<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Extreme learning machine based fast object recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Xu, Jiantao; Zhou, Hongming; Huang, Guang-Bin</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2012</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/19780">http://hdl.handle.net/10220/19780</a></td>
</tr>
</tbody>
</table>

© 2012 International Society of Information Fusion. This paper was published in 2012 15th International Conference on Information Fusion (FUSION) and is made available as an electronic reprint (preprint) with permission of International Society of Information Fusion. The paper can be found at the following official URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6289984&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxplstabs_all.jsp%3Farnumber%3D6289984. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper is prohibited and is subject to penalties under law.
Extreme Learning Machine Based Fast Object Recognition

Jiantao Xu, Hongming Zhou, Guang-Bin Huang
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
Email addresses: {xuji0010, hmzhou, egbhuang}@ntu.edu.sg

Abstract — Extreme Learning Machine (ELM) as a type of generalized single-hidden layer feed-forward networks (SLFNs) has demonstrated its good generalization performance with extreme fast learning speed in many benchmark and real applications. This paper further studies the performance of ELM and its variants in object recognition using two different feature extraction methods. The first method extracts texture features, intensity features from Histogram and features from two types of color space: HSV & RGB. The second method extracts shape features based on Radon transform. The classification performances of ELM and its variants are compared with the performance of Support Vector Machines (SVMs). As verified by simulation results, ELM achieves better testing accuracy with much less training time on majority cases than SVM for both feature extraction methods. Besides, the parameter tuning process for ELM is much easier than SVM as well.

Keywords - Extreme Learning Machine (ELM), Support Vector Machine (SVM), Object Recognition, Feature Extraction, Radon Transform

I. INTRODUCTION

Object recognition is a task of classifying objects to different categories based on their image attributes. In recent years, there have been intense activities in development of object recognition techniques based on image contents. A given object in an image is usually represented by a set of image attributes, such as color [1]-[3], texture [4]-[6] and shape [7]-[9]. Those attributes can be extracted and stored in a visual feature database and used as the training and testing data.

Images usually contain many pixels that will result in high dimensionality when digitized. For example, a small image of size 256×256 pixels has more than 60 thousand dimensions when interpreted as a vector. Therefore, the process of extracting meaningful information and removing inessential information from images becomes a critical step in object recognition. A better performance can be achieved if feature dimensionality and the complexity of the learning algorithm are optimized.

As one of the popular machine learning techniques, Support Vector Machines (SVMs) [10] have been extensively used for object recognition and achieve good performance [11], [12]. There are two main learning mechanisms for SVM. First, the training data of SVM are mapped into a higher dimensional feature space through a nonlinear feature mapping. Second, SVM uses standard optimization method to find the solution of maximizing the separating margin of two different classes in the feature space. SVMs’ performances are very sensitive to the tuning parameters. Hence the tuning process may require a long period of time.

Extreme Learning Machine (ELM), proposed by Huang et al. [13], [14] is a special type of single-hidden layer feed-forward networks (SLFNs), where the hidden node parameters are randomly generated, and the output weights are analytically determined using standard least square solutions. ELM has shown its superior generalization ability in solving regression problems and performs well in large dataset classification problems [15]-[17]. In this paper, we propose a novel learning approach based on ELM to classify various objects in images using two different feature extraction methods. In the first method, 27 features are extracted from each image based on the following features of the object: the texture features consisting of Contrast, Correlation, Energy, Homogeneity and Entropy; the color components from two types of image space: HSV (Hue, Saturation and Value) & RGB (Red, Green and Blue); intensity descriptors from the Intensity Histogram of the image. In the second method, we extract 30 dimensional features from the Radon transform of an image after a few image processing steps such as Median filter and Canny edge detector. The features extracted from the above two methods are used as the training and testing data of the ELM and SVM classifiers.

