

# Object Recognition Based on Radial Basis Function Neural Networks: experiments with RGB-D camera embedded on mobile robots

Saeed Gholami Shahbandi

LISA - University of Angers

62 av. Notre Dame du Lac, 49000 Angers, France

saeed.gholami-shahbandi@eleves.ec-nantes.fr

Philippe Lucidarme

LISA - University of Angers

62 av. Notre Dame du Lac, 49000 Angers, France

philippe.lucidarme@univ-angers.fr

**Abstract**—an object recognition strategy based on artificial radial basis functions neural networks is presented in this paper. The general context of this work is to recognize object from captures made by a mobile robot. Unlike classical approaches which always select the closest object, our method outputs a set of potential candidates if the input information is not enough discriminant. There are three main steps in our approach: objects segmentation, signature extraction and classification. Segmentation is inspired from previous works and is shortly described. Signature extraction based on global geometric and color features is detailed. Classification based on artificial neural networks is also explained and architecture of the network is justified. Finally a real experiment made with a RGB-D camera mounted on a mobile robot is presented and classification results is criticized.

## I. INTRODUCTION

Object recognition is a stimulating task for many applications, especially in the field of robotics where the interaction between robots and their environment becomes a very challenging problem. Current approaches of object recognition can be categorized by level of extraction : local features extraction are based on local neighborhood operations [2] [4] while global feature extraction considers the input data in its globality [8] [12]. Some novel approaches are based on combination of local and global features [3]. The presented work clearly belongs to global techniques.

Whether local or global, classical approaches usually involve a final classification stage that selects the best object. A large amount of classifiers can be found in the literatures. A few of the best known are support vector machine [15] [10] [16], naive Bayes [14] [11], k nearest neighbors [5], artificial neural networks [7] [6]... Most of the existing classifiers assume that input data can be separated (linearly or not) and the clusters to classified are not overlapped. The fact is that in practice, and especially in the framework of a mobile robotics, input data can be non-discriminant for classification, partially due to light-dependant input data, scene occlusion or merged objects. This usually results in a wrong classification. Two kind of mistakes can be identified :

- *False Positive* error occurs when an object that should not been classified is recognized.

- *False Negative* error occurs when an object that should been recognized is not classified.



Fig. 1: Illustration of objects used for classification

The aim of the presented work is, of course, to output the best object when it is possible. But if there is an uncertainty on the input data, this ambiguity must be detected and considered in the following steps of the system. This goal is achieved by “*Multi Labeling*”, In other words, the presented object recognition method does not always say “this is a chair”, but provide a more fine answer: “this is probably a chair, but it may also be a coffee table or a stool, therefore it is not a ball, a plant or a box”. Such an approach is unlike any other multi-class classifiers presented in the literatures. In both *One-Versus-All* and *All-Versus-All* strategies only one label is assigned to each input object. In proposed method number of assigned labels may varies from “0” to “number of classes”, this is what will be called *Multi-Label* in the following. This point of view may be disturbing for a person skilled in the art. Therefore, the reader has to consider the fact that this work takes place in a mobile robotics context; if the result of classification is ambiguous, the robot can move and observe the object from another viewpoint. By crossing and merging data, it can disambiguate the result.

The proposed classifier is composed of several levels as illustrated on the Fig. 2. Input data is provided by a RGB-D camera, the segmentation level separates objects from each other and from the background as explained in section III. For each potential object, a global feature vector is computed. This vector is a kind of distinctive signature of the object representative of it shape, color and size. Details of the features

is described in section IV. The final stage of the presented work is its classification level. Artificial neural networks has been chosen for their ability to deal with a large amount of data, generalization capabilities and a few number of parameters have to tune. Moreover *Radial Basis Function* (RBF) networks allow a gaussian approximation of each sub part of the search space which is consistent with the hypotheses of a normal distribution of the clusters. Details of the RBF network can be found in section V. Experiments have been performed with a real mobile robot, described in section VI-A, with a learning set of ten objects from everyday life. Some objects have been voluntary chosen very close to each other in order to highlight the uncertainties (see Fig. 1 and section VI-B). Two sets of parameters have been compared and performances of the methods are discussed in section VI-D.

