2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Cost-sensitive Texture Classification

Gerald Schaefer*, Bartosz Krawczyk†, Niraj P. Doshi* and Tomoharu Nakashima‡
* Department of Computer Science, Loughborough University, Loughborough, U.K.
† Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
‡ Department of Computer Science and Intelligent Systems, Osaka Prefecture University, Japan

***Abstract*—Texture recognition plays an important role in many computer vision tasks including segmentation, scene understanding and interpretation, medical imaging and object recognition. In some situations, the correct identification of particular textures is more important compared to others, for example recognition of enemy uniforms for automatic defense systems, or isolation of textures related to tumors in medical images. Such cost-sensitive texture classification is the focus of this paper, which we address by reformulating the classification problem as a cost minimisation problem. We do this by constructing a cost-sensitive classifier ensemble that is tuned using a genetic algorithm. Based on experimental results obtained on several Outex datasets with cost definitions, we show our approach to work well in comparison with canonical classification methods and the ensemble approach to lead to better performance compared to single predictors.**

I. Introduction

Texture recognition and classification play an important role in many computer vision tasks. Examples include image segmentation, content-based image retrieval, scene understanding, medical imaging, defect detection and object recognition. While in many applications no particular distinction is made in terms of importance or preference of certain textures, this is not the case in all situations. For example, for a military application the correct recognition of enemy uniforms or vehicles will be of higher relevance compared to identifying building structures or other background areas. Similarly, in medical diagnosis isolation of textures associated with tumors would be of higher priority compared to other image areas.

In this paper, we address this issue and perform cost-sensitive texture classification by reformulating the classification problem as a cost minimisation task. Following this approach, a classification system is constructed so as to lead to minimal misclassification cost rather than maximal classification accuracy, and hence able to emphasise certain textures by associating higher costs with them. In particular, we employ local binary pattern (LBP) [1] based texture descriptors which have been shown to work well for texture classification. As descriptors, we employ LBP, uniform LBP, and Compound LBP [2], which in [3] were identified as the best performing LBP features for basic texture classification.

For verification, we perform extensive experiments on the TC03, TC04 and TC05 test suites of the Outex texture benchmark database [4], which come with pre-defined costs for different texture classes. To our best knowledge, we are the first to perform dedicated cost-aware classification on these datasets. Our results show, that by taking into account the cost definitions, improved classification performance, in terms of reducing misclassification costs, is possible, and thus set a benchmark for these tested datsets.

II. Local binary pattern texture descriptors

Local binary patterns (LBP) have been shown to be a simple yet effective texture analysis technique, and have been employed in a variety of computer vision applications.

A. LBP

The original LBP [5] assigns, on a pixel basis, descriptors that describe the neighbourhood of that pixel and then forms a histogram of those descriptors. In detail, let

$$B = \begin{pmatrix} g_{(-1,-1)} & g_{(-1,0)} & g_{(-1,1)} \\ g_{(0,-1)} & g_{(0,0)} & g_{(0,1)} \\ g_{(1,-1)} & g_{(1,0)} & g_{(1,1)} \end{pmatrix} \quad (1)$$

describe the 3×3 grayscale block of a pixel at location (0,0) and its 8-neighbourhood. The first step is to subtract the value of the central pixel and consider only the resulting values at the neighbouring locations

$$LBP_1 = \begin{pmatrix} g_{(-1,-1)} - \hat{g} & g_{(-1,0)} - \hat{g} & g_{(-1,1)} - \hat{g} \\ g_{(0,-1)} - \hat{g} & & g_{(0,1)} - \hat{g} \\ g_{(1,-1)} - \hat{g} & g_{(1,0)} - \hat{g} & g_{(1,1)} - \hat{g} \end{pmatrix}, \quad (2)$$

where $\hat{g} = g_{(0,0)}$ for convenience. Next, an operator

$$s(x) = \begin{cases} 1 & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \quad (3)$$

is assigned at each location resulting in

$$LBP_2 = \begin{pmatrix} s(g_{(-1,-1)} - \hat{g}) & s(g_{(-1,0)} - \hat{g}) & s(g_{(-1,1)} - \hat{g}) \\ s(g_{(0,-1)} - \hat{g}) & & s(g_{(0,1)} - \hat{g}) \\ s(g_{(1,-1)} - \hat{g}) & s(g_{(1,0)} - \hat{g}) & s(g_{(1,1)} - \hat{g}) \end{pmatrix}. \quad (4)$$

Thus, each pixel of the 8-neighbourhood is encoded as either 0 or 1, i.e. in a binary way. The resulting 8 bit sequence then describes the texture property of the pixel at location (0,0).