The organization of this paper is as follows. In section II, we will give a brief introduction of ELM and its variants – ELM (with regularization factor $C$) and ELM (Gaussian kernel). In section III, we will give detailed information on extracting feature vectors from objects in images using two methods mentioned above. In section IV, we will discuss the performance of ELM compared with the performance of SVM in object recognition. In section V, we conclude the paper.
II. BRIEF OF EXTREME LEARNING MACHINE

A. Basic Extreme Learning Machine

Extreme Learning machine (ELM) was originally developed for the single-hidden layer feed-forward neural networks (SLFNs) and was then extended to the generalized SLFNs [18], [19]. The essence of ELM is that: the hidden layer parameters need not be tuned. The output function of ELM is:

\[ f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x) \beta \]  

where \( h(x) = [h_1(x), ..., h_L(x)] \) is the output vector of the hidden layer with respect to the input \( x \) and \( \beta = [\beta_1, ..., \beta_L]^T \) is the vector of the output weights between hidden layer nodes and output nodes. \( h(x) \) maps the data from the \( d \)-dimensional input space to the \( L \)-dimensional hidden-layer feature space \( H \).

The hidden-layer feature mapping \( h(x) \) can also be expressed as:

\[ h(x) = [G(a_1, b_1, x), ..., G(a_L, b_L, x)] \]  

where \( G(a, b, x) \) is a nonlinear piecewise continuous function and \( (a_i, b_i) \) are randomly generated according to any continuous probability distribution. A few examples of \( G(a, b, x) \) are shown below:

1) Sigmoid function

\[ G(a, b, x) = \frac{1}{1 + \exp(-(a \cdot x) + b)} \]  

2) Hard-limit function

\[ G(a, b, x) = \begin{cases} 1 & \text{if } a \cdot x - b \geq 0 \\ 0 & \text{otherwise.} \end{cases} \]  

3) Gaussian function

\[ G(a, b, x) = \exp(-b \| x - a \|^2) \]  

4) Multiquadric Function

\[ G(a, b, x) = (\| x - a \|^2 + b^2)^{1/2} \]  

In this paper, the activation function used in ELM during simulation is the sigmoid function shown in (3).

Equation (1) can also be written in the matrix form:

\[ \beta H = \mathbf{T} \]  

where \( H \) is the hidden-layer output matrix:

\[ H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \]  

and \( \mathbf{T} = [t_1, \cdots, t_N] \) is a vector of target labels. The weights that connect hidden nodes and output nodes can be estimated as:

\[ \beta = H^\dagger \mathbf{T} \]  

where \( H^\dagger \) is the Moore–Penrose generalized inverse of the hidden layer output matrix \( H \). There are a few methods that can be used to calculate the Moore–Penrose generalized inverse of a matrix such as orthogonal projection method, iterative method, and singular value decomposition [20].

B. Extreme Learning Machine (with regularization factor \( C \))

The ELM (with regularization factor \( C \)) is proposed based on the equality constrained optimization method, and the optimization problem can be formulated as:

Minimize:

\[ L_P = \frac{1}{2} \| \beta \|^2 + C \frac{1}{2} \sum_{i=1}^{N} \xi_i^2 \]  

Subject to:

\[ h(x_i) \beta = t_i - \xi_i, \quad i = 1, ..., N \]  

Based on the Karush-Kuhn-Tucker (KKT) theorem [21], to solve the above equation, it is equivalent to solve the following dual optimization problem:

\[ \begin{align*} 
  L_D &= \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^{N} \xi_i^2 - \sum_{i=1}^{N} \alpha_i (h(x_i) \beta - t_i + \xi_i) \\
  \text{Subject to:} \quad \alpha_i \geq 0, \quad i = 1, ..., N 
\end{align*} \]  

where \( \alpha_i \) is the \( i \)-th Lagrange multiplier. We can have the KKT optimality conditions as follows:

\[ \frac{\partial L_D}{\partial \beta} = 0 \Rightarrow \beta = \alpha_i h(x_i)^T \]  

\[ \frac{\partial L_D}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C \xi_i, \quad i = 1, ..., N \]  

\[ \frac{\partial L_D}{\partial \alpha_i} = 0 \Rightarrow h(x_i) \beta - t_i + \xi_i = 0, \quad i = 1, ..., N \]  

where \( \alpha = [\alpha_1, ..., \alpha_N]^T \).

