FUZZY SUPPORT VECTOR MACHINES FOR IMAGE CLASSIFICATION FUSING MPEG-7 VISUAL DESCRIPTORS

Evaggelos Spyrou, Giorgos Stamou, Yannis Avrithis and Stefanos Kollias

Image, Video and Multimedia Systems Laboratory,
School of Electrical and Computer Engineering,
National Technical University of Athens, Greece
Iroon Polytehneiou 9, 15780 Zografou, Greece
e-mail: {espyrou, iavr}@image.ece.ntua.gr,
gstam@softlab.ece.ntua.gr, stefanos@cs.ntua.gr

Keywords:SVMs,Fuzzy Systems,MPEG-7 Descriptors,Image Classification

Abstract

This paper proposes a new type of a support vector machine which uses a kernel constituted from *fuzzy basis functions*. The proposed network combines the characteristics both of a support vector machine and a fuzzy system: high generalization performance, even when the dimension of the input space is very high, structured and numerical representation of knowledge and ability to extract linguistic fuzzy rules, in order to bridge the "semantic gap" between the *low-level* descriptors and the *high-level* semantics of an image. The Fuzzy SVM network was evaluated using images from the aceMedia Repository[1] and more specifically in a *beach/urban* scenes classification problem.

1 Introduction

Many retrieval tasks, such as content-based image retrieval (CBIR) can not be performed by simply manually associating words to each image for two main reasons: Firstly, with the exponential increasing quantity of digital content, it would be a time-consuming task and secondly because "images are beyond words", [10], which means that their content can not be fully described by a list of words. As a matter of this, the extraction of visual information directly from the images is required. This usually called *low-level features extraction*.

These low-level features of an image can be efficiently captured by visual descriptors. To achieve robust classification using features captured from the whole image (global features), it is crucial to select an appropriate set of descriptors in order to capture the distinctive characteristics of the current problem. For instance, local color descriptors and global color histograms are used in indoor/outdoor classification [12] to detect e.g. vegetation (green) or sea (blue). Edge direction histograms are employed for city/landscape classification [13] since city images typically contain horizontal and vertical edges. Additionally, motion descriptors are also used for sports video shot classification [3].

However, in many cases, classification using a single visual descriptor fails to achieve satisfactory and robust results since a certain feature may be present and dominant in more than one classes. Thus, the combination of more than one visual descriptors is needed in order to further increase the efficiency of the existing techniques. For instance, in [6] by simply adding the distances between two images using various visual descriptors, the results of the classification are improved. In [4], this combination is achieved using Support Vector Machines, neural and neurofuzzy networks and fuzzy rules are extracted from the latter method. Finally, Color and Texture features presented in the form of histograms are used in [2].

Support Vector Machines are learning machines based on statistical learning theory, that can be used for pattern classification or regression. They provide high generalization performance without the need to add a priori knowledge, even when the dimension of the input space is very high. This ability results from their main difference from the other types of neural networks, that they are an exact implementation of the structural risk principle [14].

Furthermore, *Fuzzy Systems* are those systems whose variables have as domain fuzzy sets. They encode structured, empirical (heuristic) or linguistic knowledge in a numerical framework [5]. They are able to describe the operation of the system in natural language with the aid of human-like IF-THEN rules. However, they do not

[1] http://www.acemedia.org

provide the highly desired characteristics of learning and adaptation.

Although from a first sight the two aforementioned learning machines seem incompatible to be combined, we will show that a Support Vector Machine may indeed use a kernel constituted from "fuzzy basis functions" [5] and therefore construct a Fuzzy System. This way, semantic information can be extracted in the form of linguistic fuzzy rules.

The structure of this work is organized as follows: section 2 begins with the needed theoretical background in both the support vector machines and the fuzzy systems, to allow the reader to understand the notion of the "Fuzzy Support Vector Machines" that follows. Then in section 3, the MPEG-7 descriptors that capture the visual features of the images in a standardized way are presented, followed by the experimental results in section 4. Finally, conclusions are drawn in section 5.

2 Fuzzy Support Vector Machines

2.1 Support Vector Machines

Support Vector Machines are feed-forward networks that can be used for pattern classification and nonlinear regression. Their main idea is to construct a hyperplane that acts as a decision space in such a way that the margin of separation between positive and negative examples is maximized. This is generally referred as the "Optimal Hyperplane". This property is achieved as the support vector machines are an approximate implementation of the method of *structural risk minimization*[14]. Despite the fact that a support vector machine does not incorporate domain-specific knowledge, it provides a good generalization performance, a unique property among the various different types of neural networks.

