Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 40 (2007) 19–32

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error

Wing W.Y. Ng[a,b,*], Andres Dorado[c], Daniel S. Yeung[a,b], Witold Pedrycz[d], Ebroul Izquierdo[c]

[a]Department of Computing, The Hong Kong Polytechnic University, China
[b]Media and Life Science Computing Lab, Shenzhen Graduate School, Harbin Institute of Technology, China
[c]Department of Electronic Engineering, Queen Mary, University of London, UK
[d]Department of Electrical and Computer Engineering, University of Alberta, Canada

Received 31 October 2005; accepted 6 July 2006

Abstract

Image classification arises as an important phase in the overall process of automatic image annotation and image retrieval. In this study, we are concerned with the design of image classifiers developed in the feature space formed by low level primitives defined in the setting of the MPEG-7 standard. Our objective is to investigate the discriminatory properties of such standard image descriptors and look at efficient architectures of the classifiers along with their design pursuits. The generalization capabilities of an image classifier are essential to its successful usage in image retrieval and annotation. Intuitively, it is expected that the classifier should achieve high classification accuracy on unseen images that are quite "similar" to those occurring in the training set. On the other hand, we may assume that the performance of the classifier could not be guaranteed in the case of images that are very much dissimilar from the elements of the training set. To follow this observation, we develop and use a concept of the localized generalization error and show how it guides the design of the classifier. As image classifier, we consider the usage of the radial basis function neural networks (RBFNNs). Through intensive experimentation we show that the resulting classifier outperforms other classifiers such as a multi-class support vector machines (SVMs) as well as "standard" RBFNNs (viz. those developed without the guidance offered by the optimization of the localized generalization error). The experimental studies reveal some interesting interpretation abilities of the RBFNN classifiers being related with their receptive fields.

Keywords: Image classification; Radial basis functions neural networks; MPEG-7; Support vector machines; Generalization error

1. Introduction

A vast amount of digital images are become omnipresent these days call for an intensified effort towards building efficient needs for their automatic annotation and retrieval mechanisms. Classification of digital images becomes one of the fundamental activities one could view as the fundamental prerequisite for all other image processing pursuits. In image classification we could follow the general paradigm of pattern recognition. In pattern recognition, each object is described by a collection of features that forms a multidimensional space in which all discrimination activities take place. Various classifiers, both linear and nonlinear, become available at this stage including support vector machines (SVM), linear classifiers, polynomial classifiers, radial basis function neural networks (RBFNNs), fuzzy rule-based systems, etc. No matter what classifier has been chosen, a formation of a suitable feature becomes of paramount relevance. The problem of forming of the feature space in the case of images is even more complicated. On one hand, we have a lot of different alternatives. On the other

* Corresponding author. Department of Computing, The Hong Kong Polytechnic University, China. Tel.: +852 2766 4901; fax: +852 2774 0842.

E-mail addresses: cswyng@comp.polyu.edu.hk (W.W.Y. Ng), andres.dorado@elec.qmul.ac.uk (A. Dorado), csdaniel@comp.polyu.edu.hk (D.S. Yeung), pedrycz@ee.ualberta.ca (W. Pedrycz), ebroul.izquierdo@elec.qmul.ac.uk (E. Izquierdo).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.07.002

hand, the diversity of images contributes to the elevated level of complexity and difficulty. In images, we encounter a variety of images showing different shapes, colors, texture, etc yet belonging to the same class. An image could be described by an intensity of color of each pixel or even better by some descriptors. In this study, our objective is to explore and quantify the discriminatory properties of the MPEG-7 image descriptors in classification problems. Those are explored in conjunction to two main categories of classifiers such as SVMs and RBFNNs.

Unfortunately, it becomes obvious that any classifier requiring high training accuracy may not achieve good generalization capability. Since both target outputs and distributions of the unseen samples are unknown, it is impossible to compute the generalization error in a direct way. There are two major approaches to estimate the generalization error, namely, analytical model and cross-validation (CV). In general, analytical models bound above the generalization error for any unseen samples and do not distinguish trained classifiers with the same number of effective parameters but different values of parameters. Thus, the error bounds given by those models are usually loose [1]. The major problem of analytical models is the estimation of the number of effective parameters of the classifier, which could be solved by using the VC-dimensions [2]. The VC-dimension of a classifier is defined as the largest number of samples that can be shattered by this classifier [2]. However, only loose bound of VC-dimensions could be found for nonlinear classifiers, e.g. neural networks, and this puts a severe limitation on the applicability of analytical models to nonlinear classifiers, except the SVM [3]. Although CV uses true target outputs for unseen samples, it is time consuming for large datasets and *CL* classifiers must be trained for *C*-fold CV and *L* choices of classifier parameters. CV methods estimate the expected generalization error instead of its bound, thus they do not guarantee the finally built classifier to have good generalization capability [1].

In image classification, one may not expect a classifier trained using one category of images (say, animals) to correctly classify images coming from some other categories (e.g. vegetables). In this case, one may revise the training dataset by adding training samples of vegetables and re-train the classifier to include the new class of images. For example, in our dataset we have images of cow but not airplane, thus we could not expect the classifier trained using our dataset to correctly recognize an airplane. It is expected that an image classifier work well for those classes that have been used to train it assuming images belonging to the same class are conceptually similar and such that their descriptor values should also be similar. That is, unseen samples similar to the training samples, in terms of sup-type of distance in the feature space is smaller than a given threshold, are considered to be more important. Thus, in the evaluation of the generalization capabilities of the image classifiers, one may ignore those images that are totally dissimilar to those existing in the training set.

In general, image classification problems are multi-class classification problems and difficult to find a classifier with good generalization properties. In this work, we aim to find an image classifier featuring better generalization capability and interpretability with respect to domain knowledge in image classification. We concentrate on finding an optimal number of receptive fields for RBFNNs to classify the images with lower generalization error to unseen images.

We organize the study in the following manner. The starting point is a discussion on the formation of the feature space based upon the framework of descriptors being available in the MPEG-7 standard. These issues are covered in Section 2. We provide a brief introduction to image classifiers in Section 3. The localized generalization error model (R_{SM}^*) and the corresponding approach to the selection of the architecture of the network are described in Sections 4 and 5, respectively. We present a comprehensive suite of experimental studies in Section 5. Concluding comments are covered in Section 6.

2. MPEG-7 feature space

In this section, we elaborate on the feature space arising within the framework of MPEG-7. The MPEG-7 descriptors are useful for low-level matching and provide a great flexibility for a wide range of applications.

2.1. MPEG-7 descriptors

MPEG-7, formally known as multimedia content description interface is an ISO/IEC standard developed by the moving picture experts group (MPEG) for description and search of audio and video content; refer to www.chiariglione.org/mpeg/. In contrast with the earlier standards known as MPEG-1, MPEG-2, and MPEG-4 that are focused on coding and representation of audio–visual content, On the other hand, MPEG-7 moves forward and becomes more general by embracing a description of multimedia content [4].

MPEG-7 has emerged as a cornerstone of the development of a wide spectrum of applications dealing with audio, speech, video, still pictures, graphics, 3D models, and alike. In a nutshell, the MPEG-7 environment delivers a comprehensible metadata description standard that is interoperable with other leading standards such as SMPTE Metadata Dictionary, Dublin Core, EBU P/Meta, and TV Anytime; refer to www.ebu.ch/trev_284-mulder.pdf. Initially, MPEG-7 was focused more on web-based applications and annotation tools (e.g. Refs. [5,6]). Nowadays, it is being drifted to other domains such as education, video surveillance, entertainment, medicine and biomedicine.

The ultimate objective of MPEG-7 is to provide interoperability among systems and applications used in generation, management, distribution, and consumption of audio–visual content descriptions. Such descriptions of streamed (live) or

Table 1
MPEG-7 visual descriptors used to characterize still images

Color descriptors	Texture descriptors	Shape descriptors
Color layout	Texture browsing	Region-based shape
Color structure	Homogeneous texture	Contour-based shape
Dominant color	Edge histogram	
Scalable color		

stored on various media help either users or applications in identifying, retrieving, or filtering essential audio–visual information, cf. Refs. [4,7].

