Kernel based Extreme Learning Machines for Image Classification

Stevica S. Cvetković[1], Miloš B. Stojanović[2], Saša V. Nikolić[3], Goran Z. Stančić[4]

Abstract – **This paper investigates possibilities for application of Kernel based Extreme Learning Machines (K-ELM) to the problem of multiclass image classification. It is combined with Local Binary Pattern (LBP) image descriptor, to reach highly accurate results. LBP is widely used global image descriptor characterized by compactness and robustness to illumination and resolution changes. Classification is done using recently introduced K-ELM method. Experimental evaluation on a standard benchmark dataset consisting of thousand images classified in ten categories, has shown high accuracy of results comparing to other benchmark models.**

Keywords – **Image classification, Neural networks, Kernel based extreme learning machines, Local binary patterns**

I. Introduction

Image classification based on visual content is a crucial problem in computer vision research. The goal of an image classification system is to assign a category label with the most similar visual content, to the given query image. Visual similarity between images is commonly measured using robust and compact image descriptors (features).

There is a large set of visual descriptors available in the literature [1, 13]. The choice of the descriptor essentially affects the overall performance of the classification system. Local Binary Pattern (LBP) is one of the most widely used descriptor due to robustness to resolution and lighting changes, low computational complexity, and compact representation [2, 3, 4, 7]. The second crucial part of the system is machine learning technique to be applied for classification of descriptors. Support Vector Machine (SVM) is the most widely used machine learning technique for image classification purpose [5, 6].

In this study we investigate application of Kernel based Extreme Learning Machines (K-ELM) [8, 9, 10, 11, 12] for image classification, as an alternative to the commonly used SVM technique. The training of the SVM is based on solving quadric programming problem, which is usually time consuming when the number of training examples is large. Beside that SVMs are originally proposed for binary classification, while for the multi-class classification one-against-all (OAA) or one-against-one (OAO) approaches must be used in SVM implementation. On the other hand, K-ELM shows much better generalization performances for multiclass classification cases, and has better scalability and much faster training speed, compared with SVM [9].

In the rest of the paper we first give an overview of K-ELM classification method for multi-class image classification. Then we describe the process of LBP descriptor extraction. Finally, experimental evaluation and conclusion are presented.

[1]Stevica S. Cvetković is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: stevica.cvetkovic@elfak.ni.ac.rs

[2]Miloš B. Stojanović is with the College of Applied Technical Sciences Niš, Aleksandra Medvedeva 20, Niš 18000, Serbia, e-mail: milos.stojanovic@vtsnis.edu.rs

[3]Saša V. Nikolić is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: sasa.nikolic@elfak.ni.ac.rs

[4]Goran Z. Stančić is with the University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, e-mail: goran.stancic@elfak.ni.ac.rs

II. Kernel based Extreme Learning Machines (K-ELM) for multiclass classification

Let us define N training examples as $(\mathbf{x}_j, \mathbf{y}_j)$ where $\mathbf{x}_j = [x_{j1}, x_{j2}, ..., x_{jn}]^T \in \mathbf{R}^n$ denotes *j-th* training instance of dimension n and $\mathbf{y}_j = [y_{j1}, y_{j2}, ..., y_{jm}]^T \in \mathbf{R}^m$ represents *j-th* training label of dimension m, where m is the number of classes. LBP image descriptor, which will be described in the next section, will further be denoted as $\mathbf{x}_j$. As $\mathbf{y}_j$, we will denote m dimensional vector of binary class labels with value *"1"* denoting membership to the class. SLFN with activation function $h(x)$ and L hidden neurons could be defined as:

$$\sum_{i=1}^{L} \beta_i h(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{f}_j, \; j = 1, \ldots, N \qquad (1)$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ denotes the vector of weights which connects the i^{th} hidden neuron and all input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight vector which connects i^{th} hidden neuron and all output neurons, and b_i is the bias of the i^{th} hidden neuron. By ELM theory [8], $\mathbf{w}_i$ and b_i can be assigned in advance randomly and independently, without a priori knowledge of the input data. The ELM network structure is presented in Figure 1.

SLFN in (1) should satisfy $\sum_{i=1}^{L} \left\| \mathbf{f}_i - \mathbf{y}_i \right\| = 0$, i.e., there exist β_i, $\mathbf{w}_i$ and b_i such that:

$$\sum_{i=1}^{L} \beta_i h(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{y}_j, \; j = 1, \ldots, N \qquad (2)$$

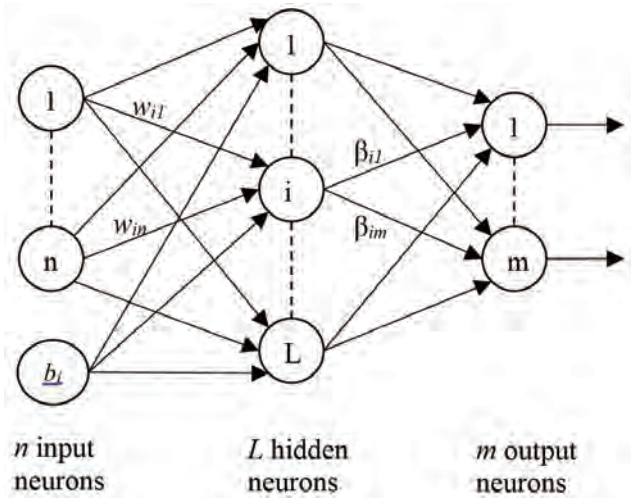

Fig. 1. Structure of an ELM network.

