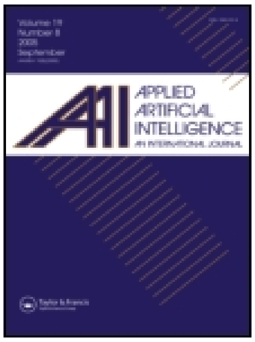

Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: http://www.tandfonline.com/loi/uaai20

Robust Boosted Parameter Based Combined Classifier for Rotation Invariant Texture Classification

A. H. El-Baz, A. S. Tolba & Sankar K. Pal

APPLIED ARTIFICIAL INTELLIGENCE
2016, VOL. 30, NO. 2, 77–96
http://dx.doi.org/10.1080/08839514.2016.1138806

Taylor & Francis
Taylor & Francis Group

Robust Boosted Parameter Based Combined Classifier for Rotation Invariant Texture Classification

A. H. El-Baz[a,b], A. S. Tolba[c], and Sankar K. Pal[b]

[a]Department of Mathematics, Faculty of Science, Damietta University, New Damietta City, Egypt; [b]Center for Soft Computing Research, Indian Statistical Institute, Calcutta, India; [c]Department of Computer Science, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt

ABSTRACT
Texture analysis and classification remain as one of the biggest challenges for the field of computer vision and pattern recognition. This article presents a robust hybrid combination technique to build a combined classifier that is able to tackle the problem of classification of rotation-invariant 2D textures. Diversity in the components of the combined classifier is enforced through variation of the parameters related to both architecture design and training stages of a neural network classifier. The boosting algorithm is used to make perturbation of the training set using Multi-Layer Perceptron (MLP) as the base classifier. The final decision of the proposed combined classifier is based on the majority voting. Experiments' results on a standard benchmark database of rotated textures show that the proposed hybrid combination method is very robust, and it presents an excellent texture discrimination for all considered classes, overcoming traditional texture modification methods.

Introduction

Texture analysis and classification have a huge variety of applications in many areas such as computer vision, remote sensing, medical image processing, defect detection, and image retrieval and classification. Although it has been widely studied, the topic remains open for research and, in fact, is one of the biggest challenges for the field of computer vision and pattern recognition. Texture classification assigns a label to every texture category using a classification rule. A texture classification system has two major components: the feature extractor and the classifier.

Early methods for texture classification focus on the statistical analysis of texture images. The representative methods include the co-occurrence matrix method (Haralick, Shanmugam, and Dinstein 1973) and filtering-based approaches (Randen and Husoy 1999), such as Gabor filtering (Manjunath and Ma 1996), wavelet transform (Laine and Fan 1993), and wavelet frames

CONTACT Dr. A. H. El-Baz, PhD elbaz@du.edu.eg Mathematics Department, Faculty of Science, Damietta University, New Damietta City, 34517, Egypt.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/UAAI.

(Unser 1995). In general, their classification results are good as long as the training and test samples have identical or similar orientations. However, the rotations of real-world textures will vary arbitrarily, severely affecting the performance of the statistical methods and suggesting the need for rotation-invariant methods of texture classification.

Kashyap and Khotanzad (1986) were among the first researchers to study rotation-invariant texture classification using a circular autoregressive model. Later models include the multiresolution autoregressive model (Mao and Jain 1992), hidden Markov model (Wu and Wei 1996), Gaussian Markov random field (Deng and Clausi 2004), and the autocorrelation model (Campisi et al. 2004). Ojala, Pietikainen, and Maenpaa (2002) proposed using a local binary pattern (LBP) histogram for rotation-invariant texture classification. Varma and Zisserman (2005) presented a statistical algorithm, MR8, in which a rotation-invariant texture on a library is first built from a training set and then an unknown texture image is classified according to its texton distribution. The LBP and MR8 methods are both state-of-the-art algorithms and yield good classification results on large and complex databases (Varma and Zisserman 2005). Scale and affine invariance is another issue to be addressed in texture classification, and some pioneer works have been proposed that use affine adaption (Lazebnik, Schmid, and Ponce 2005), fractal analysis (Xu, Ji, and Fermuller 2006) and combination of filters (Mellor, Hong, and Brady 2008).

Recently in Guo, Zhang, and Zhang (2010), a hybrid scheme, globally rotation-invariant matching with locally variant LBP texture features is proposed. Using LBP distribution, they first estimate the principal orientations of the texture image and then use them to align LBP histograms. The aligned histograms are then, in turn, used to measure the dissimilarity between images. More recently, in Li et al. (2014), a rapid-transform-based descriptor is proposed for rotation-invariant texture classification. The proposed descriptor is based on the local circular neighborhood, and the local feature vector is obtained by means of rapid-transform. The local feature vector is rotation-invariant because of the cyclic shift invariance property of rapid-transform.

However, neural classifiers can be used for feature extraction when trained on raw texture data. Neural classifiers are categorized as base classifiers or ensemble classifiers as given in Figure 1. A base neural classifier has a single learning algorithm and is trained on a single fixed training dataset. An ensemble neural classifier, however, consists of a combination of base classifiers.

Classifier combination is a popular technique that has been applied to many pattern recognition and classification tasks. A number of terminologies are used in the pattern recognition community for classifier combination, such as classifier ensembles, classifier fusion, mixture of experts, classifier integration, etc. (Kuncheva 2014). One of the main benefits of classifier combination is that the various classifiers in the combination would complement each other, resulting in better performance. The combination itself can

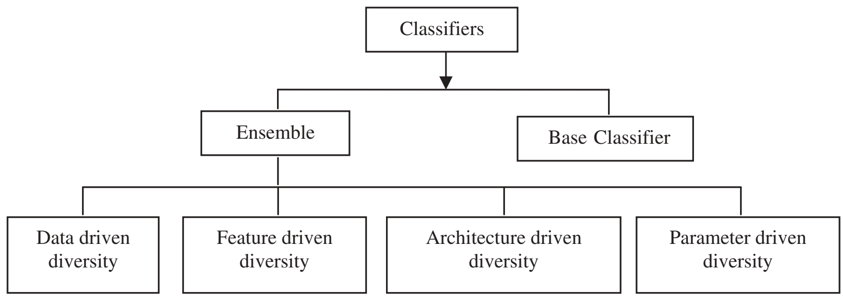

Figure 1. Classifiers categories.

be at the level of features or at the level of decisions (Kuncheva 2014). In this study, we have used the decision-level combination.

