INISTA 2009
International Symposium on INnovations in Intelligent SysTems and Applications, June 29-July 1, 2009, TRABZON, TURKEY

Wavelet based Segmentation of an Object and Classification with Intelligent Systems

Nevrez İmamoğlu
TOBB ETU
Department of Electrical and Electronics Engineering
nimamoglu@etu.edu.tr

Aydın Eresen
TOBB ETU
Department of Electrical and Electronics Engineering
aeresen@etu.edu.tr

Mehmet Önder Efe
TOBB ETU
Department of Electrical and Electronics Engineering
onderefe@etu.edu.tr

Abstract

Segmentation and classification of an object is a popular research topic in surveillance and image-based systems. In this paper, segmentation is accomplished by using Gabor Wavelets. First and second order statistics of the segmented object is used as input features to classify the Region of Interest (RoI). Nearest-Neighbor, k-Nearest Neighbor, Bayesian classifier, Levenberg-Marquardt (LM) trained Artificial Neural Networks (ANNs), and Least-Squares Support Vector Machine (LS-SVM) classifier are used to classify the object of interest whether it is a car or not. It is seen that the LS-SVM approach is superior to its alternatives.

1. Introduction

Surveillance systems use the advantages of the segmentation and classification to be able to track objects. In advanced recognition systems, segmentation of foreground image is a difficult process to handle due to sensitivity of algorithms to noise and illumination changes. There are various algorithms to segment foreground objects from background image. Thresholding methods are the most common ways to segment objects, [1]. Region-based segmentation [1], segmentation using morphological watersheds [1], motion segmentation [1,2], wavelet-based segmentation [3] and normalized cuts segmentation [2,4] are some approaches available in the literature. In this paper, segmentation is implemented using Gabor wavelets on a real time system using a stationary camera. The advantage of utilizing wavelets is the fact that they decompose the input signal into its high and low frequency components where a set of different scales could be incorporated. This aspect of the wavelet decomposition makes it a preferable method in extracting features for post-processing, such as classification.

Classification is another problem in vision-based systems. The initial stage of the recognition process is feature extraction from segmented RoIs. Extracted features are the input vectors for the classification techniques. This paper uses first and second order statistics of the RoI which includes the segmented object. Some of the statistical learning methods are examined using these extracted features. Nearest-neighbor and k-nearest-neighbor [5,6] classifiers, generalized Bayesian classifier [5,6], Levenberg-Marquardt trained ANNs [7,8], and LS-SVM classifier [9] are applied to classify segmented objects to distinguish the object as being a car or not car.

The paper is organized as follows: Section 2 explains the segmentation and feature extraction. Section 3 summarizes the experimented classification techniques. Section 4 discusses the experimental results. Concluding remarks are given at the end of the paper.

2. Segmentation and feature extraction

Segmentation process is achieved by using Gabor wavelets, thresholding and morphological operations. The main aspect during segmentation is the Gabor wavelets which enable the process being fast and robust to illumination changes. Figure 1 presents the flow chart of the car classification system in this paper.

2.1. Segmentation using Gabor wavelets

Gabor wavelet algorithm is a commonly used method in pattern recognition and image processing applications for texture segmentation due to its robustness to illumination changes, further to this, it enables analyzing an image in different frequency scales [3]. Unlike the other common wavelets, Gabor wavelets do not cause downsampling on the image and it may include more frequency scales in one level transform.

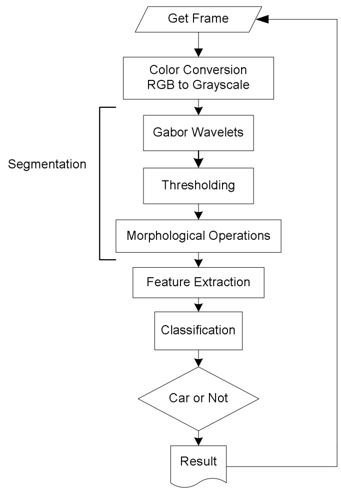

Figure 1. Flow Diagram of the Proposed System

As stated by Wei and Bartels [3], 2D Gabor wavelets can be represented as follows:

$$g_{\mu,v}(x,y)=\frac{\left\|k_{\mu,v}\right\|^2}{2\pi\sigma^2}e^{\frac{\left\|k_{\mu,v}\right\|^2\left(x^2+y^2\right)}{2\sigma^2}}\left[e^{ik_{\mu,v}(x+y)}-e^{\frac{\sigma^2}{2}}\right] \qquad (1)$$

where $k_{\mu,v}$ is the wave vector, μ is orientation and v is the scaling factor, σ is the Gaussian kernel parameter. In above, $k_{\mu,v}$ is computed as given in (Eq. 2).

