Extreme Learning Machines for Real-Time Image Classification

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

Abstract – **In this study we investigate possibilities for application of Extreme Learning Machines (ELM) to the problem of image classification. We start by extraction of Local Binary Pattern (LBP) descriptor of the image. It is widely used global image descriptor characterized by compactness and robustness to illumination and resolution changes. Classification is done using recently introduced specific single layer neural networks called Extreme Learning Machines (ELM). Our tests on a standard benchmark dataset consisting of thousand images classified in ten categories, has shown high accuracy of results while executing almost instantly during tests (< 0.1ms).**

Keywords – **Image classification, Local binary patterns, Neural networks, Extreme learning machines.**

I. Introduction

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

There is a large set of visual descriptors available in the literature [1]. 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]. 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 Extreme Learning Machines (ELM) [8, 9, 10, 11] for image classification, as an alternative to the commonly used SVM. ELM is a single hidden layer feed-forward neural network (SLFN), which overcomes an important drawback of traditional artificial neural networks (ANNs) - their slow learning speed. It increases training speed by randomly assigning weights and biases in the hidden layer, instead of iteratively adjusting its parameters by gradient based methods. As well as minimizing training error, ELM finds smallest norm of output weights and hence have better generalization performance then gradient based training algorithms, such as backpropagation. Furthermore, it can "naturally" handle multi-class classification problem using the architecture of multiple output nodes equal to the number of classes.

In the rest of the paper we first describe the process of LBP descriptor extraction. Then we give an overview of ELM for multi-class image classification. 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. Local Binary Patterns (LBP)

Local Binary Pattern (LBP) is a popular descriptor that initially captures the local appearance around a pixel. LBP descriptor of the complete image is then 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 1.). 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

158	229	229
126	**214**	198
211	250	212

b) Thresholded diff.

0	1	1
0		0
0	1	0

c) LBP

01100010

Fig. 1. Example of a LBP extraction process for central pixel of intensity 214.

If we use 3×3 neighborhood, there are 256 possible basic LBPs. Using an extension from [2], this can be further reduced into a smaller number of patterns (58), and implemented in a simple 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. Final descriptor is formed by concatenation of the previously extracted descriptors at several scales. In our method we used 3 scales, forming a 3×58=174 dimensional image descriptor.

III. Extreme Learning Machines (ELM)

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

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

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ denotes the vector of weights which connects the i^{th} hidden neuron and all input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ 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 Fig 2.

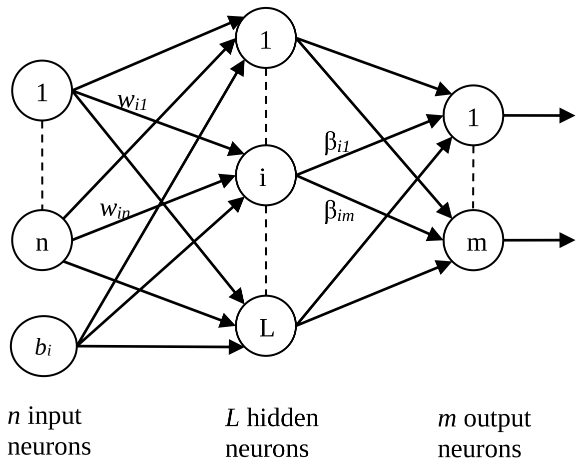

Fig. 2. Structure of the ELM network

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

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

If we denote as $\mathbf{H}$ a hidden layer output matrix of the ELM; the i^{th} column of $\mathbf{H}$ represents the i^{th} hidden neuron's output vector regard to inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$.

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}_{N \times L} \qquad (3)$$

and

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N \times m} \qquad (4)$$

Then the equivalent matrix form of (2) can be represented as:

$$\mathbf{H}\beta = \mathbf{Y} \qquad (5)$$

The output weights are then computed by finding the unique smallest norm least-squares solution of the linear system (5) as:

$$\beta = \mathbf{H}^{\dagger}\mathbf{Y} \qquad (6)$$

where $\mathbf{H}^{\dagger}$ represents the Moore-Penrose generalized inverse of the $\mathbf{H}$.

