetrical arrangement of radiating parts about a central point, or using image gradient direction convergence to locate points of high radial symmetry. We receive a reduced image where the sign is almost centered, so, for the detection of ellipsoidal signs we can use their property of symmetry. The remaining false contours of the first method could be eliminated by a simple horizontal and vertical symmetry intersection from the central point. The process is based on maintaining the points of contour located in two regions to the same distance with respect to a center line. With this fact, we maintain the points of interest and suppress the non-symmetrical points in the image. Fig 16 shows the result of the method application for different input signs.

Applying the two previous methods, now we will be able to conserve the contour points that correspond to the contour of the containing sign, so we apply now a method to fit an ellipse.



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

Figure 16. (a) Input image. (b) Canny contours. (c) False contours extraction of length inferior to 50 pixels. (d) Horitzontal and vertical symmetry.

3.1.3 Ellipse fitting

The idea of the method is to use all pixels from the image to search for the ellipse that best fits to the image points [23]. All input image contour points are considered in the fitting process. The objective is to find the parameters of the following function:

$$F(x,y)=ax^2+bxy+cy^2+dx+ey+f$$

so that they define an ellipse. First, we construct the design matrix D that has as many rows as number of contour points are considered in the ellipse fit process, and each row of six elements has the form $[x_ix_i \; x_iy_i \; y_iy_i \; x_i \; y_i \; 1]$, where (x_i,y_i) represents the coordinates of each contour point. The following step is to construct the scatter matrix:

$$S=D^TD$$

From the constraint matrix $C_{6x6:}$

$$\begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

introducing the Lagrange multiplier λ, and considering the vector a=[a b c d e f], we can find the fit ellipse parameters solving the following system:

$$Sa=\lambda Ca \quad (9)$$
$$a^TCa=1$$

This system is readely solved by considering the generalized eigenvectors of (9). The ellipse fit parameters correspond to the six elements from the only negative eigenvalue. As a result of this method, we capture the sign and correct sign distortion in order to proceed with the recognition phase. In fig 17 an exemple for ellipse fitting is shown for a given handwritten ellipse as input, and in figure 18 the method is applied for different circular signs.

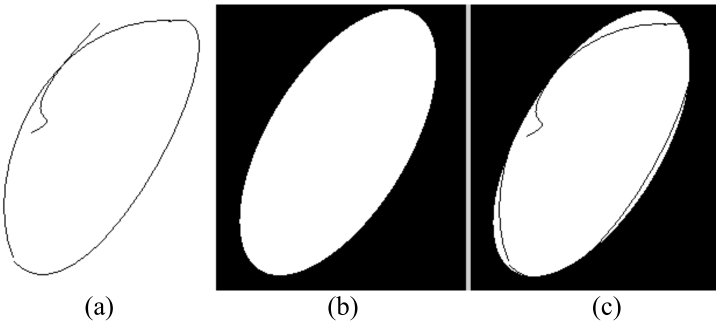


Figure 17. (a) Input contour points (b) Fitted ellipse (c) Mask applied to the input image.

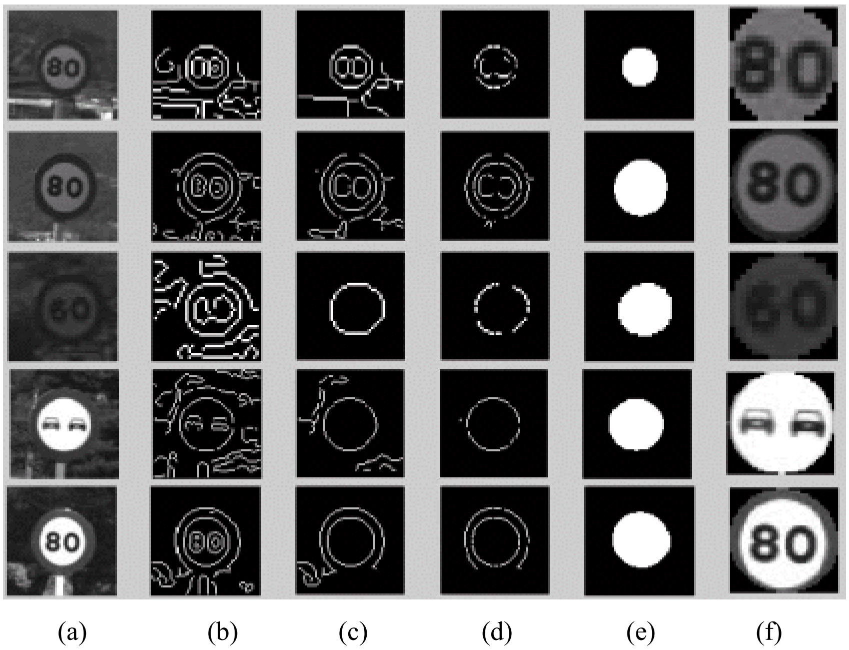


Fig. 18 (a) Input image. (b) Canny contours. (c) False contours extraction. (d) Horitzontal-Vertical Symmetry. (e) Ellipse Fitting. (f) Extracted sign

3.1.4 First database and virtual samples

In order to check the implemented methods robustness, we designed a first image database with 21 circular classes, which representative model is shown in fig 19.

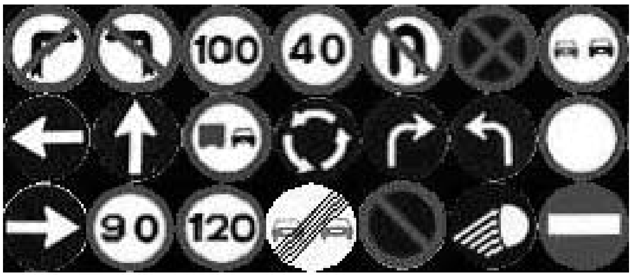


Figure 19. Database image classes.

Each of the 21 different sign classes contains 140 transformations for 10 different angles rotation, with a total of 2940 images on which we apply 50 tests**.** The samples were generated by supervising the previous commented methods. The transformations applied are: gaussian smoothing for different size and sigma values and mathematic morphology operators. The image size chosen in order to maintain the sign characteristics well defined is of 30 by 30 pixels. In fig 20 we see some samples of elements classes and their transformations. By the two possible extractions, we create more elements from each image in order to detect both types. For this task, we included circular signs for each class with edge extractions between 2 and 5 pixels depending on the type of sign, so, it will allow to fit the correlation between pixels for a successful classification independently on the sign extraction. All elements are normalized by equalization from the gray image levels, and a circular mask is applied to do the correlation only with the representative points of circular signs. We use the k-nearest neighbours algorithm as a classification method. From a simple correlation between the equalized input image resized by nearest neighbour method and the database image elements pixels, we match each input image by the class that contains most neighbours of the target object. The error parameter between pixels was selected in order to obtain the maximum difference between classes, being 0.15 in a scale between 0 and 1 gray levels. This margin error allows to grant as success the pixels of similar values that correspond to the same class. This fact does not cause classification errors since the equalization emphasizes the changes of intensity in the different sign regions. From the tests we select k=11 nearest neighbors as the experimental value for the classification.

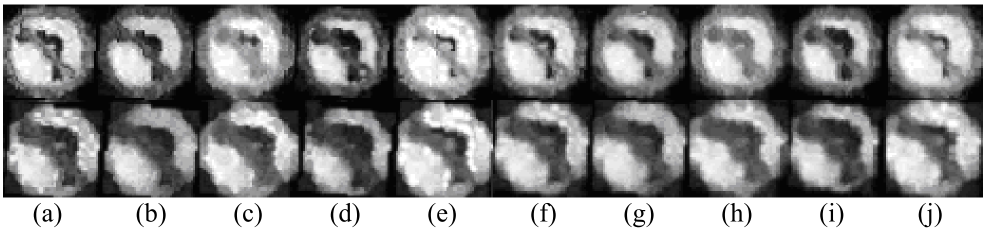


Figure 20. All images are transformations for the same angle. Top images are related to outer contour sign extraction and down images for inner contour sign extraction.
(a) Equalized input. (b) Opening image (c) Closing image (d) Eroded image (e) Dilated image
(f-j) Same transformations for different gaussian parameters.

To test the classification percentatge for the previously explained methods we used k-Nearest Neighbor and the Tangent Distance commented in the previous chapter. For Tangent Distance we calculated tangent vectors for each class to be invariant respect to rotation, x and y traslation and scaling. In fig 21 we show the tangent vectors for a given class and in fig 22 we show the samples generated using the tangent vector for rotation and an input image.

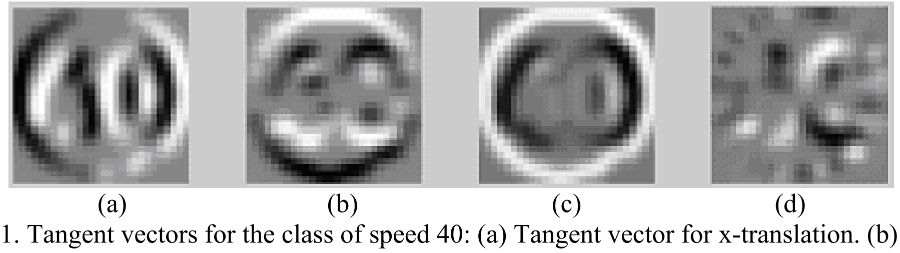


Figure 21. Tangent vectors for the class of speed 40: (a) Tangent vector for x-translation. (b) Tangent vector for y-Translation. (c) Tangent vector for scaling. (d) Tangent vector for rotation.

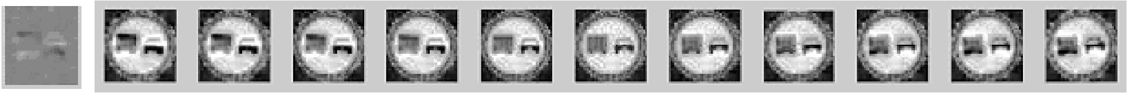


Figure 22. (a) Tangent vector for rotation. (b) Samples generated with rotation tangent vector.