$$k_{\mu,v}=k_v e^{i\phi_\mu} \qquad (2)$$

$$k_v=\frac{k_{\max}}{f^v} \qquad (3)$$

$$\phi_\mu=\frac{\pi v}{4} \qquad (4)$$

where $k_{max} = (\pi/2)$ and $f^v = \sqrt{2}$ is chosen as proposed in [3], and $v = \{0,1\}$ and $\mu = \{1,2,...,8\}$ are chosen in this paper.

After Gabor wavelet masks are created, image and masks will be convolved as follows:

$$I_{gabor}=I_{gray} * g_{\mu,v} \qquad (5)$$

Results for every orientation and scale factors are summed, and then Gabor wavelet represented image is obtained. Figure 2 shows four different frames captured from the stationary camera, and Figure 3 demonstrates Gabor wavelet representations of images captured.

Applying thresholding and some basic morphological operations on the Gabor wavelet representation of the frames, segmentation process is done by creating a mask on the segmented objects. In Figure 4, the mask images obtained from the frames in Figure 2 are shown.

Figure 2. Four Frames Captured by Camera (The images are cropped from original frames)

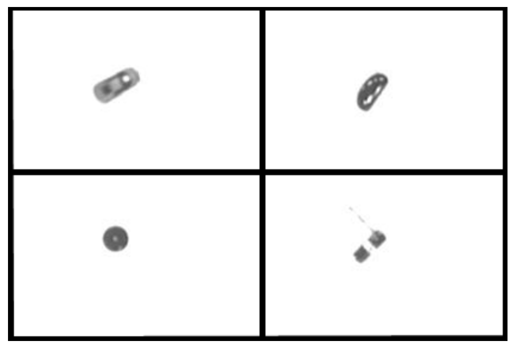

Figure 3. Gabor Wavelet Filtered Frames (The images are cropped from original results)

Figure 4. Masks of Segmented Objects

2.2. Feature extraction

The next step for segmentation process is the feature extraction from the masked images (See Fig. 4). In this paper, first and second order statistics of the images shown in Figure 4 are used to create the training data for classification. Borchani and Stamon [10] give the expressions for first and second order statistics, which are the features of the images. The first order statistics from image histogram are in (Eq. 6-9), [10].

$$M=\mu=\sum_{i=1}^{n} i h(i) \qquad (6)$$

$$SD=\sum_{i=1}^{n}(i-\mu)^2 h(i) \qquad (7)$$

$$TM=\sum_{i-1}^{n}(i-\mu)^3 h(i) \qquad (8)$$

$$E_1=-\sum_{i=1}^{n} h(i)\log\left(h(i)\right) \qquad (9)$$

where $h(i)$ is the image histogram, M is the mean of the histogram, SD is the associated standard deviation, TM is the third moment of the histogram and E_1 is the first order entropy.

The features obtained from the second order statistics are defined as given by (Eq. 10-20), [10].

$$H = \frac{1}{N_c^2} \sum_i \sum_j \left(M_c(i,j) \right)^2 \tag{10}$$

$$Cont = \frac{1}{N_c (L-1)^2} \sum_{k=0}^{L-1} k^2 \sum_{|i-j|=k} M_c(i,j) \tag{11}$$

$$E_2 = 1 - \frac{1}{N_c (L-1)^2} \sum_i \sum_j M_c(i,j) \log\left(M_c(i,j) \right) \tag{12}$$

$$Corr = \frac{1}{N_c \sigma_x \sigma_y} \left| \sum_i \sum_j (i - m_x)(j - m_y) M_c(i,j) \right| \tag{13}$$

