Automatic Artifact Identification in Image Communication Using Watermarking and Classification Algorithms

Shabnam Sodagari, G. Hossein Hajimirsadeghi, Alireza Nasiri Avanaki
Control and Intelligent Processing Center of Excellence, School of ECE, University of Tehran,
P. O. Box 14395-515, Tehran, Iran
Emails: shabnam@ieee.org, h.hajimirasdeghi@ece.ut.ac.ir, avanaki@ut.ac.ir

***Abstract*-We present a novel scheme, which enables the receiver to automatically identify the channel noise type in image communication. The method is based on embedding the image histogram, and deriving its specific statistics as descriptive features of the original image, through a robust watermarking algorithm within itself at the transmitter and extracting the hidden data at the receiver, for comparing with the corresponding features of the noise-distorted image. Then, by using a classifier, we are able to distinguish the channel noise type. Implementation results prove the efficiency of our proposed system in capably recognizing the noise type for common image transmission noise varieties.**

***Keywords*-Image communication, noise, classification algorithms, watermarking.**

I. INTRODUCTION

Images transmitted over communication channels and networks are prone to quality degradation caused by noise. Various models of noise usually used in image communications are Additive White Gaussian Noise (AWGN), salt and pepper, packet loss, to name a few. This diversity in noise categories accordingly necessitates different strategies for overcoming the effects of each distortion at the receiver. Therefore, it is important for the receiver to recognize the type of noise, to be able to select and employ the proper noise reduction technique.

Research has already been conducted to distinguish image noise varieties (e.g., [11]), but the already proposed methods [ibid] concentrate on identifying the group to which the distortion belongs to, i.e., additive, multiplicative and impulsive noises rather than exactly specifying the precise noise type.

One suitable feature of an image is its histogram, which according to our observations (proved by experiments which follow next in this paper) and also as mentioned in [11] reflects the existence of some common distortions happening to an image in a pertinent manner. Another advantage of the histogram is the low capacity it requires to be hidden within an image which diminishes the watermark visibility to a great extent. Therefore we have chosen to embed it as a characteristic to be compared to that of the received image.

In this paper, to present a solution to noise type identification at the receiver side of an image communication system. We first compute the image histogram at the transmitter and embed it through a robust watermarking scheme (which remains unchanged even after being prone to channel or network noise) within the original image. At the receiver this data is extracted and its statistics (i.e., moments) are derived. The same statistics of the histogram of the received noisy image are also computed. Both vectors are then fed to a classifier which has already been trained to tell apart various noise types. This way the receiver is equipped with a mechanism to automatically identify which noise category the received image has experienced during the transmission process.

The organization of this paper is as follows: in section II the details of our proposed scheme are discussed. The classification algorithms we have used are elucidated in section III. The data set for our classification is explained in section IV. We demonstrate our implementation results in section V. Finally section VI concludes the paper with future path of work.

II. PROPOSED METHOD

Data hiding in image communication has already been used for various purposes e.g., quality assessment [1]–[3] and error concealment [4].

In our technique, we first take the Discrete Wavelet Transform (DWT) (e.g., Haar) of the image at the transmitter. We also compute the histogram of the image, which is composed of 256 bins for grayscale images, and embed it in the horizontal and vertical detail coefficients of the wavelet decomposition. Our data hiding scheme, which is a modification of [1], is formulated as follows:

$$\begin{aligned} lh1'(4i-3) &= lh1(4i-3) + \alpha \times h(i) \\ lh1'(4i-2) &= lh1(4i-2) - \alpha \times h(i) \\ 1 \le i &< 256/2 \end{aligned} \tag{1}$$

$$hl1'(4i-3) = hl1(4i-3) + \beta \times h(i)$$
$$hl1'(4i-2) = hl1(4i-2) - \beta \times h(i) \qquad (2)$$
$$256/2 < i \le 256$$

where $lh1$ and $hl1$ denote the horizontal detail and vertical detail coefficients of one level wavelet decomposition of the image before embedding the watermark. $lh1'$ and $hl1'$ are the corresponding coefficients after embedding the watermark. $h(i)$ for $1 \le i \le 256$ denotes the histogram of the image. This way, half of the histogram data is hidden in the horizontal detail coefficients and the other half is hidden in the vertical detail coefficients. α and β, given below, are two regulating constants derived by [5], chosen to ensure invisibility (Fig. 1) and are delivered to the receiver.

$$\alpha = \frac{E_{LH}}{100E_h} \qquad \beta = \frac{E_{HL}}{100E_h} \qquad (3)$$

(a) (b)

Fig. 1. (a) original *livingroom* (b) watermarked image

E_{LH}, E_{HL} and E_h are the energies of horizontal and vertical detail coefficients and the histogram respectively. In order to eliminate the visibility of the watermark we have divided the constants by a number (here 100) which is obtained by trial and error. As is evident from Fig. 1 the watermark has no visible effects on the original image.