As k-NN is normally used as background to test, and Tangent Distance should perform affine important invariant recognition for our problem, we use them to test the actual spatial normalization techniques and the virtual database.

The 50 tests applied for classification for 3-nearest neighbor obtained a reliability of 92%, while using the tangent distance we had a hit ratio of 78%. Showing the missclass errors, we observed that tangent distance is not robust enough due to the influence of grey levels of the image and all the transformations that the signs can contain. K-nearest neighbour ran better.

The advantages of the commented spatial normalization methods are that false contours extraction and radial symmetry are robust enough since we know a priori the position of minimum perimeter of the sign.

The disadvantage is that in some cases the noise is too high and some remaining pixels of noise can produce a deformation of the fitted ellipse, obtaining a misclassification.

The actual database has an elevated number of samples that can help us to the recognition, but we observed that some of the transformations that we applied to the virtual samples created confusion between classes. At this moment, we decided to study alternative methods for spatial normalization and to extend the data base to optimize the consequent classification.

Using a Fisher Linear Discriminant Analysis projection between the classes of speed 100 and 120 we see that the projection does not separate the classes properly due to the influence of virtual samples (Fig 23).

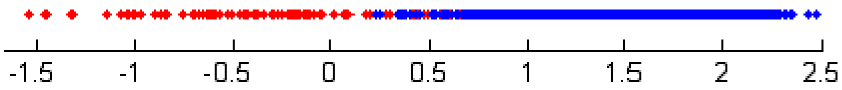


Figure 23. Fisher projection between classes of speed 100 and 120.

After that, we used 10 DVD’s from the ICC GeoMobil project with about 3000 frames each one, to extract samples to create a new real database. The classes of the new database chosen by its amount of appearance are shown in fig 24. The database contained about 3000 real samples.

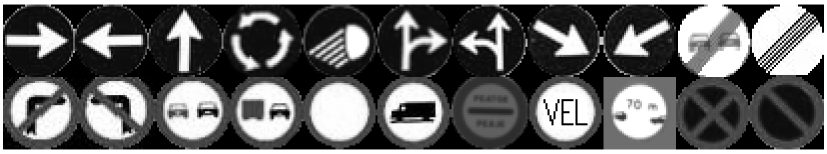


Figure 24. Circular database classes.

The class ‘VEL’ has been created due to the fact that the confusion between speed groups is too high for the resolution we work, so we first classify between the 24 classes and after that, if we obtain the ‘VEL’ label, we apply another classification techniques for this type of signs (that will be explained at next chapter).

To extract the new samples from frames and re-scale them to 35x35 pixels to database, we used a method to detect regular polygons at images called Shape Based Detection.

3.1.5 Shape Based Detection

The method is based on the extension of the fast radial symmetry transform to detect regular polygons [24]. This method operates on the gradient of a gray-scale image. Firstly, insignificant gradient elements, whose magnitudes are less than a specified threshold, are set to zero and the remaining elements normalised. Each remaining non-zero gradient element votes for a potential circle centre a distance r away (where r is the radius of the circle being targeted) along the line of the gradient vector. The vote is placed at the closest pixel to this point. The points voted for are called *affected pixels* and are defined by:

$$P_{\pm ve}(P) = P \pm round(rg(P))$$

where g(p) is the unit gradient at point p. There are positive p_{+ve} and negative p_{-ve} affected pixels corresponding to points that the gradient points towards and away from respectively. Since we do not know *a priori* whether a sign will be lighter or darker than the background we use both positively and negatively affected pixels concurrently.

However, if such information is known it can be used. To extend this voting scheme to regular polygons we define the ‘radius’ of a polygon as the perpendicular distance from an edge to the centroid. Further, rather than gradient elements voting for a single point, a line of votes is cast describing possible shape centroid positions that would account for the observed gradient element. Fig 25 shows different votes cast by a gradient element g(p) when searching for different shapes at a given radius (only the votes associated with the positively affected pixel are shown). Whereas in the case of a circle a single vote is cast per gradient element, a line of votes is cast when searching for straight-sided shapes. The white bars indicate potential centroid locations that receive a positive vote, and the dark bars indicate locations that receive a negative vote.

The negative voting is introduced to attenuate the response generated by straight lines too long to correspond to shape edges at the target radius.

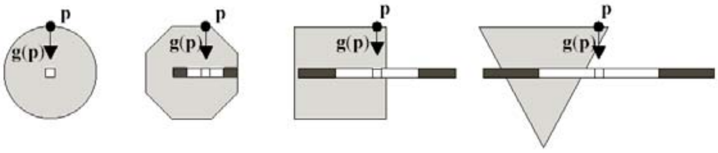


Figure 25. Voting lines associated with a gradient element when searching for different shapes.

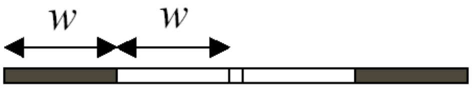


Figure 26. Line of pixels voted for by a gradient element

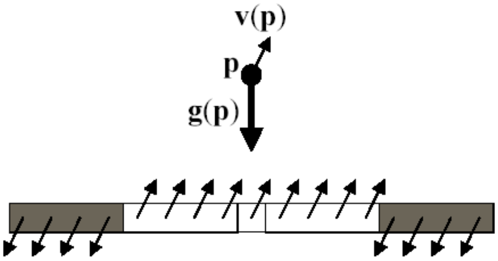


Figure 27. Example of n-angle gradient projected from a point p

The length of the line of pixels voted for is defined by w as shown in Fig 26. The width parameter w is chosen so that every point on a shape edge will cast a vote for the correct shape centroid, and is given by:

$$w = round\left(r \tan \frac{\pi}{n} \right)$$

where r is the radius and n the number of sides of the polygon being targeted. The line on which the affected pixels lie can be approximated by:

$$L(p,m) = P_{\pm ve}(p) + round(m\bar{g}(p))$$

Where $\bar{g}(p)$ is a unit vector perpendicular to g(p). The pixels receiving a positive vote are then given by:

$$L(p,m) \mid m \in [-w, w]$$

and those receiving a negative vote by:

$$L(p,m) \mid m \in [-2w, -w-1] \cup [w+1, 2w]$$

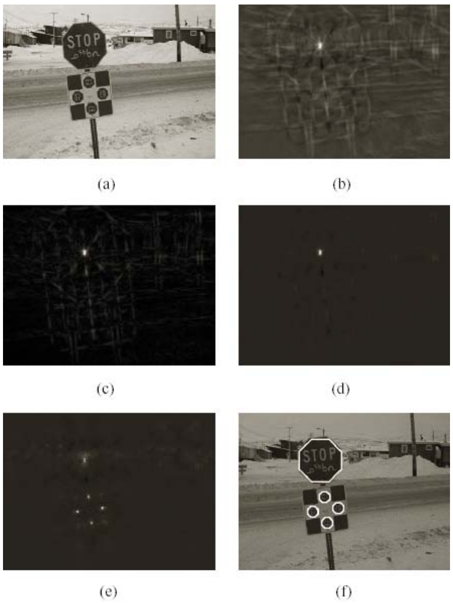


Figure 28. Searching for octagons in an image. (a) original 300 × 400 image, (b) vote image O_r for r = 40 (c) magnitude of equiangular image $||B_r||$ for r = 40 (d) result for radius r = 40, (e) total result over all radii r ∈ [6, 40], (f) detected octagons.

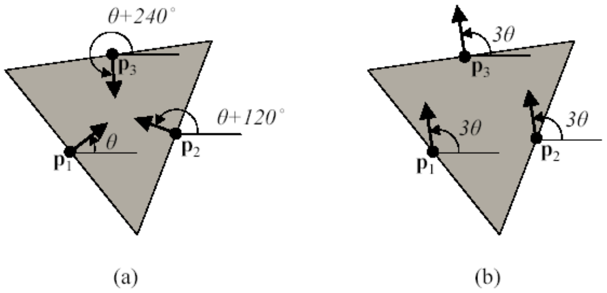


Figure 29. Example of three edge points p_i on a triangle. Showing (a) the angles of the unit gradient vectors, and (b) the resulting vectors obtained by multiplying the gradient angles by n (n = 3)

Whether targeting circles or regular polygons, all votes are accumulated into a vote image Or. Fig 28 (b) shows an example vote image for an octagonal target. Regular polygons are equiangular i.e., their sides are separated by a regular angular spacing; for an n-sided polygon this is 360/n degrees. To improve our detection of these shapes we introduce a rotationally invariant measure of how well a set of edges fits a particular angular spacing.

Define $\gamma(x,y) = n\theta(x,y)$ where $\theta = \angle g$ is the gradient angle, and n is the number of sides of the target polygon. Let v be the unit vector field such that $\angle v(x,y) = \gamma(x,y)$. For a given set of edge points p_i, the magnitude of the vector sum $\sum_i v(p_i)$ indicates how well the set of edges $g(p_i)$ fits the angular spacing defined by n.

Consider the example in Fig 29. Three edge points p_i are sampled from the sides of an equilateral triangle. The unit gradient vectors and their associated angles are shown in (a). By multiplying the gradient angles by n (n = 3 for a triangle) the resulting vectors share the same direction if, and only if, their original orientations were spaced 360/n degrees apart. Thus the magnitude of the vector sum of these n-angle vectors is maximal if the edge points occur at the targeted angular spacing.

To utilise this result we construct a vector field of projected n-angle gradients by considering each non-zero element of g and projecting its associated n-angle vector v(p) onto its voting space as shown in Fig 27 (note the sign is reversed when projecting onto negatively voted pixels). Vectors projected onto the same pixel are summed. The result is a vector field B_r, whose magnitude indicates how well the gradient elements voting on each point match the target angular spacing. Fig 28 (c) shows an example of such a magnitude image for an octagonal target. For increasing n the n-angle representation is limited by the accuracy of the gradient orientation estimate. However, it is perfectly adequate for n = 8 (e.g. Fig 28 (c)) which is the maximum required for road sign detection.

