Removal of Impulse Noise in Images by Means of the Use of Support Vector Machines

H. Gómez-Moreno, S. Maldonado-Bascón, F. López-Ferreras, and P. Gil-Jiménez

Departamento de Teoría de la Señal y Comunicaciones. Universidad de Alcalá
28871 Alcalá de Henares (Madrid). SPAIN
{hilario.gomez,saturnino.maldonado,francisco.lopez,pedro.gil}@uah.es

Abstract. In this work we present an efficient way to cancel the impulse noise in images by using the Support Vector Machines (SVMs). The suppression of impulse noise is a classic problem in nonlinear processing, and we show that the SVMs are especially useful in this processing. In this new approach we use the classification and the regression based on SVMs. By using the classifier we select the noisy pixels into the images and by using the regression we obtain a reconstruction value based on the neighboring pixels. The results obtained are comparable and, a lot of times, better than those from another "state-of-art" techniques. Besides, this new technique can be applied successfully to images with high noise ratios while maintaining the visual quality and a low reconstruction error.

1 Introduction

Sometimes the images that we receive have an added impulse noise. This impulse noise can be due to a noisy transmission channel or to imperfections of the sensor with which we obtain the images so that in some points a saturation takes place. The linear techniques are little effective in the reduction of this type of noise and as alternative the nonlinear techniques appear. A well-known nonlinear method is the median filter. The main disadvantage of this method is that it is applied on all the points of the image, noisy or not, and then a blurring is produced, specially with high rates of noise. This defect is common to other techniques dedicated to the elimination of impulse noise. In order to avoid this problem the known as "Switching Scheme" techniques appear. In these algorithms the substitution of pixels is only made in those considered as noisy. The detection of these noisy pixels and the substitution is implemented in different ways. In [1] one of these techniques is presented. The detection of noisy pixels is implemented by means of the comparison with certain thresholds. The noisy pixels are replaced with a modified median filter, that in that work is called Rank Ordered Mean (ROM) and that does not use the noisy pixel to calculate the median.

In this work, we implement a similar scheme but the detection and the substitution of noisy pixels is made with SVMs. The pixels of the noisy images are classified in "noisy" and "not noisy" using the SVMs as classifier and the substitution values are obtained with SVMs regression. Our method provides

excellent results in "Peak Signal to Noise Ratio" (PSNR), that measures the reconstruction error, and in visual quality and maintenance of the edges even for very high rates of noise.

2 SVMs classification and regression

The SVMs give a simple way to obtain good classification results with a reduced knowledge of the problem. The principles of SVMs have been developed by Vapnik [2] and have been presented in several works like [3].

The classification task is reduced to find a decision frontier that divide the data into the groups that we want to separate. The simplest decision case is when the data can be divided into two groups like in the application presented in this work.

As a generalization of the SVMs classification the SVMs regression appears. In this case the goal is to find a function that fits the sample data.

2.1 Bases of Support Vector Machines

In the simplest decision problem we have a number of vectors ($\mathbf{x} \in \mathbb{R}^N$) divided into two sets, and we must find the optimal decision frontier to divide these sets. This optimal election will be the one that maximizes the distance from the frontier to the data. In the two dimensional case, the frontier will be a line, in a multidimensional space the frontier will be an hyperplane with the form,

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \;\; \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}. \tag{1}$$

In order to obtain $\mathbf{w}$ and b we must solve the next optimization problem,

$$\begin{cases} \text{minimize} \;\; \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \\ \text{subject to} \;\; y_i((\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, l. \end{cases} \tag{2}$$

After the solution of the previous problem the decision function that we are searching for has the next form where $\mathbf{w}$ is replaced with a linear combination of example vectors,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \left\langle \mathbf{x}_i \cdot \mathbf{x} \right\rangle + b. \tag{3}$$

The y values that appear into this expression are +1 for positive classification training vectors and –1 for the negative training vectors. Also, the inner product is performed between each training input and the vector that must be classified. Thus, we need a set of training data ($\mathbf{x}$,y) in order to find the classification function. The α values are the Lagrange multipliers obtained in the minimization process and the l value will be the number of vectors that in the training process contribute to form the decision frontier. These vectors are those with an α value not equal to zero and are known as support vectors.