Classifier combinations have been extensively used for a wide variety of tasks. A useful background for using classification combinations was given in Kittler et al. (1998). In (Lotte et al. 2007), an elaborate review of classification algorithms for EEG-based brain-computer interfaces (BCIs) including classifier combinations was given. Successful applications of classifier combinations for BCI have been reported also (Rakotomamonjy et al. 2005). Classifier combinations have been used in Lu, Wang, and Jain (2003) for face recognition purposes. Besides, a rough set based ensemble classifier was used for classification of web pages (Saha et al. 2007) and an ensemble classifier was applied for web services categorization and focused crawling (Saha et al. 2010). More recently, a combination of neural network classifiers was applied in 3D hand gesture recognition (El-Baz and Tolba 2013) and in tracking of human motion (Etemad and Arya 2014). Also, a hybrid intelligent system-based rough set and ensemble classifier has been used for breast cancer diagnosis (El-Baz 2015).

Classifier combination schemes broadly consist of a set of classifiers and a combiner that combines the outputs of the classifiers to produce the final result (Figures 2 and 3). The individual classifiers can be put together in basically three architectures: serial or cascade architecture, parallel architecture, and hierarchical architecture. In this work, we have used the parallel architecture. The majority voting scheme is used to combine the outputs of the individual base classifiers.

The main contribution of this article is building a combined classifier that is based on a hybrid combination of both the parameter variation of architecture design of neural network and the AdaBoost to achieve better detection results through the concept of decision fusion. We show that by using a combination of relatively weak classifiers, we can achieve better overall performance on a standard rotated textures database.

The rest of the article is organized as follows. "Background Knowledge" describes previous work related to this study. The proposed combined

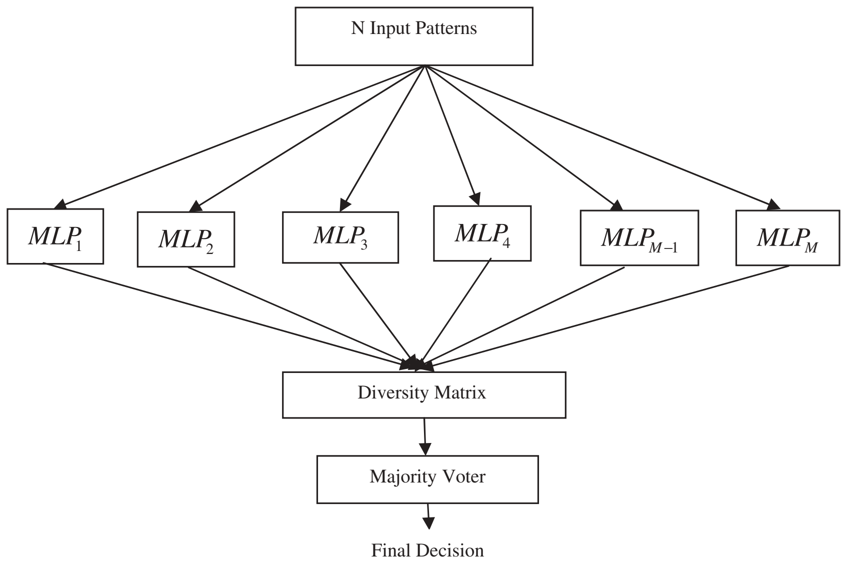

Figure 2. Architecture of parameter-based ensemble.

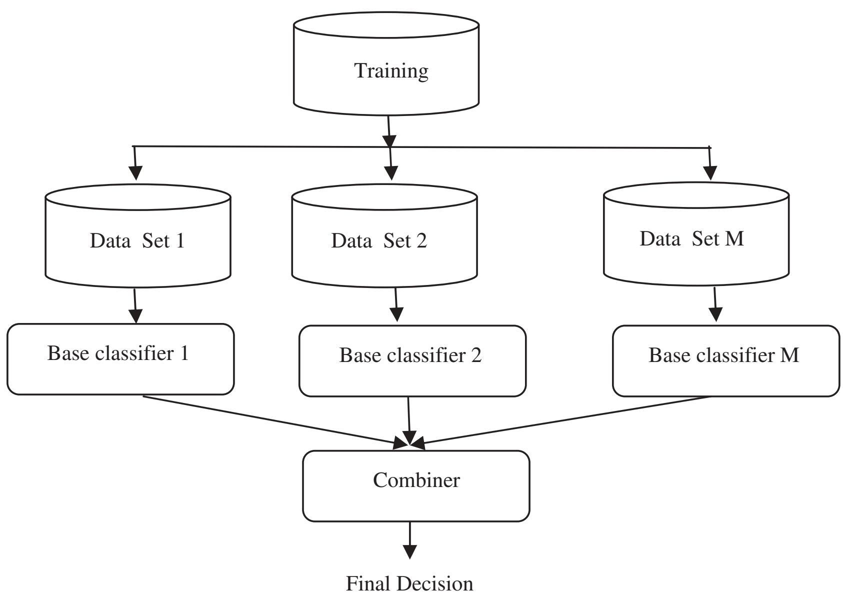

Figure 3. Architecture of data-driven ensembles.

classifier and the underpinning algorithm are presented in "Proposed Method.". In "Experiments on Rotated Texture Database," the experimental evaluations and comparison of the proposed ensemble technique on standard rotation textures databases are given. The final section concludes the article.

Background knowledge

Ensemble classifiers

Classifiers such as Multi-Layer Perceptron (MLP), learning vector quantization networks (LVQ), and Support Vector Machine (SVM) (Li et al. 2003) suffer from the problem of the selection of the suitable parameter set, and improper selection results in inferior and unstable performance.

Additionally, base classifiers are susceptible to facing the bias-variance dilemma (Geman, Bienenstock, and Doursat 1992). Combining multiple classifiers in an ensemble helps solve this dilemma by reducing both bias and variance. Bagging and Boosting algorithms are the most common techniques for diversity enforcement in classifier ensembles through the dataset. Theoretical analysis of a bagging algorithm could be found in Fumera, Roli, and Serrau (2008). Also, Eibl and Pfeiffer (2005) give explanation and theoretical analysis of the boosting algorithm. Table 1 summarizes the classification principle and weaknesses and strengths of different classification approaches.

Ensemble methods enforce diversity in performance of base classifiers using bootstrapped versions of data (Kittler et al. 1998; Etemad and Arya 2014; Li et al. 2003). The AdaBoost is the most widely used algorithm for construction of classifier ensembles. As a theoretically founded algorithm, it substitutes for the misclassifications made by weak classifiers (Schapire 1990). Table 2 shows a comparison of three different diversity enforcement methods: bagging, boosting, and parameter based diversity enforcement (PBDE).