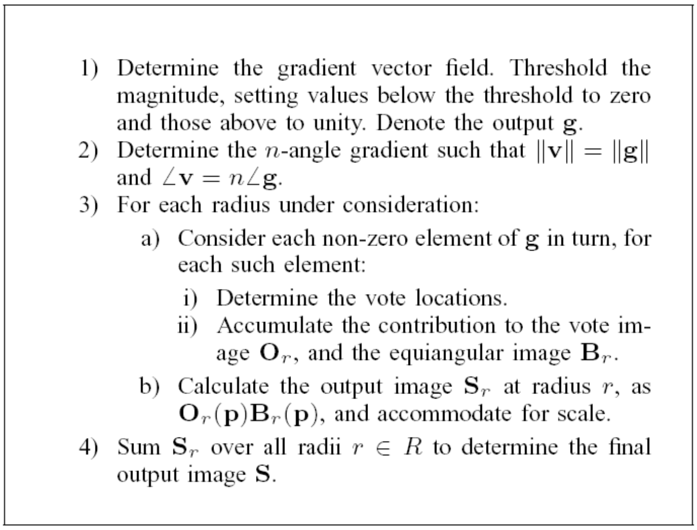
Once the vote image O_r and the equiangular image B_r have been computed the final shape response S_r is determined for the radius in question as:

$$S_r(p) = \frac{O_r(p) \| B_r(p) \|}{(2wr)^2}$$

The denominator is a scaling factor that facilitates comparisons of results across different radii. The n-angle representation is not relevant to circles since a circle has edges at all orientations. However, $||B_r||$ can still be computed by summing relevantly orientated (all) vectors voting on each point, giving $||B_r|| = O_r$, which makes the circular radial symmetry algorithm an extrema in this class of algorithms.

The transform is typically calculated over a set of radii $r \in R$, where R is the set of radii values at which the road sign is expected to appear. The combined result image S is obtained by summing over all $r \in R$.

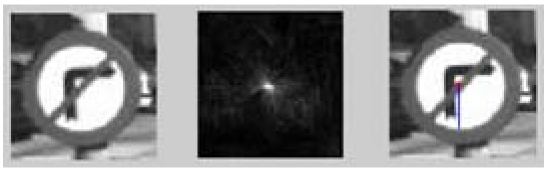
When searching across multiple radii, maxima are first identified in the combined image S, then verified to appear with sufficient magnitude in one or more of the radial results S_r from which the position and radius are determined. Fig 30 presents an overview of the algorithm, and Fig 28 shows outputs at different stages of the detection of octagons in an example image. Note that although the size and location of the shapes are recovered the orientation is not since the operator is orientation invariant (the detected octagons are drawn with zero orientation).



```
1) Determine the gradient vector field. Threshold the magnitude, setting values below the threshold to zero and those above to unity. Denote the output $\mathbf{g}$.
2) Determine the $n$-angle gradient such that $||\mathbf{v}|| = ||\mathbf{g}||$ and $\angle\mathbf{v} = n\angle\mathbf{g}$.
3) For each radius under consideration:
    a) Consider each non-zero element of $\mathbf{g}$ in turn, for each such element:
        i) Determine the vote locations.
        ii) Accumulate the contribution to the vote image $\mathbf{O}_r$, and the equiangular image $\mathbf{B}_r$.
    b) Calculate the output image $\mathbf{S}_r$ at radius $r$, as $\mathbf{O}_r(\mathbf{p})\mathbf{B}_r(\mathbf{p})$, and accommodate for scale.
4) Sum $\mathbf{S}_r$ over all radii $r \in R$ to determine the final output image $\mathbf{S}$.
```

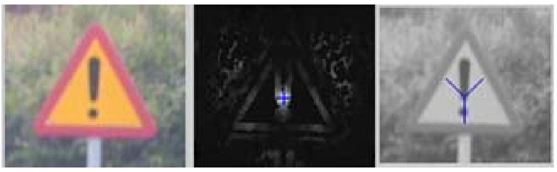
Figure 30. Summary of the algorithm

Figure 31, 32, 33 and 34 show results for the detection of different types of road signs.



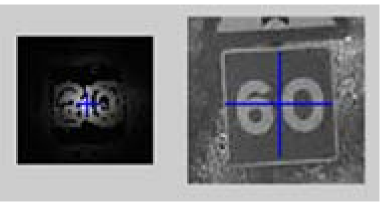
(a) (b) (c)

Figure 31. Circular detection. (a) Input image. (b) Total accumulator. (c) Detected sign.



(a) (b) (c)

Figure 32. Triangular detection. (a) Input image. (b) Total accumulator. (c) Detected sign.



(a) (b)

Figure 33. Rectangular detection. (a) Total accumulator. (b) Detected sign.

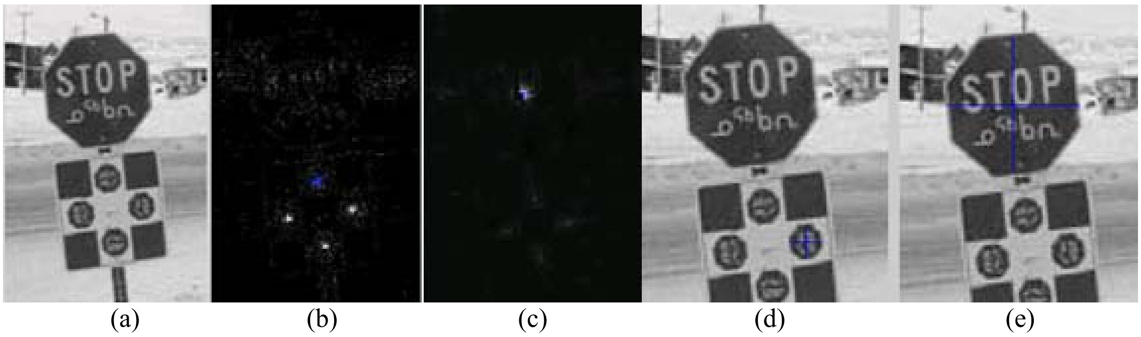


Figure 34. Octogonal detection. (a) Input image. (b) Total accumulator for small radii. (c) Total accumulator for high radii. (d)(e) Detected signs.

The previous method had been used to calculate the center and radius of the circular signs to create the database. For instance, radial symmetry is used for this type of signs. The new database contains 4000 real samples for the 22 classes of fig 35 extracted from 10 DVD's received from ICC Geomobil Project. Each DVD contained about 3000 frames and the classes that define the circular database are the most frequent from the DVD's.

The Radial Symmetry for spatial normalization of circular signs we studied proved to be more robust than the previous procedures of eliminating noise and fitting an ellipse. We only found one problem: when the sign contain noise inside and outside but next to the sign, it can displace the center of the sign. To come up with possible displacements of the center due to the image noise, we can iterate this procedure applying a circular mask to exclude near points that can displace the center of the sign, and repeat the process limiting the radius range (fig 35).

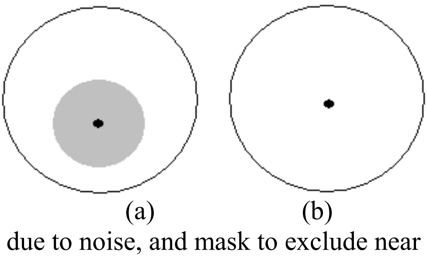


Figure 35. (a). Displaced center due to noise, and mask to exclude near points of high gradient module. (b) Center correction from the next iteration of radial symmetry.

3.1.6 Normalization

Once we extracted the sign, we equalized the image before including it in the database to do the classification invariant to illumination. Fig 36a shows the extracted normalized sign. At fig 36a we see that the regions of the sign are not homogeneous, so, classification methods could accumulate errors. To solve this problem, we use an anisotropic filter, Perona Malik [25] and Weickert [26], and we observed that Weickert runs better with our images. The result is shown in fig 36b.

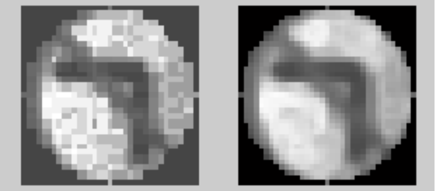


Figure 36: (a) Extracted equalized sign. (b) Anisotropic Weickert filter.

At this moment, we use different classification methods such as Tangent Distance, K-NN, Principal Components Analysis pairwise, Fisher Linear Discriminant Analysis pairwise, Boosting with Rectangular Features pairwise and multiclass Joint Boosting. The hit ratio with the new database and the new preprocessing methods: Radial Symmetry with center correction, Anisotropic Weickert and Mask is high (upon 95% in most cases). K-Nearest Neighbour delivered the best hit ratio, but we observed that in some cases the center of the detected sign could be slightly displaced (due to the ellipsoidal form of the sign or the noise). To solve this problem we selected the center and radius detected, and we extracted four slight displacements with the same radius and other four with a little variation of radius length. We classified all these extractions (fig 37 for an extracted sign) and we classified the assigned label at minimum distance. With this procedure we obtained results upon 99 % by k-NN.

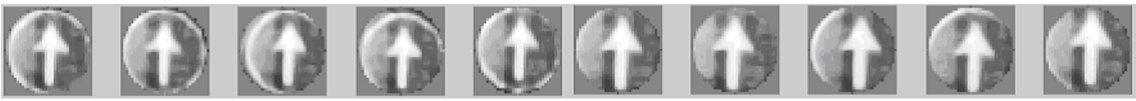


Figure 37: 10 extraccions

3.1.7 Speed signs

Now we have methods with which we obtain good results for circular signs, spatial normalization, and recognition. We grouped the speed signs at the same Velocity group. When a speed sign is detected we apply other methods to classify the correct speed sign. The problem of speed signs at the database is that they are very similar (fig 38b), we have a poor resolution, and we can receive different transformations for the same speed sign (Fig 38a).