When the data are not linearly separable this scheme can not be used directly. To avoid this problem, the SVMs can map the input data into a high dimensional feature space. The SVMs constructs an optimal hyperplane in a high dimensional space and then returns to the original space transforming this hyperplane in a non-linear decision frontier. The non-linear expression for the classification function is given in (4) where K is the kernel that makes the non-linear mapping,

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{4}$$

The choice of this non-linear mapping function or kernel is very important in the performance of the SVMs. The kernel used in our work is the radial basis function since it offers the best results with a reduced number of support vectors. This function has the next expression,

$$K(x, y) = \exp\left(-\gamma (x - y)^2\right). \tag{5}$$

The γ parameter in (5) must be chosen to reflect the degree of generalization that is applied to the data used. Also, when the input data is not normalized, this parameter performs a normalization task.

When some data into the sets can not be separated (due to noise for example), the SVMs can introduce slack variables ($\xi_i \geq 0$) to relax the minimization constraints. Then, the optimization problem takes the next form,

$$\begin{cases} \text{minimize} \quad \tau(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \\ \\ \text{subject to} \quad y_i((\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l. \end{cases} \tag{6}$$

A penalty term (C) that makes more or less important the misclassification error in the minimization process is included.

2.2 SVMs Regression

The regression process is similar to the one of classification, but in this case the values of y are real and represent the known values of the function on which we make the regression. Another modification is the use of a function that measures the required precision. The function that we use is the known as Vapnik's ε-insensitive loss function that has the next form,

$$|y - f(\mathbf{x})|_{\varepsilon} := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{7}$$

this way with the desired precision ε, one minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} |y_i - f(\mathbf{x}_i)|_{\varepsilon}. \tag{8}$$

This problem is similar to that in equation (6) but using as error measure the ε-insensitive function and taking into account the fact that y values are real. For further information about this optimization problem, [2] can be consulted.

3 Noise detection and substitution

In this work the first problem is to detect the noisy pixels. This task is treated as a classification procedure where the pixels are classified between "noisy" and "not noisy". The pixels detected as noisy must be changed. For this change we use a regression procedure based on SVMs.

In these processes we cannot simultaneously use all the pixels of the image due to the amount of calculations needed. Then, we must extract the information somehow. What we do is to form a vector for each pixel, formed by itself and by those in a window around it (except for the border). This vector will be the classifier input when we look for noisy pixels and the regression input when we obtain the pixel reconstruction value. The window size used in the scanning task depends on the quality and the speed required. A big window improves the detection of noisy pixels and the regression obtained but increases the training and execution time. In this work the results have been obtained using a 3x3 symmetric window (Figure 1).

x-1,j-1	x-1,j	x-1,j+1
x,j-1	**x,j**	x,j+1
x+1,j-1	x+1,j	x+1,j+1

Fig. 1. 3x3 Scanning window

The type of impulse noise on that we are going to work is known like salt&pepper. In this type of noise the noisy pixels take values 0 or 255 with equal probability. In this work the detection of noisy pixels is done for parts. First we detect the black pixels and we turn them into white ones. Later we look for the white pixels and replace them. This is the best option, since the detection of white and black pixels simultaneously and its regression is problematic given the disparity of values. Some experiments trying to make the regression with both noises have given an increasing number of support vectors and a reduction in the quality. This is an still open work.

In order to train the SVMs used in the detection and regression tasks we must take some example vectors of noisy images. One possibility is to take some pieces of real noisy images to obtain the vectors. But, we decided to make our own noisy images with a gray scale, a size and a noise ratio controlled. This way the training obtained is more general and it is not conditioned by the images used. We expect that the SVMs can generalize using only these reduced examples. In figure 2 some of these images are shown. They have a size of 32x64 and a percentage of noise of 30% (30% of pixels are noisy). The gray scale goes from 0 to 255 with steps depending on the image size.

As we see the images for noise detection and regression are similar but with a difference in the central edge. This difference is only because empirically spoken the results are better using these configurations. The central edges are necessary to simulate the edges present in the real images. The detection of black pixels is

(a) **(b)**

Fig. 2. Training images. (a) Detection (b) Regression

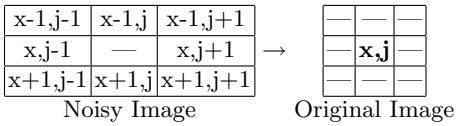

Fig. 3. Scheme of Regression procedure

make after training with images like those of figure 2(a) but with black pixels of noise.

