Robust object Recognition Using SURF feature model applied in NAO Robot

Vahid Rahmani , Vahid Rostami
Department of Electrical, IT and Computer sciences ,Qazvin Branch,Islamic Azad University, Qazvin, Iran

Abstract— Object recognition is one of the most important processes for robot soccer in the standard platform robot league. Main task of the vision system of robot soccer while playing is recognizing and tracking the objects like ball, goalposts, robots of the same team and rival robots in the game. The basic idea of several object recognition methods, especially in robot soccer and RoboCup environment, is to use the algorithms based on the color feature of pixels. One of the significant challenges in color feature-based algorithms is the illumination changes of an environment. Since colors are different in different illumination conditions and are influenced by environmental factors like noise, the vision system of a robot will be disturbed if the environment's illumination changes. Therefore, in this paper a robust object recognition approach against illumination changes of an environment is proposed by using the matching algorithm based on SURF feature. Experimental results on 5 different datasets obtained from the top 5 teams of the world shows that the proposed approach has a good result which includes a recognition rate of more than 96% and an error rate of 4%.

Index Terms— Humanoid Robot, Standard Platform League, NAO, Vision System, Neural Network
.

1 INTRODUCTION

RoboCup, "Robot Soccer World Cup", is an international project to promote artificial intelligence, robotics and other related areas. The RoboCup Standard Platform League (SPL) is one of several active soccer leagues in RoboCup competitions in which the humanoid NAO is used. As the name indicates, in RoboCup Standard Platform League, all robots and rules are identical for all participating teams as a fixed and standard platform. In Standard Platform League [1], all participating teams are only allowed to compete using the humanoid NAO. In this RoboCup league, robots play soccer fully autonomously and without interference of human. In these competitions held annually by international committee of roboCup, rules, regulations and conditions of competition environment are identical for all participating teams [2]. According to these rules, pre-determined color and size of all objects on the field are described. For example, the ball is red, the goals are yellow, the teammate robot clothing is red and opponent robot clothing is blue, and these colors will be fixed throughout the whole match. While playing soccer, the main task of the robot vision system is identifying and tracking objects such as ball, goals, teammate robot and opponent robot. The basis of many object identification methods, particularly those in soccer robots and RoboCup environment, is using algorithms based on pixel color properties. According to the report of RoboCup competitions website, annually an average of 32 teams succeed to gain the permit for participating in these competitions and according to the technical reports submitted to the competitions committee, approximately all these 32 teams use pixel color-based algorithms to identify the objects.We researched a number of alternative methods for object detection in RoboCup when designing our approach. The BHuman team uses a sequential sanity checking method for ball detection[4].The Dutch Nao Team similarly uses blob detection as an underlying mechanism for object detection, and use width- and height-based sanity checking for distinguishing goals from non-goals [5]. rUNSWift's approach differs from the blob detection methods in that they use feature descriptors and a modified ICP algorithm to map field objects to expectations [6]. The UT Austin Villa team uses a Gaussian fitness computations for simultaneous feature evaluations on detected object candidates[7].in The UPennalizers team , During calibration, a Gaussian mixture model is used to partition the YCbCr color cube into the Orange (Ball) ,Yellow (Goals) , Green (Field) ,White (Lines) colors. then, Using a number of trained images, resulting in a color look-up table. While the robot is running, the main processing pipeline segments the highest-resolution color images from the camera by classifying individual pixels based upon their YCbCr values [8]. One of the main challenges of these teams in setting matches is color recognition and learning of the robot which is time consuming and faces some problems. All the mentioned approaches require a color table that must be created and loaded to the robot before the game starts so that it can recognize the colors based on the color table while playing. Since the creation of the color table is done based on the environment conditions before the game, the robot will not be capable of recognizing colors if the illumination conditions change during the game, because the color table loaded to the robot's memory is created based on specific illumination conditions. Therefore in this paper a robust object recognition approach against illumination changes of an environment, noises, rotations and poses is proposed by using the matching algorithm based on SURF feature [9] , which is independent of color feature and illumination condition, and utilizing a neural network.