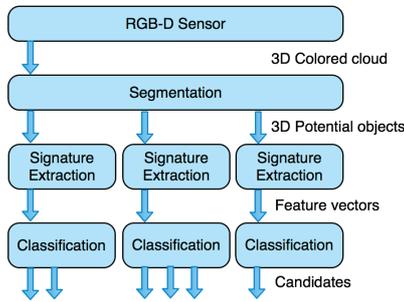
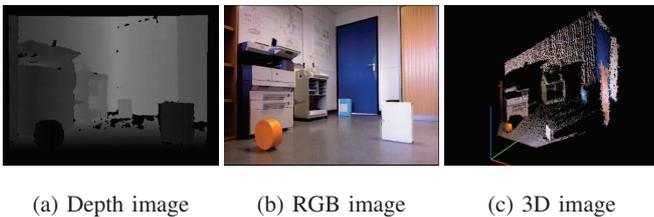


Fig. 2: Workflow

## II. RGB-D CAMERA

Novel sensors called *RGB-D cameras* are coming to attention more and more everyday, because of rich information they provide from a scene with a single shot. They provide “RGB” image and letter “D” stands for *depth* image, this combination makes the acquisition potentially a 3D colored image as illustrated in Fig. 3c. A popular solution for realization of such a device is the *KINECT*<sup>TM</sup>.



(a) Depth image (b) RGB image (c) 3D image

Fig. 3: Raw output of the RGB-D camera, and illustration of 3D image forged by combination of depth and color information. Calibrating the kinect with a fixed frame on the robot, each point in depth map will turn to a 3D Cartesian coordinate. A 3D colored image will be constructed by assigning those coordinates to corresponding pixels in RGB image. Thus one point of the 3D cloud is composed of six parameters; X, Y, Z, R, G and B.

## III. SEGMENTATION

Acquired data from the scene contains depth information auxiliary to color information which makes the isolation of the objects easy. In this work dependency of segmentation algorithm is on the depth information, those points connected to each other without a gap (Fig. 4a) make a region which is possibly an object (Fig. 4b).



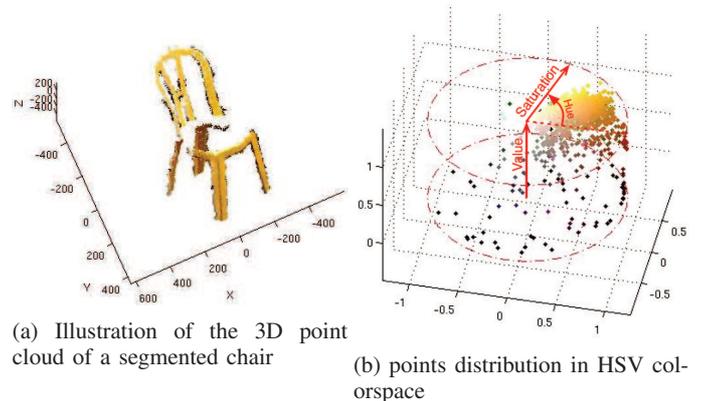
(a) Segmented regions (b) Isolated possible objects

Fig. 4: An example of object *detection* process of a similar scene illustrated in Fig. 3

All isolated regions will be considered potentially as an objects until they are assessed in the classification phase.

## IV. FEATURE SELECTION

The intrinsic of the RGB-D cameras bring the global features of an object to attention, hence only global features have been studied and exploited during this work. Two general sets of information are at hand for every single point of an object, *position* and *color* as demonstrated in Fig. 5, both in 3 dimensional space. The color space is chosen to be *HSV* after there was a transformation from RGB to HSV, that’s because after several attempts on different color spaces such as *RGB*, *YCbCr*, etc. and even their modifications, none were as distinguishing as *HSV* which is known to be more light-independent. Note that during all the calculations the *Cylindrical Coordinate System* of HSV colorspace must be taken to account.



