Texture Classification using Histogram Equalization and the Lempel-Ziv-Welch Algorithm

Leonardo Vidal Batista and Moab Mariz Meira

***Resumo*—Este trabalho apresenta um novo método de classificação de texturas usando equalização de histograma e o algoritmo Lempel-Ziv-Welch (LZW). Após equalização, cada amostra de textura é codificada pelo LZW usando dicionários pré-construídos e atribuída à classe cujos dicionários minimizam a taxa de codificação. Na avaliação experimental, o método atingiu precisão de 100%.**

***Palavras-Chave*—classificação de textura; algoritmo Lempel-Ziv-Welch; reconhecimento de padrões.**

***Abstract*— This paper presents a new texture classification method using histogram equalization and the Lempel-Ziv-Welch (LZW) compression algorithm. After equalization, each texture sample is encoded by LZW using pre-constructed dictionaries and assigned to the class whose dictionaries minimize the coding rate. In the empirical evaluation, the classifier achieved 100% accuracy.**

***Index Terms*— texture classification; Lempel-Ziv-Welch algorithm; pattern recognition.**

I. Introduction

Texture is one of the most fundamental attributes used by the human visual system and computer vision systems for segmentation, classification and interpretation of scenes [1]. There has been a great interest in the development of texture-based pattern recognition methods in many different areas, such as remote sensing [2, 3], image-based medical diagnosis [4], industrial automation [5] and biometric recognition [6, 7].

Texture classification consists in assigning an unknown texture sample **x** to one of N texture classes C_i, $i = 1, 2,\ldots, N$. Discriminating features, or structural or stochastic models, are used to characterize the classes, and classification is performed according to some measure of similarity between **x** and each class.

In supervised classification, texture samples known to belong to class C_i, $i = 1, 2, \ldots N$, are available. These pre-classified samples are used to construct models or to define discriminating features for each class, in a process known as training or learning. In order to develop efficient texture classifiers, it is crucial to discover discriminating attributes or precise models for the texture classes.

Although intuitively recognized by the human visual system, texture is not easy to characterize formally. The problem resides in the intrinsic difficulty to define what is most relevant to characterize texture, as the answer depends on subjective perceptual considerations and on particular applications. Therefore, texture feature extraction and modeling tends to be a difficult and application-driven task. A popular yet rather vague definition states that textures are spatial patterns formed by more or less accurate repetitions of some basic subpatterns. [8].

Selecting inadequate features can degrade classification performance, and selecting too many features, even appropriate ones, has negative consequences, such as increase in processing time and memory requirements, and a potential accuracy degradation due to the statistical phenomenon known as "curse of dimensionality" [9].

Early feature extraction techniques focused on spatial statistics, such as energy features and gray level co-occurrence matrices (GLCM) [10]. More recently, model-based techniques, such as Markov random fields (MRF) [9, 11], and signal processing techniques, such as Gabor and wavelet transforms [12] have been investigated. Extensive comparisons between various feature extraction methods concluded that no method is consistently superior to the others [12, 13].

After defining the features or models that characterize the texture classes, a classification method must be chosen. A large number of classifiers have been proposed in the last decades [9]. A multiple support vector machine (SVM) classifier, denominated fused SVM, with features generated by discrete wavelet frame transform, compared favorably with single SVM classifiers, Bayes classifiers using Bayes distance and Mahalanobi distance, and a learning vector quantization (LVQ) classifier [14].

The development of powerful texture classification methods with low computational complexity is a very useful direction of research [12]. In applications such as industrial web inspection the throughput is enormous and require fast methods and special hardware support [5].

Modern lossless data compression algorithms have been

Leonardo Vidal Batista and Moab Mariz Meira, Departamento de Informática, Universidade Federal da Paraíba, João Pessoa, PB, Brasil, E-mail: leonardo@di.ufpb.br. This work was supported by CNPq, a governmental Brazilian institution dedicated to scientific and technological development.

applied to classification problems, due to their ability to construct accurate statistical models, in some cases with low computational requirements [15]. A solid theoretical foundation for using LZ78 [16] and other dictionary-based compression algorithms [15] for classification is well established [17, 18]. However, despite its sound theoretical basis, the application of compression schemes to classification problems seems controversial.

A potential problem associated with lossless dictionary-based compression algorithms for texture classification is the fact that these methods search exact matches in the dictionary for strings in the message to be compressed. A precise dictionary constructed for a given texture class may present a poor performance when compressing a new sample from the same class, if this new sample presents gray-level deviations caused by digitization noise or illumination changes.