2 Object Recognition

Nao robot has two high quality (HD) 1.2 pixels cameras that are both located in the frontal part of its head. These cameras are capable of capturing 30 frames per second (fps) videos with 640×480 resolution. These cameras are located with the distance of 4cm and degree of 40 from each other and each camera's angle of view is 34.8. Also these cameras are not stereo and cannot present simultaneous images. Task of the vision system in a robot is to recognize and track the objects like ball, goalposts, robots of the same team and rival robots in the game. Dataset images are categorized in four categories and for each object in each category, 100 training images of datasets [1-2-3] are gathered. First, the feature vector of each image is extracted by SURF algorithm. Then these feature vectors are used as training data in the neural network. For training, these vectors are considered as the input and the target vector is considered as output of the neural network. Target vector is a vector with the length 5 (the number of objects) that one of its elements, according to the category that the input vector belongs to, is one and the others are zero [10]. In this step, it is determined that what objects exist in the input image, and in the next step the exact location of the object is recognized by object location finding module. Block diagram of the vision system is shown in Fig. 1. Collecting enough suitable training data is necessary to implement the identification system and achieve an acceptable result. To this aim, a set of images were captured in different lightning condition, different intervals, different positions and angles (rotation) and in noise conditions from 6 groups of objects by NAO robot's camera.

tions) that we use them as training images. Neural networks used in this paper are Multi Layered Perceptron (with a hidden layer) and the training is performed by back propagation method. Also the activation function of the neurons is Sigmoid [11]. After training, in order to recognize an object in the input image, its extracted feature vector (achieved by the steps explained in previous section) is passed to the neural network as its input and then the output is computed; the neuron whose maximum output is greater than a specific threshold defines the type of the input object. If the maximum output of all neurons is less than the threshold, then the object is defined as "unknown". The threshold value is defined differently for each object category. Table (1) shows this threshold for each object category. According to the output of the neural network, system would be capable of only recognizing the location of the detected object from the previous step and this reduces time and memory consumptions.

objects	ball	goal	Robots	field	line
Threshold	0.6	0.8	0.9	0.7	0.6

Table (1) : shows threshold for each object category

3 Finding Object Location

To decrease the calculation load and increase the processing speed, the images received from the camera are not processed completely, but a point in a far distance from the point of robot's camera is projected in the horizon and a line along the image is drawn in both sides from that projected point along the horizontal vector. Since all objects are on the field, all pixels above the horizon line are left unprocessed and only those pixels below the horizon line are processed [12].

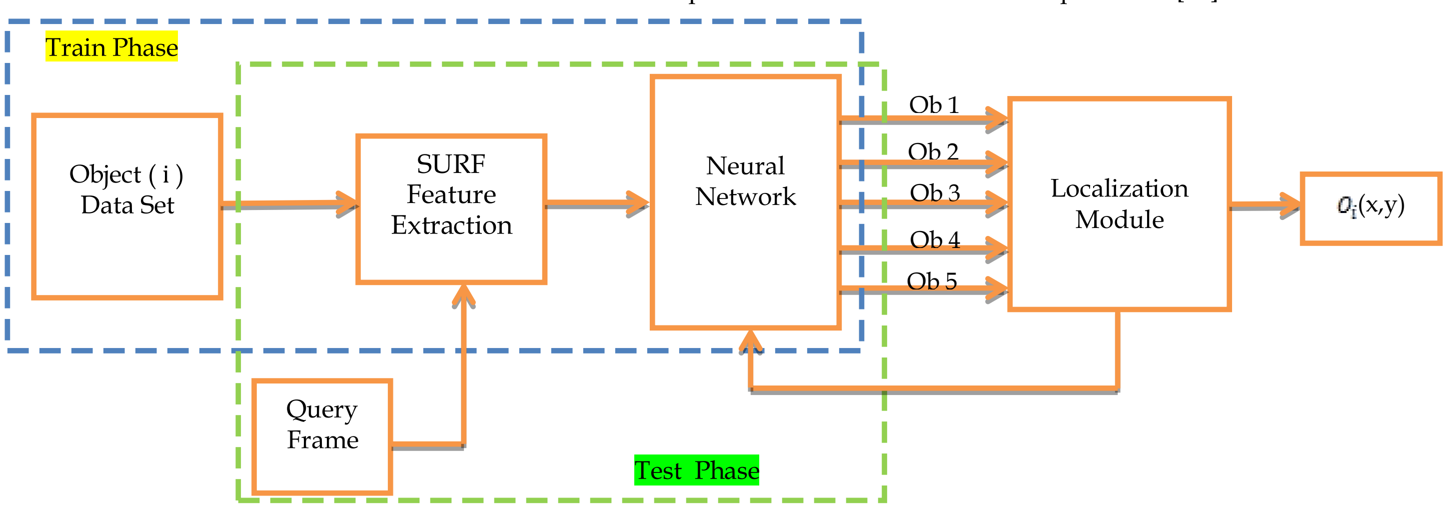


Fig1 . Block diagram of the vision system

Fig. 2 shows an example of used dataset. Now we have several images for each object (image in different distances, different poses, noisy image, image indifferent illumination condi-

