NONLINEAR FILTERS BASED ON SUPPORT VECTOR MACHINES

David A. Márquez, José L. Paredes, Winston García-Gabín
Electrical Engineering Department, Universidad de Los Andes. Mérida 5101, Venezuela
{damarquezg, paredesj, winstong}@ula.ve

ABSTRACT

In this work, a new family of nonlinear filters based on support vector machine is presented. This new filter, called support vector machine filter (SVMF), is based on the general concept of binary filters and machine learning theory. Two applications that show the potential of these filters are designed. As a first application, the proposed filter is used as an impulsive noise image denoising. The second application presents a new edge detection structure using a different point of view from the traditional ones. The results obtained for the applications at hand show that the proposed filter outperformes Center Weighted Median in the image denoising task and the traditional edge detectors. The proposed filter can be successfully applied for the processing of images corrupted with impulsive noise while maintaining the visual quality and a low reconstruction error.

Index Terms**—** Nonlinear Filter, Support Vector Machine, Image Processing

1. INTRODUCTION

Nonlinear filters have showed a great performance in signal, image and video processing tasks when the underlying noise follows an heavy-tail distribution. In particular, Boolean filters are a class of nonlinear filters widely used that encompass stack filters, weighted median filters, among others [1], [2]. They are characterized by two basic properties: threshold decomposition and a binary operation (Boolean function) performed over the observation samples. The success of these filters is based on two intrinsic properties: edge preservation and efficient noise attenuation, being robust against impulsive noise [3].

On the other hand, recently a new technique, called Support Vector Machine (SVM), has emerged as a powerful tool to solve classification and regression problems based on the statistical learning theory. This technique, originally used as an optimal classifier [4], [5], has been extended to a great variety of application [6], [7]. SVM provides a simple way to obtain good classification results with reduced *a priori* knowledge of the problem, being the most used application, the use of SVM as binary classifier where the observation vector is categorized in one out of two possible classes.

In this work, we take advantage of the classification capability of SVM in a binary filtering process, replacing the Boolean function that characterizes the Boolean filter by a Support Vector Machine, defines thus a new family of nonlinear filters.

To the best of our knowledge, it has not been reported in literature the use of Boolean filters based on SVM in which the training is based on the details and edge structure that have to be preserved. The closest work related to our approach uses SVM to classifier whether the underlying sample (pixel) is contaminated or not and apply a filtering operation [7].

A new family of non-linear filters based on SVM is proposed in this paper. The potential of these filters is showed in two image applications, image denoising and edge detection. In the latter, a new structure to detect edges using a non-traditional approach based on the proposed filters is developed. Different from the traditional methods where the derivative is approximated or complex mathematical methods are used to detect edges, the main idea, in our approach, is to train the SVM in the binary domain created by a threshold operator, so that it can recognize the presence of edges in the image of interest. The results show that our approach is better than traditional edge detectors in particular when the images are corrupted by impulsive noise and yields competitive results on clear images.

2. DEFINITIONS AND PRELIMINARIES

2.1. Threshold decomposition

Threshold decomposition is a powerful tool used for the analysis and implementation of Boolean filters including in this family median filters, weighted median filters, order static filters and stack filters [2], [8].

Consider an integer-valued sample set $X_1, X_2, \ldots, X_N$ forming a observation vector $\mathbf{X} = [X_1, X_2, \ldots, X_N]$, where $X_i \in Q$ and $Q = \{-M, \ldots, -1, 0, \ldots, M\}$. Threshold decomposition of $\mathbf{X}$ generates $2M$ binary vector, $\mathbf{x}^{-M+1}$, $\mathbf{x}^{-M+2}, \ldots, \mathbf{x}^{0}, \ldots, \mathbf{x}^{M}$ where the i-th element of $\mathbf{x}^m$ for $m = -M+1, -M+2, \ldots, 0, \ldots, M$ is defined as:

$$x_i^m = T^m(X_i) = \begin{cases} +1 & X_i \geq m \\ -1 & X_i < m \end{cases} \quad (1)$$

where $T^m(\cdot)$ is the *thresholding operator*. Thus, for each threshold level m, the i-th component of the binary vector is 1 if X_i is greater than or equal to the threshold value, otherwise it is -1.