Figure 2 shows the architecture of a parameter -based ensemble of MLP base classifiers. All base MLPs are trained on the same complete pattern set, and the decisions of all MLPs are then combined using majority voting. A stopping condition is used to stop adding more base MLPs to the ensemble as long as more MLPs do not change the performance. Figure 3 shows the architecture of the data-driven ensemble. All base MLPs are trained on different datasets and the decisions of all MLPs are then combined using majority voting. The architecture of both boosting- and bagging-based ensembles, which belong to the data-driven ensemble, is completely different from that of the parameter-based ensemble.

Diversity of ensemble base classifiers

Diverse base classifier outputs enhance the accuracy of the ensembles built around them. Different classification errors made by base classifiers are preferred. Therefore, only those base classifiers that have diverse classification results are used for getting the majority vote. To identify the most diverse set of base classifiers, the diversity matrix for M MLP base classifiers, which are used in testing N patterns, is constructed by calculating the normalized Hamming distances between the classification results of all

Table 1. Base and ensemble classifiers approaches.

Category	Summary	Examples	Reference
Base Classifiers	Classification Principle: based on a single learning algorithm Strengths: Low computational complexity Weaknesses: Low accuracy Unstable behavior High sensitivity to data or parameter variation	MLP, SVM	(Sengur et al. 2007) (Shutao et al. 2003) (Haykin 1999) (Li et al. 2003)
Data-driven Ensembles	Classification Principle: Rely on sensitivity to data variation or parameter variation Strengths: High detection accuracy Stable behavior Encompasses diversity Low sensitivity to data or parameter variation No overfitting suitable for nonparametric methods Weaknesses: High computational complexity data dependent	Boosting Other	(Breiman 1996) (Freund and Schapire 1997)
Feature-Based Ensembles	Classification Principle: Same algorithm applied on different features or feature sets Strengths: Enforce more diversity Weaknesses: Applicable only to parametric approaches	Bagging Boosting	(Breiman 1996) (Freund and Schapire 1997)
Classifier-driven Ensembles	Classification Principle: by different learning algorithms combines the features of statistical, neural, or machine learning methods Strengths: suitable for nonparametric methods overcome deficiencies of a particular classification algorithm exploit the advantages of multiple approaches while overcoming their weaknesses Weaknesses: learning algorithm dependent high computational complexity	MLP with Parzen Windows and HMMs, EM with HMMs, GBFN, JAM System Fusion scheme based on multiple SVMs with translation invariant wavelet transform features	(Hodge and Austin 2004) (Kwang et al. 2002)
Parameter-Based Ensembles	Detection Principle: by parameter change same data or features applied to detectors of different parameter sets and same learning algorithms Strengths: more room for diversity enforcement Weaknesses: depending on parameters. It is suitable for parametric methods	Parameter-based ensemble	proposed method

Table 2. Comparison of diversity enforcement methods (Kuncheva 2014).

Aspect/Diversity Enforcement	Bagging	Boosting	PBDE
Processing	Parallel	Serial	Parallel
Diversity	Bootstrap replicates of the training set.	Emphasis on misclassified patterns	Parameters variation
Variance, Bias	Reduces variance without changing the bias. However, it has been found to reduce the bias as well, for example, for high-bias classifiers.	Boosting primarily reduces bias while at the later iterations it has been found to reduce mainly variance	Reduces both bias and variance
Accuracy	High	High	Higher
Weakness	High computational complexity	Sensitive to noise and outliers, especially for small datasets High computational complexity	High computational complexity

possible classifier pairs for N patterns. The number of diversity values for an ensemble of M base classifiers is $M.(M\text{-}1)/2$. In this study, the diversity between a pair of base classifiers will be calculated using the Normalized Hamming Distance (NHD). The normalized Hamming distances $d_{\text{NHD}}(i,j)$ between two vectors i and j is given by the number of mismatching elements of the two vectors divided by the vector size N (Schulz 2008).

$$d_{\text{NHD}}(i,j) = \frac{1}{N}\sum_{k=1}^{N}[y_{i,k} \neq y_{j,k}] \tag{1}$$

The median of the NHD is then used as a threshold above which the classifier pair is added to the pool of classifiers that will be used in calculating the majority vote.

Majority voting

Given m learners (MLP neural networks) and n classes (normal, abnormal), we denote by d_{ji} the posterior probability estimate of learner j for class i given the input. Different d_j's use different parameter sets (fs_j) to represent the input. In Moglu and Alpaydin (2001), to combine the final decision of the individual classifiers in order to find fd_i, the final estimate for class i, given input fs,

$$fd_i = f(d_{\text{MLP}_1}(fs_1), d_{\text{MLP}_2}(fs_2), \ldots, d_{\text{MLP}_{mi}}(fs_m))|\psi, \tag{2}$$

f is the combining function with ψ denoting its parameters. Assuming that fd_i estimates the posterior probability of class i, and that the zero-one loss function is used, to minimize Bayesian risk, we choose the class with the highest probability:

$$i^* = \arg\max_i r_i \tag{3}$$

To fuse the decisions of a group of individual weak classifiers, the simple majority vote algorithm is used to determine in any given sequence of votes whether there is a candidate with more votes than all the others, and if so, to determine this candidate. Majority voting has many advantages such as effectiveness, robustness to noise, and avoidance of overfitting. Each classifier MLP_i is independently trained on a certain feature set (fs_j) and decides on the pattern class by its individual decision, d_{MLP_i}. Perrone (1993) gives a number of didactic examples that depict the advantage of voting. He also shows that for minimum square error, when the learners are unbiased and uncorrelated, weights should be inversely proportional to variances.

Probability of wrong prediction

An ensemble of classifiers is expected to improve the overall performance of the detection system because the individual classifiers complement each other. As a measure of performance, the probability of wrong prediction is expected to be low for an ensemble of classifiers. Assuming an ensemble of six classifiers, the probability of wrong prediction is given by (Kuncheva 2014):

$$P(\text{ensemble incorrect}) = \sum_{i=1}^{6} \binom{6}{i} \varepsilon^i (1-\varepsilon)^{6-i}, \tag{4}$$

where ε is an average error rate of the base classifier. This result is consistent with our empirical results of improved performance using an ensemble of classifiers.