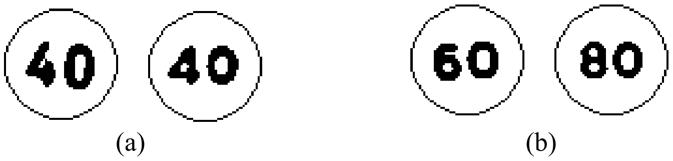


Figure 38. (a) Different representations for the same sign. (b) High similarity between classes.

3.1.7.1 Caracter segmentation for speed signs

As we have seen the best way to classify the velocity group is with a previous binarization, we thought in the segmentation of the speed digits to improve the consequent recognition by counting the number of digits and recognising them one by one. The process starts by the segmentation of the sign. After that, an up and down region are ignored. The next step is to project horizontally counting the number of pixels by row in order to find the beginning and the end of the lines that contain speed characters. With the extracted line, the process is the same but projecting vertically to segment the digits one by one. If the process runs well, we have extracted all the digits of the sign. Counting the number of extracted digits we know the possible subgroups of the sign and we are able to recognise the digits one by one to improve the recognition. In fig 39 different correct digit segmentations for different speed signs are shown.

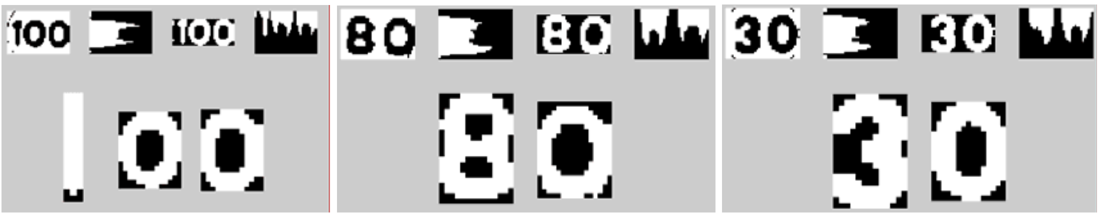
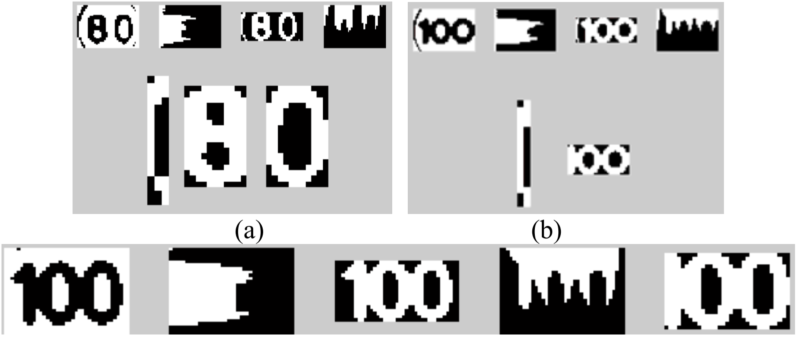


Figure 39. Different digit binarization for speed signs.

Testing the process to a complete set of signs, we met problems because in some cases the noise affects us and we count more digits than the sign contains (fig 40a). Another problem is that in some samples, the segmentation of the sign obtains a binary image where the pixels are united and with the extraction we obtain an incorrect number of digits (fig 40c). From these problems (fig 40b), we consider alternative methods for speed signs classification.



(a) (b)

(c)

Figure 40. (a) Wrong number of digits due to the noise. (c) Wrong number of digits due to the united characters. (b) Wrong extractions by the two before reasons.

3.1.7.2 Naive Boosting

To perform the recognition of speed signs, we have created a new database of 2000 real samples for the groups of fig 41. The new samples have a resolution of 41x41 pixels. Applying all classification methods and Naive Bayes Boosting, creating the probability matrix for each class, we observed that the hit ratio increases when images are binarized due to the images of the speed group are easily binarized using Otsu threshold, and we do not have the influence of the grey level that does not correspond to the principal components of the images. In this case, boosting obtains better results than the other methods because the few differences between classes have been trained by boostings, obtaining results upon 90% for the speed classification.

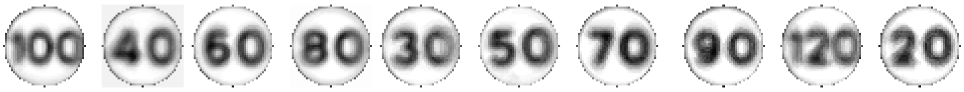


Figure 41. Speed signs.

3.2 Triangular signs

For the spatial normalization of triangular signs first we used the Shape Based Detection, but the method is not robust due to the poor resolution of the real signs and the noise of the images. We used the supervised Shape Based Detection for the generation of a real database of triangular signs. The groups found at the ICC DVDs are shown in fig 42. In fig 42a we see all considered classes, and in fig 42b we see two grouped classes since we have the same problem of very similar signs than in the case of the speed group for circular signs.

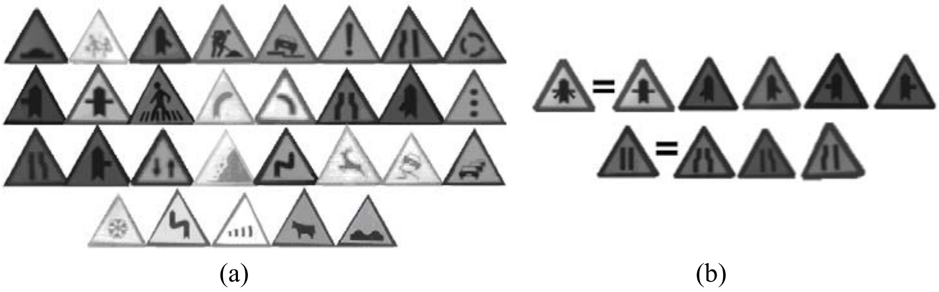

(a) (b)

Figure 42. Triangular database classes. (a) All classes. (b) Grouped classes.

The database contained 29 classes with a total of 2500 real samples extracted with supervised commented Shape Based Detection for triangles. We have used 10 ICC DVDs from the Geomobil project with about 3000 frames each one. After the extraction of the signs we have equalized and filtered with the Anisotropic Weickert filter each sample as in the case of circular signs database. The complete process to generate the supervised triangular database is shown in fig 43.

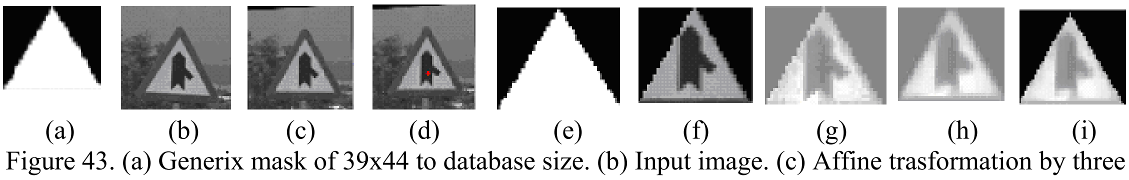

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 43. (a) Generix mask of 39x44 to database size. (b) Input image. (c) Affine trasformation by three control points. (d) Center of symmetry by Shape Based Detection for triangles. (e) Mask of the detected triangle using the detected radius. (f) Masked detected sign. (g) Equalized sign. (h) Anisotropic Weickert filter. (i) Resized image to the database size and masked with (a).

For the spatial normalization of triangular signs, we used a triangle detector by Gradient Direction Filter.

3.2.1 Gradient Direction Filter

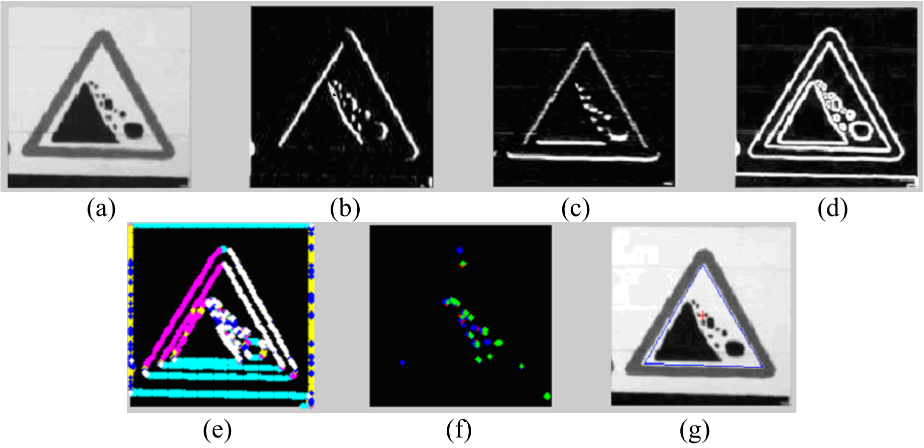
The main idea of the method is to use the gradient orientation of the image to detect the three corners of the triangle using the representatives angles of each corner.

Given an input image (fig 44a) we calculated the gradient image (fig 44d) and the angle image, where the angle image has at each point (x,y) the angle that forms the directional derivates of the input image (fig 44b-c) at point (x,y).

Now we calculated the labeled-angle image. For each image point at gradient image G(x,y)>Θ, where Θ is a threshold, we labeled the point (x,y) at labeled-angle image with the angle (x,y) from the angle image. This label is a value from 1 to 8, that corresponds to the 8 regions of the fig 45. At this moment we obtained the labeled-image of fig 44e.

The next step is to search for the three corners of the triangle. Each corner can be identified by a change of two angle orientation from the labeled-angle image. We searched for the three types of corners, for each labeled point we searched in its surroundings to find one of the three possible angle combinations to obtain the three-corner-labeled image of fig 44f.