B. Basic LBP variants

In the above procedure, the 8-neighbourhood of each pixel is utilised. Clearly, four of these neighbours are at a different distance ($\sqrt{2}$) than the other four. A circular neighbourhood can be defined on which all neighbours are equidistant from the centre pixel [1]; locations that do not fall exactly at the centre of a pixel are obtained through interpolation.

Uniform LBP patterns [1] are patterns that have either no change or two changes between 0s and 1s, that is patterns that have at most one "run" of 0s and one of 1s. For eight neighbours, there are nine rotation invariant uniform LBP codes, two without any 0-1 changes (i.e., one with all 0s and one with all 1s) and the remaining seven with $\{1, \ldots, 7\}$ 1s in sequence. It has been shown [1], that focussing on these uniform patterns while aggregating all other (i.e., non-uniform) patterns into a single group leads to improved texture descriptors.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

C. Compound LBP

The original LBP operator discards the magnitude information of difference between the centre and neighbouring grey values. In Compound LBP [2], a 2-bit code is used to encode the local texture property of an image. The first bit represents the sign of difference between the centre and neighbouring grey values, while the second bit is used to encode the magnitude of difference of neighbour pixels with respect to a threshold value M_{avg}. M_{avg} is set to the average magnitude of the difference between the centre and the neighbour grey values in the local neighbourhood. If g_p is a neighbouring pixel and g_c the centre pixel, the 2-bit code $s(x)$ is obtained as

$$b(i_p, i_c) = \begin{cases} 00 \text{ if } g_p - g_c < 0 \text{ and } |g_p - g_c| \leq M_{avg} \\ 01 \text{ if } g_p - g_c < 0 \text{ and } |g_p - g_c| > M_{avg} \\ 10 \text{ if } g_p - g_c \geq 0 \text{ and } |g_p - g_c| \leq M_{avg} \\ 11 \text{ otherwise} \end{cases}, \tag{5}$$

generating a 16-bit code for 8 neighbouring pixels. This 16-bit code is then divided into two 8-bit codes and two histograms are generated from the two codes.

D. LBP image descriptors

Applying LBP operators on the whole image gives texture descriptors at each image location. A histogram of these descriptors is then built to describe the texture characteristics of the image. These histograms then form the basis of a subsequent classification stage and are hence the features that are fed to the classifier.

III. Cost-sensitive classification

A pattern recognition algorithm Ψ maps the feature space $\mathcal{X}$ to the set of class labels $\mathcal{M}$. This is typically established on the basis of examples from a training set. The training set consists of learning examples, i.e. observations of features together with their correct classifications.

Multiple classifier systems (MCSs) can improve the performance of the best base classifier, since they can exploit the strengths and eliminate the weaknesses of the individual classifiers [6]. Let's assume that we have n classifiers $\Psi^{(1)},\ \Psi^{(2)},\ ...,\ \Psi^{(n)}$. For a given object $x \in \mathcal{X}$, each of them makes a decision regarding class $i \in \mathcal{M} = \{1,\ ..., M\}$. The combined classifier $\bar{\Psi}$ makes a decision based on

$$\bar{\Psi}\left(\Psi^{(1)}(x),\ \Psi^{(2)}(x),\ ...,\ \Psi^{(n)}(x)\right) =$$

$$\arg\max_{j \in M} \sum_{l=1}^{n} \delta\left(j,\ \Psi^{(l)}(x)\right) w^{(l)} \Psi^{(l)}(x), \tag{6}$$

where

$$\delta(j, i) = \begin{cases} 0 & \text{if } \ i \neq j \\ 1 & \text{if } \ i = j \end{cases}, \tag{7}$$

and $w^{(l)}$ is the weight assigned to the l-th classifier. Clearly, the weights play a key role in establishing the quality of $\bar{\Psi}$.

In this paper, the task we are addressing is how to select individual classifiers of an ensemble with respect to misclassification cost so as to enable cost-sensitive texture classification. Our aim is to create an ensemble with minimal classification error P within the cost bounds of a cost matrix C, where C defines the misclassification costs for all texture classes.