The training images are scanned and for each pixel a vector is formed. To each one of these vectors we assign +1 when there is noise and –1 when no. After the training process we have the α and b values and the support vectors that forms the decision function (4).

In the regression process we use synthetic images like those of figure 2(b), but in this case which we do is to form a vector around each pixel and to put like value of regression the one of the original image (without noise) that occupies the same position. In this case we exclude the central pixel since we consider it like noisy. The process is resumed in figure 3.

4 Results

In this section some results obtained with the presented technique are presented. These results show the performance of our noise reduction scheme, some problems of this technique and a comparison with some other algorithm.

4.1 Implementation

The presented algorithm has been implemented using Visual C++ 6.0 in a computer with a Pentium IV processor (2.4 GHz) and 512 Megabytes of RAM. For the implementation of the SVMs we use the LIBSVM library [4] written in C++. This library have as advantage that is easy to use and tune.

Table 1. PSNR results in dB

	Albert				**Peppers**			
	20%	**30%**	**40%**	**50%**	**20%**	**30%**	**40%**	**50%**
SDROM [1]	30.6	28.61	26.66	24.46	31.44	29.3	26.94	24.42
Median 3x3	27.11	23.13	18.95	15.3	29.05	23.84	18.94	15.17
Median 5x5	26.6	25.93	24.77	22.47	30	28.53	26.58	23.57
SVM Median 30% training	***33.9***	***31.35***	28.7	26.62	37.68	33.95	28.26	23.57
SVM 30% training	29	27.51	26.83	25.83	***38.81***	***36.32***	34.05	31.71
SVM 40% training	32.77	30.27	***29.27***	***27.91***	38.35	36.08	***34.27***	***32.15***

The SVMs noise reduction is a general framework where several implementations are possible. One possibility is to find the noisy pixels and then obtain the reconstruction values using only the values of the noisy image, this procedure is known as "non recursive". Another option is to use the previously calculated reconstruction values to find the new reconstruction values, this form is known as "recursive" and allows a good reconstruction without apply the noise reduction several times. In the results presented we use the recursive implementation for all techniques except the median due to the poor results in this case.

The training process is fundamental for the performance of the SVMs algorithm and then the values used in this process must be carefully elected. The training images used are of 32x64 with 30% and 40% of noisy pixels. The parameters that must be tuned in the SVMs are the γ of the kernel (5) and C in the minimization of (6) or (8). In our case $\gamma = 2 \cdot 10^{-6}$ for noise detection and $\gamma = 3 \cdot 10^{-5}$ for regression and C = 1000 in all cases. These values have been obtained empirically.

4.2 Tests

We use for the tests eight bits gray-scaled images of 512x512 (Figure 4). The images in this figure represent two important types of images that we can find. Albert has some textures in the hair and in the jacket and its most important characteristic is the white collar of the shirt. This zone is problematic since part of the noise is white and then this zone may be detected as noisy and replaced. The images with uniform white or black zones have this problem. The image Peppers has only gray values and no textures, then the results in this image and those like this must be better.

In table 1 we present a comparison between the SVMs noise reduction and some other techniques. The data presented are the PSNR of the reconstructed images. This measure gives (in dB) the inverse of the normalized reconstruction error. The SDROM [1] and the median have been mentioned previously. The median SVM is a modified median filter [5] that applies a median filter to the pixels detected as noisy using the SVMs. Besides, we show the results of our technique with two training noise ratios in order to see how the performance is changed according to the training parameters.