$$m_x = \frac{1}{N_c} \sum_i \sum_j i M_c(i,j) \tag{14}$$

$$m_y = \frac{1}{N_c} \sum_i \sum_j j M_c(i,j) \tag{15}$$

$$\sigma_x^2 = \frac{1}{N_c} \sum_i \sum_j (i - m_x)^2 M_c(i,j) \tag{16}$$

$$\sigma_y^2 = \frac{1}{N_c} \sum_i \sum_j \left(i - m_y \right)^2 M_c(i,j) \tag{17}$$

$$LH = \frac{1}{N_c} \sum_i \sum_j \frac{1}{1 + (i-j)^2} M_c(i,j) \tag{18}$$

$$D = \frac{1}{N_c} \sum_i M_c(i,i) \tag{19}$$

$$U = \frac{1}{N_c^2} \sum_i M_c(i,i)^2 \tag{20}$$

where M_c is the gray level co-occurrence matrices and N_c is the sum of the elements of M_c, H is the homogeneity, $Cont$ stands for the contrast, E_2 denotes the second order entropy, $Corr$ is the correlation, LH is the local homogeneity, D denotes the directionality, and U is the uniformity.

3. Classification algorithms

In this paper, the studied classifiers decide whether the extracted object is a car or not. Eleven selected features, stated in the previous section, are used as the input vector for classification. In this section, nearest-neighbor techniques, Bayesian classifier, Levenberg-Marquardt trained ANNs, and LS-SVM classifier are tested comparatively.

3.1. Nearest-Neighbor Methods

Nearest-neighbor method is one of the simplest classification techniques available in the literature. The idea behind this method is to find the shortest distance to the test point from the training data using the Euclidian distance in the feature space. The test vector belongs to the class defined by the data vector such that the distance between them is minimal. Modified form of the nearest-neighbor method is known as k-nearest-neighbor method including the same philosophy with the nearest-neighbor method. The only difference is follows: instead of finding one nearest point, one should find k nearest points to determine the class of the test data. The value of k varies from two to an upper level integer value while its value is kept smaller than the number of training data.

Nearest-neighbor methods are easy to implement and have less computational complexity than many other classification techniques. However, to be able to use nearest-neighbor methods accurately, the training data in feature space has to be well distributed and rich in number. Also, the values in feature space should be normalized to increase the efficiency of the methods [5].

3.2. Bayesian Classifier

Like the nearest-neighbor methods, Bayesian classifier is also one of the popular methods for classification purposes. The difference is that Bayesian classifier uses some statistical features of entire training data set. In [6], Theodoridis and Koutroumbas state that in feature space Bayesian method classifies the data using the value of quantity defined in (Eq. 21).

$$g_i(x) = -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)^T \sum_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) + c_i \tag{21}$$

$$c_i = \ln P(w_i) - \frac{l}{2} \ln 2\pi - \frac{1}{2} \ln \left| \Sigma_i \right| \tag{22}$$

Assuming the probabilities of the different classes are equal, Bayesian classifier can be used by choosing the shortest Mahalanobis distance (Eq. 23) in feature space [6].

$$M_{\text{distance}} = \sqrt{ (\mathbf{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \sum_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) } \tag{23}$$

3.3. Artificial neural networks

ANNs have been a successful tool for applications where there are many variables influencing the result and many data having some degree of uncertainty. Aside from these, if the data is obtained from a nonlinear process or the model being developed needs a nonlinear map among the variables involved, it becomes a tedious task to handle such a modeling problem. On powerful alternative is to utilize connectionist structures storing the hidden implications available in the data within the synaptic connections. Although many structural forms of ANNs are available, in this work we utilize the feedforward structure. The other side of neural systems research concerns the development of tuning laws to optimize the parameters of a given ANN structure. Error backpropagation, known also as the gradient descent or MIT rule, was a popular approach in 1990s, the adoption of Levenberg-Marquardt technique to training of ANNs have make it a straightforward and fast process to obtain an accurately trained network structure. The process of learning is based on the mixture of the gradient descent and Newton's method as given below.

$$p_{next} = p_{now} - \left(\mu \mathrm{I} + J^{\mathrm{T}} J\right)^{-1} J^{\mathrm{T}} E \tag{24}$$

Where p is the parameter vector, J is the Jacobian defined as $J = \partial E/\partial p$, and E is the vector of errors. For small μ, the algorithm is like the Newton search and for large μ, it behaves like gradient descent. The current form removes the rank deficiency problem and yields very accurate results. For a thorough treatment one should refer to Lourakis and Argyros [7] and Hagan and Menhaj [11].