MPEG-7 specifies standardized descriptors and description schemes for audio and video, as well as integrated multimedia content. Also standardized is a description definition language that allows for new descriptors and description schemes to be defined. MPEG-7 descriptors define syntax and semantics of features of audio–visual content. MPEG-7 allows these descriptions to be at different perceptual and semantic levels. At the lowest abstraction level, such descriptors may include shape, texture, and color. At the highest abstraction level, they may include events, abstract concepts, and so forth. Table 1 presents a list of the basic MPEG-7 visual descriptors applicable to still images.

We use the proposed feature vector structure [8] that combines features extracted from MPEG-7 color and texture descriptors. The structure keeps the original topology in order to preserve syntax and embedded semantics on these descriptors. The proposed feature space is constructed using five histogram-based MPEG-7 color and texture descriptors [9]. The color layout descriptor (CLD) effectively represents the spatial distribution of color in a very compact form by controlling the number of coefficients enclosed in the descriptor. The scalable color descriptor (SCD) is a color histogram in Hue-saturation-value color space, which is encoded by a Haar transform. The color structure descriptor (CSD) captures both color content and information about the structure of this content. In contrast to ordinary color histogram this descriptor uses a sliding window to capture color distributions by groups of pixels rather than by each pixel being treated separately. The edge histogram descriptor (EHD) captures spatial distribution of five types of edges. Semi-global and global histograms derived from this descriptor can be combined with other descriptors, e.g. color structure, to obtain accurate matching between images of a similar semantics. The homogeneous texture descriptor (HTD) is useful to represent image data as a mosaic of homogeneous textures. This descriptor provides a useful quantitative representation of the texture by using the first and second moments of the energy in the frequency domain.

2.2. *Selection of the MPEG-7 descriptor elements*

The selection of the descriptor elements becomes an important issue in defining the structure of the feature space. This is an open and difficult problem to define an absolutely optimal feature space due to the lack of evaluation criteria. Furthermore, any bias towards either color or texture descriptors may not be appropriate. Thus, in this study, we intend to use a suite of descriptors which produces a balanced feature space by including a suitable number of the texture and color descriptors. Table 2 shows the different configurations of color and textures components in the feature space.

As indicated in the last column (Settings), the CLD is set up to the recommended default, which includes six Y coefficients and three each of Cr and Cb coefficients [9]. CSD is adjusted to 64 elements, which are calculated based on approximations done when using the 184-bin descriptor. SCD is fixed to 64 by scaling the quantized representation of Haar coefficients to obtain the desired number of bits. EHD uses its default that corresponds to 80 elements. HTD uses the full-layer; that is 62 elements. The 140-element color and 142-element texture components produce a difference of two elements, which constitutes an acceptable balance. Consequently, the full-size feature vector consists of 282 elements. Afterwards, following the paradigm of pattern recognition, let us treat each image formed in the MPEG-7 feature space as a 282-dimensional numeric vector. The images will be denoted as $\mathbf{x}$, $\mathbf{z}$, $\mathbf{y}$ and alike.

3. Classifiers for image classification

In this section, we discuss several selected architectures of classifiers that are quite often encountered in image classification. It is of interest to investigate their properties in this setting and review some related development strategies.

3.1. *SVMs for image classification*

In image classification, a significant interest has been expressed in classifiers realized as SVMs [2]. In Ref. [10], Barla et al. investigated the potential of SVMs to solve image classification problems. As pointed out by Jain et al. an evident advantage of SVMs classifiers lies in their capabilities to learn from a relatively small number of samples [11]. The class (label) of a new sample is determined by a linear combination of the kernel functions evaluated on a certain subset of the examples—the support vectors and the input. The coefficients of the combination are obtained as a solution to a convex optimization problem occurring at the learning stage. The choice of the kernel relies on prior knowledge of the problem domain.

Chapelle et al. presented an approach to overcome poorly generalization on image classification tasks, because of the high dimensionality of the feature space. They used a SVMs approach because of its good generalization performance that is retained even when the dimension of the input space is very high [12]. Serrano et al. proposed an improved SVM-based approach to indoor/outdoor classification. The approach uses a low-dimensional feature set in which a low

Table 2
Setting of the dimensionality of the feature vector with the reference to different configurations of color and texture components in the MPEG-7 feature space

Descriptor	Number of elements					Settings
CLD	12					12
CSD	32	64	120	184		64
SCD	16	62	64	128	256	64
EHD	80					80
HTD	32	62				62
dim(Color) $= 140$, dim(Texture) $= 142$						
Difference in number of features						2
Total number of features						282

computation complexity is achieved without compromising the accuracy [13]. Yen et al. applied SVM ensembles to adapt binary SVMs to multi-class classification and address the high computational cost for training [14]. Tsai et al. implemented concept-based indexing by combining SOMs and SVMs in a two-stage hybrid classifier [15]. In Ref. [16] Prabhakar et al. proposed another hybrid approach. The classification of images as either picture or graphics is performed by a combination of a rule-based tree classifier and a neural network classifier. Dong and Yang combined SVMs and kNN for hierarchical classification of web images. A threshold strategy called "HRCut" is proposed as a way of handling the difficulty in applying traditional strategies (e.g. rank-based threshold) to decide the winner category for each image [17].

Several methods have been proposed to deal with multi-class problem using SVM [18], such as one-against-all, however they are still based on binary classifications. Moreover, SVM project the images to a very high dimensional kernel space in which one may find difficult to analyze the resulting classifier and relate it to any domain knowledge available when dealing any problems of image classification.

3.2. RBFNNs in image classification

In contrast, the RBFNNs [19,20] operate directly in the input space, i.e. the MPEG-7 feature space we have discussed in Section 2. In later section, we will discuss aspects of the visualization of the receptive fields and their relationships with the images and MPEG-7 descriptors, in particular. The training of RBFNNs consists of two major phases that is unsupervised and supervised learning. In the unsupervised phase, one defines the number of receptive fields and then the clustering algorithm is used to determine the center and the spread of the receptive field. The connections (weights) between the output of the RBFNN and receptive fields are computed by pseudo-inverse, which becomes possible owing to the linear relationship with respect to the parameters to be estimated. The assumed performance index is viewed as the mean squared error (MSE). A selection of a suitable number of the receptive fields of the network becomes a key architectural component to the successful performance of the resulting RBFNN. Fig. 1 illustrates the architecture of the RBFNN. The kth output of the RBFNN is described in the following form

$$f(\mathbf{x}, k) = \sum_{j=1}^{M} w_{jk} \exp\left(\frac{(\mathbf{x}-\mathbf{u}_j)^{\mathrm{T}}(\mathbf{x}-\mathbf{u}_j)}{-2v_j^2}\right), \tag{1}$$

where M, w_{jk}, $\mathbf{u}_j$ and v_j denote the number of the receptive fields, the connection (weight) between the jth receptive field and the kth output, the feature vector of the center and the width of the jth receptive field, respectively. As will be explained later on, the centers of the receptive field capture some representatives of the collection of images.

Each of the outputs $f(\mathbf{x}, k)$ represents a degree of membership of an image to the corresponding class. For the K-class problem, we end up with a network with K outputs. Given this arrangement, an image is assigned to class for which the corresponding output achieves its maximal value.