Proposed method

The base classifier

The base classifier used in this study is the MLP. It consists of three layers: an input layer, an output layer, and one hidden layer. Each layer is composed of a predefined number of neurons. The training algorithm adopted in this study is the resilient propagation algorithm (RPRO; Reidmiller and Braun 1993). Each gray-scale texture image is resized to $32 \times 32 = (1024)$ pixels and is used as a set of texture features. Each neuron in the input layer represents one of these pixels. Consequently, the input layer contains 1024 neurons. The hidden layer contains n neurons. This number is used as a varying parameter to generate base classifiers and enforce diversity. There are *13* neurons in the output layer, each of them represents a texture class in the texture database used.

Stability of base and ensemble classifiers

Neural networks' stability is crucial in the applications. Figures 4 and 5 show the oscillatory performance of the base MLP classifiers with the variation of the number of training epochs or the number of hidden neurons. This instability in performance indicates that selection of MLP parameters is a challenging task. Experimental results in this study indicate that the performance of boosted MLP is much more stable than that of the base MLP, and the

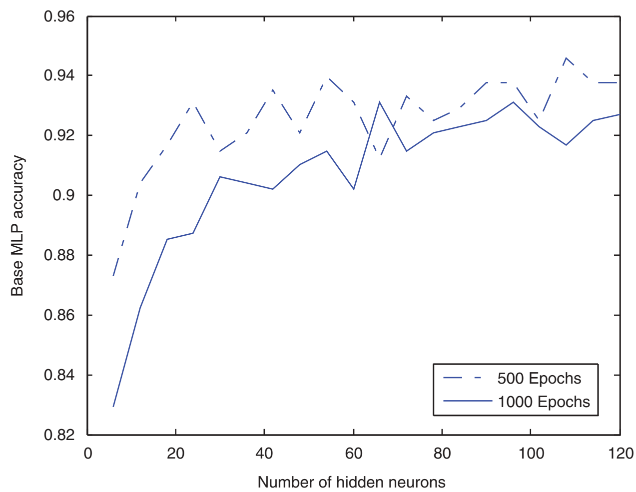

Figure 4. Accuracy of base MLP versus the number of hidden neurons.

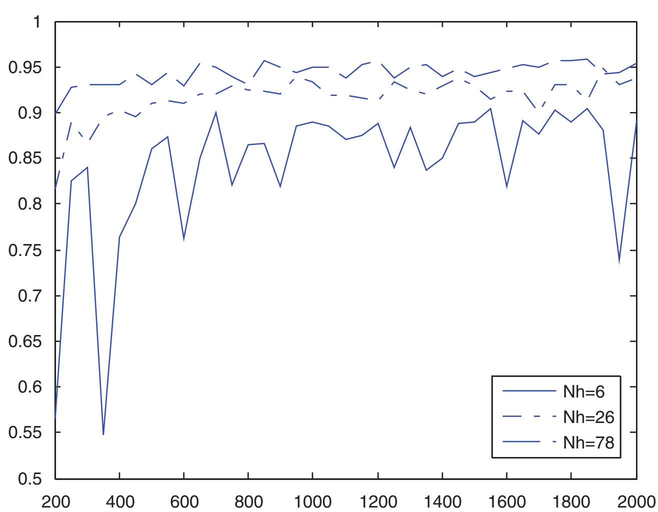

Figure 5. Accuracy of base MLP versus the number of training epochs.

parameter-boosted MLP is more stable than both. The major challenge with the MLP training is to select the suitable training parameters such as

(1) initialization of the weight vectors,
(2) number of training epochs,
(3) number of hidden neurons.

Figures 4 and 5 show the sensitivity of the MLP classifier to parameter variation. It is observed that all the previous parameters greatly affect the performance of the base MLP classifier. This weakness of the base classifier has been converted to an opportunity for enforcing diversity of the pool of base MLP, which is used for constructing the boosted parameter-based ensemble.

The variability of the base MLP performance can be due to many factors:

(1) small changes in the initial random weights can lead to large changes in the MLP representations of the input patterns (chaos theory; see Figure 6).
(2) less training results in less accurate representation of the input pattern space and more training results in overfitting (bias-variance dilemma)

Rationale behind the proposed boosted parameter based MLPS combined classifier?

The design of a combined classifier system has been widely used in neural networks. A number of researchers have applied it to improve the

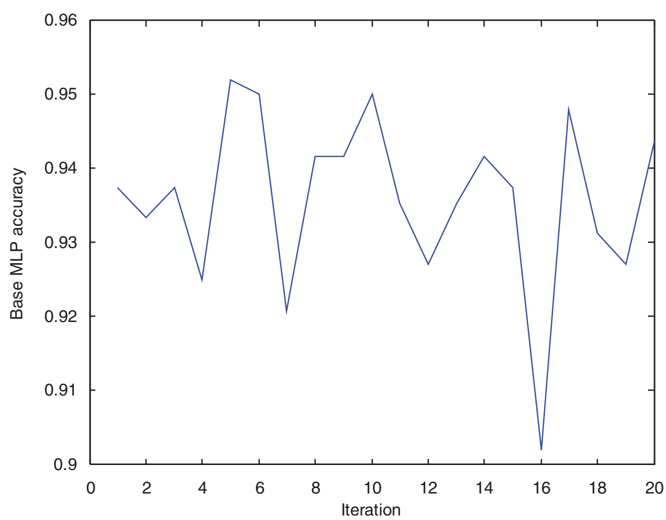

Figure 6. Variation of base MLP accuracy with different initial weights.

performance of a neural network (Tolba and Abu-Rezq 2000; Kang and Park 2009; Meynet and Thiran 2010). Figures 7–9 provide the flow chart and the algorithm of the proposed combined classifier system.