Threshold decomposition has several important properties. First, the thresholding operator is a reversible operator. Each component of the original observation vector can be exactly reconstructed from the set of its corresponding binary values as $X_i = (1/2)\sum_{m=-M+1}^{M} x_i^m$. A second important property of threshold decomposition is the partial ordering on the set of binary vectors of fixed length. That is, for all threshold level $m \geq l$, it can be shown that $x^m \leq x^l$, where the smaller than or equal to operator $(\leq)$ holds for each component of the binary vector.

2.2. Boolean Filters

Let $\mathbf{X}(n)$ be the observation vector at n-th window position, a Boolean filter, denoted by $TBF_f(\mathbf{X}(n))$, is defined as [1]:

$$TBF_f(\mathbf{X}(n)) = \frac{1}{2}\sum_{m=-M+1}^{M} f\left(T^m(\mathbf{X}(n))\right) \quad (2)$$

This work was supported in part by the Fondo Nacional de Ciencia, Tecnología e Investigación - FONACIT under grant 2005000342.

where $\mathbf{X}(\boldsymbol{n}) = [X_1(\mathrm{n}),\ldots,X_N(\mathrm{n})]$, with $X_i(n)=X(n - (N-1)/2 + i-1)$, $f:\{-1,1\}^N \rightarrow \{-1,1\}$, and $T^m(\cdot)$ is the threshold operator defined in Eq. (1). Note that the filtering operations reduces to decomposing the observation vectors in its binary representation, filtering each binary vector using the binary filter $f(\cdot)$, adding up all the binary outputs to form the filter output. This family of filters is completely specified in the binary domain through the truth table of $f(\cdot)$ or its corresponding minimum sum of products.

Boolean filters include stack filters as a special case [2], [8]. A stack filter is a particular type of Boolean filter defined by Eq. (2) where $f(\cdot)$ is a positive Boolean function that satisfies the stacking constrain [2], [8].

2.3. Support Vector Machine (SVM)

SVM principles were developed by Vapnik and presented in several works as in [5], [9]. Consider a binary classification problem, where a collection of vectors ($\mathbf{X} \in R^N$) are available. Furthermore, consider that the set of vectors are related to two different classes, y_1 and y_2, and it is desired to find an optimal hyperplane to divide these classes. The optimal decision boundary will be the one that maximizes the distance from the hyperplane to the training data. In the two dimensional case, the hyperplane will be a line, while in a multidimensional space, the hyperplane will be so that, $\langle \mathbf{W}\cdot\mathbf{X}\rangle+b=0$, where $\mathbf{W} \in R^N$ and $b \in R$ are obtained as the solution of the optimization problem [9]:

$$\begin{cases} \min\limits_{W,b} \ \frac{1}{2}\|\mathbf{W}\|^2 \\ subject\ to: \ y_i \cdot \left(\left\langle \mathbf{W}\cdot\mathbf{X}_i \right\rangle + b\right) \geq 1, \quad i=1,\ldots,l \end{cases} \tag{3}$$

The decision hyperplane can be written as:

$$f(\mathbf{X}) = \operatorname{sgn}\left(\sum\nolimits_{i=1}^{l} \alpha_i y_i \left\langle \mathbf{X}_i \cdot \mathbf{X} \right\rangle + b\right) \tag{4}$$

where $y_i \in \{+1, -1\}$ defines the class where $\mathbf{X}_i$ belongs. Thus, a set of training data is needed $\left\{(\mathbf{X}_i; y_i)\big|_{i=1}^{l}\right\}$ in order to find the classification boundary. In Eq. (4), α`s are Lagrange multiplicators obtained as part of the solution of the constrained optimization problem and l represents the number of training samples used to define the decision frontier vectors. The vectors $\mathbf{X}_i$ for $\alpha_i \neq 0$ are known as support vectors since the separation region are defined by those vectors [4].