An inner-product kernel between an input vector $\mathbf{x}$ and a *support vector* $\mathbf{x}_i$ is the main characteristic on the support vector machines. The support vectors consist of a small subset of the training set vectors that are extracted by the optimization algorithm. This kernel can be implemented in various ways, thus leading to different types of nonlinear learning machines. The most important are *Polynomial* learning machines, *Radial-Basis Function* networks and Single-hidden layer *Perceptrons*, where the kernel function is polynomial, exponential or a hyperbolic tangent function, respectively.

The concept of the optimal hyperplane for the case of linearly separable patterns is explained in figure 1. Assuming that the patterns are drawn by the training sample $(\mathbf{x}_i, d_i)_{i=1}^N$ and that the classes represented by the subsets $d_i = +1$ and $d_i = -1$ are linearly separable. A hyperplane that does the separation can be represented

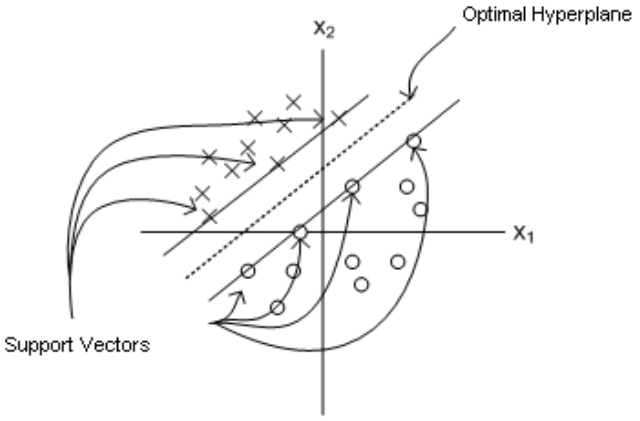

Figure 1: Optimal hyperplane for linearly separable patterns

by the equation:

$$\mathbf{w}^T\mathbf{x} + b = 0 \tag{1}$$

where $\mathbf{x}$ is an input vector, $\mathbf{w}$ is a weight vector and b a bias. Thus, for all patterns, we may write:

$$\begin{array}{ll} \mathbf{w}^T\mathbf{x}_i + b \geq 0 & \text{for } d_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq 0 & \text{for } d_i = -1 \end{array} \tag{2}$$

This assumption is relaxed in the case of non linearly separable patterns. The separation between the hyperplane and the closest data is called *margin of separation* and the goal of the support vector machine is to construct the optimal hyperplane for which this margin is maximized.

The most common method for the optimization and the construction of the optimal hyperplane is based on the Lagrange multipliers' optimization method. However, this is a very complex problem and instead, the less complex "Least squares" optimization method is applied [11]. The less accuracy of this method is compromised from its fastest speed and from the fact that it leads to a smaller number of linguistic rules. It differs from the Lagrangian because a linear system is solved instead of the quadratic optimization that is used by the traditional approach. The optimization function that needs to be minimized is:

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w}^T + C\sum_{i=1}^{N} \xi_i^2 \tag{3}$$

with the following constraints: $\psi_i(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) + b) = 1 - \xi_i, i = 1, 2, \ldots, N$. Where x_i is an input vector of the training set, ψ_i its response and C a user selected variable. The ξ_i are called "slack variables" and measure the deviation of a data point from the ideal condition of pattern separability. Using the Lagrange Multipliers, the cost function becomes:

$$\Phi(\mathbf{w}, b, \mathbf{a}, \xi) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w}^T + C\sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} a_i \left\{\psi_i(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) + b) - 1 + \xi_i\right\} \tag{4}$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_N)^T$ are the Lagrange multipliers. It should be noted that they could take both positive and negative values. By taking the derivatives, the following relations result:

$$\mathbf{w} = \sum_{i=1}^{N} a_i \psi_i \boldsymbol{\phi}(\mathbf{x}_i), \sum_{i=1}^{n} a_i y_i = 0, a_i = C\xi_i \tag{5}$$

All these equations written in the form of matrices occur to:

$$\begin{pmatrix} \Omega & \Psi \\ \Psi^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{a} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ 0 \end{pmatrix} \tag{6}$$

where the sub matrices are defined as: $\Omega_{ij} = d_i d_j \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j) + \frac{\delta_{ij}}{C}, \Psi = (\psi_1, \ldots, \psi_N)^T, \mathbf{1} = (1, \ldots, 1)^T$ and $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ By solving the linear system presented above, the Lagrange multipliers and the bias result from the relations:

$$\mathbf{a} = \boldsymbol{\Omega}^{-1}(\mathbf{1} - \boldsymbol{\Psi} b) \tag{7}$$

$$b = \frac{\boldsymbol{\Psi}^T \boldsymbol{\Omega}^{-1} \mathbf{1}}{\boldsymbol{\Psi}^T \boldsymbol{\Omega}^{-1} \boldsymbol{\Psi}} \tag{8}$$