(a) Illustration of the 3D point cloud of a segmented chair (b) points distribution in HSV color space

Fig. 5: Illustration of two set of information obtained via Kinect

Based on those two spaces, four global feature sets are proposed to be extracted; *i)*size of minimum bounding box,

ii)variance of mass, iii)average color and iv)variance of color.

PCA (*Principal Component Analysis*) is well known to be a popular feature extracting method [9] both in conventional recognition algorithms and Neural Network based algorithms. PCA calculation of position of the points provides variance of mass presenting a significant characteristic of the object. In a very similar way the variation of points in color space could be acquired. *Size* of the object is obtained by finding the smallest bounding box embodying the object and having PCA of mass in the hand, alignment of the Principal Components with axis of coordinate system will solve the problem. Finally *average color* of the object is obtained via a simple averaging of all points. Thus four 3D vectors are realized as global features, to be employed for developing a classifier in the next step.

## V. RBF NETWORKS

*Radial Basis Function* networks (Fig. 6) have been employed for several purposes from function estimation to control. RBF networks are composed of 3 layers of neurons; *input* layer, *hidden* (RBF) layer and *output* layer.

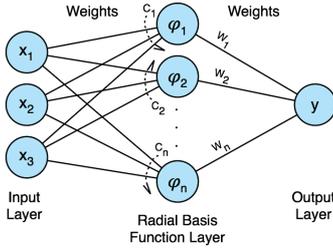


Fig. 6: Single RBF Network

Those weights from input layer to hidden layer are not multiplicative but subtractive representing the centers of RBF functions, where output weights are multiplicative in role of amplitudes of RBF functions. Mathematic definition of the network illustrated in Fig. 6 is expressed by equation (1).

$$y = RBFN(X) = \sum_{j=1}^n (w_j \times \varphi(X - C_j)) \quad (1)$$

In equation (1),  $y$  is the output of the network,  $X$  represent input vector (containing  $x_1, x_2, \dots$ ),  $RBFN$  is the functionality of network resulting  $y$ ,  $w_j$  stands for those weights from hidden layer to output node,  $\varphi$  represents the RBF functions in hidden layer and finally  $C_j$  represent the center of RBF functions.  $\varphi$  functions only differ in the position of center and amplitude.

Purpose of this work is to estimate distributions of different classes in each feature space (e.g. size, color, etc.). Distinctive RBF networks have been employed for this estimation, where *Gaussian* functions are recruited in hidden layers as RBF functions and that is expressed in equation (2).

$$\varphi(X - C_j) = \exp\left(-\frac{\sum_{k=1}^K (x_k - c_{jk})^2}{2\sigma^2}\right) \quad (2)$$

In equation( 2),  $\varphi(X - C_j)$  is the output of Gaussian function in hidden node  $j$  and  $C_j$  are those weights from input layer to that hidden node representing the center of Gaussian function,  $K$  is the dimension of input space, and  $\sigma$  is the standard deviation of the Gaussian function. The output of each network which represents the probability of a point being included in that specific distribution, will be followed by thresholding function at 0.5 to provide a binary output.

### A. Learning algorithm

There is a classical method for training RBF networks *C. M. Bishop*[17] which is only *input-driven*. *G. Bugmann* presented [13] a modified training algorithm (Algorithm 1) which is both *input-driven* and *output-driven*. With help of modified training algorithm the number of nodes in hidden layer decreases (i.e. less resources) without compromising the performance. The process of introducing training points to the network is iterative in Algorithm 1.

---

#### Algorithm 1 Modified Training Algorithm for a Gaussian RBF Network (GRBFN)

---

```

for "Epoch" times do
  for all  $point_i \in$  Training Dataset do
     $error = output^{desired} - GRBFN(point_i)$ 
    GRBFN is the network under current training process.
    if  $error < tolerance$  then
      Do Nothing.
    else
      Find the closest Gaussian function( $\varphi$ ) in the hidden
      layer and compute it's distance to  $point_i$ .
      if  $distance < threshold$  then
        Update that closest node.
      else
        Recruit a new node in hidden layer centered on
         $point_i$ .
      end if
    end if
  end for
end for

```

---

*Recruiting*: to generate a new node in hidden layer, where the input weights (center of RBF) are initialized by the introduced input itself, and the output weights will be initialized by 1, to be updated in later epochs.