Representation of rotation invariance is a challenging task from two perspectives: feature extraction and classification. The MLP has been trained on the rotated gray-level texture blocks, which makes use of the inherent feature extraction capability of the neural classifiers. Design of a robust base classifier is not possible because of the parameter selection and data dependency problems, which could result in an unstable classifier. Therefore, using

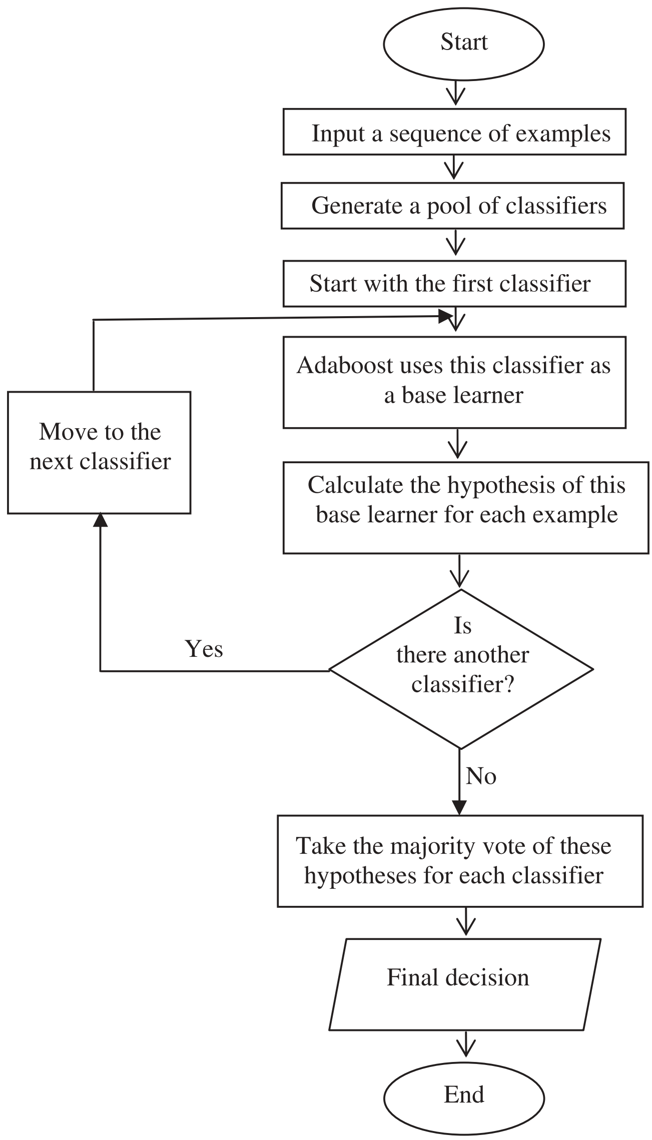

Figure 7. The flow chart of boosted parameter-based combined MLP classifier (BPBMLP).

Boosted-Parameter based MLP (BPBMLP) Ensemble Algorithm:
Input: Multidimensional data set *P* of *M* patterns with *N* features each
Output: Classified textures

BPMLP Algorithm:

for n-epochs = $\tau_{start} : \tau_{step} : \tau_{end}$
  for n-hidden = $\alpha_{start} : \alpha_{step} : \alpha_{end}$
    1- Generate a base PBMLP
    2- train the PBMLP using Adaboost (Algorithm 2 of Fig. 9)
    3- test the PBMLP on the testing set
    4- save the classes of the *N* test patterns
  end;
end;
Apply the majority voting to test results to decide the final classes

Figure 8. Algorithm 1: BPBMLP.

an ensemble is the most suitable approach for tackling the rotation-invariant texture-classification problem. A robust ensemble should encompass all possible diversity sources. In this article, diverse base classifiers have made use of diversity through data variation through the boosting algorithm and the parameter-based diversity using MLPs with different parameter sets.

This system combines the advantages of both the parameter-based diversity enforcement and the robustness of Adaboost.M1 (Freund and Schapire 1997). A parameter-based combined classifier gives its final decision by applying majority voting on the outputs of a pool of individual classifiers that are different in both their architecture and training parameters.

A pool of 16 MLP base classifiers has been generated with the following parameters: number of hidden neurons starts from 165 and is incremented by 5 until it reaches a maximum of 180. The number of training epochs starts with 1700 and is incremented with 100 until it reaches a maximum of 2000. Number of output neurons = Number of texture classes are 13 classes.

Computational complexity of an ensemble of MLP classifiers is higher than that of a single MLP classifier. The training complexity is not an issue because training is done once and offline. Computation complexity of the testing phase for an ensemble of *n* base classifiers is determined by the time of calculating the outputs of *n* MLP classifiers plus the time of calculating the collective decision using the majority vote. The parameter-based variation selection of the proposed ensemble could be implemented on a parallel system.

The complexity of the proposed combined classifier during the testing phase depends on the number of hidden neurons, the size of the input layer, and the time needed for computing the majority vote. The complexity of MLP classifier ensembles depends on the difficulties of tuning parameters of the MLP base classifiers. In Terry (2006), a measure is described that is capable of predicting the number of classifier training epochs for achieving optimal performance in

Input: Sequence of m examples $\langle (x_1, y_1),\ldots,(x_m, y_m)\rangle$ with labels $y_i \in Y = \{1,\ldots,k\}$

Initialize $D_1(i) = 1/m$ for all i.

Do while $t \le T$

1. Call base classifier (MLP), providing it with the distribution D_t.
2. Get back a hypothesis $h_t(x) = \arg\max_{y \in Y} MLP(x)$, where $MLP(x)$ is the response of the current MLP used.
3. Calculate the error of $h_t : \varepsilon_t = \sum_{i:h_t(x_i) \ne y_i} D_t(i)$.

If $\varepsilon_t > 1/2$, then set T=t-1 and abort loop.

4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.
5. Update distribution D_t

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} x \begin{cases} \beta_t & \text{if } h_t(x_i) = y_i \text{ (i.e., the texture, } x_i \text{, classified correctly)} \\ 1 & \text{otherwise (i.e., the texture, } x_i \text{, missclassified. Consequenly, } x_i \text{ assigned larger weight)} \end{cases}$$

where Z_t is a normalization constant (chosen so that D_{t+1} will be a distribution).

Save the hypothesis

$$h_j(x) = \arg\max_{y \in Y} MLP(x) = \arg\max_{y \in Y} \sum_{t:h_t(x)=y} Log \frac{1}{\beta_t}, \qquad j=1,2,\ldots,n.$$

End;

Output: the final decision by taking the majority vote of all hypotheses $h_j(x)$ for each class (texture).

Figure 9. Algorithm 2: Training the PBMLP using AdaBoost.

an ensemble of MLP classifiers. In addition, changing the number of hidden nodes has a major impact on the base classifier. Prediction of the optimal parameter set for building the base classifiers is a topic of future work (Terry 2006). In this work, selection of the most diverse set of base learners is done using Hamming distance contributes to reduce computational complexity. The processing time T for testing a new pattern is proportional to the number of hidden neurons (H), the pattern size (S), and the number of base classifiers:

$$\mathrm{T_{Ensemble}} = \mathrm{n.T_{MLP}} = \mathrm{O(n.H.S)} \tag{5}$$

Experiments on rotated texture database

We demonstrate the performance of the proposed technique on a standard rotated textures database available in the WWW USC-SIPI Image Database (The USC-SIPI Image Database 1977). The image database comprises 13 textures from the Brodatz album (Brodatz 1966) shown in Figure 10. The database contains 91 gray-scale images, including 13 distinct textures, each with seven images that vary in rotation angle (0°, 30°, 60°, 90°, 120°, 150°, and 200°).