The equivalent compact matrix form of (2) can be written as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} \tag{3}$$

where **H** in (3) represents the hidden layer output matrix of the neural network; the i^{th} column of **H** represents the i^{th} hidden neuron's output vector in regard to inputs $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$.

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & h(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \cdots & \vdots \\ h(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & h(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L} \tag{4}$$

and

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times m} \tag{5}$$

Although the output weights can be analytically determined by finding the unique smallest norm least-squares solution of the linear system (3) in order to improve the performance the constrained optimization problem can be formed for ELM multiclass classifier with multiple outputs, as shown in [9]:

$$Minimize: L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_j\|^2 \tag{6}$$

$$Subject\ to: h(\mathbf{x}_j)\boldsymbol{\beta} = \mathbf{y}_j^T - \boldsymbol{\xi}_j^T,\ j = 1,...,N$$

where $\xi_j = \left[\xi_{j1},...,\xi_{jm}\right]^T$ is the training vector of the *m* output nodes with respect to the training sample x_i, while *C* represents tradeoff parameter between model complexity and allowed errors ξ_j during training. Based on *Karush - Kuhn - Tucker* (*KKT*) theorem, the optimization problem defined in (6) is equivalent of solving the dual optimization problem:

$$\begin{aligned} L_D &= \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{j=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &-\sum_{j=1}^{N}\sum_{i=1}^{m}\alpha_{ji}(h(\mathbf{x}_j)\boldsymbol{\beta}_i - \mathbf{y}_{ji} + \boldsymbol{\xi}_{ji}) \end{aligned} \tag{7}$$

where $\alpha_j = \left[\alpha_{j1},...,\alpha_{jm}\right]^T$ are Lagrange multipliers.

After solving (7) based on *KKT* conditions, which can be found in detail in [9], the following solution is obtained:

$$\beta = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{Y} \tag{8}$$

and the decision function of ELM classifier is:

$$f(\mathbf{x}) = h(\mathbf{x})\beta = h(\mathbf{x})\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{Y} \tag{9}$$

If feature mapping $h(\boldsymbol{x})$ is unknown, we can apply Mercer's condition on ELM. We can define kernel matrix for ELM as:

$$\begin{aligned} \boldsymbol{\Omega}_{ELM} &= \mathbf{H}\mathbf{H}^{\mathrm{T}} \\ \Omega_{ELM\,i,j} &= h(\boldsymbol{x}_i)h(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned} \tag{10}$$

In KELM $\mathbf{H} = \left[h(\boldsymbol{x}_1)^{\mathrm{T}} \quad \dots \quad h(\boldsymbol{x}_N)^{\mathrm{T}}\right]^{\mathrm{T}}$ represents hidden layer output matrix which maps data $\boldsymbol{x}_i$ from the input space to the hidden layer feature space and it is irrelevant to target values y_i and number of output nodes *m*. The kernel matrix $\boldsymbol{\Omega}_{ELM} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ is related only to input data $\boldsymbol{x}_i$ and number of training samples *N*, for regression, binary classification and multi class classification.

Then, the output function of ELM classifier (9) can be written compactly as:

$$f(\mathbf{x}) = \left(\begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}}\left(\frac{\mathbf{I}}{C} + \Omega_{ELM}\right)^{-1}\mathbf{Y}\right) \tag{11}$$

In this case the feature mapping $h(\boldsymbol{x})$ does not need to be defined by users, as well as the dimensionality of feature

space L (number of hidden nodes), just its kernel $K(\boldsymbol{u},\boldsymbol{v})$. In our experiments RBF kernel is used, defined as:

$$K(\boldsymbol{u},\boldsymbol{v}) = \exp(-\gamma\|\boldsymbol{u}-\boldsymbol{v}\|^2) \quad (12)$$

where γ represents parameter of Gaussian kernel. It can be noted from (11) and (12) that optimal combination of parameters C and γ have to be obtained in order to achieve good generalization performance.

III. Local Binary Patterns (LBP)

Local Binary Pattern (LBP) is a popular image descriptor that captures local appearance around a pixel. LBP descriptor of the complete image is formed as a histogram of quantized LBP values computed for every pixel of the image. It was introduced in [4] for the texture classification problem, and extended to general neighborhood sizes and rotation invariance in [2]. Since then, LBP has been extended and applied to variety of applications [3].

For a given image I, the local LBP descriptor centered on pixel $I(x, y)$ is an array of 8 bits, with one bit encoding each of the pixels in the 3×3 neighborhood (Fig 2.). Each neighbor bit is set to 0 or 1, depending on whether the intensity of the corresponding pixel is greater than the intensity of the central pixel. To form the binary array, neighbors are scanned starting from the one to the right, at position $I(x+1, y)$, in anti-clockwise order.

a) Pixel intensities

167	221	221
147	**217**	198
132	230	212

b) Thresholded difference

0	1	1
0		0
0	1	0

c) LBP

01100010

Fig. 2. Example of a LBP extraction process for central pixel of intensity 217.