4. A concept and realization of the localized generalization error

Given the anticipated diversity of images to be classified, one could easily envision that there is no classification algorithm that is capable of carrying out a zero error classification. This straightforward and very much intuitive observation is that when it comes to images that are very much different from those the classifier was exposed during the training phase. In other words, we acknowledge that any classifier comes with some limited generalization capabilities. In terms of the geometry of the feature space, one could note that the discriminatory capabilities of the classifier are indeed confined to some limited neighborhood of the patterns we were provided to train the classifier. Moving outside such neighborhood and assessing the quality of the classifier is somewhat counterproductive. Definitely, it is known and intuitively appealing that we will not be able to improve the performance of the classifier over there. It is

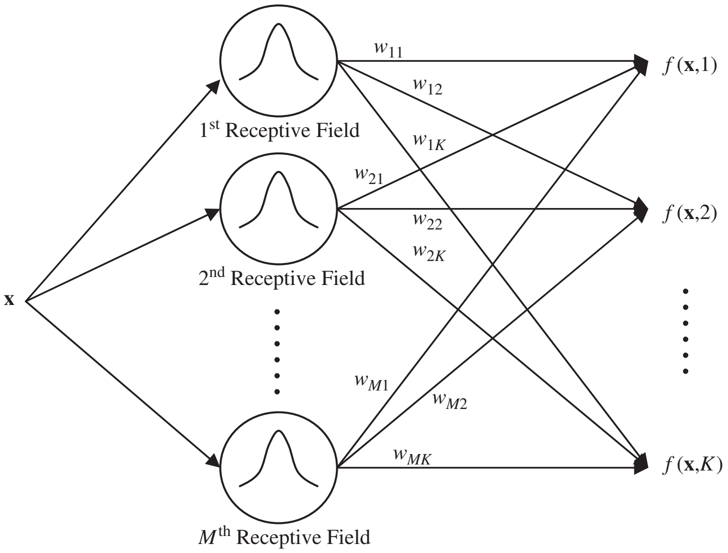

Fig. 1. Architecture of the RBFNN with M receptive fields and K outputs.

therefore better to concentrate on the neighborhood and optimize a classifier to perform the best within such regions. Assuming this point of view, the design of the classifier will be guided by the minimization of the generalization error that is confined in its nature; hence the name of the *localized* generalization error.

To explain the essence of the classifier, let us introduce some basic notation. A finite training data set formed in the n-dimensional feature vectors of the images is denoted by $\mathbf{X}$, $\mathbf{X}=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ that is dim $(\mathbf{x}_b)=n$ and N denotes the number of training samples. A classifier endowed with some vector of parameters $\boldsymbol{\theta}$ is denoted by $f(\mathbf{x};\,\boldsymbol{\theta})$ while the "true" and obviously unknown mechanism of mapping patterns on a collection of classes is denoted by $F(\mathbf{x})$.

For each $\mathbf{x}_b$ let us define its Q-neighborhood to be of the form

$$S_Q(\mathbf{x}_b) = \{\mathbf{x}|\mathbf{x}=\mathbf{x}_b+\Delta\mathbf{x};\ \ |\Delta\mathbf{x}|\leqslant Q \text{ for } \forall i=1,2,\ldots,n\}.$$

All points in $S_Q(\mathbf{x}_b)$ are regarded as unseen samples that are local (viz. similar) to $\mathbf{x}_b$. We anticipate that given the close vicinity of $\mathbf{x}_b$, it is legitimate to expect some sound performance of the classifier in this region. Obviously, the following monotonicity relationship holds: if $\hat{Q}>Q$ then $S_Q(\mathbf{x}_b)\subset S_{\hat{Q}}(\mathbf{x}_b)$.

Let us now form the union of $S_Q(\mathbf{x}_b)$ taken over all elements of $\mathbf{X}$, that is

$$S_Q=\bigcup_{b=1}^{N} S_Q(\mathbf{x}_b).$$

It will be referred to as a Q-union. The graphical illustration of this construct is visualized in Fig. 2.

Being cognizant of the limited and focused nature of any classifier, in its design and the ensuing assessment of the performance, it is legitimate to concentrate on the use of the generalization error computed within the Q-union (R_{SM}) instead of looking at the error for all possible unseen samples (call this one R_{true}). We use the notation R_{SM} to indicate that it is defined based on the stochastic sensitivity measure. This measure is computed by taking the expectation of the squared differences between the classifier outputs for the training sample and the unseen samples in its Q-neighborhood. In Fig. 2, one may notice that the R_{true} bounds for the error for any samples in the entire input space T, i.e. any point in the T in Fig. 2 is included. In contrast, the R_{SM} bounds for the generalization error for points located within the shaded area only, including the training samples represented by white crosses.

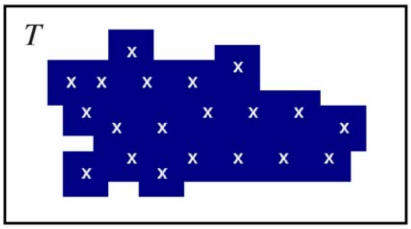

Fig. 2. An illustration of Q-union (S_Q) for some training samples. The crosses are training samples and any point in the shaded area corresponds to some unseen sample.

Thus we obtain the following straightforward relationship:

$$\begin{aligned} R_{true} &= \int_{\forall\mathbf{x}} (F(\mathbf{x})-f(\mathbf{x};\,\boldsymbol{\theta}))^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \int_{\mathbf{x}\in S_Q} (F(\mathbf{x})-f(\mathbf{x};\,\boldsymbol{\theta}))^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &\quad + \int_{\mathbf{x}\notin S_Q} (F(\mathbf{x})-f(\mathbf{x};\,\boldsymbol{\theta}))^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= R_{SM}+R_{res}. \end{aligned} \tag{2}$$

Taking advantage of the Hoeffding's inequalities [21] while not making any assumption about the distribution $p(\mathbf{x})$, the following inequality holds with a probability of $1-\eta$ [22]:

$$R_{SM} = \int_{\mathbf{x}\in S_Q} (F(\mathbf{x}) - f(\mathbf{x};\boldsymbol{\theta}))^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \leqslant \left(\sqrt{R_{emp}} + \sqrt{E_{S_Q}((\Delta y)^2)} + A\right)^2 + \varepsilon = R^*_{SM}, \quad (3)$$

where $\varepsilon = \max((F(\mathbf{x}) - f(\mathbf{x};\boldsymbol{\theta}))^2)\sqrt{\ln(\eta)/(-2l)}$, $E_{S_Q}((\Delta y)^2)$ denotes the stochastic sensitivity measure for the classifier $f(\mathbf{x};\boldsymbol{\theta})$ and $1-\eta$ stands the confidence of the bound.

Let us note that the stochastic sensitivity measure is computed by taking the expectation of the squared output perturbations of a classifier and it indicates the complexity of a classifier [22]. The output perturbation taken with respect to a training sample is defined as the differences between the classifier's outputs of the training sample and the unseen samples located within its Q-neighborhood. We have no prior knowledge as to the unseen samples and thus we assume that they are uniformly distributed around those training samples, i.e. $p(\Delta\mathbf{x})$ is governed by the uniform distribution.

A stochastic sensitivity $(E_{S_Q}((\Delta y)^2))$ is defined as keep here the first part of the expression below:

$$E_{S_Q}((\Delta y)^2) = \frac{1}{l}\sum_{b=1}^{l}\int_{\mathbf{x}\in S_Q(\mathbf{x}_b)} (f(\mathbf{x};\boldsymbol{\theta}) - f(\mathbf{x}_b;\boldsymbol{\theta}))^2 \times p(\Delta\mathbf{x})\,\mathrm{d}\Delta\mathbf{x}. \quad (4)$$