Using this database, we perform four experiments. For each experiment, three rotated images of each texture are used to construct the training set, and the remaining four rotated images of the same texture are used for testing. Figure 11 shows a sample texture with its rotated version.

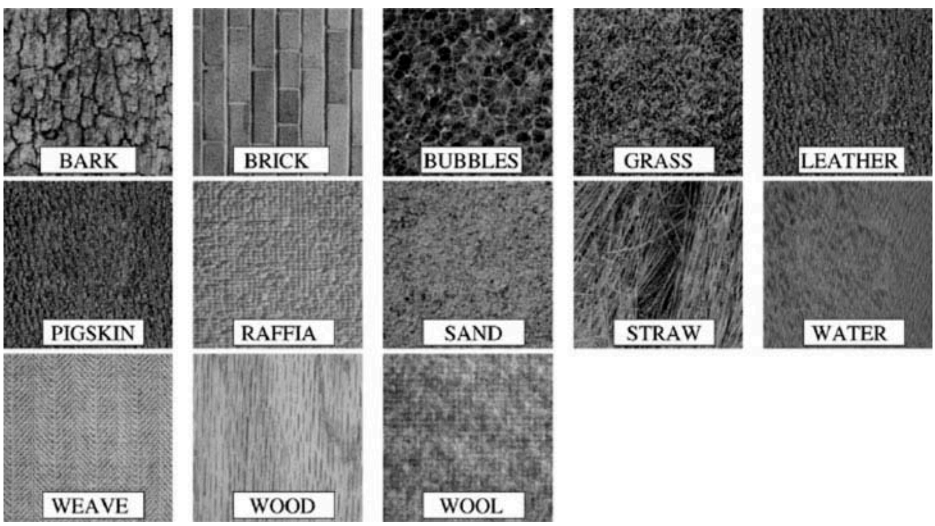

Figure 10. Texture images. Each texture was digitized at seven angles: 0°, 30°, 60°, 90°, 120°, 150°, and 200°.

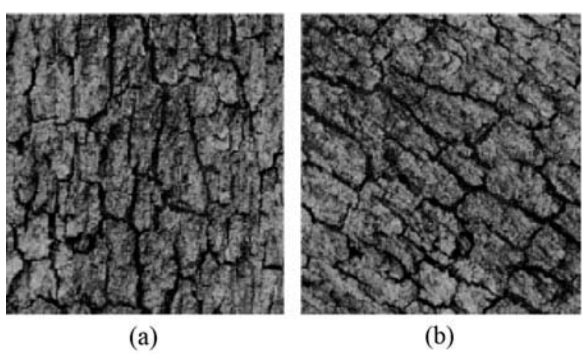

Figure 11. Sample texture image: (a) original bark image and (b) rotated bark image.

Table 3. Percentages of correct classification (PCC) of rotated textures database.

Method	PCC (%)
Best individual classifier	38.6
Committee MLPs	66.7
Boosted MLP (after 4 rounds)	56.2
The proposed method (after 2 rounds)	95
The proposed method (after 3 rounds)	97.44
The proposed method (after 4 rounds)	100

In the first experiment, we train a pool of MLP classifiers and get the performance of the best one. The best recognition rate in this case is 38.5%. In the second experiment, the majority voting rule is used to get the final decision of a committee of the base MLP classifiers, which have been trained on the same dataset at different parameters; the percentage of correct classification (PCC) is 66.7%. In the third experiment, the boosting algorithm is applied to the MLP as base classifier; the accuracy of the ensemble in this case is 56.2% at four rounds. Finally, we applied the proposed BPBMLP combined classifier, which resulted in 97.44% and 100% accuracies at three and four rounds. This result shows that the proposed method compares favorably with the state-of-the-art systems reported in the literature.

Moreover, Table 3 shows that the proposed method at only two rounds is more efficient than the best individual classifier, committee MLPs, and boosted MLP. Also, it is clear from Figures 12 and 13 that the proposed method reaches the accuracy 100% at four rounds and remains stable with increasing number of rounds, whereas the boosted MLP reaches the same

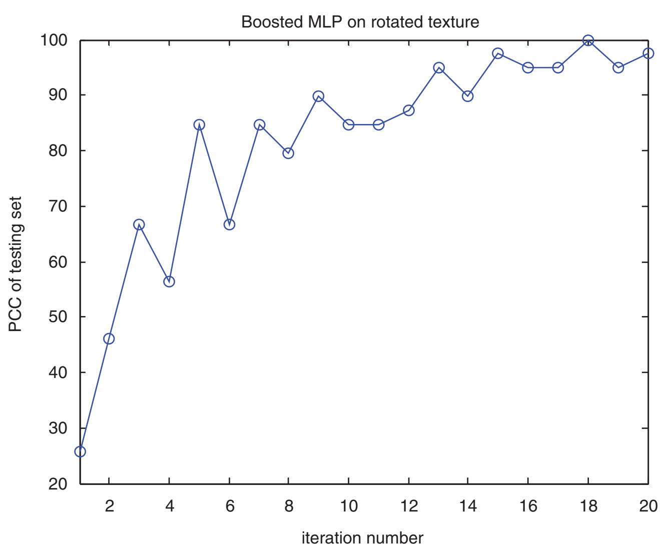

Figure 12. Accuracy of the Boosted MLP on rotated texture.

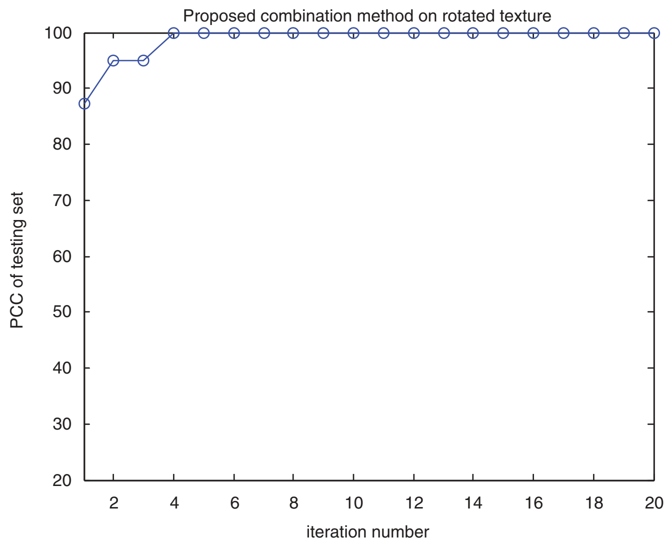

Figure 13. Accuracy of the boosted parameter-based combination method.

accuracy at 18 rounds and begins to decrease, which indicates that the performance of the proposed method is more efficient and stable than boosted MLP.