Degraded performance, even when gray-level deviations are subtle, indicates that the constructed model may not be able to adequately describe the new texture sample, and consequently classification accuracy may also degrade.

Two possible solutions to this problem are:

1. Adoption of a lossy dictionary-based compressor [19], less sensitive to small, spurious gray-level deviations;
2. Reduction of these deviations by means of image processing techniques, prior to the use of a lossless dictionary-based compression algorithm.

Histogram equalization is a well-known non-linear operation that generates an approximately uniform distribution of gray-levels over the available range [10]. Typically, discrete histogram equalization tends to map to the same value multiple gray levels that have similar values, thus reducing the small gray level deviations that tends to cause mismatches in the searching stage of lossless dictionary-based compressors. Histogram equalization also decreases the probability that a classifier discriminates texture classes by average gray level or variance, instead of by its textural properties [12]. This allows a more precise evaluation of the capabilities of the method to discriminate texture attributes.

When applied to a specific text categorization task, the *prediction by partial matching* (PPM) lossless compressor [15] correctly categorized the majority of the documents, but did not achieve accuracy competitive with state-of-the-art machine learning schemes [20]. On the other hand, another report concluded that PPM was successfully applied to various text classification problems, with performance comparable to state-of-the-art methods [21]. Some of the main drawbacks of PPM come from the large computational resources required [15].

Classifiers based on universal data compression models have several potential advantages over classical machine learning methods: since there is no feature selection, no information is discarded — the models describe the classes as a whole [20]; no assumptions about the probability distributions of the classes are required; the adaptive model construction capability of compression algorithms offers an uniform way to classify different types of sources [20]; the classification rule is very simple [21].

This paper proposes a new texture classifier based on histogram equalization and Lempel-Ziv-Welch (LZW) lossless compression algorithm [22]. The rest of this paper is organized as follows. Section II presents some fundamental concepts; section III presents the LZW algorithm; section IV describes the proposed classifier; section V presents the empirical evaluation of the proposed classifier; and section VI presents a discussion of the results and the concluding remarks.

II. Entropy and Markov Models

Let S be a stationary discrete information source that generates messages over a finite alphabet $\mathtt{A} = \{a_1, a_2, ..., a_M\}$. The source chooses successive symbols from A according to some probability distribution that, in general, depends on preceding symbols. A generic message is modeled as a stationary stochastic process $\mathbf{x} = \ldots x_{-2}\, x_{-1} x_0 x_1 x_{2\ldots}$, with $x_i \in \mathtt{A}$. Let $\mathbf{x}^n = x_1 x_2 \ldots x_n$ represent a message of length n. Since A has M distinct elements, the source can generate M^n different messages of length n. Let $\mathbf{x}_i^n$, $i = 1, 2, ..., M^n$ denote the i^{th} of these messages, according to some sorting order, and assume that the source follows a probability distribution P, so that message $\mathbf{x}_i^n$ is produced with probability $P(\mathbf{x}_i^n)$.

Let

$$G_n(P) = -\frac{1}{n}\sum_{i=1}^{M^n} P(\mathbf{x}_i^n)\log_2 P(\mathbf{x}_i^n) \tag{1}$$

$G_n(P)$ decreases monotonically with n [23] and the entropy of the source is:

$$H(P) = \lim_{n\to\infty} G_n(P) \text{ bits/symbol.} \tag{2}$$

An alternative formulation for $H(P)$ uses conditional probabilities. Let $P(\mathbf{x}_i^{n-1}, a_j)$ be the probability of sequence $\mathbf{x}_i^n = \mathbf{x}_i^{n-1} a_j$ ($\mathbf{x}_i^{n-1}$ concatenated with $x_n = a_j$) and let $P(a_j \mid \mathbf{x}_i^{n-1}) = P(\mathbf{x}_i^{n-1}, a_j)P(\mathbf{x}_i^{n-1})$ be the probability of $x_n = a_j$ conditioned on $\mathbf{x}_i^{n-1}$. The entropy of the n^{th} order approximation to $H(P)$ [23] is:

$$F_n(P) = -\sum_{i=1}^{M^n}\sum_{j=1}^{M} P(\mathbf{x}_i^{n-1}, a_j)\log_2 P(a_j \mid \mathbf{x}_i^{n-1}) \text{ bits/symbol.} \tag{3}$$

$F_n(P)$ decreases monotonically with n [23] and the entropy of the source is:

$$H(P) = \lim_{n\to\infty} F_n(P) \text{ bits/symbol.} \tag{4}$$

Eq. 4 involves the estimation of probabilities conditioned on an infinite sequence of previous symbols. When finite memory is assumed the sources can be modeled by a Markov

process of order n-1, so that $P(a_j \mid \ldots x_{n-2}x_{n-1}) = P(a_j \mid x_1 \ldots x_{n-1})$. In this case, $H(P) = F_n(P)$.