By substituting equation (13) and (14) into (15), we can simplify the above set of linear equations as:

\[ \frac{1}{C} + H H^T \alpha = \mathbf{T} \]  

From (13) and (16), we have

\[ \beta = H^T \left( \frac{1}{C} + H H^T \right)^{-1} \mathbf{T} \]  

Therefore, the output function of ELM classifier can be written as:
\[ f(x) = h(x)\beta = h(x)H^T \left( \frac{1}{C} + HH^T \right)^{-1} T \]  

(18)

**C. Extreme Learning Machine (Gaussian Kernel)**

For the cases when a feature mapping \( h(x) \) is unknown to users, Mercer’s condition can be applied on ELM. A kernel matrix for ELM can be defined as follows:

\[ \Omega_{ELM} = HH^T : \Omega_{ELM,i,j} = h(x_i) \cdot h(x_j) = K(x_i, x_j) \]  

(19)

Thus, the output function of ELM classifier can be modified as

\[
     f(x) = h(x)H^T \left( \frac{1}{C} + HH^T \right)^{-1} T \\
     = \left[ K(x, x_1) \cdots K(x, x_N) \right]^T \left( \frac{1}{C} + \Omega_{ELM} \right)^{-1} T
\]  

(20)

The Gaussian kernel \( K(u, v) = \exp (-\gamma|u - v|^2) \) is used in simulations for this paper.

**III. PROPOSED FEATURE EXTRACTION METHODS**

Feature extraction is the basis of object recognition. Within the visual feature scope, the features of an object can be classified into three major categories: Color, Texture and Shape. As mentioned in section I, two feature extraction methods will be used to extract feature vectors from objects in images in this paper.

**A. Object recognition based on color and texture**

Color is one of the most widely used features in object recognition and classification as it can be computed in a very fast manner. The attributes of color are not sensitive to the scale, rotation and location of the object. Different kinds of color attributes can be used for object recognition such as color moment, dominant color and histogram [1]. On the other hand, image texture is defined as the visual patterns and the spatial arrangement of color or intensities in an image [4]-[6]. Texture can be described by terms such as coarse, smooth, silky and rough. This section discusses the first feature extraction method which is to extract the combination of an object’s color and texture features from an image. We use the extracted features to form a feature vector to represent the object. The first image database was built by selecting 600 images from the Caltech database [22] and Corel Collection [23]. These images were arranged in 6 categories: Horse, Mountain, Food, Plane, Face and Leopard. It includes 100 images from each category and the images are in JPEG format. Sample images from the database are shown in Fig. 1.

The followings are key steps for extracting color and texture features of an object from an image:

**Step 1**: Input an image from the database

**Step 2**: Compute the average value of Red, Green and Blue components from RGB color space

**Step 3**: Convert the RGB color space into HSV color space and calculate the average value of Hue, Saturation and Value components

**Step 4**: Convert the color image into gray scale image with the size of 200×200 pixels

**Step 5**: Compute Intensity Histogram of the image and scale the Intensity Histogram into 16 gray levels

**Step 6**: Compute the gray level co-occurrence matrix (GLCM) from the gray scale image and extract the following five texture features from GLCM: Contrast, Correlation, Energy, Homogeneity and Entropy

**Step 7**: Combine the following extraction features of the object into a 27 dimensional vector: 3 features from RGB color space, 3 features from HSV color spaces,
16 gray level features from Intensity Histogram and 5
texture features from GLCM.

Fig. 2 demonstrates an example of extracting 27 features
from a cat image using the method proposed above.

Fig. 3. Sample images from the second database

Fig. 4. An overview of image processing for the second method

B. Object recognition based on shape

Shape is recognized as one of the most fundamental
characteristics for object recognition and the shape
representation can be divided into two categories: boundary
based and region based. Boundary based descriptors focus on
the contour of the shape, whereas the region based descriptors
focus on the entire shape region. For the past few decades,
many methods have been proposed for object recognition
based on shape such as Fourier Descriptor [24], Compactness
Vectors [25], Zernike moments [26] and Wavelet features [27].
In this section, we propose an effective region based descriptor
based on the Radon transform [28]-[30]. Radon transform is
the integral transform consisting of the integral of a function
over straight lines at a certain angle. It projects the target
object from the spatial domain into a projection space. An et al.
[31] proved that the extracted features from Radon transform
could be invariant to the translation, scale and rotation of the
object.