After taking into account the architecture of the RBFN, this leads to the expression [22,23]

$$E_{S_Q}((\Delta y)^2) \approx \frac{1}{3}Q^2\sum_{j=1}^{M}\upsilon_j + \frac{0.2}{9}Q^4 N\sum_{j=1}^{M}\zeta_j, \quad (5)$$

where

$$\upsilon_j = \varphi_j\left(\sum_{i=1}^{n}(\sigma^2_{x_i} + (\mu_{x_i} + u_{ji})^2)/v_j^4\right),$$

$$\mathbf{u}_j = (u_{j1}, u_{j2}, \ldots, u_{jn}),\ \ \zeta_j = \varphi_j/v_j^4,$$

$$A = (\max(F(\mathbf{x})) - \min(F(\mathbf{x}))),$$

$$\varphi_j = (w_j)^2\exp((Var(s_j)/2v_j^4) - (E(s_j)v_j^2)),$$

$$E(s_j) = \sum_{i=1}^{n}(\sigma^2_{x_i} + (\mu_{x_i} + u_{ji})^2), \quad s_j = \|\mathbf{x} - \mathbf{u}_j\|^2,$$

$$\Delta y = f(\mathbf{x};\boldsymbol{\theta}) - f(\mathbf{x}_b;\boldsymbol{\theta}),$$

$$Var(s_j) = \sum_{i=1}^{n}(E_D[(x_i - \mu_{x_i})^4] - (\sigma^2_{x_i})^2 + 4E_D[(x_i - \mu_{x_i})^3](\mu_{x_i} + u_{ji}) + 4\sigma^2_{x_i}(\mu_{x_i} + u_{ji})^2),$$

$$R_{emp} = \frac{1}{N}\sum_{b=1}^{N}(F(\mathbf{x}_b) - f(\mathbf{x}_b;\boldsymbol{\theta}))^2,$$

μ_{x_i} and $\sigma^2_{x_i}$ denote the mean and variance of the ith feature of the pattern, respectively.

The approximation of this measure is realized by removing higher order terms in its Taylor expansion. Similarly, an error of this approximation could be expressed using the Taylor remainder. If required, to improve the approximation more terms could be included in the expression.

Specifically for any RBFNN, when substituting Eq. (4) into Eq. (3), with a probability of $1-\eta$ we have

$$R^*_{SM} \approx \left(\sqrt{\frac{1}{3}Q^2\sum_{j=1}^{M}\upsilon_j + \frac{0.2}{9}Q^4 N\sum_{j=1}^{M}\zeta_j} + \sqrt{R_{emp}} + \sqrt{A}\right)^2 + \varepsilon. \quad (6)$$

From Eq. (3), one may notice that the generalization error upper bound depends on the training error (R_{emp}), stochastic sensitivity measure and some constants.

The two constants A and ε are fixed after the training dataset has been defined. The first constant (A) indicates the range of the target output; in our case it is equal to 1 (since every output represents a degree of membership to the corresponding class and therefore is positioned in the unit interval). The constant ε vanishes when the number of training samples approaches infinity. This indicates that the generalization error will be high if the training dataset consists of a very few training samples. On the other hand, the training error indicates how well the RBFNN learns from the training samples. One could not expect a RBFNN yielding good generalization capabilities if it even could not generalize the training samples. Furthermore, the stochastic sensitivity measure relates to the complexity of the RBFNN. If the RBFNN is too complex, its output changes dramatically when input is changed, thus overfitting may occur because the significant memorization effect. Alluding to the bias/variance dilemma, the classifier with good generalization capabilities should yield low classification bias (training error) and variance (complexity). Therefore the R^*_{SM} model penalizes both the training error and the RBFNN complexity. Hence the best RBFNN should achieve a sound balance between low training error and the complexity of the classifier.

5. The architecture design of RBFNNs

In the sequel, we confine ourselves to RBFNNs with Gaussian receptive fields. We apply a standard clustering algorithm (say, k-Means, self-organizing maps, etc.) to find the location of the receptive fields of the network. Typically, this is done once the number of the receptive fields has been fixed. The choice of this number is not a trivial task and its suitable selection impacts the generalization abilities of the network. To address this issue, we discuss a new algorithm which will lead to the architecture; we are referring to as a maximal coverage classifier with selected generalization (MC^{2}SG) as proposed in Refs. [22,23] For any given threshold value a shown in Eq. (7), the R_{SM} model allows to find the best classifier by maximizing the values of Q, assuming that the MSE of all samples within the Q-Union is smaller than the given threshold value. One can formulate the problem of model selection in the following manner

$$\max_{\theta\in\Lambda} Q \quad \text{s.t. constraint: } R^*_{SM}(Q)\leqslant a. \tag{7}$$

Problem (7) exhibits two design facets. The first one concerns the number of receptive fields ($\boldsymbol{\theta}$) and the second one is the Q for a fixed $\boldsymbol{\theta}$. For every fixed $\boldsymbol{\theta}$ and Q, we can determine the corresponding value of R^*_{SM}. These two parameters ($\boldsymbol{\theta}$ and Q) are independent from each other. For a given training dataset and a fixed value of $\boldsymbol{\theta}$, one could compute the maximum Q value in which the R^*_{SM}bound is less than or equal to a threshold a. So, intuitively the Q value provides an indication on how big the area of coverage of the unseen samples whose generalization errors in MSE are less than a. Thus, a larger Q indicates better generalization of a classifier (in some probabilistic sense). On the other hand, for two classifiers yielding the same value of R^*_{SM} with different values of Q, the one that yields a larger value of Q value exhibits better generalization capability. One may notice that the R^*_{SM} for RBFNN, see Eq. (6), becomes an increasing function with respect to Q. Based on Eq. (6), the value of Q is computed by solving the following fourth order polynomial equation:

$$\left(\sqrt{\frac{1}{3}Q^2\sum_{j=1}^{M}\upsilon_j+\frac{0.2}{9}Q^4N\sum_{j=1}^{M}\zeta_j}+\sqrt{R_{emp}}+\sqrt{A}\right)^2+\varepsilon = a,$$

$$Q^4\frac{0.2}{3}N\sum_{j=1}^{M}\zeta_j+Q^2\sum_{j=1}^{M}\upsilon_j-3(\sqrt{a-\varepsilon}-\sqrt{R_{emp}}-\sqrt{A})^2 = 0. \tag{8}$$

There is a maximum of four solutions for Eq. (8) and the smallest real-valued solution will be used as the final result Q^*. If no real solution exists, we choose the zero as the solution to the problem. Moreover, if the training error is larger than the selected threshold a, the generalization error should not be smaller than a and thus no value of Q could satisfy the constraint given by Eq. (7). Therefore we obtain

$$h(M,Q^*)=\begin{cases}0 & R_{emp}\geqslant a,\\ Q^* & \text{else.}\end{cases} \tag{9}$$

Thus, for the selection of the RBFNN model, Eq. (7) comes in the form

$$\max_{M\in[1,2,\ldots,l]} h(M,Q^*). \tag{10}$$

6. Experiments

In this section, we elaborate on the series of experiments. First, we elaborate on the experimental setup. Next, we report on the experimental results and focus on the interpretation of the network.

6.1. Experimental setup

The experimental material consists of 1000 color images of different size are collected from Corel stock gallery and downloaded from FreeFoto.com (www.freefoto.com). Images are manually categorized into five classes namely animal, building, city view, landscape, and vegetation. Each class comprises 200 images. Ten training and testing datasets are independently generated in a random fashion to run a 10-fold CV, with 60% of the samples being used to train the classifier and the remaining 40% samples for testing.

In general, an image is assigned to a certain class depending upon the object the camera was focused on. The following criteria were used to assign images to classes:

(1) *Animal*: A scene falls into this category when an animal of a visible size appears in the picture. The major difficulty of classifying the images in this class is that the images are the mimetic properties of several species. Some of them may be totally different from the others.
(2) *Building*: A scene is tagged with this class when a building structure of a visible size appears in the picture. Two major difficulties are identified with this category: different types of buildings, i.e. castles, residential houses, warehouses, religious facilities, etc.; and the different distances used in the close-ups.
(3) *City view*: A scene containing panoramic views of cities, specifically buildings, is assigned to this category. The different angles and small size of the man-made structures, among others, are examples of shortcomings to categorize these scenes.
(4) *Landscape*: A scene is placed in this category when the picture depicts scenery with natural elements such as mountains, forest, seashores, etc. No man-made objects are expected to be in this type of images.
(5) *Vegetation*: A scene belongs to this category when nature, i.e. plants, appears in the picture.