*Updating*: as input weights represent centers of Gaussian functions, they will be updated toward the introduced input (equation 3), while output weights behave as amplitudes of Gaussian functions and they will be updated considering the error occurred in output (equation 4).

$$c_j(t + 1) = 0.8c_j(t) + 0.2x \quad (3)$$

$$w_{ij}(t + 1) = w_{ij}(t) + learnrate \times (y_i^{desired} - y_i^{real}) \quad (4)$$

As *G. Bugmann* proposed, to ensure covering the area between training points, the variance of Gaussian function ( $\sigma$ ) should be close to the “half of the largest distance between nearest neighbors” as in equation (5) [13].

$$\sigma \rightarrow \frac{1}{2} \text{MAX}_{i=1}^{N^{obj}} \left( \min_{\substack{j=1 \\ j \neq i}}^{N^{obj}} \left( \text{distance}(p_i^{obj}, p_j^{obj}) \right) \right) \quad (5)$$

One of the important characteristics of this network along with presented training algorithm is its independency of *negative reinforcement*. This characteristic will help to set up independent classifiers for different classes as it will be explained in next section (V-B).

### B. Classification with RBFs

Membership of an object in different classes will be evaluated by individual classifiers unbiased by other classes (Fig. 7). Proposed method for classification here is a multi-label assigning scheme in which every input will be assigned with either 0, 1 or even more labels at the same time (Fig. 8), while only one of them is correct.

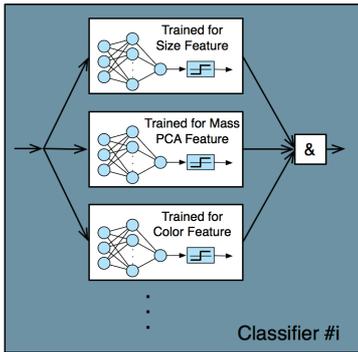


Fig. 7: A classifier trained with objects from *Class#i*

As explained in introduction two type of error may occur: *i) False Negative*: when an object is *not* assigned with correct label and *ii) False Positive*: where the object is assigned with correct label plus extra labels (*Multi-Labeling*). Although there should be a comprise between these two types of error, but it should be mentioned that the *False Positive* is remediable with further phase after classification relying on local features of the object.

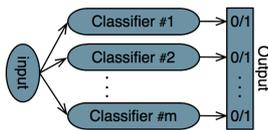


Fig. 8: Multiple label assignment schema

To verify membership of an object in a specific class, the object will be examined in all different feature spaces (e.g. size, color, etc.) separately by distinctive RBF networks in each feature space. Each network is trained in one feature

space and membership will be confirmed if all networks in different feature spaces approve. Key idea of the classification is expressed in Algorithm 2.

---

### Algorithm 2 Classification

---

```

for all Classes do
  membership = 1;
  for all Feature Sets do
    Introduce the object to the specific RBF network
    trained for current “class” and current “feature set”.
    if  $RBF(object) < 0.5$  then
      membership = 0;
    end if
  end for
  if membership == 1 then
    Assign the label of current class to object.
  end if
end for

```

---

Although proposed algorithm in current layout picks multiple labels for each object, makes it more disposed as a pre-classifier demanding a further stage of accurate decision, it could be adjusted to operate stand alone if the feature search space is discriminant enough. Such an adjustment takes place by ignoring the “*thresholding by 0.5*” operation mentioned earlier in this same section (V. RBF Networks), where analog outputs of the RBF networks stand for the probabilities of an object being a member of each class. Available probabilities makes it possible to pick the highest as winner.