Finally, the weights of the network are calculated from the equation:

$$\mathbf{w_o} = \sum_{i=1}^{n} a_{o,i} d_i \boldsymbol{\phi}(\mathbf{x}_i) \tag{9}$$

2.2 Fuzzy Sets

Each Multi-Input Multi-Output (MIMO) fuzzy system can be divided in a number of Multi-Input Single-Output (MISO) fuzzy sets. Thus, in this work, we consider only MISO Fuzzy Sets, as the conclusions can be generalized for the MIMO case: f:U$\subset \mathbb{R}^n \mapsto$V$\subset \mathbb{R}$, where U=U$_1\times$U$_2 \times \ldots$U$_n \subset \mathbb{R}^n$ is the input space and V$\subset \mathbb{R}$ is the output space.

Each fuzzy system $y = f(X) = f(x_1, x_2, \ldots, x_n), X \in U = U_1 \times U_2 \times \ldots \times U_n = U \in \mathbb{R}^n, A_{ij}$ a fuzzy set in $U_j, (j = 1, 2, \ldots, n, i = 1, 2, \ldots, N)$, may be expressed in the following form:

$$y = \frac{\prod_{j=1}^{n} y_i \left[A_{ij}(x_j)\right]}{\sum_{i=1}^{n} \prod_{j=1}^{n} \left[A_{ij}(x_j)\right]} = \sum_{i=1}^{n} \left[\frac{\prod_{j=1}^{n} A_{ij}(x_j)}{\sum_{i=1}^{n} \prod_{j=1}^{n} \left[A_{ij}(x_j)\right]} \right] y_i \tag{10}$$

Thus, a fuzzy set can be represented as a linear combination of the functions:

$$\frac{\prod_{j=1}^{n} A_{ij}(x_j)}{\sum_{i=1}^{n} \prod_{j=1}^{n} \left[A_{ij}(x_j)\right]}, i = 1, 2, \ldots, N \tag{11}$$

These functions can be defined as *basis functions* of a fuzzy system, thus are referred as "Fuzzy Basis Functions". More specifically, fuzzy basis functions are defined as:

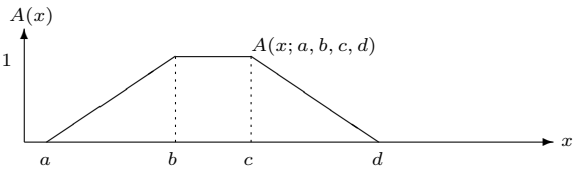

Figure 2: Pseudo-trapezoid function $A(x; a, b, c, d, h)$ with $a < b < c < d$

$$B_i(X) = \frac{A_i(X)}{\sum_{i=1}^{N} A_i(X)} = \frac{\prod_{j=1}^{n} A_{ij}(x_j)}{\sum_{i=1}^{n} \prod_{j=1}^{n} \left[A_{ij}(x_j)\right]} \tag{12}$$

where $A_i(X) = A_i(x_1, x_2, \ldots, x_n), = \prod_{j=1}^{N} A_{ij}(x_j), i = 1, 2, \ldots, N$ and a fuzzy set can be defined as:

$$f(X) = \sum_{i=1}^{N} B_i(X) y_i \tag{13}$$

Let $U \in \mathbb{R}$ the input space and the fuzzy sets $A_i, i = 1, 2, \ldots N$in U are linguistic variables in fuzzy IF-THEN rules and $A_i(x), i = 1, 2, \ldots N$ their fuzzy membership functions. A pseudo-trapezoid function is a continuous function, defined as:

$$A(x; a, b, c, d, h) = \begin{cases} I(x) & x \in [a, b) \\ h & x \in (b, c) \\ D(x) & x \in (c, d] \\ 0 & x \in U - [a, d] \end{cases} \tag{14}$$

where $a \leq b \leq c \leq d, a < d, I(x) \geq 0$ is a monotonically increasing function in $[a, b)$ and $D(x) \leq 0$ is a monotonically decreasing function in $(c, d]$. In the case where a membership function of a fuzzy set A is pseudo-trapezoid, it is then called "Pseudo-trapezoid Membership Function" (figure 2).

Fuzzy sets $A_1, A_2, \ldots, A_N$ constitute a **complete** partition in U if for each $x \in U$, exists $A_i, 1 \leq i \leq N$, with $A_i(x) \geq 0$.