Given the three-corner-labeled image, the final step is to fit the triangle using the combination of the three possible corners. We searched for each superior triangle corner, and for a different radius R, we searched at positions of fig 46 for the inferior-left and inferior-right triangle corner. For the expected corner location, we used a circular region of radius r to come up with possible displacements of the corner location due to the affine transformations. The resulting fitted triangle is shown in fig 44g.


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

(e) (f) (g)

Figure 44. (a) Input image. (b) x. (c) y. (d) Gradient. (e) Labeled-angle image. (f) three-corner-labeled image. (g) Detected triangle.

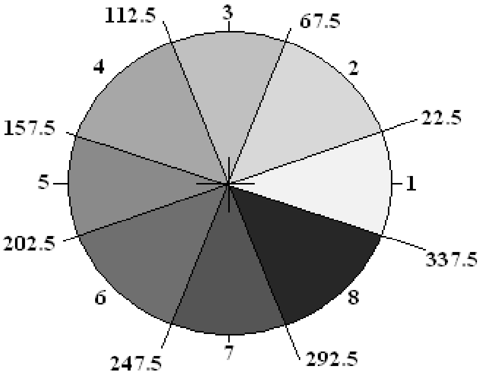


Figure 45. 8 different regions for label each angle.

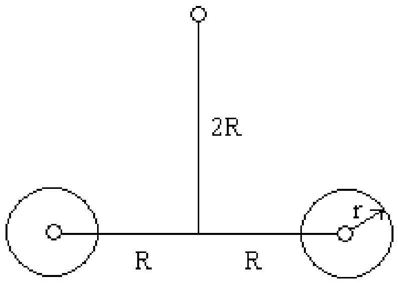


Figure 46. Given a superior corner, locations to seach for two remaining corners.

Varying the angles to label the pixels, we can extend this procedure to other geometric shapes as rectangles (fig 47).

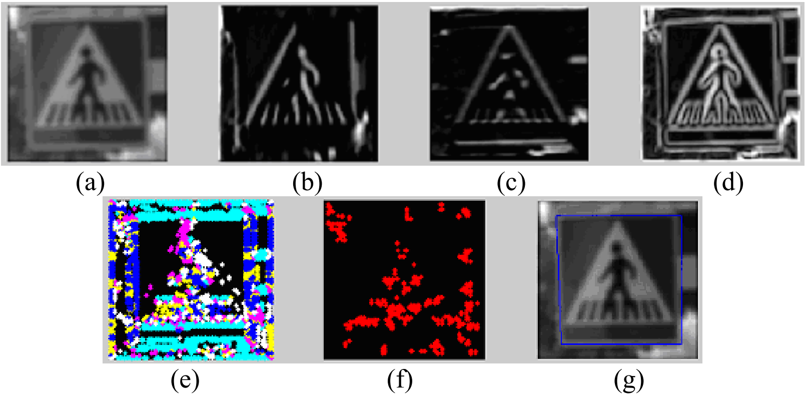


Figure 47. (a) Input image. (b) x. (c) y. (d) Gradient. (e) Labeled-angle image. (f) labeled-corners image. (g) Detected rectangle.

The advantages of the commented method is that obtains good results in most of cases and could be extended to other geometric forms.
The disadvantage is that the method is sensible to noise and we can fit a false triangle in some cases where the triangle has a poor resolution or the noise contained is too high.

After this method we considered training the three corners of the triangle by boosting with cascade of weak classifiers and to compare the results.

3.2.2 Boosting Rectangular features with cascade of weak classifiers

To train the three corners of the triangular signs, we used the boosting with rectangular features commented at the previous chapter. To optimize that procedure, we included a cascade of weak classifiers. At the first step, we obtained a 100% hit ratio and a false alarm < 50 %. The not well-classified negatives are introduced to the same procedure at next step with the same positives in order to obtain again a false < 50 % and a hit of 100%. The resulting classifier has as false alarm, the product of the results of the parcial classifiers and a hit ratio of 100%. This procedure is shown in fig 48.

To train we have extracted 1000 real samples for each corner, and we have trained with 10000 negatives samples. The results show that we obtain few false positives when we detect corners in all possible windows of the regions of interest received from the detector, but as a disadvantage, the process needs to detect the three corners of the triangle in each case to fit the sign correctly, making difficult the capture of the sign in some cases. Fig 49 shows the detection for the superior corner of the triangle at different frames.

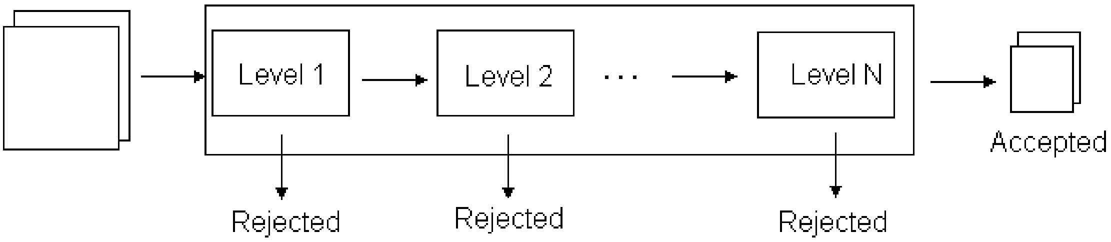


Figure 48. Cascade of weak classifiers.

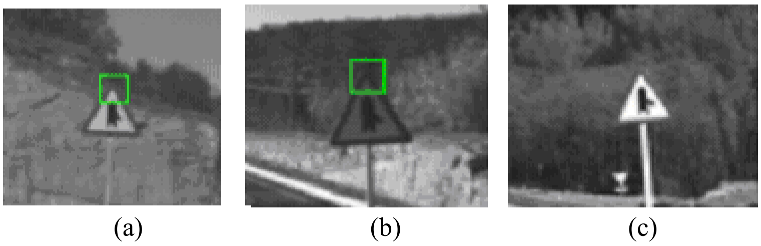


Figure 49. (a)(b) Detected superior corner for two different frames. (c) Lost corner.

3.2.3 Corner detection and density functions

The objective is to find the three corners of the triangle in order to transform the image and proceed with the recognition process. In this case, we used a corner detector. In order to eliminate irrelevant corners, we previously apply a morphological closing with a disk structure element of different sizes, depending on the size of the input image (fig 50a). After applying closing, we use Anisotropic Weickert filter to obtain homogenous regions (fig 50b). With the obtained image we apply a corner detector (fig 51a).

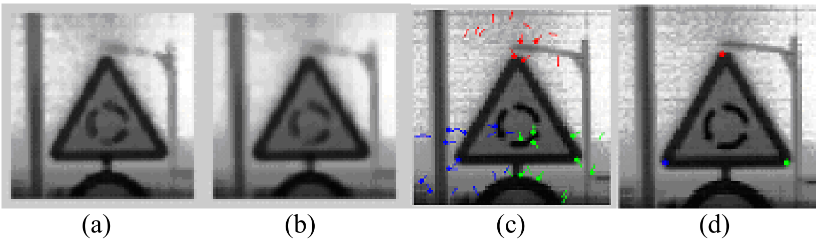


Figure 50. (a) Morphological closing. (b) Anisotropic Weickert filtering. (c) Labeled corners. (d) Result from density estimation function.

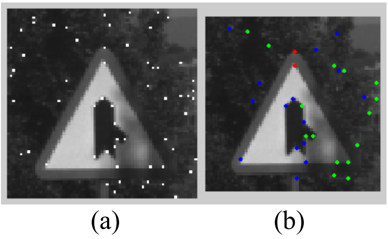


Figure 51. (a) Corners detector. (b) Reduced-labeled corners.

The next step is to label the detected corners. The main idea is that the three corners corresponding to the three inner triangle corners should have three different angle values with a certain margin error. But in practical way, we can detect the inner or the outer corner, and its corners could be slighly displazed, obtaining high variation of the hoped angle. Gradient orientations fot three corners are shown ion fig 52.

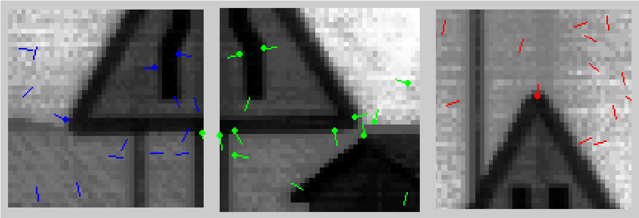


Figure 52. Gradient orientations for three triangle corners.

So, we labeled each corner depending on the directions of the pixels of its surroundings. Using the previous knowledge of the orientations of the three degrees, we eliminated false candidates of sign corners. We estimated the regions of interest of the corner through orientations smoothed input image, because we do not want to loose the strong information about orientations that could be lost at processed images. The desired angles and regions of interest to label corners are shown in fig 53. In fig 51 we can see the elimination of false corners by labeling.

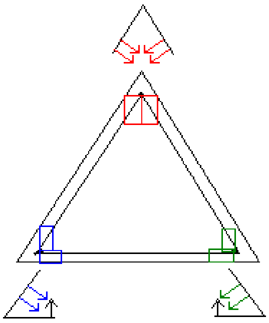


Figure 53. Desired angles for the different triangle corners and regions of interest.

At this moment we had obtained a labeled corners image. The next step was to find the combination of the three possible corners of the triangle with most probability to form part of the sign. For this task we used a density estimation function using the follows probabilities:

Step 1-Minimizing the distance of each type of corner to its respective most probable location (fig 54). At fig 55 we see the regions to consider each possible corner.

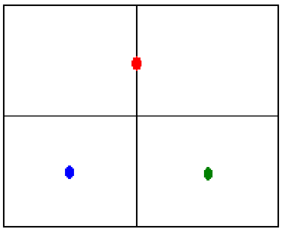


Figure 54. Points location.