For the sake of comparison, we present the results of the well-known methods, which have used the same dataset in this study. Table 4 and Figure 14 summarize the results obtained in the literature. It is clear that the proposed combination method results in superior performance compared to the other methods. The major advantage of the proposed method is the stability of its performance.

Table 4. Comparison between the proposed method and other well-known methods on the same dataset used in this study.

Reference	Method	Percentage of Correct Classification
(Haley and Manjunath 1999)	Complete space–frequency model.	96.8%
(Pietikäinen, Ojala, and Xu 2000)	LBP,	91.54%
	Combination of LBP and VAR	94.89%
(Lahajnar and Kovačič 2003)	Gabor functions and multidimensional histograms.	99.42% A major weakness is that the design and selection of the Gabor filters has a greater impact on the classifier performance.
(Ojala, Pietikainen, and Maenpaa 2002)	LBP	98.4%
proposed method	BPBMLP	100% for 4 boosting rounds

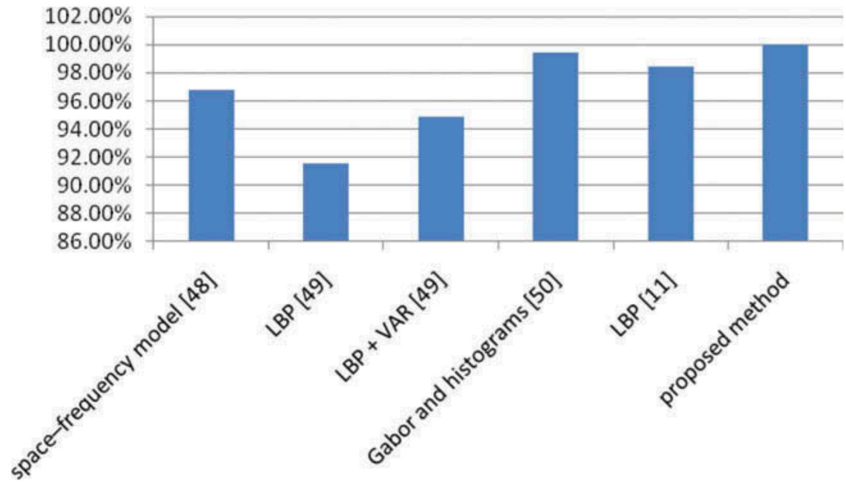

Figure 14. Comparison between the proposed method and other methods on the same dataset.

Conclusion

Rotation-invariant texture classification is a challenging problem for any base classifier. This article presents a robust combined classifier for invariant-texture classification. The proposed ensemble method is based on a hybrid combination of both the parameters variation of the architecture design of the neural network and the AdaBoost. Diversity in base classifier performance has been enforced at two successive levels. At the first level, diversity has been enforced through learning and architectural parameter variation. At the second level, the boosting algorithm has been used to make perturbation of the training set used by the multilayer perceptron (MLP) as base classifier. The final decision is based on the majority voting. Four experiments have been conducted on a standard benchmark rotated textures database and showed that the proposed combined classifier compares favorably with the other systems reported in the literature, and its overall performance is 100%. Moreover, the performance of the proposed combination method is more stable than that of the boosted MLP. As a future work, we will study the theoretical analysis behind this robust hybrid combination method and apply it to other challenging pattern recognition problems.

Funding

S.K. Pal acknowledges DST J.C. Bose Fellowship and DAE Raja Ramanna Fellowship of the Govt. of India. The research for this article was done during A.H. El-Baz postdoctoral fellowship, funded by Egyptian Ministry of Higher Education, at Center for Soft Computing Research, Indian Statistical Institute, Calcutta, India.

References

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24 (2):123–140. doi:10.1007/BF00058655.

Brodatz, P. 1966. *Textures: A photographic album for artists and designers*. New York, NY: Dover Publications.

Campisi, P., A. Neri, G. Panci, and G. Scarano. 2004. Robust rotation-invariant texture classification using a model based approach. *IEEE Transactions on Image Processing* 13 (6):782–791. doi:10.1109/TIP.2003.822607.

Deng, H., and D. A. Clausi. 2004. Gaussian MRF rotation-invariant features for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (7):951–955. doi:10.1109/TPAMI.2004.30.

Eibl, G., and K. Pfeiffer. 2005. Multiclass boosting for weak classifiers. *Journal of Machine Learning Research* 6:189–210.

El-Baz, A. H. 2015. Hybrid intelligent system-based rough set and ensemble classifier for breast cancer diagnosis. *Neural Computing and Applications* 26:437–446. doi:10.1007/s00521-014-1731-9.

El-Baz, A. H., and A. S. Tolba. 2013. An efficient algorithm for 3D hand gesture recognition using combined neural classifiers. *Neural Computing and Applications* 22 (7–8):1477–1484. doi:10.1007/s00521-012-0844-2.

Etemad, S. A., and A. Arya. 2014. Classification and translation of style and affect in human motion using RBF neural networks. *Neurocomputing* 129:585–595. doi:10.1016/j.neucom.2013.09.001.

Freund, Y., and R. E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal Computation System Sciences* 55 (1):119–139. doi:10.1006/jcss.1997.1504.

Fumera, G., F. Roli, and A. Serrau. 2008. A theoretical analysis of bagging as a linear combination of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30:1293–1299. doi:10.1109/TPAMI.2008.30.

Geman, S., E. Bienenstock, and R. Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4:1–58. doi:10.1162/neco.1992.4.1.1.

Guo, Z., L. Zhang, and D. Zhang. 2010. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern Recognition* 43:706–719. doi:10.1016/j.patcog.2009.08.017.

Haley, G. M., and B. S. Manjunath. 1999. Rotation-invariant texture classification using a complete space-frequency model. *IEEE Transactions Image Processing* 8 (2):255–269. doi:10.1109/83.743859.

Haralick, R. M., K. Shanmugam, and I. Dinstein. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* 3 (6):610–621. doi:10.1109/TSMC.1973.4309314.