3.4. Support vector machines

SVMs, proposed by Boser et al [12], are useful tools for pattern classification applications. Unlike the traditional classification techniques, minimizing empirical training error, SVMs aim to minimize upper bound of the generalization error between nearest data and separating line and therefore SVMs have a better generalization performance than ANNs, whose training is based on the minimization of the generalization error itself and this is dependent upon the data being investigated.

SVM is a linear classification method assuming linear separability of the training data. Kernel trick maps the input data set to a hyperplane in which data set is linearly separable (See Figure 5) [13]. Despite the availability of many alternatives, e.g. spline based, polynomial or hyperbolic tangent type, radial basis function (RBF) kernel given by (Eq. 25) is used in this paper [14].

$$k\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \tag{25}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are input data vectors.

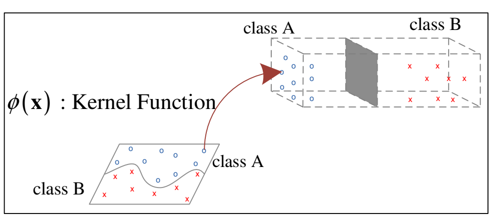

Figure 5. Mapping Dataset to Hyperplane Using Kernel Function

LS-SVM classifier, proposed by Suykens [9], is a modified form of Vapnik's SVM [12]. In Vapnik's SVM, a quadratic problem solver is used to find the optimal hyperplane. Quadratic problem solvers need long process times in most problems. In LS-SVM classifier, inequality constraints in errors are changes to equality constraints with a parameter determining the importance of the error terms. The primal form of the optimization problem is as seen in (Eq. 26), where the modification by Suykens [9] is visible.

$$\begin{aligned} &\min_{w,b,e} J_P(w,e) = \frac{1}{2} w^{\mathrm{T}} w + \gamma \frac{1}{2} \sum_{\mathrm{k}=1}^{\mathrm{N}} e_{\mathrm{k}}^2 \\ &st \quad : y_{\mathrm{k}}[w^{\mathrm{T}} \varphi(x_{\mathrm{k}}) + b] = 1 - e_{\mathrm{k}}, \mathrm{k}=1,\ldots,\mathrm{N} \end{aligned} \tag{26}$$

where N is the number of data pairs, e_k is the associated error value and γ is a parameter establishing the balance between the complexity and performance. If the dual form of the optimization problem is calculated using Lagrange multipliers, one gets the matrix equality in (Eq. 27),

$$\begin{bmatrix} 0 & \mathbf{y}^{\mathrm{T}} \\ \mathbf{y} & \mathbf{\Omega} + \mathrm{I}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \tag{27}$$

where $\mathbf{y}$ is the vector of target values (the desired class label), I is the identity matrix of appropriate dimensions, γ is LS-SVM classifier parameter, b is the bias, $1_v = [1,\ldots,1]$, $k(\mathbf{x}_k,\mathbf{x}_l)$ is the kernel function and $\mathbf{\Omega}_{kl} := y_k y_l k(\mathbf{x}_k,\mathbf{x}_l)$. Two class decision function for this setting is given by (Eq. 28),

$$y(x) = \mathrm{sgn}\left(\sum_{\mathrm{i}=1}^{\mathrm{N}} \alpha_{\mathrm{k}} y_{\mathrm{k}} k(\mathbf{x}, \mathbf{x}_{\mathrm{i}}) + b\right) \tag{28}$$

4. Experimental results

The classification algorithms are tested using 11 input features which are the first and second order image statistics extracted from RoIs. The aim is to classify the segmented object as a car, say class C, or not a car denoted by class NC. All stated classification methods give output +1 if the segmented object belongs to class C; otherwise the output is −1 meaning that the object belongs to NC. 100 images from set C and 60 images from set NC are used to learn the features for classification in all classifiers except for LM trained ANNs. In LM trained ANN classifier, 80 data pairs from set C and 45 pairs from the set NC are used to train the system, and 20 data points from set C and 15 points from NC are used for validation of the trained ANN. In all classifiers, 103 data pairs are used to test the systems. These pairs include 67 instances belonging to set C and another 36 belonging to set NC. The accuracy of the test results are shown in Table 1.