When the training data are not linearly separable, this scheme can not be used directly. In order to solve this problem, SVM turns the entry observation vector into a characteristic space of a higher dimension, solving the optimal problem in such space, and returning to the original space converting the optimal hyperplane in a non-linear decision frontier [4]. The non-linear expression to the classification function is given by:

$$f(X) = \operatorname{sgn}\left(\sum\nolimits_{i=1}^{l} \alpha_i y_i K(\mathbf{X}_i, \mathbf{X}) + b\right) \tag{5}$$

Where $K(\cdot,\cdot)$ is a kernel function that performs the non-linear transformation on the observation vectors.

In practice, an optimal separating hyperplane may not exist, in this case the optimization problem is solved by inserting non-negative slack variables ξ, reducing the optimization problem to:

$$\begin{cases} \min\limits_{W,b,\xi} \ \frac{1}{2}\|\mathbf{W}\|^2 + C\sum\limits_{i=1}^{l}\xi_i \\ subject\ to: \ y_i \cdot \left(\left\langle \mathbf{W}\cdot\mathbf{X}_i \right\rangle + b\right) \geq 1-\xi_i, \quad i=1,\ldots,l \end{cases} \tag{6}$$

where C is a penalty term that makes more or less important the misclassification error in the minimization process and, therefore, it is a tunning parameter [4].

3. SUPPORT VECTOR MACHINE FILTERS (SVMF)

For a boolean filter defined by Eq. (2), it is possible to replace the function $f(\cdot)$ that characterizes the Boolean filter by a decision function corresponding to a SVM, defining thus a new family of non-linear filters.

Let $\mathbf{X} = [X_1, X_2, \ldots, X_N]$ be the observation vector to be filtered, furthermore, let $\mathbf{x}^m = \left[x_1^m, x_2^m, \ldots, x_N^m\right]$ be its correspondent threshold decomposition at threshold level m.

The output of the Support Vector Machine filter, denoted by SVMF($\mathbf{X}$), is defined as:

$$SVMF(X_1, X_2, \ldots, X_N) = \frac{1}{2}\sum_{m=-M+1}^{M} SVM_f(x_1^m, x_2^m, \ldots, x_N^m) \tag{7}$$

where $SVM_f(\cdot): \{-1,+1\}^N \rightarrow \{-1,+1\}$, is a decision function corresponding to a SVM as in Eq. (4) and $\mathbf{x}^m = T^m(\mathbf{X})$, is the threshold decomposition of input vector $\mathbf{X}$.

Substituting $SVM_f(\cdot)$ for the decision structure from Eq. (5), Eq. (7) reduces to:

$$SVMF(\mathbf{X}) = \frac{1}{2}\sum_{m=-M+1}^{M}\left(\operatorname{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i \cdot K(\mathbf{x}^m, \mathbf{x}_i) + b\right)\right) \tag{8}$$

At first look, it seems that the computational cost of Eq. (8) is expensive, this, however, can be notably reduced if it is noticed that for any $m \in \left(-\infty, X_{(1)}\right]$ or $m \in \left(X_{(i-1)}, X_{(i)}\right], i = 2,\ldots,N$ or $m \in \left(X_{(N)}, +\infty\right]$, threshold decomposition outputs the same binary vectors. Therefore, there are at least $N+1$, different binary vectors $\mathbf{x}^m$. Thus, after some simplifications, Eq. (8) reduces to:

$$SVMF = \frac{X_{(1)} + X_{(N)}}{2} + \frac{1}{2}\sum_{i=2}^{N}\left(X_{(i)} - X_{(i-1)}\right)\operatorname{sgn}\left(\sum_{i=1}^{n} y_i\alpha_i \cdot k\left(\mathbf{x}^m, \mathbf{x}_i\right) + b\right) \tag{9}$$

where $X_{(i)}$ is the i-th smallest sample of the set $\{X_1, X_2, \ldots, X_N\}$, with $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(N)}$. The filter representation in Eq. (9) provides us with an interesting interpretation of the SVM filter. The filter output is computed by the sum of the midrange of the signed samples $(X_{(1)}+X_{(N)})/2$ and a linear combination of the differences between successive order statistics $(X_{(i)}-X_{(i-1)})$, multiplies by a factor ($\pm 1/2$) whose signed depends on the training samples. This is another way to see the proposed SVM filter and, as expected, output the same results as the one obtained by Eq. (8).