The RBFNN that was trained using MC^{2}SG is compared with multi-class SVMs [18] endowed with Gaussian kernels.

Table 3
Experimental results for 10 independent runs

	RBFNN by MC^{2}SG	RBFNN by sequential learning	SVM with Gaussian Kernel
Training accuracy	91.38 ± 1.46	92.25 ± 1.14	89.01 ± 0.06
Testing accuracy	83.90 ± 1.81	79.73 ± 1.03	78.18 ± 1.77
Training time (s)	49.02 ± 1.44	49.85 ± 4.87	706.08 ± 76.52

Reported are mean values and the standard deviations.

The RBFNN consists of five output neurons each of them corresponding to one of the classes occurring in the problem. The receptive fields are formed with the use of the *K*-Means algorithm. A "standard" RBFNN is also considered whose the number of receptive fields is selected through sequential learning [24]. In this mode of learning, one receptive field is being added to the RBFNN until the highest training accuracy has been reached or some pre-selected maximal number of receptive fields has been reached. Among those RBFNNs already formed, the one which yields the highest training accuracy will be selected.

The determination of the constant a in Eq. (9) is realized according to the classifier's output schemes for classification. In our case, the output value of the RBFNN has the range of [0, 1] and misclassification occurs when the squared difference between the RBFNN output and target output is larger than 0.25. Therefore, we select the value of a to be 0.25. Similarly as encountered in other methods, $h(M, Q^*)$ is generally not differentiable with respect to M. Therefore in order to find the optimal solution one must experiment with all possible values of M. Our experimental results show that $h(M, Q^*)$ approaches zero when the classifier becomes too complex, i.e. M becomes too large. Some useful heuristics would be to stop the process once Q approaches zero. Following this observation, we stop the search when the values Q drop below 10% of the maximum value of Q during the experiments.

6.2. Experimental results

The high dimensionality of feature vectors, large number of samples and multiple classes are common in image classification. Both SVM and the proposed RBFNN training with MC^{2}SG are scalable to the high dimensionality of features.

As shown in Table 3, the RBFNN trained with the use of MC^{2}SG outperforms both the RBFNN developed through sequential learning and the SVM returning higher accuracy on the testing set. This is somewhat expected as the traditionally trained RBFNN minimizes only the training accuracy and thus it has no guarantee as to the generalization abilities of the classifier. The average accuracy on the testing set of the RBFNNs trained by MC^{2}SG is 5.73% and 4.17% higher than the multi-class SVM with Gaussian kernel and RBFNN trained by sequential learning which minimizing the training error, respectively. The 5% testing accuracy indicates the RBFNNs trained by MC^{2}SG correctly classify 20 more images out of 400 when being compared with other methods.

Table 4
Distribution of receptive fields among classes

Animal	Building	City view	Landscape	Vegetation
10.30 ± 0.8	9.70 ± 0.6	10.20 ± 0.9	10.10 ± 1.1	9.30 ± 0.8

Furthermore, SVM is originally designed for binary classification and the sound theoretical foundation of SVM is built for the binary classification only. In contrast, RBFNN is suitable for multi-class classification problems and both the R^*_{SM} and MC^{2}SG are designed for multi-class problems. Furthermore, SVM minimizes the generalization error for all unseen samples while the MC^{2}SG minimizes the generalization error for those samples similar to the training samples only. Minimizing the generalization error for unseen samples that are totally different from the training samples may not be meaningful in image classification problems because similarity of images in the same class is based on conceptual matching in addition to low-level matching. On the other hand, if a large group of future images is dissimilar to all of the training samples, one may consider revising the training dataset and retrain the classifiers. In the retraining, the clusters represented by those receptive fields in the RBFNN may be merged together or changed to adapt to the new elements in the class (e.g. *animal* images in a different scenario).

Moreover, the training time of SVM is 14 times longer than the proposed one. The optimization problem solved by the MC^{2}SG is linear with respect to the number of samples while the one for SVM is more than quadratic.

In RBFNN, each of the receptive fields corresponds to a region in the input space. Higher activation values of the receptive field reported with respect to a certain image indicates that the image is closer (similar) to this particular receptive field. The classification result of the RBFNN is based on the weighted activation values of all the receptive fields and thus the similarity between images and the abstract images captured by the receptive fields plays an important role in the image classification using RBFNN. In Table 4, we summarize the distribution of the number of receptive fields with respect to each concept class. A receptive

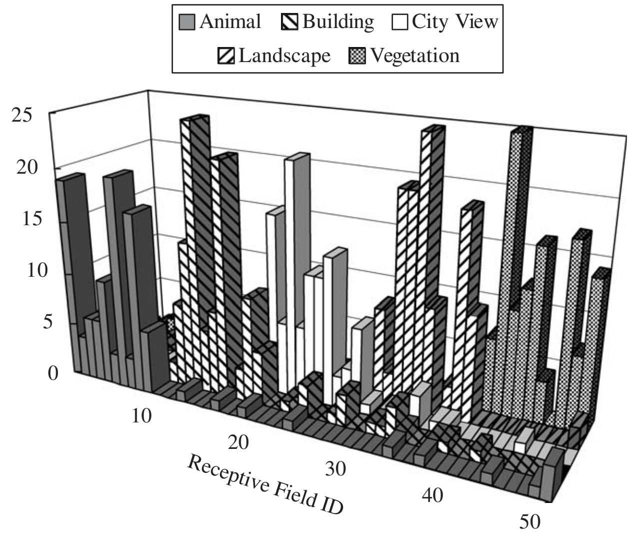

Fig. 3. Receptive fields against the number of images with highest activation. Groups of nodes within the hidden layer resemble the semantic categorization of data.

field represents the class of *Buildings* if the training image that is the closest to it is belongs to the *Building* class. One may observe that the number of receptive fields in classes *Animal*, *City View* and *Landscape* are slightly larger than those for the classes of *Building* and *Vegetation*. This indicates that the varieties of these two categories represented in terms of color and texture descriptors are less diversified than reported for the remaining three classes. For examples, images that belong to the category of *Buildings* usually consist of many vertical and horizontal edges. The color of the images belonging to the category *Vegetation* is predominantly green. In contrast, images in the category of *Animals* exhibit different shapes (edges and texture). The color also exhibits more diversity when more types of animals are involved in this class. On average, the numbers of receptive fields in all classes are nearly the same and this indicates that the complexity of recognizing image belonging to different categories could be equally challenging.

When using SVM, the images are projected to a very high dimensional kernel space and it is very difficult to visualize the SVM or relate it to the problem domain knowledge. Instead, the RBFNN works in the original feature space in the following section, we provide further analysis to the resulting RBFNN using the eighth training dataset, which yields the highest testing accuracy among all.

For each image being fed to the RBFNN, one of the receptive fields yields the highest activation level. Fig. 3 shows the number of images yielding the highest activation for each receptive field. For example, the first receptive field comes with 19 images in the Animal class yielding the highest activation but zero for other classes, therefore this receptive field forms an abstraction for the Animal class. It offers an insight into the relation between nodes in the hidden layer and the expected categorization of images produced by the network.