## VI. EXPERIMENT

### A. MiniRex

This work takes place in a larger project called *Cart-O-matic*. Our team was involved in a robotics competition (*Défi-CAROTTE*) founded by the French Research Agency (ANR) and the General Delegation for Armaments (DGA). The aim of this contest was to map and locate objects in a structured environment similar to an apartment. The particularity of our team was the use of a multi-robot strategy [1]. During the navigation, an object is usually seen from several viewpoints by many robots (Fig. 9). This also justify our choice of a classifier that can provide several candidates as output, by crossing information from each robots, ambiguities can usually be solved. Our team designed and built seven identical mobile robots called *MiniRex* (MINIature Robot for Exploration) illustrated in Fig. 9. Each robot is composed of an Embedded PC (proc. Atom 1.6GHz), inclinometer and ultrasonic sensors for navigation, LIDAR for localization and mapping, and an RGB-D sensor (Microsoft Kinect) for object recognition. The built-in actuator of the Kinect has been locked and the sensor was calibrated with a new reference frame merged with the robot’s coordinate system.

### B. Database

Ten classes of objects (Fig. 1) are picked out of those proposed by *Défi-CAROTTE*. Initially a very small database was

constructed by capturing objects with the robots as illustrated on Fig. 9, and eventually grew bigger by appending those captures from several missions launched during the process of development, that is to say the current version of the database contain real case data and the results are very reliable.

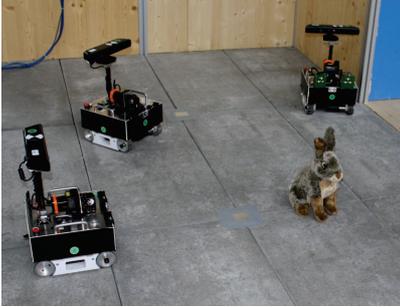


Fig. 9: Capturing an object with several robots

The database was randomly distributed into *Training* and *Testing* datasets, delivering more to the training set. TABLE I presents all classes as well numerating the objects included.

Class	Number of Items in the dataset		
	Total	Training	Testing
Black Chair	57	31	26
Wheeled Chair	170	116	54
Yellow Chair	121	86	35
White Box	94	65	29
Blue Box	62	39	23
Orange Cylinder	87	50	37
Green Cylinder	71	42	29
Fan	85	50	35
Red Ball	74	39	35
Plant	85	43	42
OVERALL	906	561	345

TABLE I: DataBase

### C. Parameters settings

As mentioned in the section V-A the parameter  $\sigma$  should be close to the “half of the largest distance between nearest neighbors”[13]. Based on this hypothesis estimated  $\sigma$  for each network comes in TABLE II. In this experiment due to inadequate number of training points and non-exhaustive dataset,  $\sigma$  will be a multiplication of the estimated value in order to expand the frontier of the distribution’s estimation. In configurations (TABLE III) a coefficient to these  $\sigma$  values will be adjusted.

*Bugmann* proposed in [13] values for those variables mentioned in Algorithm 1 (*learnrate*, *threshold*, *tolerance* and Number of “*epochs*”) as well as the proposal for  $\sigma$  estimation. Based on experiments turned out only the  $\sigma$  affects the performance of the classifier critically, therefore performance of classifier has been assessed by observing the errors while all parameters except the coefficient of  $\sigma$  were adjusted to those values of TABLE III and coefficient of  $\sigma$  was varying from 0.2 to 8.

	Estimated $\sigma$ in distinctive networks			
	Size	Mass PCA	Color	Color PCA
Black Chair	50.00	15.49	4.03	4.22
Wheeled Chair	50.00	20.28	3.34	3.96
Yellow Chair	50.00	20.11	3.92	2.51
White Box	35.71	8.68	15.22	14.51
Blue Box	25.71	3.75	5.42	5.14
Orange Cylinder	39.78	7.56	7.25	8.83
Green Cylinder	11.24	2.67	10.41	10.46
Fan	49.38	16.10	13.40	13.09
Red Ball	12.89	3.77	15.10	13.45
Plant	23.13	6.50	2.76	3.38