4. SVM FILTER DESIGN BASED ON DETAIL STRUCTURE

Several 3x3 masks were designed generalizing some particular cases that may appear in a filtering process. Thus, a 9-component vector are formed and used to train the SVM that, in turns, defines the filtering characteristic function.

Figure 1 depicts the designed masks. These masks were created trying to get a good prediction model, expecting the SVM to generalize to other possible cases that may appear during the filtering process. On the upper part of each mask is the assigned label that represents the desired output for that particular mask. For instance, the training vector for the first mask of Fig. 1, is $[\mathbf{X}; y] = [1,1,-1,1,1,-1,1,1,-1;1]$. The assigned label to this particular mask is "+1" which indicates that the white zone is generalized. Thus a vertical line is preserved during the filtering operation.

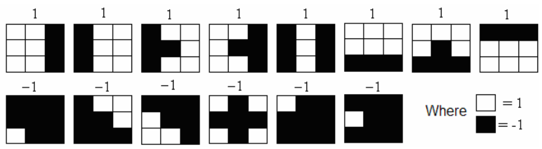

Figure 1. Example of SVM training mask.

5. APPLICATIONS

5.1. Impulsive noise removing using SVM filter

As a first application, the proposed filter is used in an image denoising task, more precisely to mitigate the impulsive noise in images. SVM filters are trained using masks like the ones show in Fig. 1, for the linear kernel of the type, $K(\mathbf{X}, \mathbf{Y}) = \langle \mathbf{X} \cdot \mathbf{Y} \rangle$ and a radial base function, $K(\mathbf{X}, \mathbf{Y}) = \exp(-|\mathbf{X} - \mathbf{Y}|/2\sigma^2)$.

The performance of the proposed filter is compared to the performance obtained using Center Weighted Median (CWM) filters [3]. The best CWM filter with minimum Mean Square Error (MSE) and Mean Absolute Error (MAE) is chosen and used in the comparison. Thus the performance of the SVM filter is compared to the performance of the best CWM filter.

Figure 2 shows the performance of the SVM filter with RBF kernel, for this case the parameter C changed from $C = 10^{-2}$ to $C = 10^2$ obtaining the best result for $C = 1$ and $\sigma = 10$. Figure 2 also shows a zoom-in of the image part enclosed by a rectangle. As can be seen, the SVM based filter has a better performance, eliminating impulsive noise efficiently, while preserving details and features present on the original image.

Figure 2. SVM filter performance. (a, e) original image (b, f) image with impulsive noise density of 10% (c, g) SVM filter output (d, h) CWM filter output. Bottom a zoom-in of the rectangular area.

The proposed approach was tested using a bank of images, both natural and artificial. Table 1 presents selected results of the MSE and MAE for the filtering process for a noise density of 10%.

It can be observed comparing the images and the errors values that the SVM filter shows a better performance compared to that yielded by the CWM filter. Table 1 reveals in a quantitative way what was presented in Fig. 2. The SVM filter not only removes impulsive noise effectively, but also keeps details well, acting as a competitive filter for the elimination of impulsive noise in images.

Table 1. Mean Square Error and Mean Absolute Error.