Haykin, S. 1999. *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Hodge, V. J., and J. A. Austin. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22 (2):85–126. doi:10.1023/B:AIRE.0000045502.10941.a9.

Kang, S., and S. Park. 2009. A fusion neural network classifier for image classification. *Pattern Recognition Letters* 30 (9):789–793. doi:10.1016/j.patrec.2008.06.009.

Kashyap, R. L., and A. Khotanzad. 1986. A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (4):472–481. doi:10.1109/TPAMI.1986.4767811.

Kittler, G., M. Hatef, R. P. W. Duin, and J. Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20:226–239. doi:10.1109/34.667881.

Kuncheva, L. I. 2014. *Combining pattern classifiers: methods and algorithms* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Kwang, K., J. Keechul, H. P. Se, and J. K. Hang. 2002. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (11):1542–1550. doi:10.1109/TPAMI.2002.1046177.

Lahajnar, F., and S. Kovačič. 2003. Rotation-invariant texture classification. *Pattern Recognition Letters* 24:1151–1161. doi:10.1016/S0167-8655(02)00285-4.

Laine, A., and J. Fan. 1993. Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (11):1186–1191. doi:10.1109/34.244679.

Lazebnik, S., C. Schmid, and J. Ponce. 2005. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (8):1265–1278. doi:10.1109/TPAMI.2005.151.

Li, C., J. Li, D. Gao, and B. Fu. 2014. Rapid-transform based rotation invariant descriptor for texture classification under non-ideal conditions. *Pattern Recognition* 47:313–325. doi:10.1016/j.patcog.2013.05.003.

Li., S., J. T. Kwok, H. Zhu, and Y. Wang. 2003. Texture classification using the support vector machines. *Pattern Recognition* 36:2883–2893. doi:10.1016/S0031-3203(03)00219-X.

Lotte, F., M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. 2007. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal Neural Engineering* 4 (2):R1–13. doi:10.1088/1741-2560/4/2/R01.

Lu, X., Y. Wang, and A. K. Jain. 2003. Combining classifiers for face recognition. *Proceedings of the International Conference on Multimedia (ICME'03)* 3:13–16.

Manjunath, B. S., and W. Y. Ma. 1996. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8):837–842. doi:10.1109/34.531803.

Mao, J., and A. K. Jain. 1992. Texture classification and segmentation using multi-resolution simultaneous autoregressive models. *Pattern Recognition* 25 (2):173–188. doi:10.1016/0031-3203(92)90099-5.

Mellor, M., B. Hong, and M. Brady. 2008. Locally rotation, contrast, and scale invariant descriptors for texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (1):52–61. doi:10.1109/TPAMI.2007.1161.

Meynet, J., and J.-P. Thiran. 2010. Information theoretic combination of pattern classifiers. *Pattern Recognition* 43 (10):3412–3421. doi:10.1016/j.patcog.2010.04.013.

Moglu, F. A., and E. Alpaydin. 2001. Combining multiple representations for pen-based handwritten digit recognition. *Turkish Journal of Electronic Engineering* 9 (1):1–12.

Ojala, T., M. Pietikainen, and T. Maenpaa. 2002. Multi-resolution, gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7):971–987. doi:10.1109/TPAMI.2002.1017623.

Perrone, M. 1993. Improving regression estimation: averaging methods for variance reduction with extension to general convex measure optimization. PhD thesis. Brown University, Providence, RI.

Pietikäinen, M., T. Ojala, and Z. Xu. 2000. Rotation-invariant texture classification using feature distributions. *Pattern Recognition* 33:43–52. doi:10.1016/S0031-3203(99)00032-1.

Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvardo. 2005. Ensemble of SVMs for improving brain computer interface (p300 speller performances). International Conference on Artificial Neural Networks, Warsaw, Poland, September 11–15.

Randen, T., and J. H. Husoy. 1999. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21 (4):291–310. doi:10.1109/34.761261.

Reidmiller, M., and H. Braun. 1993. A direct adaptive method for faster backpropagation learning: The RPRO algorithm (1:586–591), Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, March 28–April 1. doi:10.1109/ICNN.1993.298623.

Saha, S., C. A. Murthy and S.K. Pal. 2007. Rough set based ensemble classifier for web page classification. *Fundamenta Informaticae* 76 (1–2):171–187.

Saha, S., C. A. Murthy and S.K. Pal. 2010. Application of rough ensemble classifier to web services categorization and focused crawling. *Web Intelligence and Agent Systems* 8 (2):181–202.

Schapire, R. E. 1990. The strength of weak learnability. *Machine Learning* 5:197–227. doi:10.1007/BF00116037.

Schulz, J. 2008. Introduction to data similarity. http://www.code10.info/index.php (accessed June 2014).

Sengur, I. Turkoglu, M. C. Ince, and A. Sengur. 2007. Wavelet packet neural networks for texture classification. *Expert Systems with Applications* 32:527–533. doi:10.1016/j.eswa.2005.12.013.

Shutao, L., T. K. James, Z. Hailong, and W. Yaonan. 2003. Texture classification using the support vector machines. *Pattern Recognition* 36 (12):2883–2893. doi:10.1016/S0031-3203(03)00219-X.

Terry, W. 2006. Accuracy/diversity and ensemble mlp classifier design. *IEEE Transactions on Neural Networks* 17 (5):1194–1211. doi:10.1109/TNN.2006.875979.

The USC-SIPI Image Database. 1977. Volume 1: Rotated Textures. Signal and Image Processing Institute of the University of Southern California, Los Angeles. http://sipi.usc.edu/services/database/data-base.cgi?volume=rotate. (accessed February 2014).

Tolba, A. S., and A. N. Abu-Rezq. 2000. Combined classifiers for invariant face recognition. *Pattern Analysis Applications* 3 (4):289–302. doi:10.1007/s100440070001.

Unser, M. 1995. Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing* 4 (11):1549–1560. doi:10.1109/83.469936.

Varma, M., and A. Zisserman. 2005. A statistical approach to texture classification from single images. *International Journal of Computer Vision* 62 (1–2):61–81. doi:10.1007/s11263-005-4635-4.

Wu, W.-R., and S.-C. Wei. 1996. Rotation and gray-scale transform-invariant texture classification using spiral resampling, subband decomposition, and hidden Markov model. *IEEE Transactions on Image Processing* 5 (10):1423–1434. doi:10.1109/83.536891.

Xu, Y., H. Ji, and C. Fermuller. 2006. A projective invariant for texture. Paper presented at the International Conference on Computer Vision and Pattern Recognition, New York, NY, June 17–22, pp. 1932–1939.