U.P.B. Sci. Bull., Series C, Vol. 77, Iss. 4, 2015 ISSN 2286-3540

CHOOSING THE MOST EFFICIENT CLASSIFIER FOR A FAST LOGO RECOGNITION SYSTEM

Raluca BOIA[1], Radu DOGARU[2]

In this paper we address the challenging problem of logo recognition. As logos appear in various scales, colors and with illuminations coming from source lights of various intensities, positions or colors, their recognition is a tedious task. The system is based on a Bag-of-Words framework with Scale Invariant Feature Transform (SIFT) features that are matched using a Random Sample Consensus (RANSAC) homography detection. The logo recognition system is completed by a classifier and we aim to choose the one providing the best compromise between time and quality performance from a set of five hardware-oriented implementations of classifiers in the context of FlickrLogos-32 database.

Keywords: logo recognition, machine learning, neural networks, classifier, image stitching

1. Introduction

Logo recognition is a challenging object recognition problem with a large area of applicability starting from modern marketing, advertising, and trademark registration to vehicle recognition. It is also useful for e-business applications, in retrieving and classifying products according to their logos, inspection of industrial goods, identifying the source of documents etc.

The challenges of logo recognition arise firstly from perspective deformations, varying background and occlusions. Also, the pattern might suffer different deformations, such as warping, since many logos are situated on non-planar surfaces. A great difficulty encountered consists in detecting very small logos, as their resolution can go down to 20x20 pixels. Furthermore, the intra-class variability is high, as a certain brand logo can suffer from variations in the color scheme or even in shape.

Although the interest in the domain of pattern recognition is extremely high, logo recognition has gained rather limited attention. The first approaches showed only partial success [1], [2], as they were limited in handling large image collections. A moment of breakthrough in scalable image object recognition was

[1]PhD Student, Doctoral School on Electronics, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: rboia@imag.pub.ro

[2]Professor, Department of Applied Electronics and Information Engineering, Natural Computing Laboratory, University POLITEHNICA of Bucharest, Romania, e-mail: radu_d@ieee.org

the introduction of the bag-of-visual words approach [3] that also lies at the basis of our system. In the past years, a number of works have addressed the logo detection using Bag-of-Words (BoW) techniques. Kleban et al. [4] do logo recognition by performing frequent item-set mining to discover association rules in spatial pyramids of visual words. Revaud et al. [5] use a bag-of-words-based approach coupled with learned weights whose purpose is to down-weight visual words that appear across different classes, while Romberg et al. [6] have developed a system inspired by the bag of visual words approach, by embedding spatial knowledge into the cascaded index. Romberg and Lienhart [7] later created a novel bundling technique on min-hashing of SIFT based visual words.

Since the number of brands having personalized logos increases every day, logo recognition systems must have a large processing capability and must be able to support a high number of classes and to handle large image collections. Existing methods for logo recognition still have issues when dealing with these requirements. Time performance is also an issue, if we take into consideration the need to process large amounts of data in short periods of time. A good example is the analysis of long sports videos for building statistics of the displayed trade-marks.

The aim of our research is to build a fast logo recognition system, designed to efficiently distinguish between large numbers of classes with high precision. The method relies on Difference-of-Gaussians (DoG) keypoint extractor and Scale Invariant Feature Transform (SIFT) for image description. The first novelty of the paper relies in the feature selection phase, which is done by pairing each two images from the training set for every class. This is accomplished with the use of homography computation using Random Sample Consensus (RANSAC) and has the purpose of ensuring a fair selection of the features situated in the logo area, by discarding those features corresponding to locations on areas of occlusions or background regions. The machine learning stage of the system is based on the bag-of-words technique, which was modified from the default version by replacing the linear SVM default classifier by a better-performing classifier. Our final purpose is to find the best option among five hardware-oriented classifiers. For this we adapted some of the classifiers such that they are more appropriate for embedded systems.

The remainder of the paper is structured as following: section 2 deals with feature extraction, selection and image descriptor creation. Section 3 presents the classification stage and classifier presentation, followed by section 4 depicting the experiments performed. The last two sections analyze the results obtained.

2. Image description

2.1. Feature extraction and preprocessing

In order to extract the most important features in the images we use Scale-Invariant Feature Transform (SIFT) algorithm, one of the most robust image descriptors. The locator of keypoints is the Difference-of-Gausssians (DoG) extractor and the description of the keypoints' vicinity is done by the SIFT local features [9].