The second image database was built by selecting 400
images from the Caltech database and Corel Collection. These
images were arranged in 4 categories: Bus, Dinosaur and
Flower and Face. It includes 100 images from each category
and the images are in JPEG format. Sample images from the
database are shown in Fig. 3.

The followings are key steps for extracting shape features
of an object from an image. Fig. 4 gives an overview of image
processing (Step 1 to Step 5) for the second method.

Step 1: Input an image from the database
Step 2: Convert the color image into gray scale image to
reduce complexity
Step 3: Apply Median filter to blur the image
Step 4: Apply Canny edge detection method to find edges of
image
Step 5: Perform the following morphological operations to
the image:
Dilation – to remove gaps in edges
Erosion – to remove the noise and smooth the object
Step 6: Perform Radon transform to the image from angles of
0 degree to 170 degrees with a 10 degrees interval,
which results in 18 histograms of Radon transform
Step 7: Sample 30 line integral values from each histogram
of Radon transform. And sum the respective line
integral values of all the 18 histograms of Radon
transform to form 30 features.

IV. EXPERIMENTAL RESULTS

The Simulation of ELM and SVM algorithms on all the
data sets are carried out in MATLAB 2007b environment
running in Core 2 Duo, 2.2-GHZ CPU with 2-GB RAM. For
color and texture based classification, 300 training images and
300 testing images are randomly selected from the first image
database for each simulation. For shape based classification,
200 training images and 200 testing images are randomly
selected from the second image database for each simulation.

For SVM (Gaussian kernel) and ELM (Gaussian kernel),
the cost parameter $C$ and kernel parameter $\gamma$ need to be
chosen appropriately in order to achieve good generalization
performance. We use 15 different values of $C$ and 15 different
values of $\gamma$ resulting in a total of 225 pairs of $(C, \gamma)$ to look
for the best performance. The values of $C$ and $\gamma$ are shown
below:

$C = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100,
1000, 10000\};$

$\gamma = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100,
1000, 10000\};$
Fig. 5 and Fig. 6 show the effect of tuning parameters on the testing accuracy of SVM (Gaussian kernel) and ELM (Gaussian kernel) for Food category respectively. We can observe that the performances of SVM (Gaussian kernel) and ELM (Gaussian kernel) are sensitive to the parameters chosen. The best generalization performance of SVM (Gaussian kernel) and ELM (Gaussian kernel) are usually achieved in a very narrow range of \((C, \gamma)\). Thus, the best combination of \((C, \gamma)\) for SVM (Gaussian kernel) and ELM (Gaussian kernel) need to be chosen properly for each data set.

For basic ELM, there is only one parameter to be tuned, which is the number of hidden nodes \(L\). For each database we use 15 different values of \(L\) and the values of \(L\) are shown below:

\[L = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100\}\.

For ELM (with regularization factor \(C\)), the user specified parameters are \((C, L)\). However, ELM (with regularization factor \(C\)) can achieve good generalization performance as long as the number of hidden nodes \(L\) is large enough (In this paper, \(L = 400\)). Therefore, only parameter \(C\) needs to be tuned. For each database we use 15 different values of \(C\) specified below:

\[C = \{0.001, 0.01, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 1000, 10000\}\.

Fig. 7 and Fig. 8 show the effect of tuning parameters on the testing accuracy of basic ELM and ELM (with regularization factor \(C\)) for Food category respectively. We can search along with different values of \(L\) or \(C\) to look for the best testing accuracy. Compared to SVM, ELM can be used easily and effectively by avoiding time consuming parameters tuning. Fifty trails have been conducted for each object recognition task, and the average of the training time and testing accuracy are reported in this section.

Table I presents the performance comparison of ELM and SVM based on the first feature extraction method in Section III. We can see that ELM outperforms SVM in all the simulation in terms of training time. The testing accuracy of ELM (Gaussian kernel) is higher than that of SVM (Gaussian kernel) in the majority of object categories. Table II presents the performance comparison of ELM and SVM for the second feature extraction method. We can observe that ELM (Gaussian kernel) has better testing accuracy on 3 out of 4 categories with much faster training speed.