Fuzzy sets $A_1, A_2, \ldots, A_N$ are **consistent** if $A_i(x_0) = 1$ for some $x_0 \in U$, then for every $j \neq i, A_j(x_0) = 0$.

A Fuzzy set A is **normal** in U, if $A(x) \geq 0$ for every $x \in U$and there exists $x_0 \in U$ that $A(x_0) = 1$.

A normal, consistent and complete fuzzy set can be shown in figure 3.

Theorem: If the fuzzy sets $A_i(x), i = 1, 2, \ldots, N$, defined in $U = [a, b]$ are normal, consistent and complete, with pseudo-trapezoid membership functions $A_i(x) = A_i(x; a, b, c, d), i = 1, 2, \ldots, N$ and $A_1 < A_2 < \ldots < A_N$ and

$$f(x) = \sum_{i=1}^{N} B_i(x) y_i \tag{15}$$

a fuzzy set, where $B_i(x), i = 1, 2, \ldots, N$ are fuzzy basis functions, then, for a given function $g(x)$, continuous in

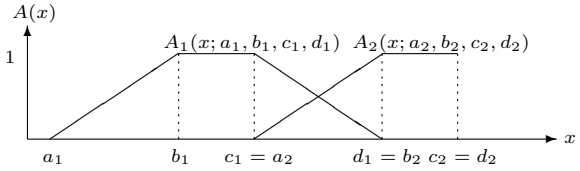

Figure 3: Normal,consistent and complete fuzzy set, with pseudo-trapezoid membership functions $A_1(x; a_1, b_1, c_1, d_1)$ and $A_2(x; a_2, b_2, c_2, d_2)$

$U = [a, b]$ and $\epsilon > 0$ a real number, there exists a fuzzy set $f(x)$, such as [15]:

$$sup_{x \in U} |g(x) - f(x)| < \epsilon \tag{16}$$

2.3 Kernel of the transformation

Let m be the dimension of the input space of the Support Vector Machine and n the dimension of the feature (output) space. In order to find a kernel $\mathrm{K}(\mathbf{x}, \mathbf{x}_i)$ that can be used in a Support Vector Machine, there are two different approaches [14]:

- Find directly the kernel $\mathrm{K}(\mathbf{x}, \mathbf{x}_i)$ that satisfies the Mercer's Theorem
- Find the function $\boldsymbol{\phi(x)}$ that performs the mapping from the input to the feature space

Following the second approach, the kernel will result from the mapping function as:

$$K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\phi(x)} \cdot \boldsymbol{\phi(x}_i) = \sum_{j=0}^{n} \boldsymbol{\phi}_j(\mathbf{x}) \cdot \boldsymbol{\phi}_j(\mathbf{x}_i) \tag{17}$$

where $\mathbf{x}_i$ is one of the n support vectors and $\mathbf{x}$ is an input vector. The optimal hyperplane in the feature space is given by [14]:

$$\sum_{j=0}^{n} w_j \boldsymbol{\phi}_j(x) = 0 \tag{18}$$

and a fuzzy system can be represented as linear combination of fuzzy basis functions as is proved in [16]:

$$f(\mathbf{x}) = \sum_{i=1}^{N} B_i(\mathbf{x}) y_i = 0 \tag{19}$$

A fuzzy basis function $B_i(\mathbf{x})$ is given by:

$$B_i(\mathbf{x}) = \frac{\prod_{j=1}^{k} A_{i_j}^j(x_j)}{\sum_{i_1 i_2 \ldots i_k \in I} \prod_{j=1}^{k} A_{i_j}^j(x_j)} \tag{20}$$

since the denominator is a function of N vectors and not of a single vector, as $\boldsymbol{\phi}(\mathbf{x})$, there seems to be an inconsistency. However, by combining the two relations, the result is:

$$f(\mathbf{x}) = \sum_{i=1}^{N} B_i(\mathbf{x}) y_i = B_1(\mathbf{x}) y_1 + B_2(\mathbf{x}) y_2 + \ldots + B_N(\mathbf{x}) y_N = 0 \tag{21}$$

and since a membership function $A_i(\mathbf{x})$ is given by:

$$A_i(\mathbf{x}) = A_i(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} A_{ij}(x_j) \tag{22}$$

we are lead to:

$$\frac{1}{\sum_{i=1}^{N} A_i(\mathbf{x})} \big(A_1(\mathbf{x}) y_1 + A_2(\mathbf{x}) y_2 + \ldots + A_N(\mathbf{x}) y_N \big) = 0 \tag{23}$$

Since $\sum_{i=1}^{N} A_i(\mathbf{x}) \neq 0$, we conclude:

$$\sum_{i=1}^{N} A_i(\mathbf{x}) y_i = 0 \tag{24}$$

2.4 Clustering of the input space

It is now obvious that the fuzzy basis functions may be used to perform the transformation to the feature space. Since the property of the Fuzzy Systems as "universal approximators" [15] should be satisfied as the optimal hyperplane needs to be able to "learn" any function and construct the optimal hyperplane in the feature space, the support vectors should cluster the input space in a way that the defined fuzzy sets be normal, consistent and complete with pseudo-trapezoid membership functions.