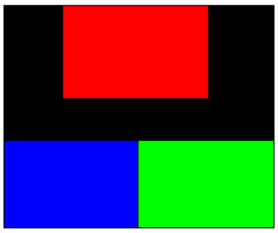


Figure 55. Locations for search each possible angle.

Step 2-Gradient magnitude. This probability is important due to the fact that corners of the triangular signs have high gradient values.

Step 3-Density of points of orientation desided at surroundings of each type of corner.

Step 4-Minimize distance between combinations of the corners to form a triangle (fig 56).

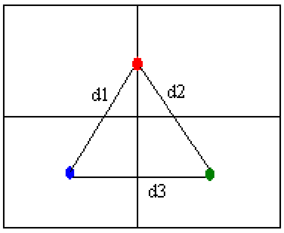


Figure 56. Distances to minimize.

Next step is to combine all possible corners and to use the comemented density estimation function with all probabilities normalized to estimate the combination with high probability to be part of the triangle. The result of the process is shown in fig 50.

The corner-labeling using density estimation functions has the advantage that the three corners we fit are normally visible, and we can label the three corners and use the information we know a priori about the sign to find the best combination of vertex.
The disadvantage is that in some cases we lost one of the corners due to the poor quality of the received image, and false labeled corners can modify the correct combination of vertex.

To solve the disadvantages of the previous method we combine it with the Hough Transform to detect triangles.

3.2.4 Hough Transform to detect triangles

To complement the previous procedure, we used the Hough Transform to detect the three sides of the triangle. Given an input image (fig 58a) we used Canny detector to obtain tne contours map (fig 58b). We estimated

$$x \cdot \cos\theta + y \cdot \sin\theta = \rho$$

with the parameters of fig 57 for each pixel to obtain the space of Hough for lines (fig 58d). (x,y) are the pixel localition, ρ is the distance to the origin by a perpendicular line, and θ is the angle.

Given a region of interest that contains a sign, we know a priori that the three possible angles of each side of the triangle and an error margin. With this information we searched at three possible angles range for the three three representative lines at Hough space (fig 58e). Once we had the three lines, we only needed to calculate its intersection to find the three corners of the triangle (fig 58c). With the three corners of the triangle we can transform the image to correct the affine transformations to proceed with the classification procedure.

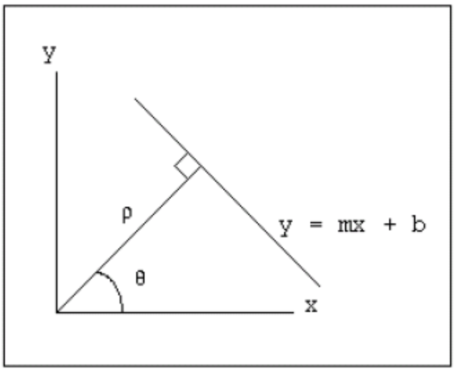
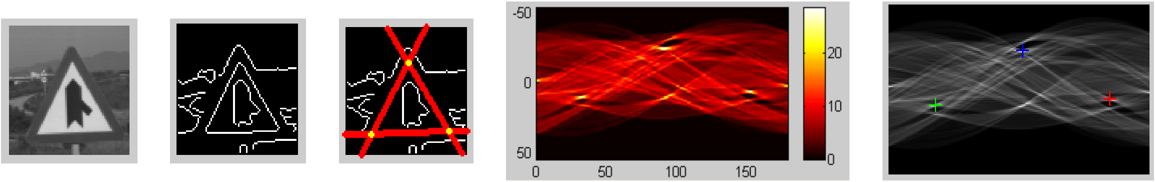


Figure 57. Correspondence to Hough



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

Figure 58. (a) Input image. (b) Canny contours. (c) Detected triangle lines and intersections. (d) Hough space. (e). Three detected lines for three known angles and a margin error.

The Hough procedure has a high reliability compared to the previous procedure, so the final process of spatial normalization for triangular signs is first to use the Hough transform. If we obtain the three possible corners at regions of probability of each corner, we transform the image and classify, in other case, if Hough transform does not find correctly the three corners, we apply the corner detection with density estimation functions.

The advantage of the commented method is that the sign contained at region of interest should be larger compared to the noise contained. So, in the space of Hough we normally detect the three representatives lines with the expected angle.
The disadvantage is that in some cases where the sign have a poor resolution, the noise can simulate a side of the triangle if it has a similar angle and a considerable length.

3.2.5 Improving the classification

In order to test the classification of triangles using the previos spatial normalization techniques, we used 9-different extractions by displasing the fitted sign to improve the classification step, classifying for the sign at minimum distance from all different extractions (fig 59).

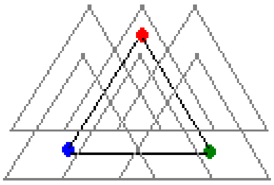


Figure 59. Different classification extractions for the fitted sign.

The classification of triangles using the spatial normalization techniques and different extractions to apply the classification methods offers a reliability upon 91 %.

In the next chapter we show all percentages of classification using the different classification techniques previously analized for the recognition of circular and triangular signs. The classification techniques are compared using another well-known databases.

4-Validation of the traffic signs recognition methods

To validate the methods of our system, we created a database of 4000 circular real samples, 1000 real speed samples and 2000 triangular samples. The spatial normalization methods are the previously commented, and the classification techniques are:

- 3-Nearest Neighbour
- Tangent Distance with invariant tangent planes related to x-traslation, y-traslation, scaling and rotation.
- Principal Components Analysis keeping 99.95% of data variancefollowed by 3-Nearest Neighbour.
- Fisher Linear Discriminant Analysis after applying Principal Components Analysis keeping 99.95% of data variance.
- Support Vector Machine with projection kernel Radial basis Function $K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|^2\right), \gamma > 0$.
- Real Adaboost with Rectangular Features
- Joint Boosting
- Naive Boosting
- Real Adaboost by 40% of Sampling using as simple learner Fisher Linear Discriminant Analysis after applying Principal Components Analysis keeping 99.95% of data variance.

The first statistics are related to the reliability of the classification methods using 80% of the database to train and the 20% to test in a cross-validation of 3 iterations. The confidence range at cross-validation is calculated by:

$$Confidence = 1.96 \cdot \left(std\left(p_1, ..., p_n\right) / \sqrt{n} \right)$$

4.1 Circular signs

The results for the 22 circular groups are shown in table 1. In this table the Naive Boosting is not used since this method is used for binary images. In fig 60 we see the graphic with the mean of each method and the confidence range.

Knn	**98.72±0.60**
Tangent Distance	62.21±4.02
PCA+Knn	92.68±1.72
FPCA	97.23±1.27
SVM	98.20±1.02
Boost with Rectangular Features	92.24±2.94
Joint Boosting	95.10±1.91
Boost sampling FPCA	93.32±2.20

Table 1. Classification methods and reliability for circular signs.

Now, as k-nearest neighbor obtain the best hit ratio, we use this method with the preprocessing for circular signs of radial symmetry and the 9 different extractions of the sign commented at previous chapter to do the recognition on new regions of interest received from the detector using the DVD's of the Institut Cartografic of Catalunya' Mobile Mapping Process.

Using a total of 2045 regions of interest, the radial symmetry preprocessing and classifying by the extraction of minimum distance, we obtain a 98.72% for circular signs, having used less than a second for the spatial normalization, and about 3 second for the 10 classifications of the sign for the different extractions, by a Pentium IV 3.2Ht.

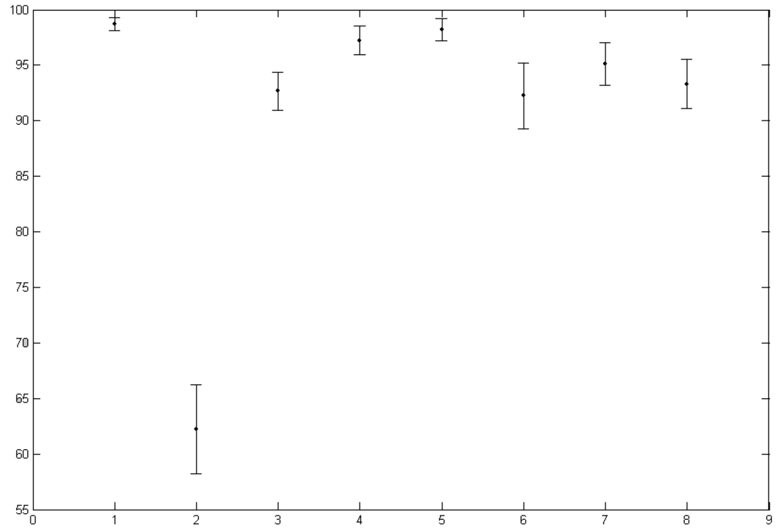


Figure 60. Hit ratio and confidence rang for circular group. From left to righ: Knn, Tangent distance, PCA+KNN, FPCA, SVM, Boost with Rectangular Features, Joint Boosting, Boosting sampling FPCA.

4.2 Speed signs

For the speed groups of circular signs, we use the binary database previously commented, and the classification methods used are the previous and Naive Boosting. The results are shown in table 2. In fig 61 we see the graphic with the mean of each method and the confidence range.

Knn	70.53±1.70
Tangent Distance	46.37±2.30
PCA+Knn	68.23±1.50
FPCA	**88.89±1.30**
SVM	79.84±2.00
Boost rectangular	85.93±2.10
Joint Boosting	80.36±1.50
Boost sampling FPCA	88.85±1.90
Naive Boosting	82.78±1.80

Table 2. Classification methods and reliability for the speed group.