TABLE II:  $\sigma$  Estimation; As proposed in section V-A the estimation of  $\sigma$  is based on the values of points in training dataset. Size of an object in any dimension varies from  $2.0 \times 10^{-4}$  to  $1.05 \times 10^3$  with an average of 327.2, similarly variance of mass varies from  $7.0 \times 10^{-5}$  to  $2.83 \times 10^2$  with an average of 77.5, and limitations of color space would be  $0^\circ < H < 360^\circ$ ,  $0 < S < 1$  and  $0 < V < 1$

Coefficient of $\sigma$	1st Configuration	2nd Configuration
		2.1
Tolerance		0.15
Number of “Epoch”		5
Learning rate		0.51
Threshold		$0.5 \times \sigma$

TABLE III: RBF Network Configurations

### D. Results

Fig. 10 presented results of observation proposed after setting the parameters in subsection VI-C.

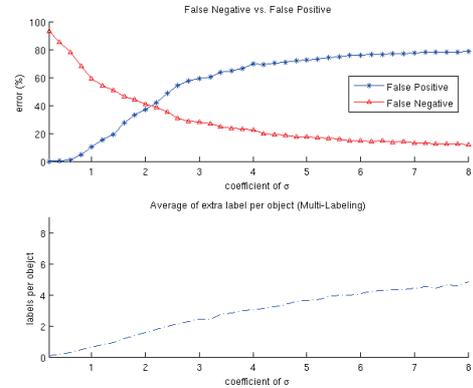


Fig. 10: Performance of classifier with different  $\sigma$  values. In the first plot two types of error are compared while coefficient of  $\sigma$  increased from 0.2 to 8, under the same circumstance average number of multi label per object is illustrated in second plot.

Increment of  $\sigma$  which is the width of Gaussian functions recruited in RBF networks will cause an expansion of the distribution’s estimation. As it can be seen in Fig. 10 the consequence of increasing  $\sigma$  is the decrement of false negative and increment of false positive. Less false negative error is desirable, but it should be considered that at the same time multi labels per object is increasing and this will cause a

post-processing computational cost. Therefore there should be a trade off between false negative error and number of multi labels per object with respect to the application and available resources. Assessment have been followed in details by picking two values for the coefficient of  $\sigma$ , presented in TABLE III and detailed results are available at TABLE IV.

Class	Items	1st Configuration		2nd Configuration	
		C R	F P	C R	F P
Black Chair	26	8	4	17	17
Wheeled Chair	54	23	12	43	34
Yellow Chair	35	20	0	30	7
White Box	29	22	17	26	26
Blue Box	23	8	7	18	18
Orange Cylinder	37	28	20	35	35
Green Cylinder	29	23	11	28	28
Fan	35	20	18	34	34
Red Ball	35	29	24	34	34
Plant	42	28	22	39	39
<b>TOTAL</b>	<b>345</b>	<b>209</b>	<b>135</b>	<b>304</b>	<b>272</b>
Correct Recognition (%)		60.57%		88.11%	

TABLE IV: Classification results (CR: Correct Recognition, FP: False Positive). Correct recognition represents those objects assigned with correct label and false positive represents number of correctly recognized objects with multi-labels

These results confirm the influence of the  $\sigma$  parameter. In the first configuration 209 objects have been recognized correctly and 135 (65%) of them have several candidates (multi labels). The second configuration offers a better classification rate (304 objects correctly recognized over 345), but the rate of multi labels is higher (89%). For a better understanding of these results, we must keep in mind that *False Positive* is definitively not classification error. It must be considered as extra information: *“It’s most probably a black chair, but it is not excluded to be a wheeled chair”*.

## VII. CONCLUSION

Most interesting features of proposed classifier are the “multi-labeling” mechanism and un-necessity of “negative reinforcement”. These features provide classifiers independent from each other and capable of handling uncertainty. Proposed classifier as well has advantage of fine tuning carrying out an adaptive system for variety of feature sets and also other applications. Dynamics, high speed and low computational cost of the method makes it suitable for integration into a more exhaustive recognition system as a pre-classifier in order to decrease computational costs in supplementary accurate recognition stage.