As base classifier, we utilise a cost-sensitive classification tree that has its roots in the idea of the EG2 algorithm [7]. EG2 uses the information cost function (ICF) to select an attribute. For each attribute, the ICF is calculated. The attribute with the highest ICF value is then used to partition the data at a certain level of the tree. This is based on the misclassification cost rate proposed in [8]. To have a representative pool of classifiers we need to create a set of them. To do this, we use a random subspace approach [9], which randomly divides the feature space into several subspaces and trains individual classifiers on each of them. This ensures that the pool is diverse and contains heterogeneous rather than homogenous classifiers.

For selecting and combining individual classifiers for the ensemble, we employ a genetic algorithm (GA) [10]. An individual in the GA population represents a classifier ensemble $Ch = [W]$ where component W represents the weights assigned to each of the base classifiers in a form $W = [W_1, W_2, ..., W_L]$ and is a real-valued vector with values in [0;1]. When a classifier is not selected in a particular

Algorithm 1 Cost-sensitive ensemble algorithm.

Input:
U → set of classifiers

Output:
Q → ensemble
W → set of weights assigned to classifiers
P → ensemble error
B → best solution

$P = 1.0$
$B =$ empty

Create initial population
Select individuals for evaluation
for all selected individuals **do**
  Evaluate individual's fitness with weights from **W**
  if fitness $< P$ **then**
    replace P
    replace B
  end if
end for

while termination conditions not satisfied **do**
  Select pairs for crossover from best-ranked individuals
  Apply crossover operator
  Apply mutation operator
  Select new individuals
  for all selected individuals **do**
    Evaluate new individual's fitness with weights from **W**
    if fitness $< P$ **then**
      replace P
      replace B
    end if
  end for
  Create new population
end while

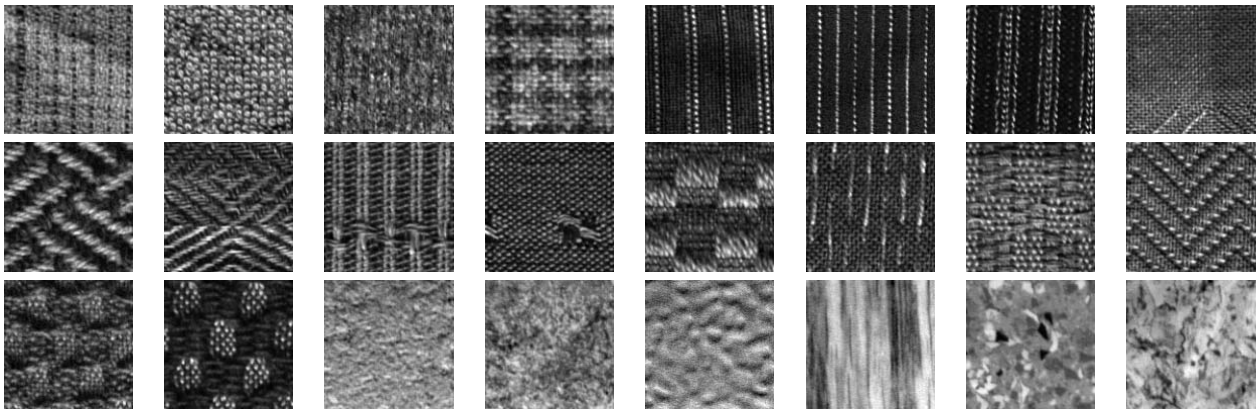

Fig. 1. Sample texture images of each of the classes of the Outex test suite.

individual, its weight is set to 0.

The GA, which is outlined in Algorithm 1, proceeds in the following stages:

- *Population generation:* The initial population is generated randomly.
- *Population assessment:* For each member of the population, a value of the fitness function is calculated.
- *Choosing elite members:* Members with the highest fitness values are taken from the population and are carried over to the descendant population without mutation, crossover or selection.
- *Mutation:* The mutation operator changes a selected (one at a time) member of the population by applying some random changes to its chromosome. The chromosome is altered with a probability that is changed during the optimisation progress. In the early phase of the optimisation, a special emphasis is put on searching for possibly best areas of weight values, while later on attention is shifted to exploring the most promising area for optimal settings. Mutation involves adding a vector of numbers randomly generated according to a normal density distribution (with mean of 0 and standard deviation of Δ_m).
- *Crossover:* The crossover operator generates one offspring member from two parents. Offsprings are obtained according to the two-point rule.
- *Formation of new population:* A selection of individuals from the population is formed by merging the descendant population and a set of individuals created by mutation and crossover. The probability of selection P_s of a particular individual is proportional to the value of its fitness. A tournament selection scheme is employed.