Table 5
Confusion matrix of RBFNN trained by MC^{2}SG

63.80 ± 6.3	2.80 ± 1.7	3.20 ± 1.6	3.70 ± 1.8	7.30 ± 3.4
2.80 ± 2.0	**65.90 ± 6.4**	5.70 ± 2.1	3.70 ± 2.0	1.40 ± 0.8
4.20 ± 2.1	8.90 ± 3.2	**57.60 ± 5.4**	6.70 ± 2.6	0.30 ± 0.8
1.40 ± 0.9	0.50 ± 0.8	1.60 ± 1.2	**75.60 ± 10.4**	1.50 ± 0.9
6.30 ± 1.7	0.00 ± 0.0	1.00 ± 1.4	1.40 ± 1.3	**72.70 ± 5.2**

Table 6
Confusion matrix of RBFNN trained by sequential learning

60.10 ± 5.4	3.70 ± 2.0	4.10 ± 1.5	4.80 ± 1.6	8.40 ± 3.1
3.60 ± 1.6	**61.80** ± 7.2	8.00 ± 3.2	4.20 ± 2.2	2.00 ± 1.3
4.50 ± 1.8	9.60 ± 2.8	**54.70 ± 5.7**	8.40 ± 2.5	0.70 ± 0.9
1.90 ± 1.2	1.40 ± 1.0	2.10 ± 0.9	**72.60 ± 10.5**	2.10 ± 1.4
6.90 ± 2.1	0.80 ± 0.6	1.50 ± 1.4	2.60 ± 1.5	**69.70 ± 4.9**

Table 7
Confusion matrix of SVM

54.90 ± 6.2	3.10 ± 1.2	5.10 ± 2.1	9.00 ± 2.6	8.70 ± 2.8
2.40 ± 0.8	**55.30 ± 4.8**	13.80 ± 1.7	6.30 ± 2.7	1.70 ± 1.0
3.70 ± 2.2	3.50 ± 2.0	**59.50 ± 5.5**	10.30 ± 2.6	0.70 ± 1.2
1.70 ± 1.0	1.10 ± 1.2	2.60 ± 1.1	**74.10 ± 9.3**	1.10 ± 1.1
9.60 ± 2.5	0.30 ± 0.4	0.40 ± 0.5	2.20 ± 1.6	**68.90 ± 3.8**

Tables 5–7 present the average confusion matrices for the testing dataset for RBFNN trained by MC^{2}SG, RBFNN optimized by sequential learning and SVM, respectively. RBFNN trained by MC^{2}SG achieves higher accuracy than its counterparts when classifying images belonging to classes: *Animal*, *Building*, *Landscape*, and *Vegetation*. The accuracy for images ascribed to class *City view* is higher in comparison with RBFNN trained by sequential learning but lower than the one obtained by the SVM classifier. However, the confusion between classes is lower in most of the cases when using RBFNN optimized by MC^{2}SG than using the other two classifiers.

From the confusion matrices, one may notice that the classifiers develop some confusion between the following pairs of classes: Animal&Vegetation, Building&City view, Building&Landscape, Landscape&City view and Vegetation&Animal. These errors are somewhat anticipated because of the commonalities of color and texture descriptions occurring in these classes. Furthermore, such possible misclassifications could be expected from the semantic point of view. For example, one can observe in Fig. 4 that images belonging to City view consist of many Buildings and Animal images usually having the background similar to Vegetation's foreground. The pair of Landscape and City view are similar in cases which the lower part of the image is water while the upper part is either mountain or buildings. It is also noted that animals exhibiting mimicry as the last four images depicted at the last row of Fig. 4 are frequently misclassified as vegetation.

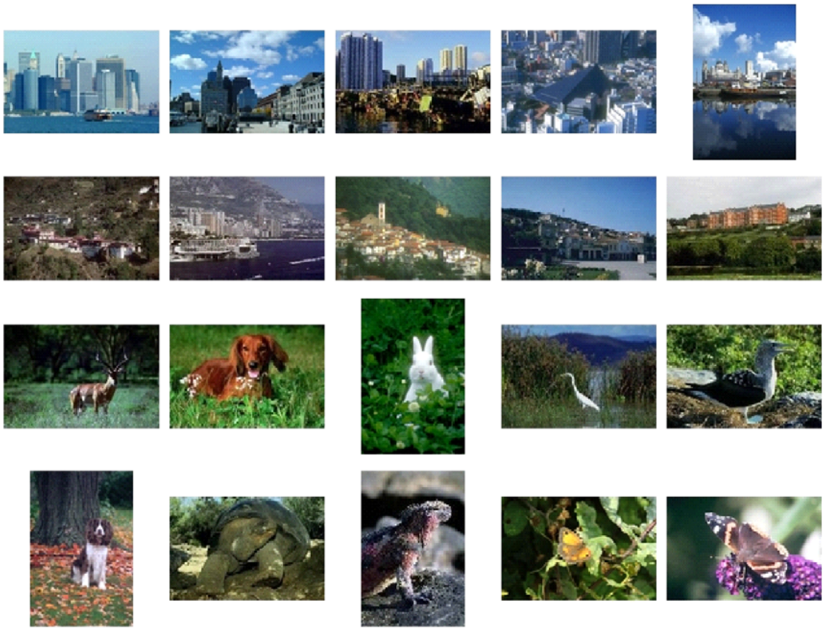

Fig. 4. Samples of expected misclassified images occurring due to semantic overlap.

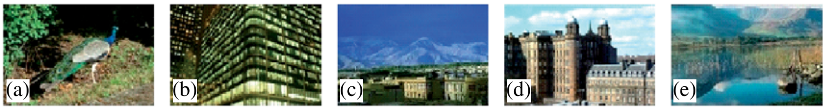

Fig. 5. Selected samples of misclassified images.

It is instructive to take a look at some selected samples to identify situations that derive in misclassification. Low differentiation between background and foreground in Fig. 5(a) causes that the *animal* becomes classified as *vegetation*. Close-up and picture's angle in Fig. 5(b) reduces the salient properties of *building* image such as vertical and horizontal edges. Fig. 5(c) displays a picture that could have been easily assigned to class *landscape*. However, the beholder is able to recognize manmade structures and assign this picture to class *city view*. The classifier is not equipped with the same criterion and fails in the attempt of classifying the image. Fig. 5(d) shows a shot of a *building* image captured with a wide focus that puts the picture in the *city* view class. Again the classifier lacks of criteria to recognize this picture as belonging to the expected class. Fig. 5(e) presents a *landscape* image in which the mountains and sky reflect in the lake. It produces an effect in the color and texture distribution that must have misled the classifier.

These subjective observations motivate a closer look at the feature vectors, abstract images, and matching results placed at the same level of the classifier decisions. Fig. 6 provides a transition in this approximation presenting the actual images closest to some selected abstract images (centers of the receptive fields). The corresponding features vectors of these images along with the models are depicted in Fig. 7.

Fig. 6 not only presents the closest images to the selected abstract images but also can be used to get some indications of the kind of images that represent each class. We selected one of the receptive fields in each image class and plot the corresponding histograms in the left hand side of Fig. 7. This histogram indicates the values of the receptive field in the 282-dimensional feature space. Moreover, we plot the images from the training dataset, in the right hand side of Fig. 7, which are nearest to the receptive field in each class. In addition to the criteria presented above in assigning images to each class, aided by observation from Figs. 6 and 7, it is noticeable that:

(1) *Animal* images are correctly classified when there is a clear contrast between color in the foreground and background. Note the first half (color component) in the models and the closest images at Fig. 7. Note also the gaps between color distribution in the color layout and color structure. *Animal* and vegetation present the same pattern but it is more common in the former one.
(2) *Building* images are easily recognized when the angle used to capture the structure reduces the presence of non-directional edges in the edge histogram component, which is common in the other classes. In addition, the focus determines the amount of sky or neighboring

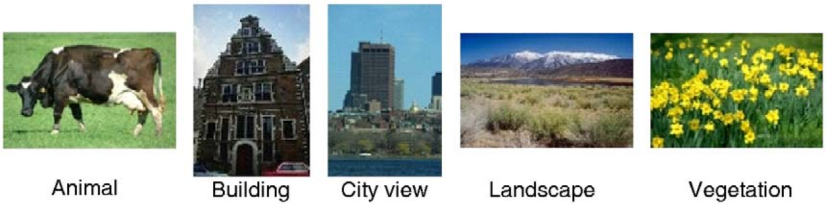

Fig. 6. Closest images to selected abstract images presented in Fig. 7.