Define the *coding rate* of a coding scheme as the average number of bits per symbol the scheme uses to encode the source output. A *lossless compressor* is a uniquely decodable coding scheme whose goal is to achieve a coding rate as small as possible. The coding rate of any uniquely decodable coding scheme is always greater than or equal to the source entropy [23]. Optimum coding schemes have a coding rate equal to the theoretical lower bound $H(P)$, thus achieving maximum compression.

For Markov processes of order n-1, optimum encoding is reached if and only if symbol $x_n = a_j$ occurring after $\mathbf{x}_i^{n-1}$ is coded with $-\log_2 P(a_j \mid \mathbf{x}_i^{n-1})$ bits [15, 23]. However, it may be impossible to accurately estimate the conditional distribution $P(. \mid \mathbf{x}_i^{n-1})$ for large values of n, due to the exponential growth of the number of different contexts, which brings well-known problems, such as context dilution [15].

III. The LZW Algorithm

Even though the source model P is generally unknown, it is possible to construct a coding scheme based upon some (possibly implicit) probabilistic model Q that approximates P. The better Q approximates P, the smaller the coding rate achieved by the coding scheme.

In order to achieve low coding rates, modern lossless compressors rely on the construction of sophisticated models that closely follows the true source model. *Statistical compressors* encode messages according to an estimated statistical model for the source. *Dictionary-based compressors* replace strings of symbols from the message to be encoded with indexes into a dictionary of strings, which is generally adaptively constructed during the encoding process. When *greedy parsing* is used, at each step the encoder searches the current dictionary for the longest string that matches the next sequence of symbols in the message, and replaces this sequence with the index of the longest matching string in the dictionary.

Dictionary-based compressors with greedy parsing, such as LZW, are highly popular because they combine computational efficiency with low coding rates. It has been proved that each dictionary-based compressor with greedy parsing has an equivalent statistical coder that achieves the same compression [15]. In dictionary-based coding, the dictionary embeds an implicit statistical model for the source.

The initial LZW dictionary contains all possible strings of length one. The LZW algorithm finds the longest string, starting from the first symbol of the message, that is already present in the dictionary. This string is coded with the index for the matching string in the dictionary, and the string is extended with the next symbol in the message, x_i. The extended string is added to the dictionary and the process repeats, starting from x_i [15].

LZW achieves optimum asymptotic performance for Markov sources, in the sense that its coding rate tends to the entropy of the source as the length of the message to be coded tends to infinity [24]. It means that LZW algorithm learns a progressively better model for the source during encoding, and a perfect model for the source is learned when an infinite message has been coded. In practice, since the messages to be compressed are finite, LZW only learns an approximate model for the source.

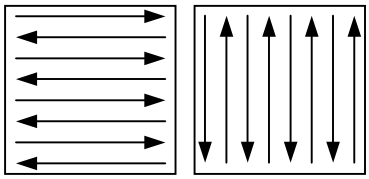
IV. The Proposed Method

Due to their capability to build accurate models, modern lossless compressors can be used as model-based classifiers. Any efficient lossless compressor could be used, but LZW algorithm was chosen, due to its good compromise between coding efficiency and computational requirements [15].

A. The Learning Stage

In the learning stage, the number N of classes is defined, and a training set T_i of texture samples known to belong to class C_i, is selected, $i = 1, 2, \ldots N$. The samples are n x n images extracted from histogram-equalized images. The LZW algorithm sequentially compresses all samples in T_i, following the horizontal scanning order shown in Figure 1.a, and the resulting dictionary H_i is kept as a model for the horizontal structure of textures in C_i, $i = 1, 2, \ldots N$.

The LZW algorithm then compresses all samples in T_i following the vertical scanning order shown in Figure. 1.b, and the resulting dictionary V_i is kept as a model for the vertical structure of textures in C_i, $i = 1, 2, \ldots N$.