The ensemble misclassification cost, calculated on the training set, serves as fitness function, and we thus reformulate the classification problem into a cost minimisation problem. Termination conditions can in principle be adjusted; we use the number of iterations without result improvement.

IV. Experimental results

For our experiments, we used the TC03, TC04, and TC05 databases of the Outex texture benchmark suite [4]. Each database comprises 24 texture classes captured under the same conditions and without rotation, and 20 samples are provided for each class. From Figure 1, which shows one sample image for each of the 24 classes, it can be seen that the dataset is not simple as several texture classes are rather similar. A classifier is trained on half of the dataset (i.e., on 240 images) and the remaining half is used for testing. For each database, 100 database splits are defined, while the image resolution is different between the three different datasets.

What makes the selected three benchmark databases distinct from other texture classification datasets, is the fact that, for each fold, costs for each of the texture classes are specified, so that we can consequently take this cost information into account during the classification task. To the best of our knowledge, ours is the first work that explicitly uses the cost information available in Outex TC03, TC04, and TC05.

In our experiments, we employed three types of LBP features, standard (circular) LBP calculated at 3 radii (1,3,5), uniform LBP calculated at the same radii, and Compound LBP, which in the study of [3] were found to be the most effective LBP features for basic texture classification.

For our ensemble, we trained a total of 50 cost-sensitive decision tree classifiers using the random subspace approach, consisting of 40% of the original feature space. The parameters used for the weight optimisation were set as follows: N_c (the upper limit of algorithm cycles) to 1000, N_p (the population quantity) to 50, β (the mutation probability) to 0.7, γ (the crossover probability) to 0.3, Δ_m (the mutation range factor) to 0.2, and V (the upper limit of algorithm iterations without quality improvement) to 20.

To put our results into context, we also performed classification using a single cost-sensitive decision tree, and a single C4.5 decision tree classifier [11]. We further implemented an ensemble of C4.5 decision trees that is created in the same fashion as our cost-sensitive ensemble, i.e. by optimising the weights through a genetic algorithm but with classification accuracy as fitness criterion.

Classification results are given, in terms of classification accuracy and misclassification cost as averages over 10 runs,

TABLE I. CLASSIFICATION RESULTS FOR THE TC03 DATASET. FOR EACH METHOD RESULTS ARE GIVEN IN TERMS OF CLASSIFICATION ACCURACY AND MISCLASSIFICATION COSTS. EVERY SECOND LINE INDICATES THE METHODS COMPARED TO WHICH THE EXAMINED ONE WAS FOUND TO BE STATISTICALLY SIGNIFICANTLY BETTER.

		$\text{LBP}_{R=\{1,3,5\}}$	$\text{LBP}^{u2}_{R=\{1,3,5\}}$	CLBP
C4.5	accuracy	85.32	86.11	88.29
		-	-	-
	cost	90.60	103.20	85.09
		-	-	-
CSTree	accuracy	93.25	92.91	94.05
		C4.5,C4.5Ens	C4.5,C4.5Ens	C4.5,C4.5Ens
	cost	48.34	58.29	44.35
		C4.5,C4.5Ens	C4.5,C4.5Ens	C4.5,C4.5Ens
C4.5 Ensemble	accuracy	90.23	91.28	92.61
		C4.5	C4.5	C4.5
	cost	60.11	65.25	59.04
		C4.5	C4.5	C4.5
Cost-sensitive Ensemble	accuracy	99.41	99.29	99.97
		all other methods	all other methods	all other methods
	cost	21.96	24.50	20.02
		all other methods	all other methods	all other methods

in Table I for TC03, Table II for TC04, and Table III for TC05. We also carried out a statistical test, namely an F-test with a significance level of 0.05, to judge which methods work statistically better than others, and present the results in the same tables.

Looking at the results for the TC03 dataset in Table I, we can first of all notice that the standard C4.5 classifiers did not do very well, neither in terms of classification accuracy nor in terms of misclassification costs. Application of a cost-sensitive decision tree (CSTree) for classification immediately leads to a significant drop of misclassification cost, while also leading to improved classification accuracy. The results of the C4.5 ensemble demonstrate that a multiple classifier system can yield significantly better classification performance. At the same time, we can notice that nevertheless even a single CSTree classifier leads to lower misclassification costs than the C4.5 ensemble. Finally, inspecting the results of our proposed cost-sensitive classifier ensemble, it is apparent that it clearly gives the best results for all feature types, both in terms of costs and accuracy while the results of the F-test indicate that our approach is statistically superior to all other tested methods.