We adapted the classical locator algorithm such as to extract more keypoints on the surface of the logos, by increasing the value of the edge threshold that eliminates peaks of the DoG scale space whose curvature is too small. Another technique of increasing the number of features extracted in the case of small sized logos is to enlarge the images containing small logos, while keeping the aspect ratio. The idea is similar to the one in [5], where the images are doubled in size prior to keypoint extraction. However, we choose to upscale just the small images (under 200x200 pixels), as we are interested in keeping a low running time of the algorithm. The upscaling size was selected empirically to be 200 pixels, as it gives the best increase in number of features while keeping the quality of the features and not introducing resizing artifacts.

2.2. Robust feature selection using homography detection

As we eventually want to describe the spatial layout of visual features specific to each class, we need to build class models, by finding the most relevant features. Consequently, we take all the possible image pairs from one class and try to find the full-correspondences of features – more exactly features that correspond spatially and as descriptor values, as suggested in [8]. In order to attain this, we use homography estimation between training images with RANSAC, using the open source library VlFeat [10].

Two images of the same planar surface are connected by a homography transform that has the property of being able to move points from one image plane to the second image's plane. Its computation requires at least 4 pairs of corresponding points from those images. The first stage consists in matching the SIFT descriptors, by using the algorithm suggested by Lowe [11], that rejects the ambiguous matches.

RANSAC algorithm deals with the outliers and finally outputs the homography transform between the images. A small number of inliers can lead to a degenerate transform, thus the correctitude of the homography needs to be tested and the criteria chosen is the one from [12].

After the matching, we will keep only the points in the common regions of the crops, meaning that we reject those coming from occlusions or background areas. For *n* image crops per class, *n(n-1)/2* image pairs are matched. Often it happens that the matching procedure does not work for a certain training image pair. Due to occlusions, inverted colors, large variations or distortions in shape, not all the pairs of images have enough matching points to output an appropriate homography. In addition, many logos are on reflecting background which is known to distort descriptors. However, the advantage of this exhaustive matching procedure is that images that cannot be matched with some of the images can still be matched to others. Thus, all images in a class are interconnected and the algorithm is able to select features from all training images.

2.3. Bag-of-words classes description

Each image will output a number of descriptors that represent it in the context of the class it is part of. Since the number of features is variable for each image, the only way to deal with this classification problem is to create a vocabulary of features that can accurately depict the images, through the method of bag-of-words.

This is a representation technique inspired from text classification [3]. In this model, a text is described as the multiple set of its words, disregarding grammar and even word order, but keeping count of the multiplicity. In the same manner, for image classification, local image features are treated as words and their distribution will characterize the entire image.

In this view, the features extracted and filtered for each image become the visual words. In order to form the visual vocabulary that will describe the entire dataset, the most representative features must be identified. To do so, the features coming from the training images are clustered with k-means. The choice of the number of words in the vocabulary is extremely important, since a large vocabulary can be too discriminative between the images in the same class, while a small vocabulary will not have enough power to represent each class differently. Our study includes several tests for different values of vocabulary size.

3. Classification stage

3.1. Training phase

Once the vocabulary is established, the features from all the images in the training dataset are labeled as the closest visual word, using k-dimensional trees, the results being that the logo images will be represented by frequencies of visual words.

Then, a classifier is trained to separate the images into the corresponding classes. The computed histograms of visual words of the images in each class are given as input to the classifier, which now has the task of learning the classes' particularities.

3.2. Testing phase

For a test image, using the already built vocabulary, the histogram of visual words is computed for that image. This is then fed to the previously trained classifier, which offers the label of the corresponding class.

3.3. Choice of classifier

We chose to change the default classifier used in the bag-of-words method, by replacing it with a more hardware-oriented learning system. While many choices exist for classification systems [13], we evaluate five hardware-oriented adaptations of widely used classifiers, since we aim to build a fast robust system of logo recognition. We further discuss their quality performance versus the time consumption and select the best compromise.

3.3.1. Adaline network

The Adaline (ADAptive LInear Neuron) network is the simplest linear adaptive network [14].

