The Real-Time Object Detection System on Mobile Soccer Robot using YOLO v3

Hendawan Soebhakti
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
hendawan@polibatam.ac.id

Senanjung Prayoga
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
senanjung@polibatam.ac.id

Rifqi Amalya Fatekha
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
rifqi@polibatam.ac.id

M. Buana Fashla
Department of Electrical Engineering
Batam State Polytechnic
Batam, Indonesia
mbuanafashla@gmail.com

***Abstract*—In a Soccer robot, each robot must be able to detect the object such as ball, goal and circle line in the game field through the vision system using a camera. The object detection in the past decade was used color filtering method, then it's methods was develop into neural network based. Neural network detection has also a lot of progress, start from R-CNN, fast R-CNN, and new method is YOLOv3. In this paper we will explain the implementation of YOLO V3 to detect object at Barelang mobile soccer robot. This system is run on SHUTTLE X1 MINI PC has Octa Core Intel Core i7-7700HQ processor, 16 GB of RAM and with 3GB of NVIDIA GeForce GTX 1060 graphics card has cuda core score 1152. The experiments result show the object detection get 28.3 fps. Datasets performance of the proposed method is IOU 71.76%, recall 0.92, precision 0.92 and mAP 87.07%. YOLO v3 capable to detect and distinguish objects in the different lighting condition, with max distance 3 m for ball object and 8 m for goal object.**

Keywords—YOLO V3, Mobile Soccer Robot, Object Detection.

I. INTRODUCTION

The Indonesian robot contest is an annual event as a venue for engineering and design competition in the field of robotics organized by the Indonesian Ministry of Higher Education. In this competition there are 6 division. One of them it is The Indonesian Wheeled Robot Contest. This divison adopts the rules of the Middle Size League (MSL), which is one of the robot matches in the Robot Soccer World Cup Competition (Robocup). In MSL matches, there are two robots team, each team consisting of 5 robots. Each team must be able to play football autonomously. Like the real soccer match, the team that produces the most goals, the team will wins. To be able to play autonomously, at least each robot must have real time object detection system capabilities. The minimum capability of object detection here is that it can at least detect ball objects, goal objects, and landmarks in the form of a circle line in the middle of the field that can be used as a marker for positioning robots.

In the previous year (2018) the detection system used at the Barelang robotic team was color filtering method and contour extraction [1]. With this method, it can detect objects in the form of a goal with a maximum distance of 8.5 m. The weakness of the method is that it is very vulnerable to changes in light intensity. If the light intensity changes, then the color threshold setting must be changed as well. And that is very troublesome when the match takes place.

In object detection, besides using the color threshold method, it can also use object recognition methods with the artificial neural network. The detection of objects using neural networks starting in the early 1990s, Le Cun et al have conducted research on handwritten digit recognition with a back-propagation network [2]. Then Redmon et al. in 2016 at IEEE conference on computer vision and pattern recognition introduced a new method in detecting objects using a neural network known as You Only Look Once (YOLO v1) [3]. YOLO v1 is a developing object classification method from DNN (Deep Neural Network). This first version was tested at PASCAL VOC 2007 + 2012 resulting in 63.4% mAP with 45 fps using the Titan X computer. Then at the end of 2016 on December 25, 2016, Josep Redmon again published the development of YOLO called YOLO v2. In this second version, Josep added batch normalization in the YOLO v1 detection process, it give a 2% increase in mAP, so the mAP for the 2007 + 2012 PASCAL VOC at YOLO v2 was 78.6% at resizing 544 × 544 with fps 40 on Titan X computers [4]. That results is very fast to be used as a detection of real time objects. In 2018, Josep Redmon re-published the latest version of YOLO, called YOLOv3. In this version there are not many changes but there are additions to the convolutional layer to process feature map [5].

Based on this, in order to the Barelang mobile soccer robots can detect various objects in real-time and with various changing environmental conditions, YOLO v3 is implemented to detect objects such as goal, ball and circle line in the middle of the field.