(a) (b)

Fig. 1. Scanning order. (a) Horizontal; and (b) Vertical

B. The Classification Stage

In the classification stage, LZW operates in static mode. In this mode, one of the dictionaries generated in the learning stage is used, and no new strings are added to the dictionary during the encoding process.

An n x n texture sample $\mathbf{x}$ from an unknown class is coded by the LZW algorithm with static dictionary H_i, following the horizontal scanning order shown in Figure 1.a., and the corresponding coding rate h_i is registered, $i = 1, 2, \ldots N$. Then the LZW algorithm with static dictionary V_i encodes $\mathbf{x}$, following the vertical scanning order shown in Figure 1.b., and the corresponding coding rate v_i is registered, $i = 1, 2, \ldots N$. As in the learning stage, all samples are extracted from histogram-equalized images.

Let

$$r_i = \frac{h_i + v_i}{2} \tag{5}$$

Sample **x** is assigned to C_i if $r_i < r_j$, $j = 1, 2, \ldots, N$, $j \neq i$. The rationale is that if **x** is a sample from class C_i, the dictionaries H_i and V_i probably embeds the model that best describes its horizontal and vertical structure, thus yielding the smallest coding rates.

V. Experimental Results

Thirty natural textures from the Brodatz album [25], obtained from a public archive, were selected to evaluate the performance of the proposed method. In the experiments, each Brodatz texture constitutes a separate class. All textures have 640 x 640 pixels, with 8 bits/pixel. This corpus is the same used in [14], thus allowing direct comparison with several state-of-the-art texture classifiers from the literature. The Brodatz textures used are shown in Figure 2.

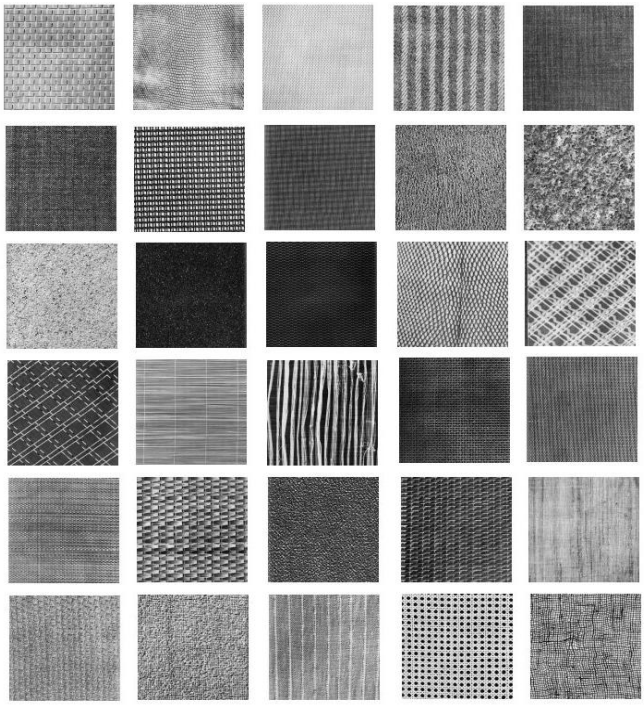


Fig. 2. Brodatz textures used in the experiments. From left to right and from top to bottom: D1, D3, D6, D11, D16, D17, D20, D21, D24, D28, D29, D32, D34, D35, D46, D47, D49, D51, D52, D53, D55, D56, D57, D65, D78, DD82, D84, D85, D101, D104.

Each Brodatz texture was histogram equalized and partitioned in n x n non-overlapping subimages, which were taken as texture samples. The samples were separated in two disjoint sets, one for training and the other for testing the classification accuracy. It is important to notice that only with disjoint sets for training and testing it is possible to reach accurate results. Nevertheless, as pointed out in [12] and in [14], the use of overlapping sets is quite common in the texture classification literature. In these cases, reported results are normally overoptimistic and not attainable in a realistic situation.

Classification accuracy is assessed by the *correct classification rate* (*CCR*):

$$CCR = \frac{c}{t} \text{ x } 100\,\% \tag{6}$$

where c is the number of texture samples correctly classified, and t is the number of classified samples. For comparison, tests were made with and without applying histogram equalization in the learning and in the classification stage. In this section and in the next one, the proposed classifier with and without histogram equalization will be identified as CLZWHE and CLZW, respectively.