Each constituent of an input vector belongs to a fuzzy set A_{ij}^j, $(i = 1, \ldots, m), (j = 1, \ldots, n)$, where n is the number of the support vectors. Every fuzzy set A_{ij}^j has a triangular membership function $A_{ij}^j(x_j)$. The points of these functions for which $A_{ij}^j(x_j) = 1$ are those that correspond to the constituents of the support vectors. The width of the triangular functions is not predefined, but defined by the support vectors. Thus, considering 3 support vectors, with consecutive coordinates, the width of the membership function whose center is defined by the "middle" support vector, is defined by the centers of the two other membership functions. This representation is chosen in order that the fuzzy sets be normal, consistent and complete.

The construction of the fuzzy membership functions is done in the following way: After the optimization algorithm is performed in a given training set, the support vectors are selected. Let $\mathbf{x}_i$ one of them. Then, each fuzzy set $A_i^j(x_j)$ has a triangular membership function $A_i^j(x_j) = \triangle(a_{i-1}^j, a_i^j, a_{i+1}^j)$. All the fuzzy membership functions are constructed this way, using all n support vectors. The fuzzy basis functions which will be used for the transformation kernel turn up from these membership functions.

A clustering of the input space by a fuzzy set is depicted in figure 4.

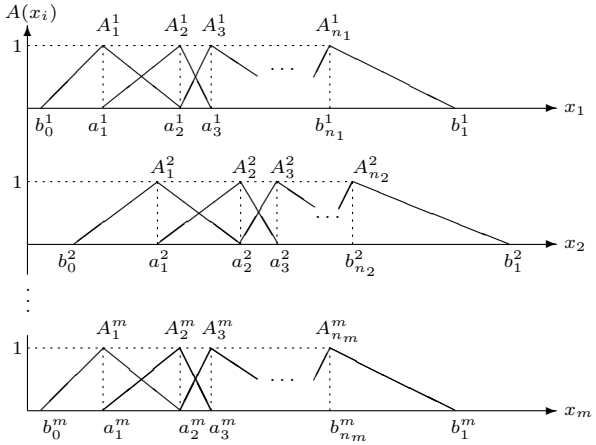

Figure 4: Clustering of the input space by a fuzzy set A

2.5 Definition of the Transformation Function

The membership function of a fuzzy set $A_{i_1 i_2 \ldots i_n} = A_{i_1} \times A_{i_2} \times \ldots \times A_{i_n}, i_1 i_2 \ldots i_n \in I$, with $I = i_1 i_2 \ldots i_n$ where $i_j = 1, 2, \ldots, N_j, j = 1, 2, \ldots, n$ is defined as:

$$A_{i_1 i_2 \ldots i_m}(\mathbf{x}) = A_{i_1}^1(x_1) \cdot A_{i_2}^2(x_2) \cdot \ldots \cdot A_{i_m}^m(x_m) \quad (25)$$

using the product inference operator and $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ is an input vector. It should be noticed that only those membership functions that are defined by the support vectors are used for the construction of the kernel.

Assuming triangular fuzzy membership functions, and the product operator, The transformation function $\boldsymbol{\phi(x)}$ is constructed in the way that will be described.

Each fuzzy membership triangular function has the following general form:

$$\triangle(x; a, b, d) = \begin{cases} I(x) & x \in [a, b) \\ 1 & x = b \\ D(x) & x \in (b, d] \\ 0 & x \in U - [a, d] \end{cases} \quad (26)$$

where $U \subset \mathbb{R}$ is the input space and $I(x) = \frac{x-a}{b-a}, I(x) = \frac{x-d}{b-d}$ and comprises a special case of the trapezoidal membership function with $b = c$. Replacing a, b, d with the coordinates of the support vectors, leads to:

$$A_{i_j}^j(x; a_{i_j-1}^j, a_{i_j}^j, a_{i_j+1}^j) = \begin{cases} \frac{x - a_{i_j-1}^j}{a_{i_j}^j - a_{i_j-1}^j} & x \in \left[a_{i_j-1}^j, a_{i_j}^j\right) \\ 1 & x = a_{i_j}^j \\ \frac{x - a_{i_j+1}^j}{a_{i_j}^j - a_{i_j+1}^j} & x \in \left[a_{i_j}^j, a_{i_j+1}^j\right) \\ 0 & x \in U - \left[a_{i_j-1}^j, a_{i_j+1}^j\right] \end{cases} \quad (27)$$

and obviously,$i_1 i_2 \ldots i_m \in I$ and $I = \{i_1 i_2 \ldots i_m | i_j = 1, 2, \ldots, n\}$