The structure of this paper is organized as follows. Section 2 the method of proposed approach for object detection using YOLO v3 in detail. Section 3 describes the design hardware used in robots system. The results of experiments are presented in section 4. And in section 5 contains the conclusion and future work from the results obtained.

II. OBJECT DETECTION

YOLO v3 is an accurate and fast object detection method [5]. This is very important in detecting objects for mobile robot soccer. YOLO v3 is different from the similar method of detecting objects such as the R-CNN series which based on regional detection: Fast R-CNN [6] and Faster R-CNN [7]. Fast R-CNN and Faster R-CNN based on the region target detection method, it contains all sorts of potential regional generating parts and various feature layers. This causes the

978-1-7281-2807-8/19/$31.00 ©2019 IEEE

image processing to be long, so it is not suitable for use in real time applications.

Whereas the YOLO method, training and detection, feature extraction and regression classification, are all done in one network. YOLO regard object detection as a regression problem. Once the image is input into the network, the position of all objects in the image, their categories and corresponding confidence probabilities can be obtained [8].

YOLO uses probabilities, convolutional neural networks and non-max supression to detect objects. YOLO takes a very different approach. YOLO really sees images only once but in a smart way. This system only uses 3 processes to detect objects as shown in Fig. 1, which is changing the size of the image from the frame to be detected to a size of 416 × 416 pixels, then using convolutional network to detect objects, and after that the detection results are unified into 1 box by using Non-Maximum Suppression [1].

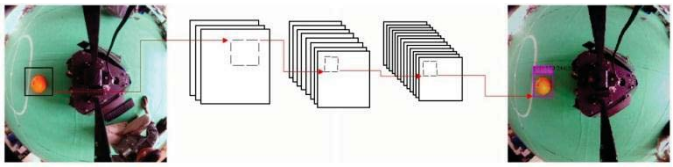

Fig. 1. YOLO Object Detection

In the process of resizing an image to 416 × 416 pixels the image is divided into a grid × grid cell. In each cell it is responsible for predicting the object that is believed. Then each cell will predict a bounding box to describe a rectangle that wraps an object. For each box that is believed to be an object, the cell also predicts the class of the object. This works like classification: it provides a probability distribution for all possible classes in the box. Then the results of each probability class form a lot of boxes and put together using NMS (Non-Maximum Suppression) based on the threshold of the object [9], as shown in Fig. 2.

Fig. 2. Non Maximum Suppression on YOLO
Source: https://appsilon.com/object-detection-yolo-algorithm/

YOLO v3 predicts the bounding boxes use dimension clusters as anchor boxes. Anchor boxes initialize the width and height size, which is closest to the size of the object and then be changed to the size of the object using the output neural network (final feature map). This network predicts 4 coordinates in each bounding box, namely t_x, t_y, p_h, p_w and also object class [4]. Fig. 3 is a bounding box of the detected object. The prediction of width and height of the box as offsets from cluster centroids.

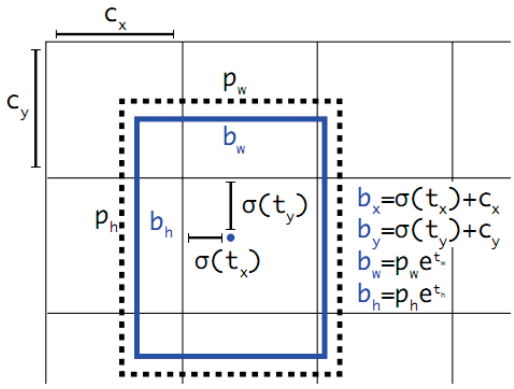

Fig. 3. Bounding boxes with dimension priors and location prediction [4]

YOLO predicts several bounding boxes, but YOLO sets bounding boxes based on the largest IOU (Intersect Over Union). IOU is the overlapping rate of the generated candidate bound and ground truth bound, that is the ratio of their intersection and union. [4,10].