For a given training set $T=\{(\mathbf{x}_j, \mathbf{y}_j)\} | \mathbf{x}_j \in \mathbf{R}^n, \mathbf{y}_j \in \mathbf{R}^m, j = 1, \ldots, N\}$ with N instances of n-dimensional descriptors, sigmoid activation function $g(x)$, and hidden number of neurons L, ELM algorithm for classification problems can be summarized as follows:

Training:

(a) Assign random input weights $\mathbf{w}_i$, and biases b_i, $i = 1, \ldots, L$.
(b) Compute the hidden layer output matrix $\mathbf{H}$ using (3).
(c) Compute the output weights β using (6).

Testing:

(a) Compute the hidden layer output matrix $\mathbf{H}_{test}$ for instances from the test set, using (3)
(b) Compute the output matrix $\mathbf{Y}_{test}$ according to (5) using the β obtained in step 3 of the training.
(c) For every row in $\mathbf{Y}_{test}$ (i.e. every test instance), compute a class label as the index of the maximal value in that row.

IV. Experimental evaluation

To test the proposed method for image classification, we used publicly available Corel1000 dataset [7]. It 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. 3. The dataset is characterized by large variations of images inside a category, what makes this dataset close to the real world image classification scenario.

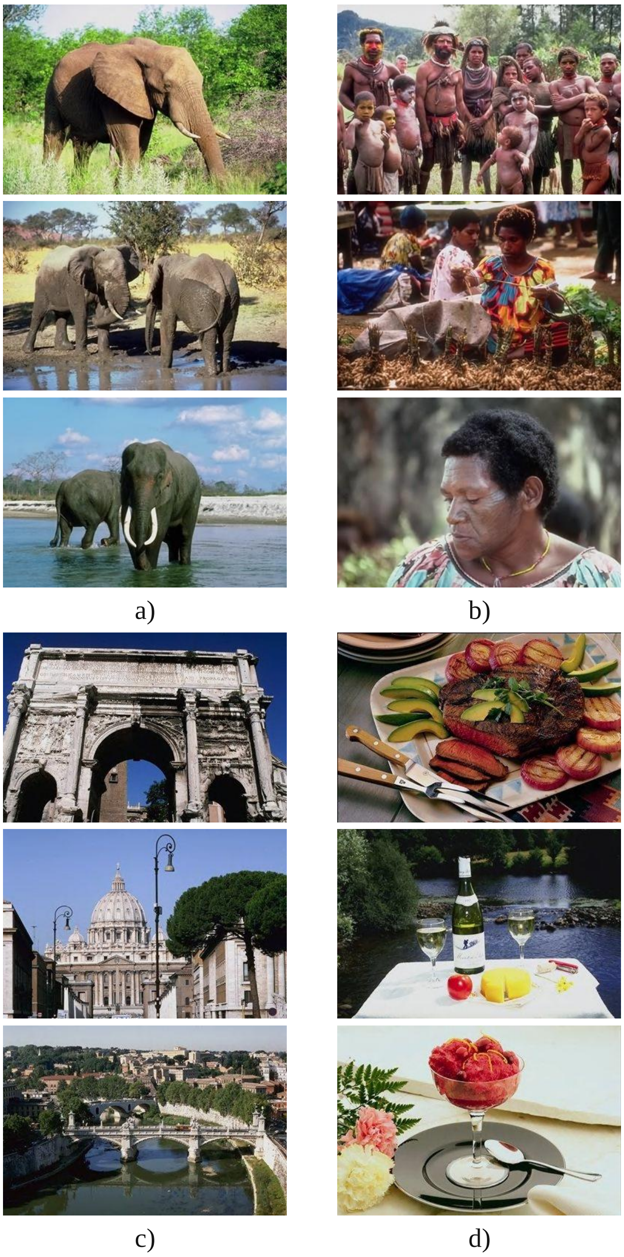

Fig. 3 Example images of several categories from *Corel1000* dataset: a) Animals, b) Africa people, c) Buildings, d) Food.

For the tests, we implemented a method in MATLAB and used it to measure the classification accuracy and speed. Only grayscale image information is used by first converting an image into *YCbCr* color space and using only *Y* channel for LBP extraction. Final descriptor is formed by concatenation of the LBP histograms extracted at 3 scales (original + 2 downsampled). To achieve correctness of results, tests were repeated for 50 times over random partitions of every category, where we randomly selected 50 images for training and other 50 for testing. Classification accuracy results are presented in Fig 4.