Since we need only the functions that are determined by the support vectors, it necessary to define an index set that satisfies this demand, in order to select the correct subset of the fuzzy membership functions. Using the previous assumptions for the partition of the input space, the following relationship occurs that describes explicitly the index set I':

$$I' = \left\{ i_1 i_2 \ldots i_m | i_j = 1, 2, \ldots, n | b_{i_1}^1, b_{i_2}^2, \ldots, b_{i_m}^m \in X_S \right\} \quad (28)$$

where $X_S = \{\mathbf{x}_i\}_{i=1}^n$ is the set of the support vectors and obviously $I' \subset I$.

Fuzzy membership functions are given by:

$$B_{i_1 i_2 \ldots i_m}(\mathbf{x}) = \frac{\prod_{j=1}^m A_{i_j}^j(x_j)}{\sum_{i_1 i_2 \ldots i_m \in I'} \prod_{j=1}^m A_{i_j}^j(x_j)} \quad (29)$$

From the definition of these functions becomes obvious that exist n functions, equal to the number of the support vectors. Thus, the index set I' has n members. Since $\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}_0(x)\boldsymbol{\phi}_1(x) \ldots \boldsymbol{\phi}_n(x)]^T$, replacing $\boldsymbol{\phi}_i(\mathbf{x})$ with $A_{i_1 i_2 \ldots i_m}(\mathbf{x})$ using:

$$\boldsymbol{\phi}_i(\mathbf{x}) = A_i(\mathbf{x}) \equiv A_{i_1 i_2 \ldots i_m}(\mathbf{x}) \quad (30)$$

where $i_1 i_2, \ldots, i_m | i_j = 1, 2, \ldots, m | (b_{i_1}^1, b_{i_2}^2, \ldots, bi_m{}^m) \equiv \mathbf{x}_i$ is a support vector. Finally,

$$\boldsymbol{\phi}(\mathbf{x}) = [\boldsymbol{\phi}_0(x)\boldsymbol{\phi}_1(x) \ldots \boldsymbol{\phi}_n(x)]^T = [A_0(x)A_1(x) \ldots A_n(x)]^T \quad (31)$$

which leads us to the kernel function:

$$K(\mathbf{x}, \mathbf{x}_i) = \boldsymbol{\phi}(\mathbf{x}) \cdot \boldsymbol{\phi}(\mathbf{x}_i) = \sum_{j=0}^n \boldsymbol{\phi}_j(\mathbf{x}) \cdot \boldsymbol{\phi}_j(\mathbf{x}_i) = \sum_{j=0}^n A_j(\mathbf{x}) \cdot A_j(\mathbf{x}_i) \quad (32)$$

Since $A_j(\mathbf{x})$ are triangular functions, constructed as previously described, the final relation for the kernel is:

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^n \prod_{j=1}^m \triangle^j(x_{i-1}^j, x_i^j, x_{i+1}^j) \cdot \triangle^j(x_{i-1}^{`j}, x_i^{`j}, x_{i+1}^{`j}) \quad (33)$$

where $\mathbf{x}_i = (x_i^1, x_i^2, \ldots x_i^m), i = (1, 2, \ldots, n)$

2.6 Satisfaction of the Mercer's theorem

A basic requirement for a chosen kernel $K(\mathbf{x}, \mathbf{x}_i)$ is to satisfy *Mercer's* theorem [14], in order to be able to be analyzed in a series with positive coefficients:

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{i=1}^{\infty} \lambda_i \boldsymbol{\phi}_i(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}_i) \quad (34)$$

Since the kernel has been constructed in the way presented above, the permutation and the ability to be analyzed in a series is a priori secured. However, this can be proved if the following relation stands: $\int_a^b \int_a^b K(\mathbf{x}, \mathbf{x}') \Psi(x) \Psi(x') dx dx' \geq 0$