$$IOU(Pred, Truth) = \frac{area(box_{truth}) \cap area(box_{pred})}{area(box_{truth}) \cup area(box_{pred})} \quad (1)$$

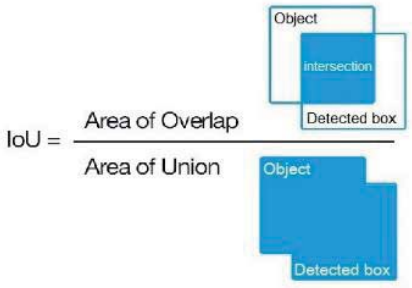

Fig. 4. Representation of the IoU Formula

Each prediction will be more convinced to predict the size, aspect ratios, or class of the object, and increase each recall and precession. Recall is a positive true proportion of output, while precession is a positive true proportion of the detection box.

$$Prec(Pred, Truth) = \frac{area(box_{truth}) \cap area(box_{pred})}{area(box_{pred})} \quad (2)$$

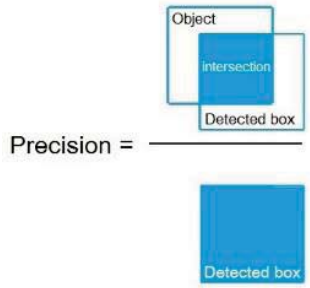

Fig. 5. Representation of the Precision Formula

$$Recall(Pred, Truth) = \frac{area(box_{truth}) \cap area(box_{pred})}{area(box_{truth})} \quad (3)$$

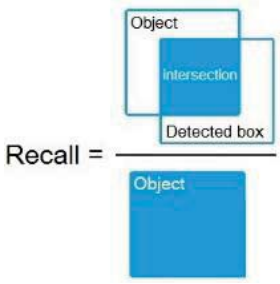

Fig. 6. Representation of Recall Formulas

To test the learning outcomes used Average Precision (AP). Average Precision is calculated based on the average of the maximum precision of 11 recall levels.

$$AP = \frac{1}{11} \sum_{r\epsilon\{0.0,\ldots,1.0\}} AP_r = \frac{1}{11} \sum_{r\epsilon\{0.0,\ldots,1.0\}} Pinterp^{(r)} \quad (4)$$

AP used here is AP on PASCAL VOC 2007, which will be true positive if IoU > 0.5 (50%). The results of the AP will be made into MAP (Mean Average Precission). MAP is the average of the AP of each object class [11].

III. Hardware Design

In this section, the embedded process from object detection results to mobile soccer robot will describe. Fig. 7 shows the block diagram of the construction hardware of the mobile soccer robot. Barelang mobile robot soccer team hardware construction consists of Digital Omni Camera, Mini Computer, Motor DC Controller, Motor DC, Kicker Selenoid Controller, Kicker Selenoid. For the vision system, this robot is equipped with ELP Camera 180 Degree with a fisheye lens that makes it possible to produce images with a wider range and the image results is in radial shapes. This camera is placed at the top of the robot facing downward, so that it gets 360 degrees of view on all sides of the robot to detect and find out the position of object. Mini Computer used in robots as the core for controlling the system that will process the proposed method, SHUTTLE X1 MINI PC is selected as a mini computer board in this system. YOLO V3 Deep-Learning Neural Network needs a combination of Central Processing Unit (CPU) and Graphical Processing Unit (GPU) work together to speed up calculation during the process. SHUTTLE X1 MINI PC has Octa Core Intel Core i7-7700HQ processor, 16 GB of RAM and with 3GB of NVIDIA GeForce GTX 1060 graphics card has cuda core score 1152, with its compact dimensions the mini pc Shuttle is easy to implant inside the robot. When the results of deep-learning process has been done in the mini computer, then all the coordinate and shape of object will be sent to the sub controller of the motor dc controller. This information is used to move the motor dc and let the robot knows the position of the object, for example an object detected by a robot is ball, goal, enemy, and itself. When the system has detected and decides the robot to do a kick to enemy's area, the system will give a signal to the kicker selenoid controller and then execute by the actuator.