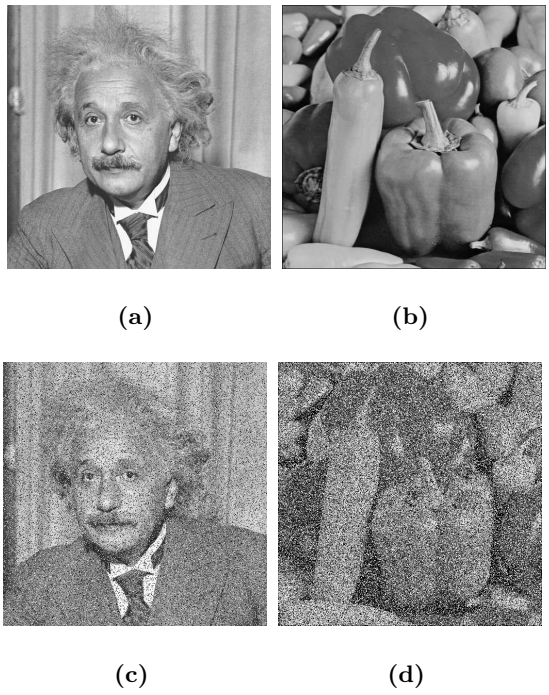

Fig. 4. Test images. (a) Albert (b) Peppers (c) Albert 20% noise (d) Peppers 50% noise

We can see how the SVMs noise reduction gives the bests results except for the Albert image with 20% and 30% of noise. This fact is due to an effect that can be seen in figure 5(c). The collar of the shirt is treated as noise and then some zones near the border have been blurred, this way the PSNR decreases. This effect can be reduced (Figure 5(d)) if we use training images with more noise ratio. In figures 5(a) and 5(b) we can see visual examples of how good our algorithm is for even high noise ratios.

The only drawback of our algorithm is that it takes more time than other techniques for the same images. For example, the median or the SDROM takes about 1 second for our tests images (20%) while the SVMs takes 12 seconds. This time is increased with the noise ratio and for 50% of noise can be 22 seconds.

5 Conclusions

In this work a new method of reduction of impulse noise that is very efficient for very high rates of noise and that is even superior than other techniques considered like standard is presented. This technique is based on the use of Support Vector Machines in the detection of the noisy pixels and for obtaining the values of reconstruction of these pixels.

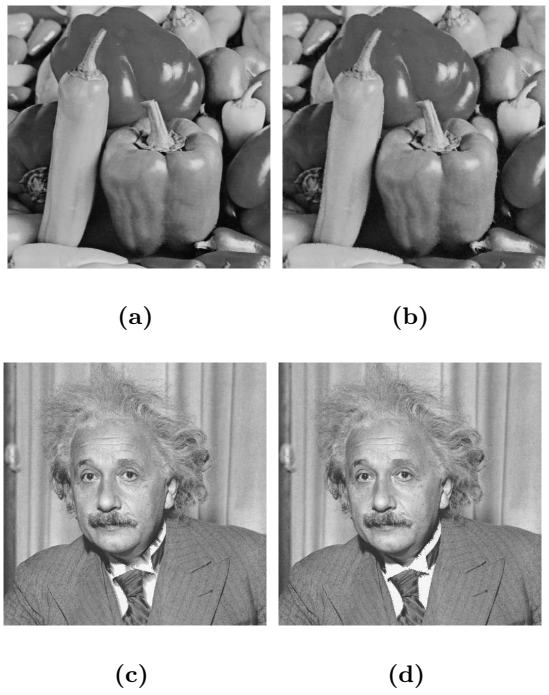

Fig. 5. Examples of noise removal. (a) Peppers (20%) recovered with 30% noise training (b) Peppers (50%) recovered with 30% noise training (c) Albert (20%) recovered with 30% noise training (d) Albert (20%) recovered with 40% noise training

With this work the number of applications of this new technique of learning is extended and in addition it demonstrates that they can be used successfully in the image processing task.

References

1. Abreu, E., Lightstone, M., Mitra, S.K., Arakawa, K.: A new efficient approach for the removal of impulse noise from highly corrupted images. IEEE Transactions on Image Processing **5** (1996) 1012–1025
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (2000)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Methods. Cambridge University Press, Cambridge, U.K. (2000)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
5. Gómez-Moreno, H., Maldonado-Bascón, S., López-Ferreras, F., Utrilla-Manso, M., Gil-Jiménez, P.: A Modified Median Filter for the Removal of Impulse Noise Based on the Support Vector Machines. In: Advances in Signal Processing and Computer Technologies. WSEAS Press (2001) 9–14