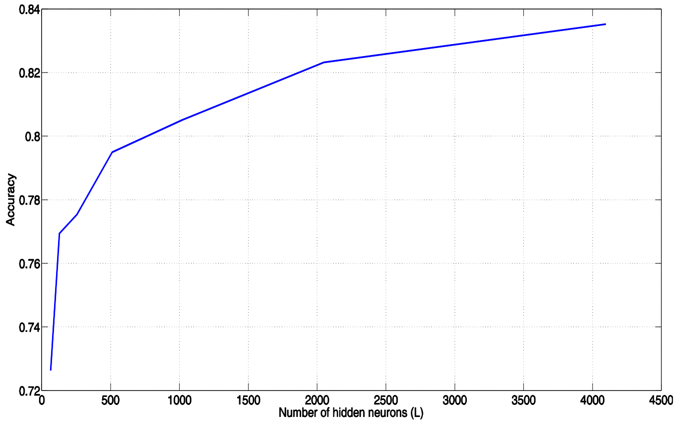

Fig. 4. Classification accuracy on Corel1000 dataset depending on the number of hidden neurons in ELM.

We further measured average training and testing time of the method on an *Intel Core i5 2.9GHz* computer. Results are presented in Fig. 5 and Fig. 6. Training time for all 500 images is only about 1 second (Fig. 5.), while test image classification is done instantly (< 0.1ms). These results demonstrate high performances in terms of training and test speed on this dataset.

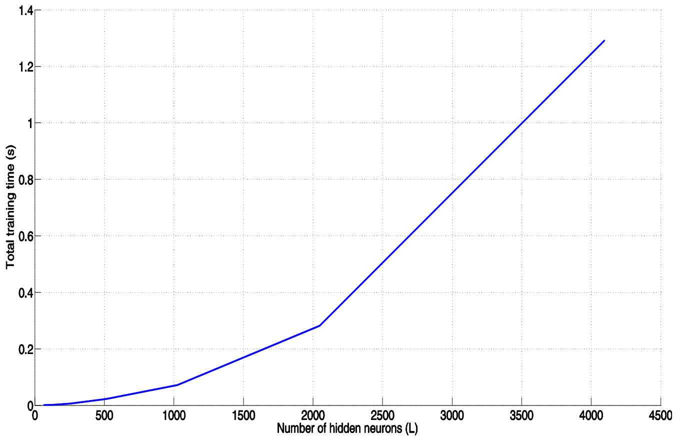

Fig. 5. Total ELM training time for all 500 training images.

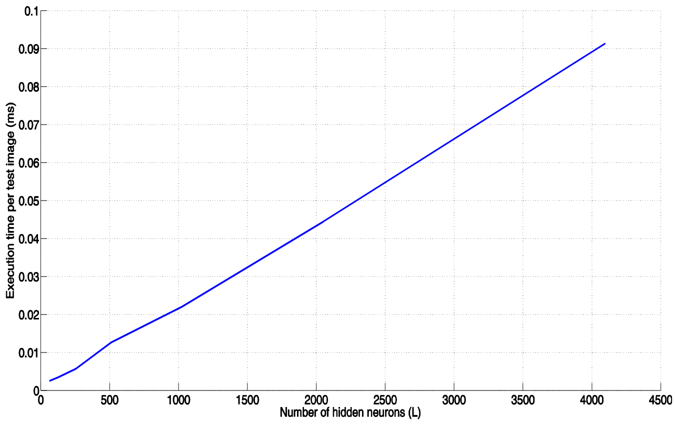

Fig. 6. Average ELM test time per image (in milliseconds).

It can be observed that increased number of neurons constantly improves classification results at the cost of increased training/testing time. Top achieved results are near to 83.52% for L = 4000 neurons in ELM hidden layer. For practical applications of the proposed method, one should experiment with values L>2000.

In order to compare results of the ELM with other common classification techniques, we measured accuracy of the Linear SVM and kernelized RBF SVM [12], on the same dataset. Linear SMV accuracy was 81.64%, while RBF SVM reached 83.49%. It can be noted that ELM outperforms Linear SVM in terms of accuracy, having the similar algorithm complexity. On the other side, ELM reaches results comparable to the kernelized SVM, while operating significantly faster during the training and testing.

V. Conclusion

In this paper we presented results of our research in the field of automatic image classification. Standard LBP image descriptor is used in combination with fast and powerful ELM classifier. Average accuracy of around 84% is acceptable result for rapid image classification. It can be concluded that combination of LBP descriptor with ELM classifier is reasonable choice for image classification applications. ELM classifier could be used as an alternative to the commonly used SVM. In the future, we plan to investigate performance of other types of image descriptors combined with ELM classifier, particularly integration 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] 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.