Nearest-neighbor, 3-nearest-neighbor and 5-nearest-neighbor algorithms result in unsatisfactory performance with respect to Bayesian, LM trained ANN and LS-SVM classifiers. Nearest-neighbor algorithms have poor performance with inputs more than 2. To be able to use the nearest-neighbor and k-nearest-neighbor methods efficiently, the dimensions must be normalized and number of training data must have a sufficient distribution over the feature space. *k*-nearest-neighbor, in fact, should give better performance, however with given feature space k-nearest-neighbor methods gave least performance due to the fact that the stated conditions are not satisfied.

Table 1. Classification Results

	#Correct Classification	#Misclas.	Percent Accuracy
Nearest-Neighbor	85	18	82.52%
3-Nearest-Neighbor	80	23	77.67%
5-Nearest-Neighbor	83	20	80.58%
Bayesian	100	3	97.09%
LM	96	7	93.20%
LS-SVM	103	0	100.00%

LM trained ANN algorithm gives better performance than nearest-neighbor and k-nearest-neighbor methods with 93.20% accuracy. However, computational complexity is high, and finding the best performing network structure requires considerable a trial-and-error process.

Bayesian classifier using Mahalanobis distance has reliable results with 2.91% error rate in total. Computational complexity is less than LM trained ANN and LS-SVM classifiers with obtained feature space. Bayesian method may lead to a better performance if the number of points in the feature space is increased and a better distribution is observed.

LS-SVM classifier has demonstrated the best performance with no misclassifications. An increase in the number of training data in LS-SVM increases the computational complexity more than that of the Bayesian classifier.

5. Conclusions

This paper compares six methods to classify objects in a given image to find a car in the RoI. The RoI is segmented from the given image by utilizing Gabor wavelets. The results obtained after the tests emphasize that nearest-neighbor and k-nearest-neighbor methods result in relatively poor performance with given inputs and feature definitions. LM trained ANN gives better accuracy than nearest-neighbor methods; on the other hand, its performance is poorer than Bayesian and LS-SVM classifiers. Bayesian classifier and LS-SVM have the best performances, and LS-SVM provides 100% accuracy with given input data. This result is attributed to the fact that SVM approach transforms the problem into an optimization problem and utilizes the conventional approaches to achieve the minimum.

6. Acknowledgements

This work is supported by TÜBİTAK grant no 107E137. The authors acknowledge the facilities of the Unmanned Aerial Vehicles Lab of TOBB ETÜ.

7. References

[1] R. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey, 2008.

[2] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2003.

[3] H. Wei and M. Bartels, "Unsupervised Segmentation Using Gabor Wavelets and Statistical Features in LIDAR Data Analysis", *18th Int. Conf. on Pattern Recognition*, 2006, Vol. 1, pp. 667-670.

[4] J. Shi and J. Malik. "Normalized Cuts and Image Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000, Vol. 22 No. 8, pp. 888-905.

[5] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2003.

[6] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, Elsevier USA, 2006.

[7] M.I.A. Lourakis and A.A. Argyros, "Is Levenberg-Marquardt the Most Efficient Optimization Algorithm for Implementing Bundle Adjustment?", *Proc. of the 10th IEEE Int. Conf. on Computer Vision*, 2005, Vol. 2, pp. 1526-1531.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, New Jersey, 1999.

[9] J. A. K. Suykens, T. G. Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Publishing Co. Pte. Ltd., Singapore, 2002.

[10] M.N. Sarlashkar, M. Bodruzzaman and M.J. Malkani, "Feature Extraction Using Wavelet Transform For Neural Network Based Image Classification", 1998, *IEEE Proc. of the 13th Southeastern Symposium on System Theory*, pp. 412-416.

[11] M.T. Hagan and M.B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm", *IEEE Trans. on Neural Networks*, 1994, Vol. 5, No. 6, pp. 989-993.

[12] B.E. Boser, I.M. Guyon and V. Vapnik, "A training algorithm for optimal margin classifiers", *In 5th Annual Workshop on Computational Learning Theory*, ACM, Pittsburgh, 1992.

[13] B. Schölkopf, and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond (Adaptive Computation and Machine Learning), The MIT Press, Cambridge, Massachusetts, London, England, 2002.

[14] J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery 2*, 1998, Vol. 2, pp. 121-167.