The embedded histogram which remains unchanged during transmission can be extracted at the receiver by taking the DWT of the received image and following the procedure below:

$$h(i) = \frac{lh1'(4i-3) - lh1'(4i-2)}{2\alpha} \qquad (4)$$
$$1 \le i < 256/2$$

$$h(i) = \frac{hl1'(4i-3) - hl1'(4i-2)}{2\beta} \qquad (5)$$
$$256/2 < i \le 256$$

This watermarking scheme is based on the assumption that the variations in neighboring wavelet coefficients are not considerable and therefore it does not introduce much error if their difference is neglected.

Next, the statistics of the extracted histogram and those of the histogram of the noise corrupted image are derived and fed into the classifiers.

III. Classification Algorithms

In this study a number of statistical and structural classification algorithms are employed. Artificial Neural Networks (ANN) is a popular structural means of classification which is capable of massive parallel processing and model representation [12]. A Multi Layered Percepton (MLP) neural network is a simple structure of feedforwrd ANN and is widely used for pattern recognition purposes. In ANN, the structure and hidden layers of the networks play an important role to cope with non-linear classification problems [12], [19]. However, the exhaustive required time for training and calibration in some applications necessitate pursuing faster algorithms.

On the other hand, statistical classification algorithms have usually less computational cost at the training stage, but unsatisfactory results might be obtained, depending on the distribution of data sets. Furthermore, they are more sensitive to errors in their data [13]. To balance the relative influence of data set on the model [14], [15], extraction of the most significant features, and also avoiding irrelevant structures, some data pre-treatments are essential before applying statistical classification algorithms.

Data pre-treatment is often performed in Principle Component Analysis (PCA) [16]. PCA is a linear transform from original features onto principle component axes. These orthogonal axes make a space in which the distance between classes is the greatest, whereas the inner distance for each class is the least. This space provides a high potential of separability for good performance of statistical classification methods. In PCA, a matrix known as separability matrix is introduced and the importance of the PCA reduced features are commonly measured by eigenvalues of the separability matrix.

Other classifiers which are used in this experiment are Bayesian, *k*-Nearest-Neighbor (*k*NN), Minimum Mean Distance (MMD), linear discriminant, and Support Vector Machine (SVM) classifiers.

In Baysian classifiers, using Bayes rule, the classifier selects the most probable class for unseen feature vectors. In this approach the distribution of samples for each class should be determined. Thus, a number of techniques are employed to estimate the probability density functions (pdf's) for classes. In this paper we use Gaussian estimator as a parametric approach and *k*-Nearest-Neighbor and Parzen estimators as nonparametric techniques.

*k*NN classifier is one of the simplest classification techniques which also provides successful results in most

cases. In *k*NN, the *k*-nearest objects to the unknown sample are selected and a majority voting is conducted to predict the class of that sample. Another algorithm for classification is MMD, in which the distance of the unknown sample from each class is calculated, and then the sample is associated with the nearest class. Linear classifiers are a group of well-known statistical classifiers. In linear discriminant classifiers, linear discriminant functions are constructed to distinguish the regions between classes.

Support Vector Machine (SVM) is a supervised learning approach, used to solve both classification and regression problems. SVM is based on statistical learning theory [17], [18], and in recent years a considerable amount of attention has been paid to extend this algorithm. Considering the two class problem, the goal is to find the optimal hyperplane boundary that exactly separates both classes. The closest vectors to the boundary are support vectors and the margin is the minimal distance between the hyperplanes and support vectors. In some cases, soft margins are employed to make the separating hyperplane to increase the margin and generalization at the cost of misclassification for some of samples in the training set. Note that when classes are separated by a nonlinear boundary, the kernel method is applied. Some of the common kernels are Linear, Polynomial, and RBF kernels.

IV. Data Set for Noise Classification

Here we use the standard deviation and the third to seventh moments of both the watermarked and the noisy histograms as the feature vector to be the input of our classifier for distinguishing the noise type. Note that mean is the same for all image histograms and is equal to 1024.

Lena, cameraman, mandrill, lake, jetplane, house, walkbridge, peppers, pirate, and *livingroom* are used in this study to prepare the data set for classification. Four types of noise including salt & pepper (with density range of 1%-70%), JPEG (with quality factors 35-90), packet loss (loss probabilities of 2%-70%), and AWGN (PSNR's ranging from 8-29 dB), were added to each picture.