Turning to the results for TC04 (Table II) and TC05 (Table III), we see that they are similar to those obtained for TC03. While in general the performance is lower, indicating that these datasets are somewhat more challenging, for all features and both datasets, our cost-sensitive ensemble is shown to give the best results, both for misclassification costs and classification accuracy, and to statistically outperform all other methods. This impressively demonstrates the efficacy of our proposed technique and the usefulness of cost-sensitive methods for texture classification.

TABLE II. CLASSIFICATION RESULTS FOR THE TC04 DATASET, LAID OUT IN THE SAME FASHION AS TABLE I.

		$\text{LBP}_{R=\{1,3,5\}}$	$\text{LBP}^{u2}_{R=\{1,3,5\}}$	CLBP
C4.5	accuracy	87.21	88.03	89.61
		-	-	-
	cost	120.89	108.00	99.75
		-	-	-
CSTree	accuracy	91.24	90.75	95.06
		C4.5	C4.5	C4.5,C4.5Ens
	cost	57.50	60.25	39.50
		C4.5,C4.5Ens	C4.5,C4.5Ens	C4.5,C4.5Ens
C4.5 Ensemble	accuracy	94.28	94.42	94.11
		C4.5,CSTree	C4.5,CSTree	C4.5
	cost	89.75	87.55	54.90
		C4.5	C4.5	C4.5
Cost-sensitive Ensemble	accuracy	98.60	98.72	99.02
		all other methods	all other methods	all other methods
	cost	32.60	33.50	27.80
		all other methods	all other methods	all other methods

TABLE III. CLASSIFICATION RESULTS FOR THE TC05 DATASET, LAID OUT IN THE SAME FASHION AS TABLE I.

		$\text{LBP}_{R=\{1,3,5\}}$	$\text{LBP}^{u2}_{R=\{1,3,5\}}$	CLBP
C4.5	accuracy	80.24	80.97	82.64
		-	-	-
	cost	185.90	174.50	159.80
		-	-	-
CSTree	accuracy	88.24	87.86	90.66
		C4.5	C4.5	C4.5
	cost	68.45	70.00	53.75
		C4.5	C4.5	C4.5
C4.5 Ensemble	accuracy	92.63	93.12	94.57
		C4.5,CSTree	C4.5,CSTree	C4.5,CSTree
	cost	66.90	62.50	50.55
		C4.5	C4.5,CSTree	C4.5,CSTree
Cost-sensitive Ensemble	accuracy	95.34	95.02	96.12
		all other methods	all other methods	all other methods
	cost	45.80	47.25	40.30
		all other methods	all other methods	all other methods

V. CONCLUSIONS

In this paper, we have addressed the problem of cost-sensitive texture classification which associates misclassification costs with texture classes to prioritise the correct identification of certain textures. We approached this by constructing an ensemble of cost-sensitive decision tree classifiers where classifier selection and fusion is optimised using a genetic algorithm. Experimental results on three Outex test dataset with defined misclassification costs confirm the efficacy of our proposed approach and show it to statistically outperform both single cost-sensitive classifiers and a canonical decision tree ensemble.

REFERENCES

[1] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[2] F. Ahmed, E. Hossain, A. S. M. H. Bari, and A. Shihavuddin, "Compound local binary pattern (CLBP) for robust facial expression recognition," in *12th IEEE Int. Symposium on Computational Intelligence and Informatics*, 2011, pp. 391–395.

[3] N. P. Doshi and G. Schaefer, "A comparative analysis of local binary pattern texture classification," in *Visual Communications and Image Processing*, 2012.

[4] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, "Outex – new framework for empirical evaluation of texture analysis algorithms." in *16th Int. Conference on Pattern Recognition*, 2002, pp. 1:701–706.

[5] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study for texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

[6] L. I. Kuncheva, *Combining pattern classifiers: Methods and algorithms*. Wiley-Interscience, New Jersey, 2004.

[7] M. Nuenz, "The use of background knowledge in decision tree induction," *Machine Learning*, vol. 6, pp. 231–250, 1991.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Chapman and Hall, 1984.

[9] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 832–844, 1998.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Mitchigan Press, 1975.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.