3.3.2. Neo-Fuzzy Neuron

The second low complexity classifier tested is the Neo-Fuzzy Neuron (NFN). The NFN [15] is produced by a fusion between fuzzy logic and neuroscience. Basically, a set of fuzzy membership functions expands the input feature space into a higher dimension and a simple Adaline is trained to optimize accuracy. Its advantages are the high-speed learning (100 times faster or more than the conventional multi-layer neural networks) and the guarantee for the convergence to the global minimum on the error-weight space. Although it was originally used for approximating nonlinear dynamic functions, the network is extremely useful for classification problems, as showed in [16], [17]. As an improvement, instead of using the Gaussian function from the original version [15], the triangular function is used, just as for the other hardware-oriented networks.

3.3.3. Fast Support Vector Classifier

A fast and compact network that aims to match the quality of the SVM is the Fast Support Vector Classifier (FSVC). Basically a SVM with the Gaussian kernel definition, the radial basis function (RBF) network can be improved such as to have a lower complexity and be more hardware-oriented. The modifications introduced in [18] achieve a decrease in time consumption of this network for both training and testing phase.

The first adaptation for decreasing the complexity is using a more simple kernel, a hardware oriented function, namely a piecewise linear function. Moreover, a completely different, much faster approach (when compared to SVM) is considered for training, through which new RBF neurons can be constructively created during one single training epoch with centers selected from the training dataset, based on a novelty criterion. Also, instead of using the classical Euclidean distance, the Manhattan distance (easier to compute in hardware-oriented systems) is used, since it was proved that the overall performance is not affected. The selection of support vectors step of the training algorithm has a complexity of $O(N)$ where N is the number of samples in the dataset. The remainder of the training algorithm is simply an Adaline system trained in an m dimensional space formed by the outputs of the RBF neurons.

3.3.4. Modified Naïve Bayes Classifier

The next classifier to test is a modified version of Naïve Bayes Classifier. The Bayes classifiers were introduced [19], [20] as an application of the Bayes theorem. In spite of the assumptions they make, they proved useful in many real-world problems, especially when the data set is large and the training time must be fast (e-mail filtering for example).

In our study, we used a modified version of the Naïve Bayes (NB) algorithm [21], in which changes were performed to make it more suitable for embedded systems. The modifications target the reduction of computational complexity being inspired from [22]: i) A tuning parameter was added which works similarly to the radius for radial basis neural networks; ii) The exponential function in the definition of the Gaussian basis is approximated with a simpler-to-implement piecewise linear (triangular) function. The modified NB classifier (MNB) allows improving the accuracy over the "standard" NB by proper tuning, and the speed is also improved.

3.3.5. Support Vector Machines

The Support Vector Machines (SVMs), introduced by Vapnick [23], are considered to be among the top methods of supervised learning. Vedaldi et al. [24] introduced the idea that large scale non-linear support vector machines can be approximated by linear ones using a suitable feature map. This greatly improves the performance of classification compared to the linear SVM but with equivalent time consumption.

4. Experiments and results

4.1. Logo Image Database

For a realistic evaluation of our proposed method, we chose the FlickrLogos-32 database [8] that depicts brand logos, formed by carefully selecting images from collections of photos in a real word environment.

The dataset contains 32 known logo classes. The database creators used a division of the dataset into disjoint subsets for training, validation and testing, each containing natural images of all 32 classes. We used their partitions for training and validation and the testing partition was kept as the original. As we aim at classifying the logos and not to solve the problem of localization, we use only crops containing the logos, extracted from the natural images given in the database. Thus, the training set holds approximately 53 logo images per class, while the test partition holds approximately 44 logos per class (1715 images for training and 1409 for testing).

Compared to other databases for object detection, FlickrLogos-32 can be considered a small-object dataset, by taking into account the average object size. This adds up to the other challenges brought by the database: the great variance of object sizes- from tiny logos in the background to image-filling views, perspective tilt, important rotations, color and shape changes inside the same class of objects, different occlusions and variable background. Moreover, the difficulty arises from the need to do a multi-class recognition on this large number of classes. Some examples that partly reveal the issues mentioned can be seen in Fig. 1.

Fig. 1. Sample images from FlickrLogos-32 containing logos from the classes Coca Cola, FedEx, Ferrari and Paulaner. Note the variability in logo appearance or due to shadowing, color balance, warping etc.

4.2. Results

For each classifier we compute the accuracy of the system, which is equal to the sum of true positives and true negatives over the number of all images. The first tests performed using the Adaline network vary the learning rate and the vocabulary size. The results are presented in table 1.