ISSN 2229-5518

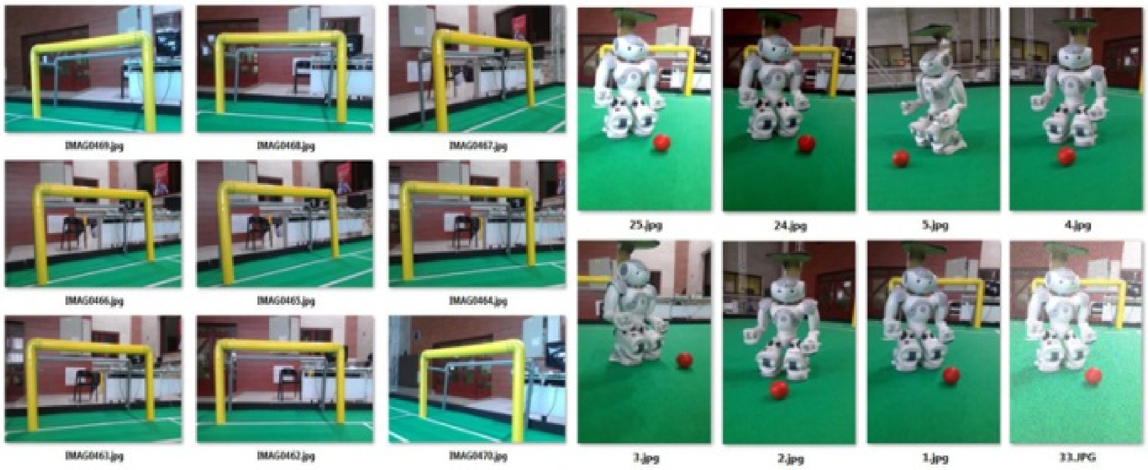


Fig 2. an example of used dataset.

In order to find the exact location of a recognized object by the previous section, firs we grid the input image whose objects are recognized in the previous step and then compute the histogram resulted from each grid's SURF features and then we set these histograms as input of the neural network and compute the output. If a part of an object is in grid i, that object's corresponding neuron in the neural network's output is activated with value 1, and the grid i is labeled. This approach is done for all the grids and finally those grids with the same labels are considered as the location of the object. Figure (3) shows an example of detecting ball and robot location.

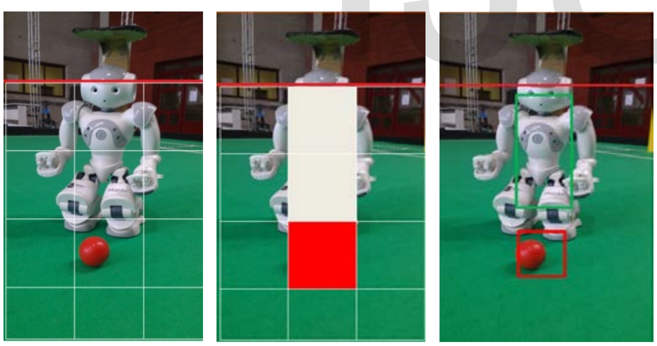
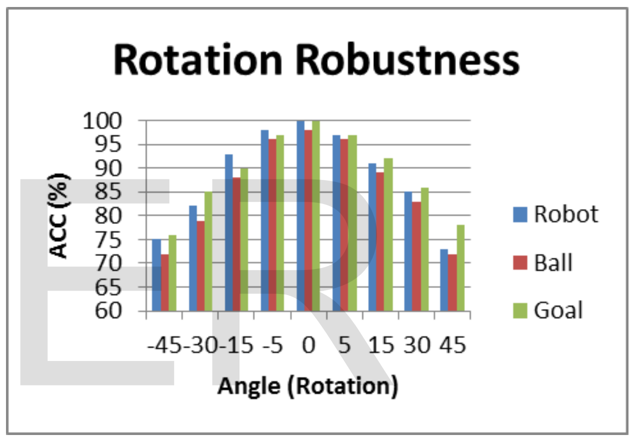


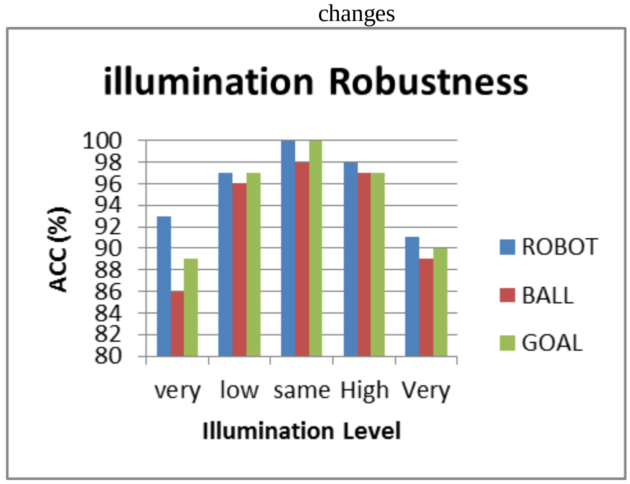
Fig 3 . sample of ball and robot detection

4 Conclusion

Since the extracted feature from the previous step is independent of scale, rotation, noise and illumination changes, the proposed method is capable of object recognition in different scales, poses and also illumination conditions. Also with training the neural network by extracted feature vectors obtained from noisy images, recognition is approximately robust to noise. The following plots show the robustness of the proposed approach against illumination and pose changes.