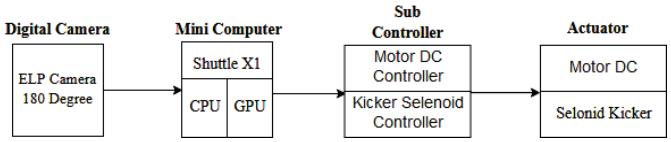

Fig. 7. Block diagram of hardware construction.

Barelang mobile soccer robot has a mechanical design that can be seen on Fig. 8. The material used is in the form of aluminum plate for the base and robotic cover, while for the support poles use hollow stainless material. This design makes it possible to make the vision of the Omni camera to be quite wide, so it can improve the quality of detection of objects. This robot uses three omnidirectional wheel which will make the movement of the robot more flexible in all directions.

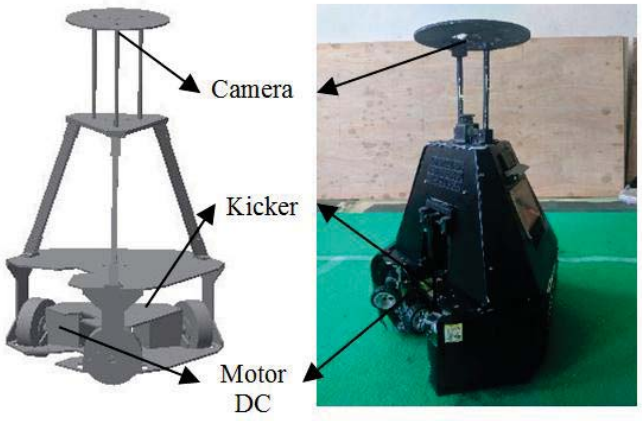

Fig. 8. Mechanical robot design

IV. Experiments Results

To use YOLO v3 software requirements are as follows. CMAKE version 3.8 for modern CUDA, CUDA 10.0, OpenCV version 2.4, cuDNN version 7.0 in this experiments using a Linux 16.04 as a operating system. After the software has been installed, the next step is to take pictures to be used as YOLO v3 learning data. Learning data was taken using an omni camera that embedded on top of robot body. Fig. 6 is example of the learning data. There is no specific method in collecting training data. Data collection is adjusted to the needs by paying attention to the light factors in the room. More variations of data taken it will be better the results of training. The amount of learning data used reaches 52.000 data.

After retrieving the data, the next step is to label the data according to the desired object name using the software provided by the darknet, Yolo Mark. The marking process will produce a text file containing the coordinates of the bounding box on the object labeled. The next step is the training process, the things needed for learning are darknet53.conv.74 (pre-trained weights for the convolutional layers) [5], calculating the anchor value, determining the number of

classes or objects that want to learn and creating a configuration file for YOLO v3.