The data set was divided into 3 subsets, a classifier training set, a validation set, and a classifier test set. Except for "part a" in section V, in all the simulations, the training set and validation sets were derived from *lena, cameraman*, and *mandrill* and the test set was constructed by feature vectors of the seven remaining images. For MLP classifier, linear scaling of the features was performed, whereas for all the statistical techniques, whitening and PCA were conducted. After applying PCA, three combined features were obtained with eigenvalues λ=[0.5229,0.2063,0.0966], indicating that the first and the second reduced features have the highest separability. In Fig. 2 and Fig. 3, the distributions of the original features of the two first noisy features and the distributions of PCA reduced features for all four classes are depicted, respectively. These figures imply that ideally a nonlinear classifier is needed to solve this classification problem.

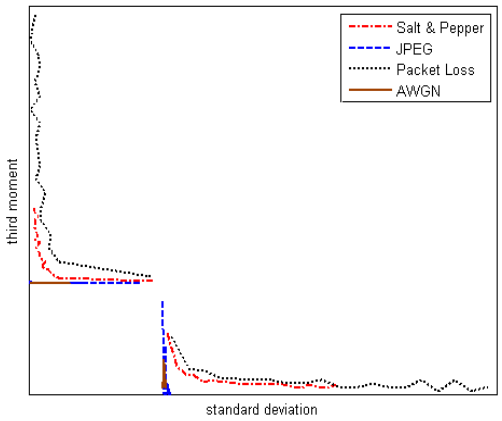

Fig. 2. Distribution of the two first features for each class before PCA.

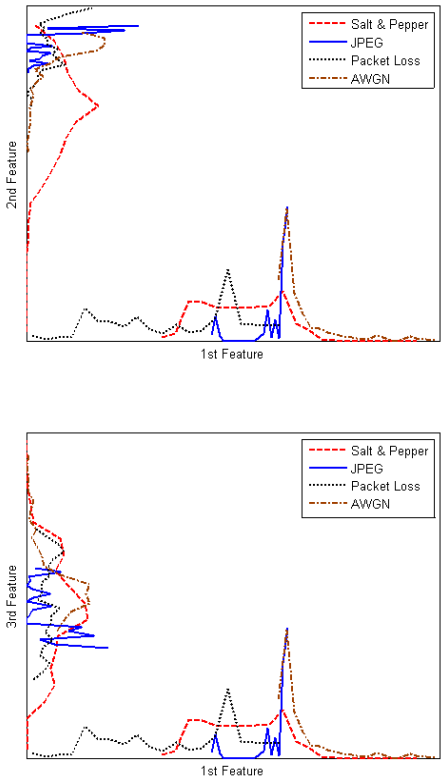

Fig. 3. Distribution of the (top) first and second and (bottom) first and third PCA reduced features.

V. Implementation Results

A. Training MLP with Lena Feature Vectors

First, the statistics derived from *lena* were used to construct the training, validation, and test data. Easily, average correct classification rate of approximately 1 was achieved for all the

structures with four to twelve features. It means that the second and the third moments of both the watermarked and the noisy histograms were enough to identify the noise in *lena*. The best results were obtained for the structure with four features and three neurons in the hidden layer. Evaluating this model on other images, the classifier failed to detect the noise in the test vectors.

In the second step, we trained MLP with len*a* data set and validation set was extracted from *cameraman*. Like all neural networks training runs in this study, early stopping approach was employed to have an acceptable generalization (Fig. 4). The best model was achieved with the same structure, described in the previous paragraph and with the average correct classification ratio of 0.90. But, only the mean rate of 0.18 was resulted for feature vectors of other images with this classifier. It means that using only one image as a training source is not sufficient for designing a general classifier.

B. Training MLP Neural Network

A MLP Neural Network with feature vectors of three different images i.e., *lena*, *cameraman*, and *mandrill* was constructed. A number of different structures were examined and the best accuracy (86%) was obtained with second and third moments of both the watermarked and the noisy histograms as feature vectors, indicating that these features are of the highest importance for the purpose of classification. The results for this classifier accompanied by other classifiers are provided in Table I.