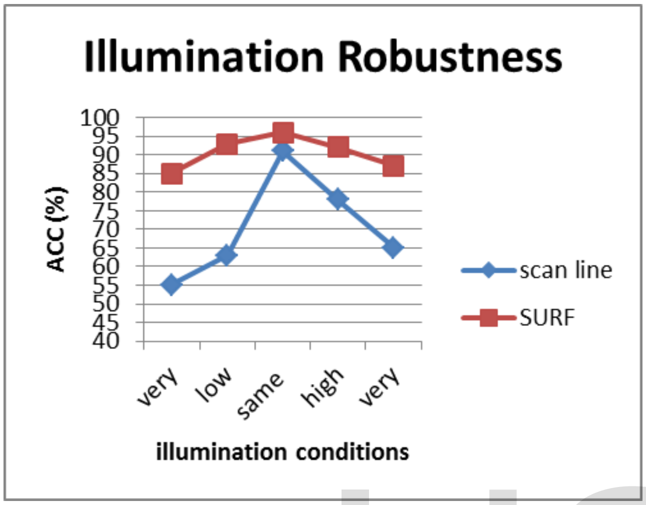


Plot (1) : show the robustness of the proposed approach against pose changes



Plot (2) : show the robustness of the proposed approach against illumination changes

As it is shown in plot (1), the proposed method is robust against rotation and poses changes and has a good performance. Also according to the results obtained from plot (2), the proposed method is robust against illumination changes and has a reasonable accuracy. We tested the proposed method and the linear scanning method (introduced in the introduction) in five different illumination conditions as well in order to investigate the accuracy of the proposed method. Experimental results show that the proposed method is performing better from the aspects of accuracy and robustness against illumination changes in comparison with the linear scanning method. Plot (3) shows both algorithms' performances in different illumination conditions.



Plot (3) : shows the performances of both methods in different lighting conditions

This paper presents a real-time auto-adjusting vision system for robotic soccer. During testing the system seems to be able to meet its requirements. The robots were able to localize themselves on the field and to handle the ball, goal, line, robot and other landmarks in an accurate manner.

5 References

[1] RoboCup SPL, "The RoboCup Standard Platform League," 2014, http://www.tzi.de/spl

[2] RoboCup Technical Committee. RoboCup Standard Platform League (Nao) rule book, 2014- 05-08. available on ttp://www.tzi.de/spl/pub/Website/Downloads/Rules 2014. pdf.

[3] S. Barrett, K. Genter, M. Hausknecht, T. Hester, P. Khandelwal, J. Lee, M. Quinlan ,A. Tian, P. Stone, andM. Sridharan. Austin Villa 2010 standard platform team report. Technical Report UT-AI-TR-11-01

[4] Rofer, T., Laue, T., et. al B-Human Team Report and Code Release 2010, http://www.bhuman.de/file_download/33/bhuman10_coderelease.pdf.

[5] C. Verschoor, A. Wiggers, D. ten Velthuis, A. Keune, M. Cabot,S. Nugteren, E. van Egmond, H. van der Molen, R. Iepsma, M. vanBellen, M. de Groot, E. Fodor, R. Rozeboom, and A. Visser, "Dutch nao team - technical report," 2011.

[6] S. Harris, P. Anderson, B. Teh, Y. Hunter, R. Liu, B. Hengst,R. Roy, S. Li, and C. Chatfield, "Robocup standard platform league- runswift 2012 innovations," in Australasian Conference on Robotics and Automation, 2012.

[7] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs,R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operatingsystem," in ICRA Workshop on Open Source Software, 2009.

[8] Christopher Akatsuka, Alan Aquino, Sarah Dean,Yizheng He, Tatenda Mushonga, Yida Zhang, andDr. Daniel Lee, "Robocup 2013 Standard Platform LeagueTeam Description Paper"

[9] Lowe, D.G. 1999. Object recognition from local scale-invariant features.The proceedidng of the seventh IEEE International Conference on Computer Vision 1999. 2: 1150-1157.

[10] R. Jain, R. Kasturi and B.G. Schunck, Machin Vision, McGraw-Hill, Singapore, 1995

[11] L. Fausett, Fundamentals of Neural Networks, Prentice-Hall, New Jersey, 1994.

[12] Ehsan Hashemi, Maani Ghaffari Jadidi, et. al MRL-SPL Code Release and Team Report 2012.