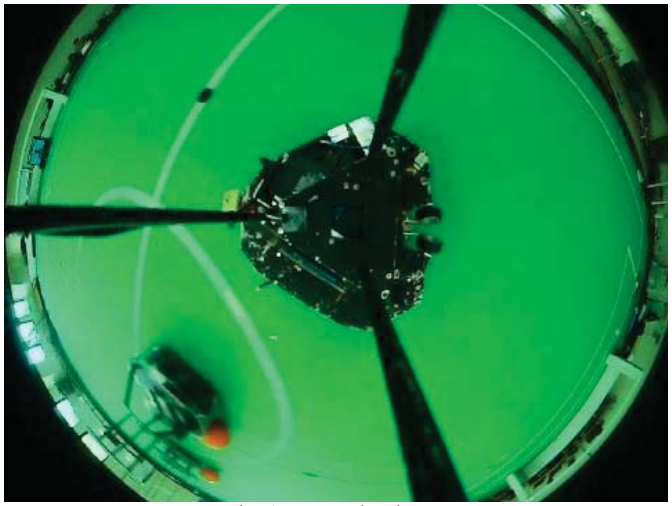

Fig. 9. Example Picture

Darknet53.conv.74 has an increase in speed of detection and data analysis compared to the previous version (YOLO v2). Training is carried out continuously at the highest resolution of the image except that the method used in YOLO v3 is multi-scale [5]. The duration of the learning process is in accordance with the capabilities and resilience of the pc used, the higher the computer specifications used, the faster and better the learning process is, CPU and GPU usage will cause the computer to heat up so that every few hours pause for a few minutes the goal is to rest on the computer and then resume. If the mean average precisions (mAP) and the percentage of each object are above 80%, the learning process can be stopped. Fig. 10 and Fig. 11 is resource of CPU and GPU that used for learning YOLO v3.

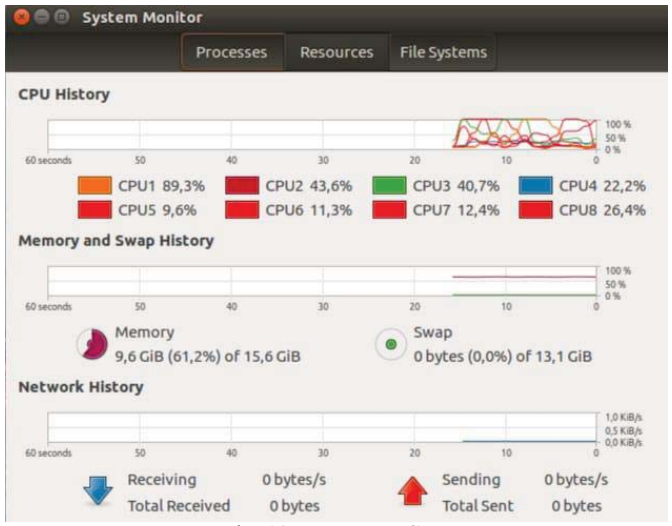

Fig. 10. Resources CPU

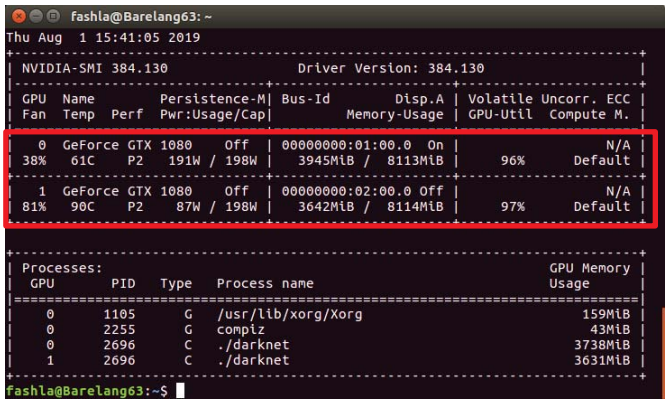

Fig. 11. Resources GPU

From the results of this experiment, it was found that Frame Per Second (FPS) of 28.3 fps using live-stream from a webcam (Fig. 11). If YOLO v3 is tested using AVI or MP4 video files, fps will get increase.