As described in V-B(Classification with RBFs) it’s possible to alter the layout of the network to produce analog outputs, which will be exploited by a simple inference system picking the most probable label considering the possibility of a rejection for those objects not belonging to any predefined classes.

Those feature sets studied and proposed here have advantage of simplicity in extraction without compromising discriminative characteristic of global features.

Designed RBF network is an ensemble of Gaussian functions all having the same standard deviation ( $\sigma$ ) in all direc-

tions, makes them homogeneous Normal Distribution functions with different amplitudes. This is compromising the precision of rendering an estimation of the distribution, as the resulting RBF network would not predict the characteristic of distribution but only estimating the profile of it, as well even for such a goal it needs a considerable number of nodes in hidden layer. Further development of this proposal would be realizing a network recruiting Gaussian functions with independent variance ( $\sigma$ ) in different directions taking to account their orientation.

## ACKNOWLEDGMENT

This work has been partially supported by the *French National Research Agency (ANR)* and *General Delegation for Armaments (DGA)* through the *Cart-O-matic* project in the *CAROTTE* challenge.

## REFERENCES

- [1] A. Bautin, O. Simonin and F. Charpillet, *Towards a communication free coordination for multi-robot exploration*, CAR 2011, 6th National Conference Control Architecture of Robots, May 2011, Grenoble, France, 8p.
- [2] H. Bay, A. Ess, T. Tuytelaars, L. Gool, *SURF: Speeded Up Robust Features*, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 2008, .
- [3] L. Wang , J. Shi , G. Song and I-fan Shen *Object Detection Combining Recognition and Segmentation*, ACCV’07 Proceedings of the 8th Asian conference on Computer vision, Part I, pp 189-199, 2007.
- [4] J. Mutch and D. G. Lowe, *Multiclass Object Recognition with Sparse, Localized Features*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p.11-18, June 17-22, 2006.
- [5] Kilian Q. Weinberger, John Blitzer and Lawrence K. Saul, *Distance metric learning for large margin nearest neighbor classification*, In NIPS, MIT Press, 2006.
- [6] L. Chen, *Pattern Classification by Assembling Small Neural Networks*, IEEE International Joint Conference on Neural Networks, IJCNN ,vol. 3, pp. 1947-1952, 2005.
- [7] K. Fukushima, *Neural Network Models for Vision*, International Joint Conference on Neural Networks, IEEE, vol. 4, pp. 2625-2630, 2003.
- [8] S. Hinz, *Integrating Local and Global Features for Vehicle Detection in High Resolution Aerial Imagery*.ISPRS Archives, Vol. XXXIV, Part 3/W8, Munich, 17.-19. Sept. 2003.
- [9] G. P. Zhang, *Neural Networks for Classification: A survey*, IEEE Trans. Systems, Man and Cybernetics - Part C: Application and Reviews, vol. 30, No. 4, pp 451-462, 2000.
- [10] O. Chapelle, P. Haffner, V. N. Vapnik, *Support vector machines for histogram-based image classification*, IEEE Transactions on Neural Networks, vol. 10, Issue. 5, pp. 1055-1064, 1999.
- [11] D. J. Moore, I. A. Essa, M .H .Hayes, *Exploiting human actions and object context for recognition tasks*, The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1 ,pp. 80-86, 1999.
- [12] C R. Shyu, et al, *Local versus Global Features for Content-Based Image Retrieval*, IEEE Workshop on Content-Based Access of Image and Video Libraries, 1998,
- [13] G. Bugmann, *Normalized Gaussian Radial Basis Function Networks*, *Neurocomputing* [special issue on Radial Basis Function Networks] vol. 20 , pp 97110, 1998.
- [14] B. Krebs, F. M. Wahl, *Automatic generation of Bayesian nets for 3D object recognition*, Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 126-128, 16-20 Aug 1998.
- [15] Christopher J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Appeared in Data Mining and Knowledge Discovery Vol. 2, pp. 121-167, 1998.
- [16] E. Osuna, R. Freund, F. Girosit, *Training support vector machines: an application to face detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130-136, 1996.
- [17] C. M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, Oxford, UK, pp 170, 1995.