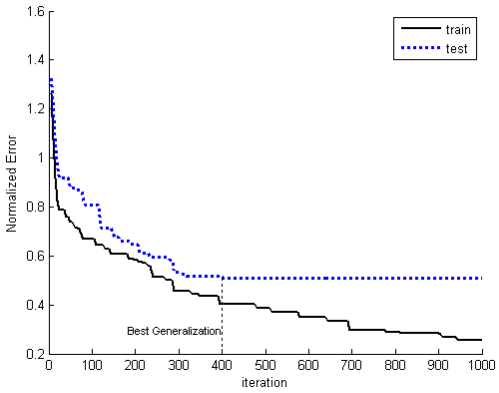

Fig. 4. Training process and early stopping approach. (Note that the normalized best error is illustrated)

The confusion matrix [20], which contains information about actual and predicted classifications and shows the degree of confusion between classes for the best MLP classifier is illustrated in Table II.

C. Bayesian Classification

To implement Bayesian classifiers, Gaussian estimator, *k*NN estimator, and Parzen estimator were employed to estimate the pdf's. For *k*NN, $k = 5$, and for Parzen h = 0.8 were selected because of having the best validation.

D. kNN and MMD Classifiers

*k*NN and MMD classifiers were also applied to this problem. Through validation process $k = 13$ was selected for *k*NN. It is obvious that *k*NN has a better performance for classification of the noise and as it is a fast algorithm it can be easily employed for our purpose.

TABLE I
PERFORMANCE OF THE PROPOSED CLASSIFIERS

Classifier	Accuracy
MLP	**0.864**
Bayesian (Gaussian)	0.824
Bayesian (*k*NN)	0.826
Bayesian (Parzen)	0.772
***k*NN**	0.829
MMD	0.528
Linear (l)	0.768
Linear ($l(l-1)/2$)	0.802
SVM	0.84

TABLE II
CONFUSION MATRIX FOR THE BEST MLP CLASSIFIER

		Predicted			
		Salt & Pepper	JPEG	Packet Loss	AWGN
Actual	Salt & Pepper	373	10	1	43
	JPEG	0	338	77	12
	Packet Loss	0	29	398	0
	AWGN	0	61	0	366

E. Linear Discriminant Classifiers

Two methods were used for evaluating our linear classifiers. In the first one, l linear classifiers were constructed, each one discriminating between one of the l classes in the problem with $l-1$ remaining classes. In the second method $l(l-1)/2$ classifiers were developed to discriminate between each of the l classes in the problem. In both approaches the majority voting was conducted to assign class labels to unknown samples. The results verify that using $l(l-1)/2$ linear discriminant functions have better results in this experiment.

F. SVM

In this study, we used soft margin support vector machine with linear kernels to classify the noise in the images. Indeed, both hard and soft margin SVM's were examined and better results were obtained by soft margin SVM classifier with

average correct classification ratio of 0.84. The plots of using linear SVM for classifying the noise are illustrated in Fig. 5 and Fig. 6. Fig. 5 is an example of soft margin classifier, while Fig. 6 is an example of hard margin SVM.

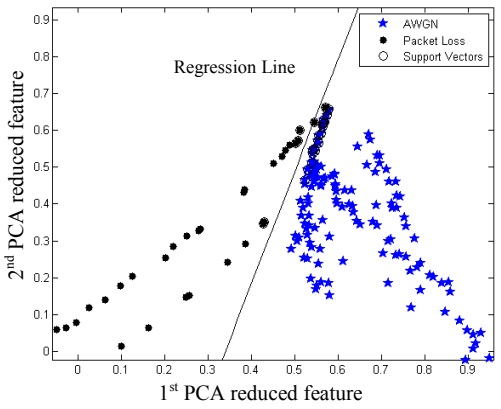

Fig. 5. Soft Margin SVM for discriminating between Packet Loss and AWGN, using the first two PCA reduced feature vectors.

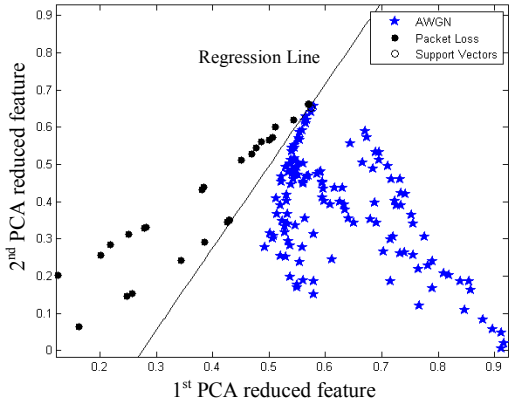

Fig. 6. Hard Margin SVM for discriminating between Packet Loss and AWGN, using the first two PCA reduced feature vectors.