In the first experiment, the training set comprised the first three quadrants of each texture, and the test set comprised the last quadrant of each texture. The effect of the sample size in *CCR* was then evaluated for n = 4, 8, 16, 32. Results are shown in Table 1.

TABLE 1
CCR ACHIEVED BY CLZW AND CLZWHE FOR VARIOUS SAMPLE SIZES.

Sample size (n x n)	*CCR* (%)	
	CLZW	CLZWHE
4 x 4	80.6	99.9
8 x 8	97.9	100
16 x 16	99.7	100
32 x 32	100.0	100

In the second experiment, the sample size was fixed in 32 x 32. Consequently, there are 400 non-overlapping texture samples in each class. The effect of the training set size in the classifier accuracy was then assessed. The proportion of training samples in each class was set to 1.25% (5 samples), 2.5% (10 samples), 3.75% (15 samples), 5% (20 samples), 6.25% (25 samples), 7.5% (30 samples), 8.75% (35 samples) and 10% (40 samples). In each case, all samples that were not used for training were used for testing. Table 2 summarizes the results of CLZWHE and CLZW, along with the results [14] for single and fused SVM, Bayes classifier using Bayes distance, and the LVQ classifier.

TABLE 2
CCR WITH CLZWHE, CLZW, BAYES CLASSIFER WITH BAYES DISTANCE (BAYES-BAYES), LVQ, SINGLE SVM AND FUSED SVM, FOR VARIOUS LEARNING SET SIZES.

Number of learning samples per class	*CCR* (%)					
	Bayes-Bayes	LVQ	Single SVM	Fused SVM	CLZW	CLZWHE
5	79.5	69.7	78.2	78.4	93.8	99.3
10	86.5	80.0	87.5	87.9	96.2	99.6
15	87.9	83.9	89.6	90.2	98.4	99.9
20	90.3	87.0	91.8	92.5	99.2	100
25	91.2	88.2	91.4	92.9	99.3	100
30	91.8	89.6	94.6	95.1	99.3	100
35	92.1	90.2	94.9	95.9	99.7	100
40	92.7	90.4	95.8	96.3	99.7	100

Figure 3 presents a graphical comparison of the *CCR*

achieved by CLZWHE, CLZW and fused SVM *versus* training set size.

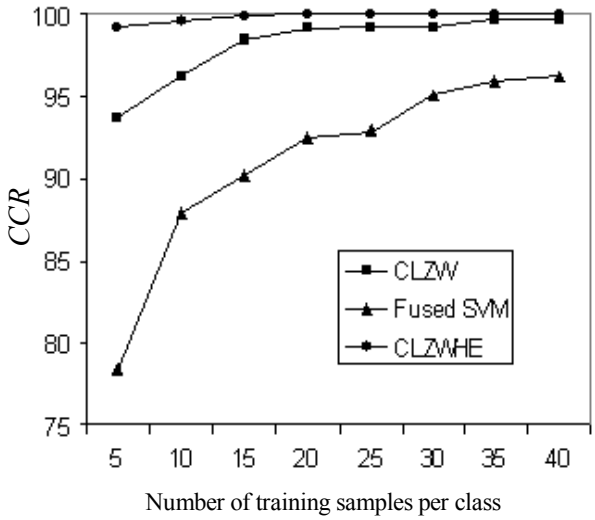


Fig. 3. *CCR* achieved by CLZWHE, CLZW and fused SVM *versus* training set size.

VI. Discussion and Conclusion

This paper proposed a new, simple and highly accurate texture classification scheme based on the LZW algorithm.

In the evaluation of the effect of sample size over classification accuracy, Table 1 shows that histogram equalization has a positive impact in the classifier performance. CLZWHE achieved *CCR* = 100% with samples as small as 8 x 8 pixels, while CLZW achieved *CCR* = 100% only with 32 x 32 or greater samples. The improvement derived from histogram equalization for 4 x 4 samples was impressive.

The second experiment, summarized in Table 2 and Figure 3, shows the superiority of the proposed method over single and fused SVM, Bayes classifier using Bayes distance and the LVQ classifier. This superiority is still more remarkable for very small training sets. With only 5 training samples for each class, CLZWHE achieved *CCR* = 99.3% and CLZW achieved *CCR* = 93.8%, while the third best method in this case, Bayes classifier with Bayes distance, achieve *CCR* = 79.5%. This shows the capability of the proposed method to learn and generalize from very small training sets. This is ratified by noticing that the proposed method achieved *CCR* = 100% with only 20 training samples.