From Fig. 9, we can observe that the learning speed of ELM is much faster than SVM especially when the size of training data becomes larger. The training time spent by SVM increases sharply when the number of training data increases. However, the training time spent by ELM increases very slowly when the number of training data increases.
Fig. 8. Performance of ELM (with regularization factor $C$) for Food category.

Fig. 9. The relationship of training time against the number of training data.

Table I. Performance comparison of ELM and SVM (based on color and texture features) – 1st database

<table>
<thead>
<tr>
<th>Object Categories</th>
<th>Basic ELM</th>
<th>SVM (Gaussian Kernel)</th>
<th>ELM (with regularization factor $C$)</th>
<th>ELM (Gaussian Kernel)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>Training Time (Seconds)</td>
<td>Testing Rate (%)</td>
<td>(C, $\gamma$)</td>
</tr>
<tr>
<td>Food</td>
<td>50</td>
<td>0.0209</td>
<td>96.3</td>
<td>(10,5)</td>
</tr>
<tr>
<td>Leopard</td>
<td>100</td>
<td>0.0143</td>
<td>95.1</td>
<td>(20,1)</td>
</tr>
<tr>
<td>Horse</td>
<td>90</td>
<td>0.0278</td>
<td>94.6</td>
<td>(100,1)</td>
</tr>
<tr>
<td>Face</td>
<td>80</td>
<td>0.0402</td>
<td>92.3</td>
<td>(20,1)</td>
</tr>
<tr>
<td>Mountain</td>
<td>90</td>
<td>0.0505</td>
<td>95.0</td>
<td>(100,2)</td>
</tr>
<tr>
<td>Plane</td>
<td>60</td>
<td>0.0159</td>
<td>99.9</td>
<td>(100,1)</td>
</tr>
</tbody>
</table>

Table II. Performance comparison of ELM and SVM (based on shape features) – 2nd database

<table>
<thead>
<tr>
<th>Object Categories</th>
<th>Basic ELM</th>
<th>SVM (Gaussian Kernel)</th>
<th>ELM (with regularization factor $C$)</th>
<th>ELM (Gaussian Kernel)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>Training Time (Seconds)</td>
<td>Testing Rate (%)</td>
<td>(C, $\gamma$)</td>
</tr>
<tr>
<td>Dinosaur</td>
<td>45</td>
<td>0.0109</td>
<td>87.8</td>
<td>(10,2)</td>
</tr>
<tr>
<td>Bus</td>
<td>40</td>
<td>0.0059</td>
<td>84.4</td>
<td>(100,1)</td>
</tr>
<tr>
<td>Flower</td>
<td>80</td>
<td>0.0271</td>
<td>82.9</td>
<td>(50,2)</td>
</tr>
<tr>
<td>Plane</td>
<td>25</td>
<td>0.0006</td>
<td>84.0</td>
<td>(1,1)</td>
</tr>
</tbody>
</table>

V. CONCLUSION

This paper proposes two effective feature extraction methods for fast object recognition based on Extreme Learning Machine (ELM). The first method extracts 27 features from the color (RGB and HSV values, Intensity Histogram) and texture (Contrast, Correlation, Energy, Homogeneity and Entropy) attributes of an object. The second method extracts 30 features based on the line integral values generated from Radon Transform. The second method works well especially when the shape of the objects are clear. The features extracted by using the two proposed methods are fed to ELM classifier to form an effective and accurate object recognition system.

The simulation results of the ELM based fast object recognition system are also compared to the results when using SVM classifier. From the simulation results, we can see that ELM tends to have better testing accuracy in majority cases with very fast training speed. In addition, ELM is more effective in implementation as the parameter tuning process for ELM is much simpler than SVM. In the future, one could try the ELM based fast object recognition system on more complex object recognition tasks and more scalable advantages can be expected.
ACKNOWLEDGMENT
This research was sponsored by the grant from Academic Research Fund (AcRF) Tier 1 of the Ministry of Education under project no. RG 22/08 (M52040128, Singapore).

REFERENCES