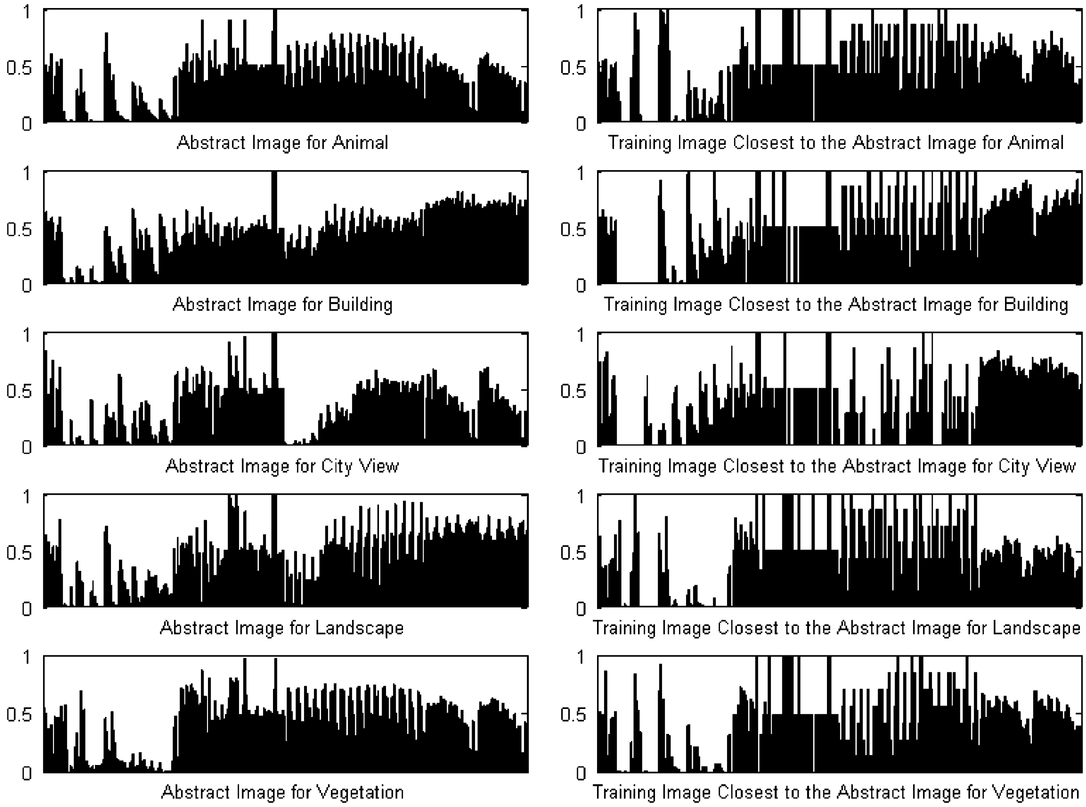

Fig. 7. The abstract image (i.e. the center of the receptive field) captured by the receptive fields yielding highest activation responds to each semantic class and the closest training image to each of them. (The x-axis represents the feature ID from 1 to 282 and the y-axis represents the value of the feature.)

objects to the structure. It has an effect on the mosaic of homogeneous textures as can be observed in Fig. 7 second row. The energy and energy deviation resulting from the last banks of filters is higher in comparison with the remaining classes, especially *city view* with which there is a semantic overlapping.

(3) *City view* images normally consist of small-size man-made structures and certain amount of sky, water, or both, producing edges and texture distribution that depart from the other classes. It can be noted in the second half of the abstract image and closest image in the third row of Fig. 7.

(4) *Landscape* images can be differentiated from images in other classes (mainly *city view*) based on texture descriptions, especially homogeneous textures as can be observed in the fourth row of Fig. 7.

(5) *Vegetation* images have descriptions similar to *animal* images. The color structure and edge histogram components contribute to the separability of these classes.

Fig. 8 includes a number of selected images coming from each class present in the training dataset. The images located in each row of this figure belong to *Animal*, *Building*, *City view*, *Landscape* and *Vegetation*, respectively. Fig. 8 as well as Fig. 7 also shows that the variety of images in each class and consequently the variability of feature values in each descriptor justify the usage of multiple neurons to represent a broader class. The abstract images can be seen as prototypes that capture the average representation of a sub-class. Thus, models focus on groups of feature vectors even if there is not a high difference between them for the beholder in terms of the semantic categorization.

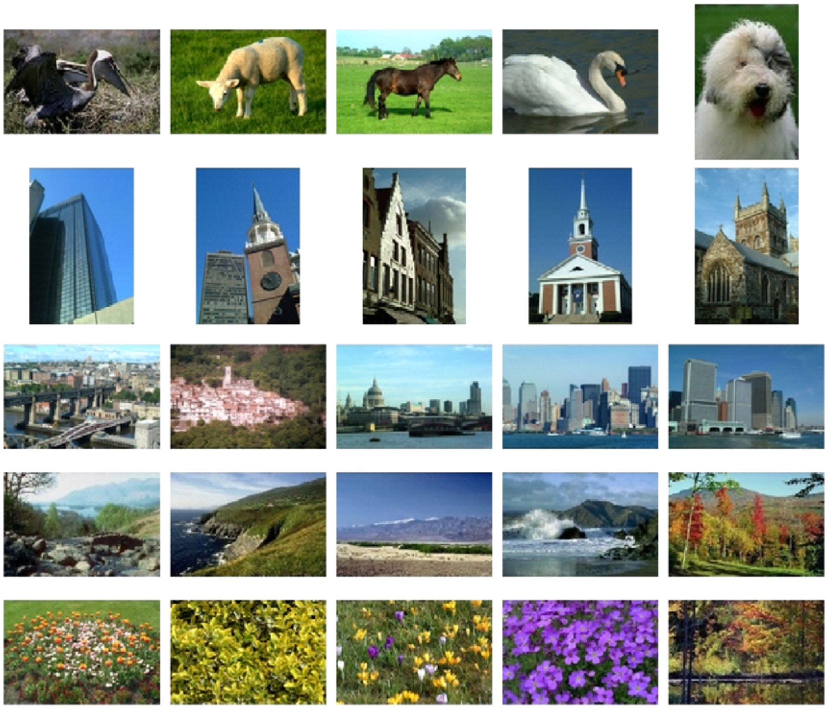

Fig. 8. More samples of closest images to the abstract images.

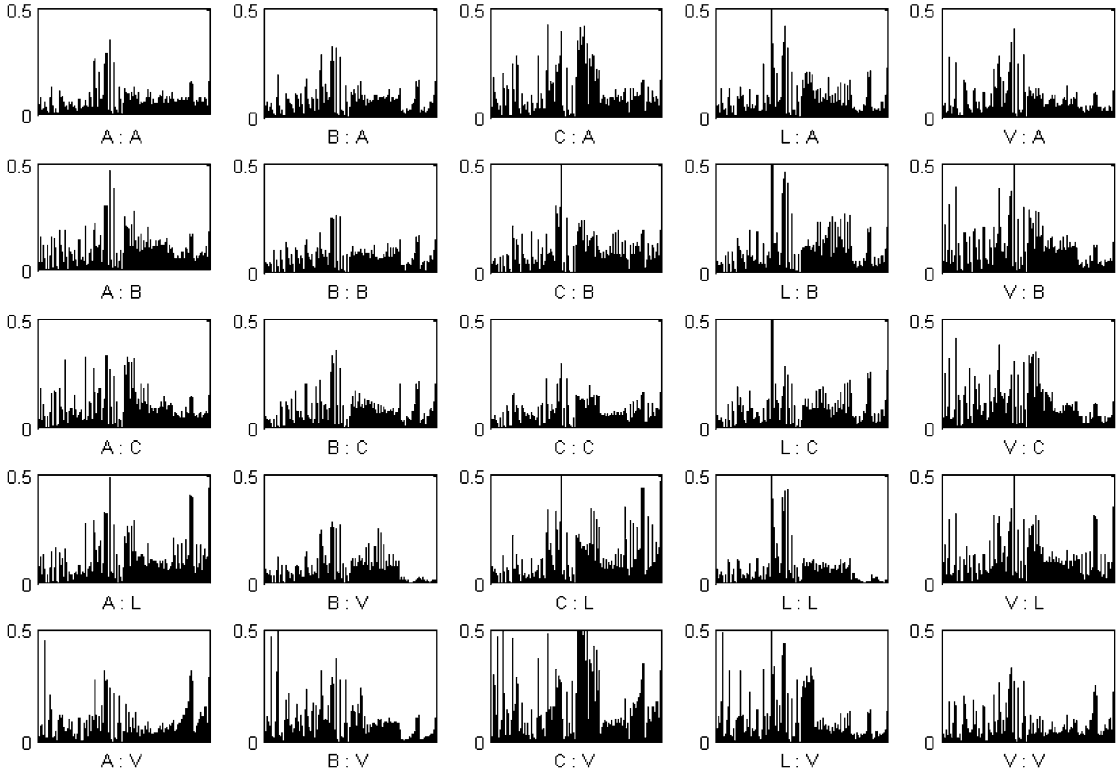

Fig. 9. The plots of the average squared differences between testing images and the abstract image in the selected receptive fields.