	MSE SVM Filter	**MSE CWM Filter**	**MAE SVM Filter**	**MAE CWM Filter**
Cameraman (linear kernel)	65.69	75.09	1.74	2.31
Cameraman (RBF kernel)	60.65	75.09	1.61	2.31
Circuit (RBF kernel)	48.49	49.13	1.25	1.27
Lena (RBF kernel)	43.05	46.19	2.18	2.71

5.2. Image edge detection based on SVMF

As a second application, an approach for edge detection is presented using a different point of view from the traditional [10]. In our case, we do not try to approximate the derivative or use other mathematical methods to detect edges in an image. The main idea, in this case, is to train the SVM filter to recognize the presence of edges in an image. This application derives from the design based on edge and structures to be preserved.

In order to get an appropriate generalization, a new set of 3x3 masks are designed. Figure 3 depicts some of the designed masks that are used to train the SVM corresponding to the decision structure filter. The masks are designed such that the edge information of the image is captured by the filter while the impulsive noise is removed. Three groups of training vectors were created to detect: vertical, horizontal and diagonal edges and they are used to train three SVM. The kernel functions used in the SVM is the RBF kernel.

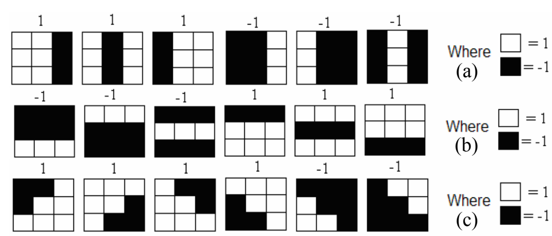

Figure 3. Example of SVM training masks to detect edges and elimination of noisy component. (a) horizontal edges, (b) vertical edges, (c) diagonal edges.

The designed masks allow that edge detection using the SVM filter can be generalized to any kind of image either non-noisy or noisy ones.

For a SVM filter denoted by Eq. (8), it is possible to define a function that permits to detect edges in images. SVM filter for edge detection output is defined as:

$$SVM_{EDGE} = \begin{cases} +1 & if \quad \sqrt{(E_H)^2 + (E_V)^2 + (E_D)^2} \geq Th \\ 0 & otherwise \end{cases} \qquad (10)$$

where E_H, E_V and E_D, correspond to the output of the three SVM filters, as in Eq. (8), trained using masks like the ones show in Fig. 3, that detect horizontal, vertical and diagonal edges, respectively. The value of *Th* is a tunable parameter that can be adjusted as a tradeoff between the among of edges to be detected and the noise immunity. High threshold values yield robustness to impulsive noise but lose some of the real edges, whereas low values result in many false edges induced by the impulse noise. The outputs of the three SVM are combined to find the total amount to which any edge exist, this value is the compared to the threshold *Th* to determine the existence of an edge.

The performance of the proposed edge detection approach is compared to those yielded by traditional edge detection methods, among them: Sobel, Canny, Prewitt and Roberts, which have been widely used in digital image processing [11].

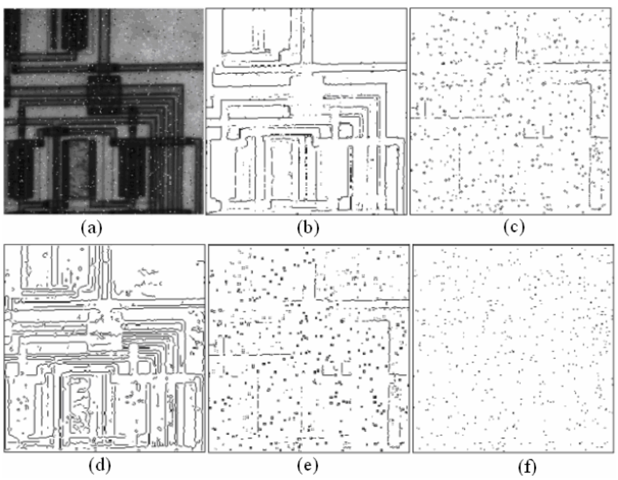

Figure 4. Edge detection in images (a) Original with noise 5%, (b) SVM filter, (c) Sobel edge detector, (d) Canny edge detector, (e) Prewitt edge detector, (f) Roberts edge detector.