Table 1

System's accuracy using Adaline network

	Learning rate				
# of words	**0.00001**	**0.0001**	**0.00015**	**0.0002**	**0.0003**
1000	79.41%	87.23%	86.87%	86.51%	85.09%
2000	80.26%	87.57%	87.93%	87.79%	86.44%
3000	86.58%	**91.62%**	91.55%	91.55%	90.48%
4000	80.55%	87.86%	84.67%	81.61%	86.08%

The experiments performed using the Neo-Fuzzy Neuron implied a tuning of the number of fuzzy membership functions and the learning rate, as depicted in table 2.

Table 2

System's accuracy using NFN

# of fuzzy membership functions	**# of words**	**learning rate 0.00005**	**learning rate 0.00008**	**learning rate 0.0001**	**learning rate 0.0002**
5	**1000**	87.78	89.56	89.63	88.71
	2000	89.42	89.92	90.06	90.41
	3000	91.55	**91.76**	91.62	91.05
	4000	91.41	91.12	90.98	88.43

For the Fast Support Vector Classifier, the radius (R) and the learning rate (η) are tuned in order to test the capabilities of system in the issue of logo recognition. The tests were performed using the implementation of FSVC[3] and they also include a dimensionality reduction using PCA**.** Table 3 indicates that using a smaller-sized descriptor increases the accuracy of the classifier. An important remark is that the best result obtained by the FSVC and PCA (88.07%) is better than the one obtained with simple linear SVM i.e. 81.87%.

Table 3

System's accuracy using FSVC

	PCA size 100			**No PCA**	**PCA size 100**				**No PCA**
	η = 0.1				**η = 0.01**				
# of words	**R = 10**	**R = 8**	**R = 6**	**R = 50**	**R = 10**	**R = 8**	**R = 6**	**R = 5**	**50**
1000	80.05	83.53	74.16	66.78	84.74	84.67	76.36	61.17	66.78
2000	75.94	81.9	83.74	58.05	83.39	85.3	85.66	80.62	58.05
3000	62.52	84.67	86.08	31.58	78.92	86.44	87.65	87.08	31.58
4000	49.04	72.6	85.37	19.94	72.39	86.01	87.86	**88.07**	19.94

For the Modified Naïve Bayes classifier, in order to achieve the best accuracy in classifying logos, various values of the tuning parameter were tested. The results using vocabularies sized at 1000 – 4000 words prove to be poor.

[3] http://www.mathworks.com/matlabcentral/fileexchange/49695-fast-support-vector-classifier--a-low-complexity-alternative-to-svm-

However, if the data is reduced to a smaller dimensionality using Principal Component Analysis (PCA), the accuracy of the system increases considerably. This proves that the MNB classifier has a better classification power when working on smaller dimensions. Table 4 depicts the best results of the network obtained for a value of the tuning parameter of 30.

Table 4

System's accuracy using Modified Naive Bayes

	PCA Size			
# of words	**50**	**70**	**100**	**No PCA**
1000	79.63%	79.28%	78.92%	48.9%
2000	81.83%	81.48%	80.91%	16.32%
3000	81.9%	82.04%	81.48%	8.3%
4000	83.75%	**83.89%**	83.53%	6.17%

The tests for the Support Vector Machines were performed by varying the homogeneity factor of the χ2 kernel. The importance of porting the data into a higher dimension using the feature map is proved by the fact that without this adaptation the performance of the system is lower. These accuracies can be analyzed in table 5.

Table 5

System's accuracy using linear SVM

	Feature map homogeneity factor						No feature map
# of words	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**1**	
1000	88.37%	89.22%	89.56%	89.68%	89.37%	89.28%	78.78%
2000	90.68%	90.90%	91.15%	91.06%	90.87%	90.87%	80.18%
3000	91.09%	91.96%	92.06%	91.78%	91.40%	91.15%	82.37%
4000	91.18%	91.84%	**92 .31%**	92%	91.96%	91.84%	81.87%

5. Discussion

5.1. Time Consumption and Complexity

Since a logo recognition system must be able to easily learn new classes, not only the testing time of the learning system has a high importance, but also the training time. Table 6 holds the measured time consumptions for the Matlab implementations of the classifiers tested on an Intel i7 4770 3.4 GHz processor, 16Gb RAM machine. The SVM system was implemented in C++, thus its time consumption is not reported, since the comparison would not be fair.