For instance, the color component—first half of the feature vector in *vegetation* presents a high variety of instances as reported in the differences between the model and the average of images ascribed to this class at Fig. 9 (fifth row, fifth column). In the case of the homogeneous texture descriptions there are two groups of frequency filters. One may notice that the values of the low frequency filters yielding a very high variance produces high differences between the images and abstract images of their classes (diagonal form left to right in Fig. 9). The exception is class *landscape* (fourth row, fourth column), which as mentioned above, relies on this descriptor to discriminate from other classes.

In Fig. 9, we use the notation of "A:B" to denote the plot of the average differences between the abstract image of animal and testing samples of building. (A: Animal, B: Building, C: City view, L: Landscape and V: Vegetation.) Columns correspond to the model, while rows refer to the testing sample. The diagonal shows the highest matching.

Finally, it is worth to stress the contribution of the localized generalization error model in selecting the RBFNN architecture that produces significant separability and discrimination between classes, though the description of images does not capture enough criteria to resemble semantic characterization of these classes.

7. Conclusions

In this study, being motivated by the concept that semantically similar images should exhibit similarity in the feature space, we proposed an application of the localized generalization error model to image classification. This model captures the generalization error for unseen samples that are similar to the training samples. Experimental results show that the RBFNN trained using the minimization of the localized generalization error outperforms "standard" RBFNN and multi-class SVM. Moreover, we showed how the RBFNN could be helpful in the visualization of the receptive fields with respect to the categories of images to be classified.

The selection of the MPEG-7 descriptors (the feature space) for describing images is still an open problem. In this work, we formed the MPEG-7 feature space to describe images by balancing the number of color and texture descriptors. However, further detailed investigations on the formation of the optimal feature space become necessary. It is very likely that some classes of images may be distinguished well by some MPEG-7 descriptors however the choice of such descriptors could depend upon the character of the classes. While here we decided to consider 5 predefined categories of images, one should become aware that more work is necessary to fully explore the capabilities of the MPEG-7 descriptors as a basis of the most suitable feature space. One interesting alternatives would be to form the feature space with the aid of the localized generalization error model.

Acknowledgments

This work is supported by a Hong Kong Polytechnic University Interfaculty Research Grant No. G-T891 and Canada Research Chair (W. Pedrycz).

References

[1] T. Hastie, R. Tibshirani, J. Friedman, The Element of Statistical Learning, Springer, New York, 2001.

[2] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[3] V. Cherkassky, X. Shao, F.M. Mulier, V.N. Vapnik, Model complexity control for regression using VC generalization bounds, IEEE Trans. Neural Networks 10 (5) (1999) 1075–1089.

[4] B.S. Manjunath, P. Salembier, T. Sikora, Introduction to MPEG-7 Multimedia Content Description Interface, Wiley, New York, 2002.

[5] E. Izquierdo, I. Damnjanovic, P. Villegas, X. Li-Qun, S. Herrmann, Bringing user satisfaction to media access: the 1st busman project, IEEE Proceedings of the International Conference on Information Visualisation, 2004, pp. 444–449.

[6] V. Mezaris, H. Doulaverakis, R.M.B. de Otalora, S. Herrmann, I. Kompatsiaris, M.G. Strintzis, A test-bed for region-based image retrieval using multiple segmentation algorithms and the MPEG-7 experimentation model: the schema reference system, Image and Video Retrieval, Lecture Notes in Computer Science, 2004, pp. 592–600.

[7] S.-F. Chang, T. Sikora, A. Puri, Overview of the MPEG-7 standard, IEEE Trans. Circuits Systems Video Technol. 11 (6) (2001) 688–695.

[8] A. Dorado, W. Pedrycz, E. Izquierdo, An MPEG-7 learning space for semantic image classification, Proceedings of the 31st Latin American Computing Conference, Cali, Colombia, 2005, pp. 297–307.

[9] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Trans. Circuits Systems Video Technol. 11 (6) (2001) 703–715.

[10] A. Barla, F. Odone, A. Verri, Old fashioned state-of-the-art image classification, IEEE Proceedings of International Conference on Image Analysis and Processing, 2003, pp. 566–571.

[11] A.K. Jain, P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.

[12] O. Chapelle, P. Haffner, V.N. Vapnik, Support vector machines for histogram-based image classification, IEEE Trans. Neural Networks 10 (5) (1999) 1055–1064.

[13] N. Serrano, A. Savakis, A. Luo, A computationally efficient approach to indoor/outdoor scene classification, IEEE Proceedings of International Conference on Pattern Recognition, 2002, pp. 146–149.

[14] R. Yan, Y. Liu, R. Jin, A. Hauptmann, On predicting rare classes with SVM ensembles in scene classification, IEEE Proceedings of International Conference on Acoustics, Speech, and Signal Processing, 2003, pp. 21–24.

[15] C.-F. Tsai, K. McGarry, J. Tait, Image classification using hybrid neural networks, ACM SIGIR Proceedings of International Conference on Research and Development in Information Retrieval, 2003, pp. 431–432.

[16] S. Prabhakar, H. Cheng, J.C. Handley, Z. Fan, Y.W. Lin, Picture–graphics color image classification, IEEE Proceedings of International Conference on Image Processing, 2002, pp. 785–788.

[17] S.-B. Dong, Y.-M. Yang, Hierarchical web image classification by multi-level features, in: Proceedings of International Conference on Machine Learning and Cybernetics, Beijing, China, November 2002, pp. 663–668.

[18] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass Kernel-based vector machines, J. Mach. Learn. Res. 2 (2) (2002) 265–292.

[19] Y. Cheng, J. Lu, T. Yahagi, Car license plate recognition based on the combination of principal components analysis and radial basis function networks, IEEE Proceedings of International Conference on Signal Processing, 2004, pp. 1455–1458.

[20] Y. Fan, M. Paindavoine, Implementation of an RBF Neural Network on embedded systems: real-time face tracking and identity verification, IEEE Trans. Neural Networks 14 (5) (2003) 1162–1175.

[21] W. Hoeffding, Probability inequalities for sums of bounded random variables, J. Am. Stat. Assoc. 58 (1963) 13–30.

[22] W.W.Y. Ng, D.S. Yeung, D. Wang, E.C.C. Tsang, X.-Z. Wang, Localized generalization error and its application to RBFNN training, in: Proceedings of International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2005, pp. 4667–4673.

[23] D.S. Yeung, W.W.Y. Ng, D. Wang, E.C.C. Tsang, X.-Z. Wang, Localized generalization error model and its application to architecture selection for radial basis function neural network, IEEE Trans. Neural Networks, submitted for publication, second review.

[24] L. Yingwei, N. Sundararajan, P. Saratchandran, A sequential learning scheme for function approximation using minimal radial basis function neural networks, Neural Comput. 9 (1997) 461–478.