Future directions of research include making the classifier invariant to rotation, scale and illumination changes; investigating the use of lossy dictionary-based compressors; adapting the method for other image classification and segmentation problems and for texture synthesis.

References

[1] M. Porat, Y. Y. Zeevi., "Localized Texture Processing in Vision: Analysis and Synthesis in the Gaborian Space," *IEEE Transactions on Biomedical Engineering*, vol. 36, issue 1, pp. 115-129, 1989

[2] F. Dell'Acqua, P. Gamba, "Texture-based Characterization of Urban Environments on Satellite SAR Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, issue 1, pp. 153-159, 2003

[3] M. F. Augusteijn, L. E. Clemens and K. A. Shaw, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, issue 3, pp. 616-626, 1995

[4] T. E. Southard, K. A. Southard, "Detection of Simulated Osteoporosis in Maxillae Using Radiographic Texture Analysis," *IEEE Transactions on Biomedical Engineering*, vol. 43, issue 2, pp. 123-132, 1996

[5] A. Kumar, G. K. H. Pang, "Defect Detection in Textured Materials Using Optimized Filters Systems," *IEEE Transactions on Man and Cybernetics, Part B*, vol. 32, issue 5, pp. 553-570, 2002

[6] A. K. Jain, A. Ross, S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, issue 1, pp. 4-20, 2004

[7] X. He, Y. Hu, H. .Zhang, M. Li, Q. Cheng, S. Yan, "Bayesian Shape Localization for Face Recognition Using Global and Local Textures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, issue 1, pp. 102-113, 2004

[8] S. Baheerathan, F. Albregtsen, H. E. Danielsen, "New Texture Features Based on the Complexity Curve," *Pattern Recognition*, vol. 32, issue 4, pp. 605-618, 1999

[9] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd ed., New York: Wiley Interscience, 2000

[10] A. Bovik (ed.), *Handbook of Image and Video Processing*, San Diego: Academic Press, 2000

[11] F. S. Cohen, Z. Fan, M. A. Patel, "Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, issue 2, pp. 192–202, 1991

[12] T. Randen, J. H. Husøy, "Filtering for Texture Classification: a Comparative Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 21, issue 4, pp. 291-310, 1999

[13] J. M. H. du Buf, M. Kardan, M. Spann, "Texture Feature Performance for Image Segmentation," *Pattern Recognition*, vol. 23, issues 3-4, pp. 291-309, 1990

[14] S. Li, J. T. Kwok, H. Zhu, Y. Wang, "Texture Classification Using the Support Vector Machines," *Pattern Recognition*, vol. 36, issue 12, pp. 2883-2893, 2003

[15] T. C. Bell, J. G. Cleary, J. H. Witten, *Text Compression*, Prentice-Hall, Englewood Cliffs, 1990

[16] J. Ziv, A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, vol. 24, issue 5, pp. 530-536, 1978

[17] J. Ziv, "On Classification with Empirically Observed Statistics and Universal Data Compression," *IEEE Transactions on Information Theory*, vol 34, issue 2, pp. 278-286, 1988

[18] J. Ziv, N. Merhav, "A Measure of Relative Entropy Between Individual Sequences with Application to Universal Classification," *IEEE Transactions on Information Theory*, vol. 39, issue 4, pp. 1270-1279, 1993

[19] W.A. Finamore, M. A. Leister, "Lossy Lempel-Ziv algorithm for large alphabet sources and applications to image compression," *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 235-238, 1996

[20] E. Frank, C. Chui, I. H. Witten, "Text Categorization Using Compression Models," *Proceedings of the Data Compression Conference*, Salt Lake City, pp. 555, 2000

[21] W. J. Teahan, D. J. Harper, "Using Compression Based Language Models for Text Categorization," *Workshop on Language Modeling and Information Retrieval*, pp. 83-88, 2001

[22] T. A. Welch, "A Technique for High Performance Data Compression," *IEEE Computer*, vol. 17, issue 6, pp. 8-19, 1984

[23] C. E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, (1948)

[24] S. A. Savari, "Redundancy of the Lempel-Ziv-Welch Code," *Proceedings of the Data Compression Conference*, Salt Lake City, pp. 191-200, 1997

[25] P. Brodatz, *Textures: A Photographic Album for Artists and Designers.* Dover, New York, 1966.