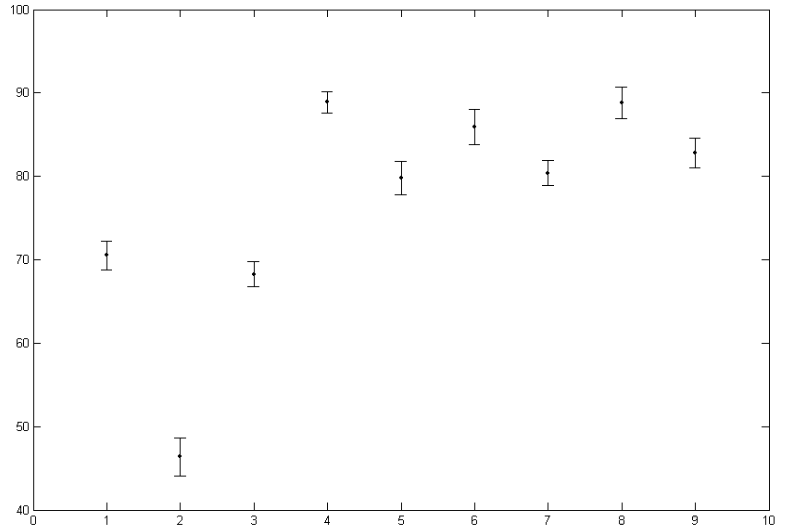


Figure 61. Hit ratio and confidence rang for speed groups. From left to righ: Knn, Tangent distance, PCA+KNN, FPCA, SVM, Boost with Rectangular Features, Joint Boosting, Boost sampling FPCA, Naive Boosting.

In the case of the speed group, FPCA runs better. We see that the classic classification methods obtain a lower reliability compared with FPCA and boosting. Boostings offer a high reliability too, it is due to the few differences between samples have been trained by boosting methods, and the classical methods use all the image accumulating errors in some cases.

With a total of 500 new regions of interest, the mean hit ratio in this case is 88.89%.

4.3 Triangular signs

The results for the triangular classes without preprocessing and using 80% to train and 20% to test in a cross-validation of 3 iterations are shown in Table 3. In this table the Naive Boosting is not used since this method is used for binary images.

Knn	**98.01±0.60**
Tangent Distance	67.38±1.20
PCA+Knn	93.82±1.32
FPCA	97.02±0.80
SVM	96.01±0.70
Boost rectangular	95.83±2.10
Joint Boosting	93.27±1.70
Boost sampling FPCA	94.02±1.90

Table 3. Classification methods and reliability for triangular signs.

Since the k-Nearest Neighbor obtains the best hit ratio for the case of triangular signs too, we use this method with the preprocessing for triangular signs of Hough transform and corner detection and the 9 different extractions of the sign commented at previous chapter to do the recognition on new regions of interest received from the detector using the DVD’s of the Institut Cartografic of Catalunya’ Mobile Mapping Process.

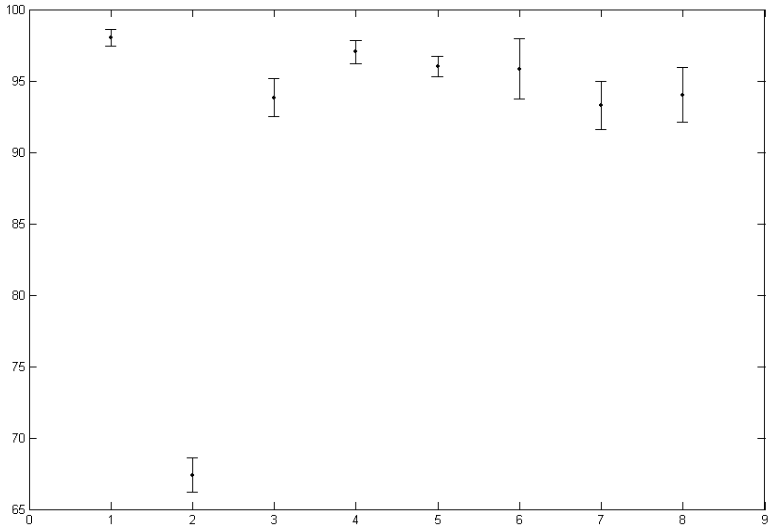


Figure 62. Hit ratio and confidence rang for speed groups. From left to righ: Knn, Tangent distance, PCA+KNN, FPCA, SVM, Boost with Rectangular Features, Joint Boosting, Boost sampling FPCA.

Using a total of 1022 total regions of interest, Hough tranform and corner detection followed by the classification of the extraction at minimum distance, we obtain a 88.52% for triangular signs. This result is lower since the preprocessing for spatial normalization fails in some cases. The time for spatial normalization for this type of signs is 1 second, and about 3 seconds for the classification.

4.4 UCI Repository database

To see the reliability of the classification method used in our system, we use some of the most known databases from the UCI repository [27]. As we receive the samples of these databases as a vector of features, we do not apply some classification techniques as tangent distance because we need derivable images, Real Adaboost with Rectangular Features because we do not have rectangular regions and Naive Boosting because the vector of features are not binary. The results are shown in table 4.

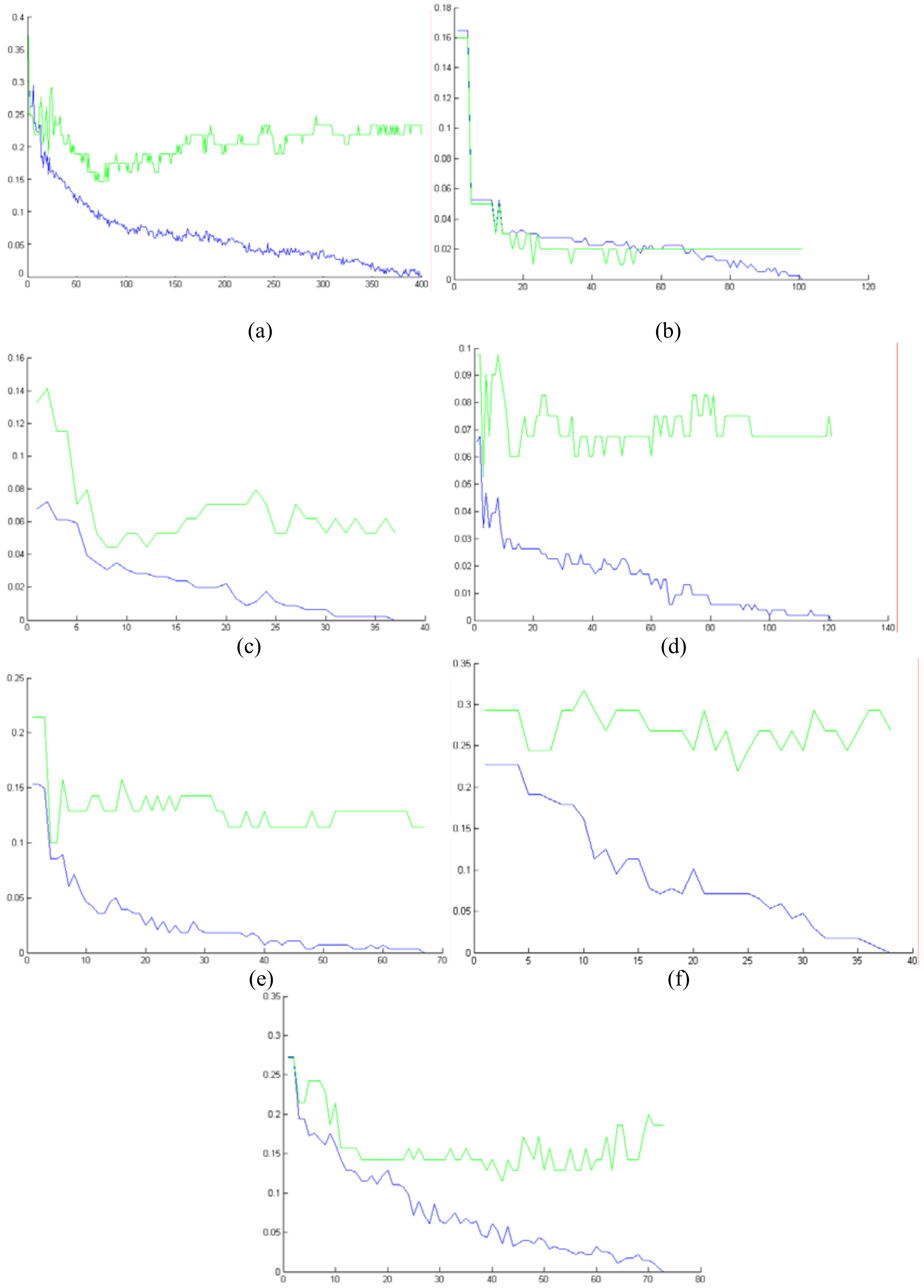
Database	**Features**	**3-KNN**	**PCAKNN**	**FisherPCA**	**Boosting Sampling FPCA**	**Joint Boosting**	**SVM**
BUPA Liver disorder	6	68.12±1.64	69.57±1.64	63.29±0.95	65.22±2.84	**74.86±3.77**	68.60±3.41
Wisconsin diagnostic breast cancer	30	92.92±2.00	91.91±2.40	85.55±4.17	89.09±2.52	**94.69±4.01**	93.49±3.03
8-D Banana shaped data	8	**98.67±1.31**	98.33±1.31	84.67±6.23	84.33±5.58	94.67±3.97	96.00±0.00
Wisconsin breast cancer	9	96.99±0.85	96.49±1.30	95.99±1.97	95.74±3.22	93.73±8.36	**97.24±0.49**
Cleveland heart disease	13	66.10±3.33	58.76±2.93	75.14±5.86	76.84±2.21	**79.66±8.36**	77.97±7.67
German database	24	67.00±2.04	70.00±0.57	73.00±4.27	70.50±0.57	72.25±3.80	**76.67±3.12**
Inosphere database	34	85.24±2.47	88.10±8.30	89.52±3.37	88.10±6.73	**90.44±1.94**	87.14±1.62
Sonar signals database	60	79.68±6.95	82.93±7.30	65.04±8.87	72.36±1.60	78.64±6.17	**84.55±4.23**
SPECTF heart	44	75.24±6.12	70.95±4.07	78.10±7.98	86.20±6.73	**89.53±7.98**	85.72±5.60

Table 4. Reliability of the classification methods using 3 iterations of cross-validation for different UCI repository databases.