For an objective comparison of the learning systems, a computation of their training and recall complexities is necessary. For the Adaline network, for an input size of n, N samples and M classes, the training time is *nr epochs* $* N * (n+1) * M$. Thus, the training complexity is $O(n \cdot N \cdot M)$. For the recall phase (one single epoch) the number of operations needed is $(n+1)*M$, giving a complexity of $O(n \cdot M)$. Since the system does not require any additional computation of various bases, its recall complexity is the lowest among all investigated classifiers. Moreover, for both Adaline and Neo-fuzzy-neurons, low complexity hardware-oriented solutions are provided by replacing multiplication (requiring a large number of basic logical gates) with comparisons (with a minimal requirement of logical gates) while no significant changes in performance are observed [17].

Table 6

Comparative performance in the context of Logo recognition on the FlickrLogo-32 database

Classifier	Accuracy [%]	Training + Testing time [s]
MNB - PCA	83.89	0.45
FSVC	88.07	315.22
NFN	91.76	518.69
Adaline	91.62	87.83

For the neo-fuzzy neuron with *nf* fuzzy logic membership functions, the training complexity *is* $O(nf \cdot n \cdot N \cdot M)$, while the recall complexity (for one sample) is $O(nf \cdot n \cdot M)$. Basically, in this case the complexity is *nf* times larger that of the Adaline plus some additional resources needed to compute the fuzzy membership functions.

The Modified Naïve Bayes involves computing *2nM* means and dispersions based on *N* training samples, leading to a training complexity of $O(n \cdot N \cdot M)$ plus the complexity of using M radial basis function units. Although the complexity is comparable to that of training the Adaline, it is actually faster since it requires *only one training epoch* (compared to several tens in the case of Adaline or Neo-fuzzy neuron). The recall complexity of MNB is comparable to that of Adaline i.e. $O(n \cdot M)$. In terms of training, the MNB classifier is the fastest among the investigated classifiers.

For the linear SVM network, the training and test complexity are the same as for the Adaline classifier, since recent advances have made it possible to learn linear support vector machines in time linear with the number of training examples [25]. However, to this we must add the time needed for transporting the data from its dimension to the Vedaldi feature map used in our classification system. Consequently, compared to Adaline, the SVM and feature expansion solution is generally slower.

For FSVC, the training complexity is generally larger than for Adaline, since it includes an Adaline system operating in an *m*-dimensional feature space (generally *m* is smaller than *n* for our problem but larger than *M*) but it also needs to compute *m* distances and RBF functions among *n-sized* vectors. Finally, the recall complexity is $O((n+M) \cdot m)$ now proportional to the number of RBF units (usually larger than the number of classes *M*).

5.2. Quality performance

Regarding the quality, the best accuracy of 92.31% was obtained using the SVM network with a feature map and a 4000 words vocabulary. The next obtained accuracy was of 91.76% of Neo-Fuzzy Neuron trained on a vocabulary of 3000 words. The Adaline network gives 91.62% accuracy for the vocabulary of 3000 words. Some classifiers work better if the data dimensionality is reduced, thus through the use of PCA we obtained high accuracies for FSVC, namely 88.07% and for the modified Naïve Bayes classifier 83.89%.

When the classification speed is an issue, accepting a small loss of performance (-0.7%) allows the use of the fastest solution, in our case the Adaline network, which requires no additional feature expansion and once trained has a very fast speed in ensuring a decision.

6. Conclusions

In conclusion, if the size of the vocabulary is optimally chosen, the problem becomes almost linearly separable and the simplest classifier (Adaline) will have enough power to separate the classes of logos with only a very small loss in performance compared to a more sophisticated solution (linear SVM plus nonlinear expansion).

The tests performed have proven that the vocabulary size should not be very small, as the generalization power will be lower, especially when there is large intra-class variability. However, a very large vocabulary will introduce quantization errors and offer a less appropriate class description.

Acknowledgements

This work was partially supported by the Romanian Sectoral Operational Programme Human Resources Development 2007-2013 through the European Social Fund Financial Agreements POSDRU/159/1.5/S/132495.

R E F E R E N C E S

[1] *A. Bagdanov*, *L. Ballan*, *M. Bertini*, and *A. Del Bimbo*. “Trademark matching and retrieval in sports video databases”. In Proc. of the ACM International Workshop on Multimedia Information Retrieval, MIR, 2007, pp. 79-86.

[2] *A. Joly* and *O. Buisson*. “Logo retrieval with a contrario visual query expansion”. In Proc. of the ACM International Conf. on Multimedia, ACM MM, 2009, pp. 581-584.