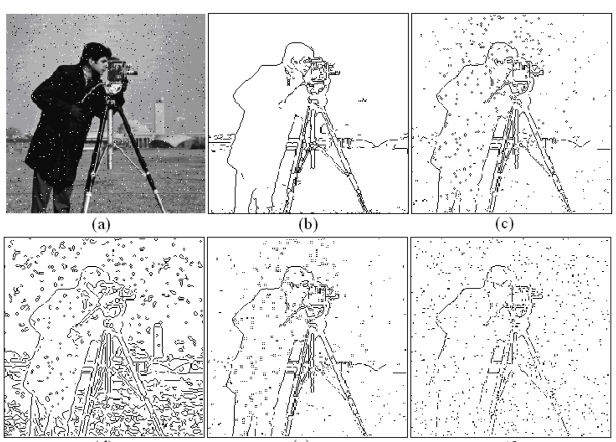

Figure 5. Edge detection in images (a) Original with noise 5%, (b) SVM filter, (c) Sobel edge detector, (d) Canny edge detector, (e) Prewitt edge detector, (f) Roberts edge detector.

Figure 4 shows SVM filter performance in edge detection in noise images. Edge detection using SVM filter is superior to the rest of the methods. The image obtained in Fig. 3 (b) reflects the generalization capabilities of the proposed method since SVM filters not only eliminate impulsive noise effectively, but also many details (edges) from the original image are suitably detected.

A second example that shows the performance of the proposed filter in edge detection in noise images is shown in Fig. 5, as before, it is easy to see the superior performance of the edge detection method using SVM filter compared to those yielded by traditional edge detection method. Note that most of the edges are preserved while the noise is removed.

6. CONCLUSSIONS

In this work, a new family of non-linear filters based on SVM is presented. The proposed filter is based on the general concept of binary filters where the characteristic function of the Boolean filter is replaced by a SVM. The obtained results for the new filtering approach show a better performance than traditional methods either in the noise elimination or the edge detection, behaving as a competitive filter for digital image processing.

7. REFERENCES

[1] Ki Dong Lee, Yong Hoon Lee "Threshold Boolean filters" IEEE Trans on Acoustics, Speech, and Signal Processing, vol. 42, no. 8 pp. 2022-2036, August 1994.

[2] José L. Paredes, Gonzalo R. Arce. "Stack Filters, Stack Smoothers, and Mirrored Theshold Decomposition" IEEE Transactions on Signal Processing, vol. 47, no. 10, pp. 2757-2767, Oct. 1999.

[3] L. Yin, R. Yang, M. Gabbouj, Y. Neuvo. "Weighted Median Filters: A Tutorial", IEEE Trans. on Circuits and Systems, vol. 43, no. 3, pp. 157-192, Mar. 1996

[4] Christopher J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". Microsoft Research (formely Lucent Technologies), 1998.

[5] N. Cristianini and J. Shawe-Taylor. "An Introduction to Support Vector Machines", Cambridge University Press, Cambridge, 2000.

[6] Y. LeCun, L. D. Jackel and V. Vapnik. "Learning algorithms for classification: A comparison on handwritten digit recognition". Neural Networks, pp. 261-276, 1995.

[7] H. Gómez Moreno. "Removal of Impulsive Noise in Images by Means of the Use of Support Vector Machines" Computer Science, pp. 536-543, 2003.

[8] Wendt, Peter D.; Coyle, Edward J. y Gallagher, Neal C. "Stack filters". IEEE Transactions on Signal Processing, vol. 34, no. 8, pp. 898-911, Ago. 1986.

[9] V. N. Vapnik, "Statistical Learning Theory", John Wiley & Sons Inc., New York, 1998.

[10] Hou, Z.J., Koh, T.S., "Robust edge detection". Pattern Recognition, vol. 36, no. 9, pp. 2083-2091, 2003.

[11] Jae S. Lim. "Two-dimensional Signal and Image Processing". Prentice Hall. New Jersey, 1990.