The results of table 4 show that in most of cases Boosting, and Joint Boosting in particular run better (5 of 9 databases). 3-KNN only is the best classifier in one of the databases, and Support Vector Machines also obtain good results for three of the databases.

In our problem, SVM and Boostings also run well, but the spatial normalization techniques improve the correlation between the model to recognize and the database, so that only when images are too similar as in the case of the speed group, Boosting outperform the k-NN.

At fig 63 we see the progress of iterations of training (blue) and test (green) from different databases from the UCI repository in one iteration of cross-validation for the Joint Boosting algorithm.



(a) (b)

(c) (d)

(e) (f)

(g)

Figure 63. Progress of iterations of train (blue) and test (green) from different databases from the UCI repository in one iteration of cross-validation for the Joint Boosting algorithm. (a) BUPA Liver disorder. (b) 8-B Banana shaped data. (c) Wisconsin diagnostic breast cancer. (d) Wisconsin breast cancer. (e) Inosphere database. (f) Sonar signals database. (g) SPECTF heart.

5. Conclusions and future work

We have designed a system to a complex real problem: the road sign recognition. Since the first approaches of this work are related to autonomous driving or assisted driving, our mean intention is the creation of cartographic maps. The process is divided in three phases: detection, spatial normalization, and recognition. From the detection by Adaboost with cascade of weak classifiers, we receive a region of interest to do the consequent spatial normalization and recognition.

We have studied different spatial normalization techniques for the different geometric forms of the signs. From the different analyzed methods, we have found the radial symmetry as the most robust for the circular signs. This method allows to find the center of the sign and its approximated radius. Optimizing this procedure with different extractions of the signs and classifying by the extraction at minimum distance we obtain a robust approach.

We have studied different methods for the spatial normalization of triangular signs. As we only work with the information of the grey level of the image with a poor resolution, it is difficult to find a robust preprocessing for our problem. Hough transform for triangles combined by a corner-labeled-detector using the gradient orientations is the best preprocessing we have found for the spatial normalization of triangles.

For the recognition step, we analyzed an extensive set of classification methods, from the most classical: K-nearest neighbors, Tangent Distance, Principal Components Analysis, Fisher Linear Discriminant Analysis, or Support Vector Machines, to the most novel ones as boosting methods.

Boosting algorithms as part of multiclassifier methods have achieved a lot of popularity showing their strong advantage compared to the classical pattern classification algorithms. We have studied different feature extraction techniques for boosting algorithms using share features, rectangular features or Naive boosting.

We have estimated robust classifiers, by comparing the classical methods of classification with different variants of feature extraction for boosting. We show their utility when classes are very similar, and that the training of the most discriminant features split more properly the different types of classes. We have extended the comparison of the classic and novel methods of classification in our road sign traffic system and in other known databases as the UCI repository.

We have seen that our system allows a reliability upon 99% on circular signs, and a percentage upon 90% for the speed groups and triangular classes.

The results show that the presented method offers good robustness in case of high variance of sign appearance as noise, affine deformation, and reduced visibility without need to use color information. At the same time it allows a successful multi-class recognition. In addition, its low time-cost allows to be applied to real-time vision tasks.

The road sign detection study is still open. We are currently extending the proposed methods to recognize a set of any geometric forms of road signs and to improve the spatial normalization in signs as triangles, where the localization is not yet robust enough. The Institut Cartogràfic of Catalunya Mobile Mapping Process is using a color camara at Geomobil at this moment. Our next step will be to use the color information to theshold at color space, obtaining the representatives parts of the sign, and beeing able to capture it most properly for the consequent recognition.

We will separate the grouped types of triangles due to its similarity, doing the same procedure in the case of the speed group, where the samples are too similar and we need another preprocessing and more resolution of the images.

Anyway, we will be working still with the information of grey level of the image. We plan extracting distintive invariant features from images that can be used to perform reliable matching between different views of object [28], and perform the spatial normalization and recognition using boosting, training distances using contextual information into boosting, representing the object as a constellation of generalized correlograms that integrate both information of local parts and their spatial relations [29]. We consider that this procedure could be interesting to obtain a robust spatial normalization and recognition for object advanced classification.

6. References

[1] H. Akatsuka and S. Imai. “Road signposts recognition system.” In *Proc. SAE vehiclehighway infrastructure: safety compatibility*, pages 189–196, 1987.

[2] D. Kellmeyer and H. Zwahlen. “Detection of highway warning signs in natural video images using color image processing and neural networks.” In *IEEE Proc. Int.Conf. Neural Networks 1994*, volume 7, pages 4226–4231, 1994.

[3] M. de Saint Blancard. “Road sign recognition: A study of vision-based decision making for road environment recognition.” In *Vision-based Vehicle Guidance*, Springer Series in Perception Engineering. Springer-Verlag, 1992.

[4] Arturo de la Escalera, Luis E. Moreno, Miguel Ángel Salichs and José María Armingol “Road Traffic Signs Detection and Classification,” *CICYT Project TAP94-0711-C03-02*, jul. 1996.

[5] G. Piccioli, E. D. Michelli, and M. Campani. “A robust method for road sign detection and recognition.” In *Proc. European Conference on Computer Vision 1994*, pages 495–500, 1994.

[6] G. Piccioli, E. D. Michelli, P. Parodi, and M. Campani. “Robust road sign detection and recognition from image sequences.” In *Proc. Intelligent Vehicles’94*, pages 278–283, 1994.

[7] Huang, Chung-Lin and Hsu, Shih-Hung. “Road Sign Interpretation using Matching Pursuit Method”. *IEEE International Conference on Pattern Recognition (ICPR2000)*, pages 1329-1333. 2000. Barcelona, Spain.

[8] Piccioli, G., et al., "Robust Road Sign Detection and Recognition from Image Sequences", *Intelligent Vehicles Symposium*, pages 278-283, Paris, 1994.

[9] R. Ghica, S. Lu, and X. Yuan. “Recognition of traffic signs using a multilayer neural network,” In *Proc. Can Conf. on Electrical and Computer Engineering*, 1994.

[10] Y. Aoyagi and T. Asakura, "A Study on Traffic Sign Recognition in Scene Image using Genetic Algorithms and Neural Networks," *Proceedings of the IEEE IECON International Conference on Industrial Electronics, Control,and Instrumentation*, Taipei, Taiwan, pp. 1838-1843, 1996.

[11] M. Lalonde and Y. Li, "Detection of Road Signs Using Color Indexing," Technical Report CRIM-IT-95/12-49, Centre de Recherche Informatique de Montreal, *http://www.crim.ca/sbc/english/cime/ publications.html*, 1995.

[12] Naser EL-SHEIMY, "An Overview of Mobile Mapping Systems", Canada 2002.

[13] R. Alamús, A.Baron, E.Bosch, J.Casacuberta, J.Miranda, M.Pla, S.Sànchez, A.Serra, J.Talaya, "On the accuray and performance of the GEOMÒBIL system,"*Institut Cartogràfic de Catalunya (ICC), Parc de Montjuïc, E-08038 Barcelona*, Commission II, WG V/2.

[14] Jordi Vitrià and Xavier Baró, "Traffic Sign Detection on Greyscale image," In *Recent Advances in Artificial Intelligence Research and Development, IOS Press*, Spain, 2004.

[15] Ayse Küçükyilmaz, "Pattern Classification: A Survey and Comparison", Department of Computer Engineering, Bilkent University, Turkey, April 2005.

[16] Patrice Y. Simard, Yann A. Le Cun, John S. Denker, Bernard Victorri, "Transformation Invariance in Pattern Recognition - Tangent Distance and Tangent Propagation", 2000.

[17] P. Viola, J. Jones. "Robust Real-Time Face Detection". Technical Report 2001/01, Compaq CRL, February 2001.

[18] R. Lienhart and J. Maydt. "An extended set of Haar-like features for rapid object detection". In Proc. of the IEEE Conf. on Image Processing (ICIP '02), pages 155 - 162, USA, 2002.

[19] Antonio Torralba, Kevin P. Murphy and William T. Freeman, "Sharing Visual Features for Multiclass and Multiview Object Detection". Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. April 2004.

[20] Lindsay I Smith , "A tutorial on Principal Components Analysis", February, 2002

[21] MaxWelling , "Fisher Linear Discriminant Analysis", Department of Computer Science, University of Toronto, 2002.

[22] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification", Department of Computer Science and Information Engineering, National Taiwan University, 2002.

[23] Andrew Ditzgibbon, Maurizio Pilu, and Robert B. Fisher, "Direct Least Square Fitting of Ellipses," *Tern Analysis and Machine Intelligence*, vol. 21, no. 5, may. 1999.

[24] Gareth Loy, Nick Barnes, "Fast Shape-based Road Sign Detection for a Driver Assistance System", 2004.

[25] Perona P., Malik J., "Scale space and edge detection using anisotropic diffusion", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, pp. 629-639, July 1990.

[26] Weickert J., “Anisotropic diffusion in image processing”, *European Consortium for Mathematics in Industry (B.G. Teubner,Stuttgart).*, 1998.

[27] C. Blake and C. J. Merz. UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/_mlearn/MLRepository.html]., 1998.

[28] David G. Lowe, “Distinctive Images Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, 2004.

[29] J. Amores, N. Sebe, P. Radeva, Th. Gevers, and A. Smeulders, “Boosting Contextual Information in Content-based Image Retrieval”, *6th ACM SIGMM International Workshop on Multimedia Information Retrieval, New York, NY USA, pp.31-3,* October 15-16, 2004.