Simulation results show that the features we extracted produce excellent results for many classification algorithms, and also our method outperforms other relevant techniques (e.g. [11]) in that it targets identifying the exact noise category than a class of similar noise types with average correct classification ratio of 0.86.

VI. Conclusions

We presented a method for automatic dominant noise type identification at the receiver of an image transmission system and demonstrated its efficiency of implementation by putting forward the high correct classification ratios we achieved. We obtained the best results (about 86% accuracy) by the neural network classifier. This technique can be extended to video communication applications and color images and in addition, to measuring the attributes of the detected noise. It can be investigated if instead of using the histogram of the whole image a rough estimate of the histogram of each block can be used. The solution to telling apart all noise categories when two or more noise types affect the image is a topic of future work.

Furthermore, other advanced neural network structures and learning approaches that offer more generalization and adaptation can be used such as Locally Linear Model Tree (LoLiMoT), Adaptive Network Based Fuzzy Inference System (ANFIS), Takagi–Sugeno (TS), and Piecewise Linear Networks (PLN) [6]–[10].

References

[1] M. H. Kayvanrad, S. Sodagari, A. Nasiri Avanaki, and H. Ahmadi-Noubari, "Reduced reference watermark-based image transmission quality metric," in *Proc. 3rd Int. Symp. Communication, Control and Signal Processing,* pp. 526-531, March 2008.

[2] T. Brandão, and P. Queluz, "Towards objective metrics for blind assessment of images quality," in *Proc. Int. Conf. Image Processing*, pp. 2933-2936, Oct. 2006.

[3] M. Holliman and M.M. Yeung, "Watermarking for automatic quality monitoring," in *Proc. SPIE Security and Watermarking of Multimedia Contents*, vol. 4675, pp. 458-469, Jan. 2002.

[4] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, and M. Carli, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 15, no. 11, pp. 1394-1406, Nov. 2005.

[5] M. Barni, *Document and Image Compression*, CRC Press, pp. 255-278, 2006.

[6] W. Eppler, H. N. Beck, "Piecewise Linear Networks (PLN) for function approximation," in *Proc. IJCNN'99,* pp. 388–391, 1999.

[7] J. S. R. Jang, "Adaptive-network-based fuzzy inference system," *IEEE Transactions on System, Man and Cybernetics*, vol. 23, pp. 665–685, 1993.

[8] O. Nelles, "Orthonormal basis functions for nonlinear system identification with local linear model trees (LoLiMoT)," in *Proc. of IFAC symposium on system identification*, Kitakyushu, Fukuoka, Japan, 1997.

[9] M. Sugeno, G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, pp. 15–33, 1988.

[10] G. Giacinto, F. Roli, L. Bruzzone, "Combination of neural and statistical algorithms for supervised classification of remote-sensing images," *Pattern Recognition Letters*, vol. 21, no. 5, pp. 385-397, 2000.

[11] L. Beaurepaire, K. Chehdi, B. Voze, "Identification of the nature of noise and estimation of its statistical parameters by analysis of local histogram," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2805 -2808, Apr. 1997.

[12] I. A. Basheer, M. N. Hajmeer, "Artificial neural networks: fundamentals, computation, design and application," *Journal of Microbiological Methods*, vol. 43, pp. 3–31, 2000.

[13] D. M. Jackson, L. J. White, "Stability problems in non-statistical classification theory," *The Computer Journal*, vol. 15, no. 3, pp. 214-221, 1972.

[14] K. H. Esbensen, *The Introducer Package: Multivariate Data Analysis in Practice,* 5th ed. Camo, Oslo, ISBN 82-993330-3-2, pp. 1–11, 2006.

[15] G. W. Johnson, R. Ehrlich, "State of the art report on multivariate chemometric methods in environmental forensics," *Environmental Forensics*, vol. 3, pp. 59-79, March 2002.

[16] M. Turk, A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no.1, pp. 71-86, 1991.

[17] C. Cortes, V. Vapnik, "Support-vector networks," *Mach. Learn*, vol. 20, pp. 273–297, 1995.

[18] Y. Xu, S. Zomer, R.G. Brereton, "Support vector machines: a recent method for classification in chemometrics," *Critical Reviews in Analytical Chemistry*, vol. 36, pp. 177-188, 2006.

[19] E. C. Goncalves, L. A. Minim, J. S. R. Coimbra, V. P. R. Minim, "Modeling sterilization process of canned foods using artificial neural

networks," *Chemical Engineering and Processing*, vol. 44, no. 12, pp. 1269-1276, Dec. 2005.

[20] R. Kohavi, F. Provost, "Glossary of terms," *Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process*, vol. 30, pp. 2–3, 1998.