If 3×3 neighborhood is used, there are 256 possible basic LBP codes. Using an extension from [2], this can be further reduced into a smaller number of patterns (58), which forms in a rotation-invariant descriptor. The extension is inspired by the fact that some binary patterns occur more frequently than others.

To describe the complete image, the quantized LBP patterns are grouped into histograms. The image could be divided into blocks, with a histogram computed for every block and concatenated to form the final descriptor. In our method we used only one image block, i.e. a global histogram is computed for the complete image.

To include image details at multiple scales, we extracted LBP histograms over the original image and several times resized image. Resizing is done to the half width and height of the original image using bicubic interpolation method.

Color image information is exploited by first converting an image into *YCbCr* color space and using all three color channels for LBP extraction. Final descriptor is formed by concatenation of the LBP histograms extracted at 3 scales (original + 2 downsampled) and 3 color channels. The computed image descriptor contains 3×3×58=522 dimensions.

IV. Experimental evaluation

Test of the proposed method is performed using publicly available Corel1000 dataset [7]. The dataset consists of 1000 images classified into following 10 categories: *Africa people, Beach, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Food*. An example image for every category is presented in Fig. 2. The dataset is characterized by large intra-category variations, which makes this dataset close to the real world image classification scenario.

We implemented the method in MATLAB and used it to measure the classification accuracy and time performance. To achieve correctness of results, tests were repeated for 50 times over random partitions of every category. We experimented with different number of training images per class. Classification accuracy results are presented in Table 1.

Table 1. Comparison of classification accuracy on Corel1000 dataset depending on the number of images per class. Standard deviations are given in brackets.

	% of training images per class		
	80%	50%	20%
K-ELM	90.86 (±1.32)	89.58 (±1.24)	85.49(±1.07)
SVM (kernel) [14]	90.79 (±1.92)	88.26 (±1.62)	84.04 (±0.87)
SVM (linear) [14]	89.62 (±1.27)	87.65 (±1.03)	83.41 (±1.25)

We further measured average training and testing time of the method on an *Intel Core i7 3.5GHz* computer. Training time of the complete training set was less than 1 second, while classification of a test image is done instantly (< 0.1ms). These results demonstrate high performances in terms of training and test speed on the test dataset.

In order to compare results of the ELM with other common classification techniques, we measured accuracy of the Linear SVM and kernelized RBF SVM [14], on the same dataset. Linear SVM parameter C was set to value 0.1. RBF SVM optimal parameters were determined by grid-search and 5-fold cross-validation, where C was examined in range $[2^{-4}, \ldots, 2^{10}]$, and γ in range $[2^{-10},.., 2^{4}]$. On the other hand, parameters of K-ELM were set to fixed values $C=10$ and $\gamma=2^{10}$. It can be noted from experimental results that in terms of accuracy ELM constantly outperforms both Linear SVM and kernelized RBF SVM, without additional computational costs.

V. CONCLUSION

In this study we presented results of our research in the field of automatic image classification using Kernel based ELM classifier (K-ELM) combined with LBP image descriptor. K-ELM classifier could be used as an effective alternative to the commonly used SVM methods. We reached classification accuracy of over 90%, on a test dataset containing 1000 images in 10 categories, what is high quality result on this dataset. It can be concluded that combination of K-ELM classifier with the LBP image descriptor is reasonable choice for image classification applications. In the further research, we plan to investigate integration of other image descriptors combined with K-ELM classifier. Particularly we will focus our research on fusion of color and texture descriptors.

REFERENCES

[1] Xin Zhang, Yee-Hong Yang, Zhiguang Han, Hui Wang, and Chao Gao, "Object class detection: A survey," *ACM Computing Surveys*, 46, 1, Article 10, July 2013.

[2] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), pp. 971-987, 2002.

[3] M. Pietikäinen and G. Zhao, "Two decades of local binary patterns: A survey," In: E Bingham, S Kaski, J Laaksonen & J Lampinen (eds) Advances in Independent Component Analysis and Learning Machines, Elsevier, 2015.

[4] T. Ojala, M. Pietikäinen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", *Pattern Recognition*, vol. 29, pp. 51-59, 1996.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169–2178, 2006.

[6] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[7] James Z. Wang, Jia Li, Gio Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 9, pp. 947-963, 2001.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[9] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 42, no. 2, pp. 513-529, 2012.

[10] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational Learning with Extreme Learning Machine for Big Data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31-34, December 2013.

[11] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in Extreme Learning Machines: A Review," *Neural Networks*, vol. 61, no. 1, pp. 32-48, 2015.

[12] Stevica Cvetković, Miloš B. Stojanović, Saša V. Nikolić, "Multi-channel descriptors and ensemble of Extreme Learning Machines for classification of remote sensing images," *Signal Processing: Image Communication*, vol. 39, 2015, pp. 111-120.

[13] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011.

[14] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 27:1–27:27, May 2011.