[3] *J. Sivic* and *A. Zisserman*. “Video Google: a text retrieval approach to object matching in videos”. In Proc. of International Conf. on Computer Vision, ICCV, 2003, pp. 1470 - 1477.

[4] *J. Kleban*, *X. Xie*, and *Wei-Ying Ma*. “Spatial pyramid mining for logo detection in natural scenes”. In Proc. of IEEE ICME, 2008.

[5] *J. Revaud*, *M. Douze*, *C. Schmid*. “Correlation-Based Burstiness for Logo Retrieval”. In Proc. of the ACM MM, 2012, pp. 965-968.
[6] *S. Romberg*, *L. Garcia Pueyo*, *R. Lienhart*, and *R. van Zwol*. “Scalable Logo Recognition in Real-World Images”. In Proc. of International Conference on Multimedia Retrieval, ICMR, 2011.
[7] *S. Romberg*, *R. Lienhart*. “Bundle Min-Hashing for Logo Recognition”. In Proc. of International Conf. on Multimedia Retrieval, ICMR 2013, 113-120.
[8] S. *Romberg, L. Garcia Pueyo*, *R. Lienhar*t, *R. van Zwol*, .: “Scalable logo recognition in real-world images”. In: ACM ICMR. (2011) 965-968
[9] *D. Lowe*. “Distinctive image features from scale-invariant keypoints”. International Journal of Computer Vision, **vol. 60(2),** pp. 91–110, 2004
[10] *A. Vedaldi, B. Fulkerson*. “VLFeat: an open and portable library of computer vision algorithms”. In Proc. of ACM MM, 2010, pp. 1469-1472.
[11] M. *Brown, D. Lowe,*.: “Automatic panoramic image stitching using invariant features. International Journal of Computer Vision, **vol. 74(1)**, 2006, pp. 59-73
[12] *Richard Hartley* and *Andrew Zisserman*, Multiple View Geometry in Computer Vision Second Edition, Cambridge University Press, March 2004.
[13] *C. Bishop*, "Pattern recognition and machine learning", 2006, Springer.
[14] *B. Widrow* and *M. E. Hoff, Jr*., “Adaptive switching circuits,” in IRE WESCON Conv. Rec., pt. 4, pp. 96–104, 1960.
[15] *T. Miki* and *T. Yamakawa*. Analog implementation of neo-fuzzy neuron and its on-board learning. In N.E. Mastorakis, editor, Computational Intelligence and Applications, pages 144–149, Piraeus, 1999. WSES Press
[16] *Bodyanskiy Ye., Zaychenko Yu., Pavlikovskaya E., Samarina M., Viktorov Ye.*: “The neo-fuzzy neural network structure optimization using the GMDH for the solving forecasting and classification problems”. Proc. Int. Workshop on Inductive Modeling,. 77-89, 2009.
[17] *R. Dogaru, L.O. Chua*, "The comparative synapse: a multiplication free approach to neuro-fuzzy classifiers", in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Vol. 46, Nr. 11, November 1999, pp. 1366 - 1371.
[18] *R. Dogaru*, “A hardware oriented classifier with simple constructive training based on support vectors”, in Proceedings of CSCS-16, the 16th Int’l Conference on Control Systems and Computer Science, May 23 - 26, 2007, Bucharest, Vol.1, pp. 415-418.
[19] *M. E. Maron*, “Automatic Indexing: An Experimental Inquiry”, Journal of the ACM (JACM), 8(3):404–417, 1961.
[20] *M. Minsky*, “Steps toward artificial intelligence”, Proceedings of the Institute of Radio Engineers, **vol. 49**, pp. 8-30, 1961.
[21] *R. Dogaru*, “A Modified Naïve Bayes Classifier for Efficient Implementations in Embedded Systems”, International Symposium on Signals, Circuits and Systems - ISSCS, 2011.
[22] *R. Dogaru*, *A.T. Murgan*, *S. Ortmann*, *M. Glesner*, “A modified RBF neural network for efficient current-mode VLSI implementation”, Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (Micro-Neuro’96), IEEE Computer- Press, pp. 265-270, 1996.
[23] *V. Vapnick*, “The Nature of Statistical Learning Theory”, Springer-Verlag, New-York, 1995.
[24] *A. Vedaldi* and *A. Zisserman*, Efficient Additive Kernels via Explicit Feature Maps, Proc. CVPR, 2010.
[25] *T. Joachims*, “Training linear SVMs in linear time,” in Proc. KDD, 2006.