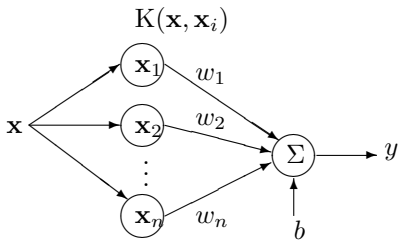

Figure 5: The fuzzy support vector machine network:K is the kernel of the transformation, $\mathbf{x}_i$ is a support vector, $\mathbf{x}$ is an input vector, $\mathbf{w}_i$ a weight, b the bias and y the output of the network.

for every $\Psi(x)$. It is:$\int_a^b \int_a^b K(\mathbf{x},\mathbf{x}')\Psi(x)\Psi(x')dxdx' = \int_a^b \int_a^b \left[\sum_{i=1}^n A_i(\mathbf{x})\cdot A(\mathbf{x}')\right]\Psi(x)\Psi(x')dxdx' = \sum_{i=1}^n I_j = \mathbf{I}$

Since $I_j = \int_a^b \int_a^b A_j(x)\Psi(x)d\mathbf{x}A_j(x')\Psi(x')d\mathbf{x}' = \int_a^b A_j(x)\Psi(x)d\mathbf{x}\cdot\int_a^b A_j(x')\Psi(x')d\mathbf{x}' \geq 0$ It becomes obvious that $I \geq 0$ as a sum of non-negative quantities. Thus, Mercer's theorem is satisfied not only from the chosen kernel, but from *every* kernel that can be written in the form of an *inner product*.

2.7 Learning/Network's Construction

Each neuron in the hidden layer can be viewed as a result of a fuzzy "IF-THEN" rule. This means that knowledge can be easily extracted from a fuzzy SVM network. The network has an input layer, a single hidden layer and an output layer and can be seen on figure 5.

Each support vector represents a fuzzy rule. Let $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ an input vector, $\mathbf{x_i} = [x_{i1}, x_{i2}, \ldots, x_{im}]$ a support vector and ψ_i its response. A fuzzy rule can be stated as:

> IF x_1 is around x_{i1} AND x_2 is around x_{i2} AND ... x_m is around x_{im} THEN $\mathbf{x}$ belongs to ψ_i

3 Feature Extraction

In order to provide standardized descriptions of audio-visual (AV) content, MPEG-7 standard [1] specifies a set of descriptors, each defining the syntax and the semantics of an elementary visual low-level feature *e.g.*, color, shape. In this work, the problem of image classification is based on the use of three MPEG-7 visual descriptors which are extracted using the aceToolbox, developed within the aceMedia project[7] and is based on the architecture of the MPEG-7 eXperimentation Model [9]. A brief overview of each descriptor is presented below and more details can be found in [8]

Color Layout Descriptor (CLD) is a compact and resolution-invariant MPEG-7 visual descriptor defined in the YCbCr color space and designed to capture the spatial distribution of color in an image or an arbitrary-shaped region. The feature extraction process consists of four stages.

Scalable Color Descriptor(SCD) is a Haar-transform based encoding scheme that measures color distribution over an entire image, in the HSV color space, quantized uniformly to 256 bins. To reduce the large size of this representation, the histograms are encoded using a Haar transform.

Edge Histogram Descriptor (EHD) captures the spatial distribution of edges. Four directions of edges (0°, 45°, 90°, 135°) are detected in addition to non-directional ones. The input image is divided in 16 non-overlapping blocks and a block-based extraction scheme is applied to extract the five types of edges and calculate their relative populations.

As it will become obvious in section 4, where sample images from the dataset we used, these descriptors were selected in order to efficiently capture the specific features of the beach/urban problem. A typical beach photo contains the characteristic blue of the sky and the sea and the grey of the sand, where in urban images the sky is still present, but also some other characteristic colors, e.g. from the road or the vegetation. Apart from that, urban images typically contain horizontal and vertical edges, as buildings are present and the textures of e.g. sand and sea are different than those of e.g trees and road. All these features are efficiently captured by the selected MPEG-7 descriptors and fused by the proposed network architecture.

4 Experimental Results

In order to create an input vector for the network, all the descriptors that are used are merged into a unique vector. This method is called *merging fusion*. Let $D_1, D_2, \ldots, D_M$ the M considered descriptors, where each one is represented in a form of a vector. The *merged* descriptor is formed as:

$$D_{merged} = [D_1|D_2|\ldots|D_M]$$

In order to avoid scale effects, all features should have more or less the same numerical values. However, in our case, the MPEG-7 descriptors are already scaled to integer values of equivalent magnitude.

The aceMedia content repository database [2] was used during these experiments. More specifically, content of the Personal Content Services database was used. It consists of 767 high quality color images divided in two classes *beach* and *urban*. All the results using a single descriptor are presented in table 1, while in table 2 results using all 4 combinations of the 3 selected descriptors are presented. For the training dataset 40 images from the

[2]http://driveacemedia.alinari.it/

Figure 6: Representative Images - First Row:Beach Images, Second Row: Urban Images

Descriptor	Classification Rate
EH	82.5%
CL	83.5%
SC	83.6%

Table 1: Classification rate using different MPEG-7 descriptors: edge histogram (EH), color layout (CL) and scalable color (SC)

beach category and 20 from the *urban* were used. The remaining 707 (406 from *beach* and 301 from *urban*) were used for the evaluation. A few representative images from both categories can be seen in figure 6.

The results are better compared to those in [4] with the use of the Falcon-ART neural network but again do not reach those of the "Back-propagation fusion" method. However, the use of the Fuzzy Support Vector Machine network is able to extract linguistic fuzzy rules. An example of an extracted fuzzy rule for the case of the Edge Histogram descriptor follows:

> IF the number of 0 ° edges on the *upper* part of the image is *low* AND the number of 45 ° edges on the *upper* part of the image is *medium* AND ... AND the number of non-directional edges on the *lower* part of the image is *high*, THEN the image belongs to class *Beach*

5 Conclusions and Future Work

The proposed network was applied successfully to the problem of image classification using and fusing MPEG-7 descriptors. Best results were achieved using all three descriptors. However fusion was useful as it can provide a linguistic description of the underlying classification mechanism. Future work will aim to use more MPEG-7 descriptors and more classes. Additionally, this classification method may be extended in matching the segments of an image with predefined object models.

6 Acknowledgements

This work was supported by the EU project aceMedia "Integrating knowledge, semantics and content for user centered intelligent media services" (FP6-001765).

Descriptor	Classification Rate
EH+CL	83.4%
EH+SC	86.6%
CL+SC	87.7%
EH+CL+SC	91.2%

Table 2: Classification rate using all combinations of the selected MPEG-7 descriptors: edge histogram (EH), color layout (CL) and scalable color (SC)

References

[1] Shih-Fu Chang, Thomas Sikora, and Atum Puri. Overview of the mpeg-7 standard. *IEEE trans. on Circuits and Systems for Video Technology*, 11(6):688–695, 2001.

[2] Ya-Chun Cheng and Shu-Yuan Chen. Image classification using color, texture and regions. *Image and Vision Computing*, (21):759–776, 2003.

[3] S. Gao W.-K. Sung D.H. Wang, Q. Tian. News sports video shot classification with sports play field and motion features. *ICIP04*, pages 2247–2250, 2004.

[4] E.Spyrou, H.LeBorgne, T.Mailis, E.Cooke, Y.Avrithis, and N.O'Connor. Fusing mpeg-7 visual descriptors for image classification. In *International Conference on Artificial Neural Networks (ICANN)*, 2005.

[5] Bo Yuan George J. Klir. *Fuzzy Sets and Fuzzy Logic - Theory and Applications*. Prentice Hall, 1995.

[6] J.Stauder, J.Sirot, H.Le Borgne, E.Cooke, and N.O'Connor. Relating visual and semantic image descriptors. In *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, London, U.K.*, 2004.

[7] I. Kompatsiaris, Y. Avrithis, P. Hobson, and M.G. Strinzis. Integrating knowledge, semantics and content for user-centred intelligent media services: the acemedia project. Proc. of WIAMIS 04, Portugal, April 21-23, 2004., 2004.

[8] B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE trans. on Circuits and Systems for Video Technology*, 11(6):703–715, 2001.

[9] MPEG-7. Visual experimentation model (xm) version 10.0. ISO/IEC/ JTC1/SC29/WG11, Doc. N4062, 2001.

[10] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE t. PAMI*, 22(12):1349–1380, 2000.

[11] J.A.K. Suykens, J. De Brabanter, L. Lukas, and J Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Elsevier Neurocomputing*, 48:85–105, 2002.

[12] M. Szummer and R.W. Picard. Indoor-outdoor image classification. In *IEEE international workshop on content-based access of images and video databases,*, 1998. Bombay, India.

[13] A. Vailaya, A. Jain, and H.-J Zhang. On image classification: City images vs. landscapes. *Pattern Recognition*, 31(12):1921–1936, 1998.

[14] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[15] Xiao-Jun Zeng and Madan G. Singh. Approximation theory of fuzzy systems - siso case. *IEEE Transactions on Fuzzy Systems*, 2(2), 1994.

[16] Xiao-Jun Zeng and Madan G. Singh. Decomposition property of fuzzy systems and its applications. *IEEE Transactions on Fuzzy Systems*, 4(2), 1996.