Fig.12. Data frame per second of YOLO v3

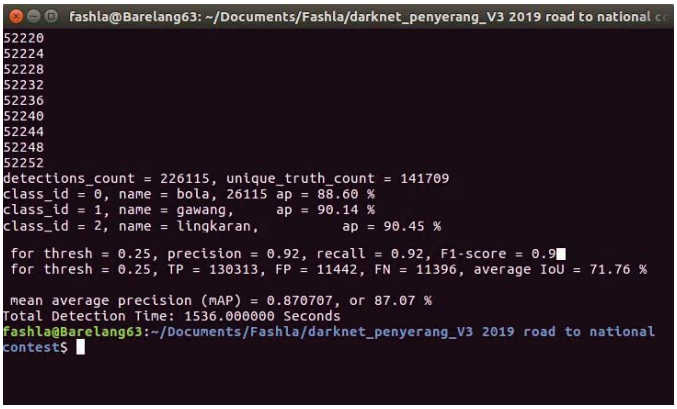

Fig. 13. Datasets Performance of YOLO v3

The training process used the pre-trained model of the darknet, darnet53.conv74, which means a pre-trained model that has been trained with 74 convolution layers followed by average pool-layer with validation accuracy of imagenet single crop top-5 of 93.8%. Then the model is converted to performance detection and a convolutional layer is added. In the training process using a linear activation function for the last layer and another layer using the leacky rectified activation

function. Fig. 10 show the datasets performance of the proposed method which the IOU 71.76%, recall 0.92, precision 0.92 and mAP 87.07%.

To determine the effectiveness of the proposed method, the YOLO v3 system is embedded into Barelang mobile soccer robot in real-time condition. YOLO v3 system is utilized to detect the ball, goal, and circle line landmark on soccer field.

In this section it is intended to show the results of the test using YOLO v3 on the ball object (Fig. 14), the goal (Fig. 15) and the circle line (Fig. 16). The test is carried out in a room that is exposed to sunlight from outside. While in Fig. 17 it can be seen that the ball object can still be detected even though the light conditions in the environment are not bright enough. This is very difficult to do with the color threshold method, but by using YOLO v3, this is very easy to do. The quality of object detection is very dependent on the quality of the learning data, process of labeling the data, and the variation lighting in the room. The more variations of data that are learned, the better the results of the detection.

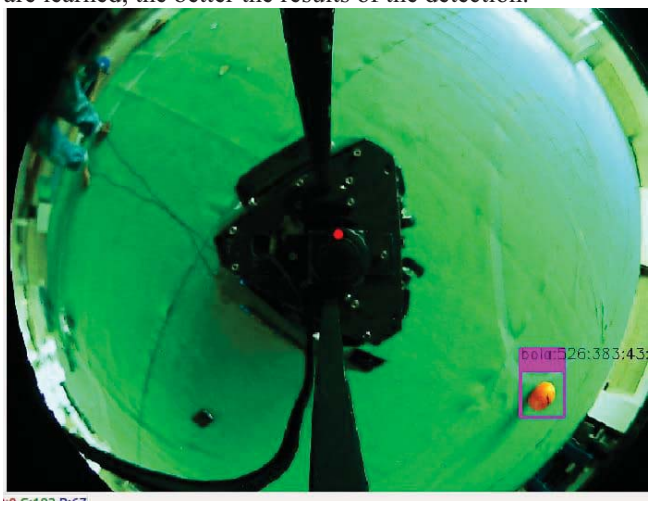

Fig. 14. Ball object detected

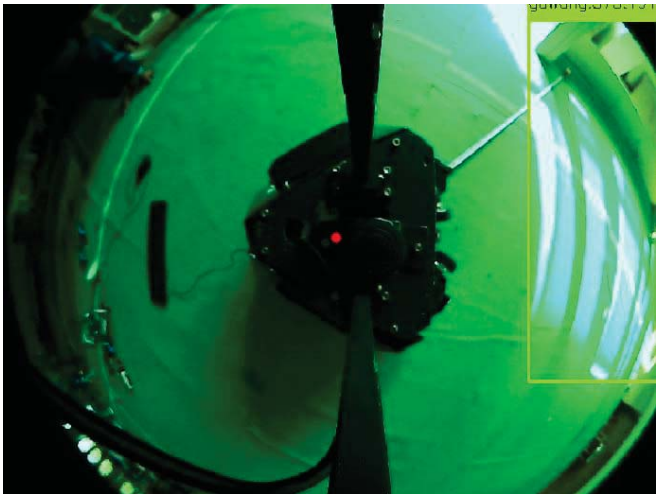

Fig. 15. Goal object detected

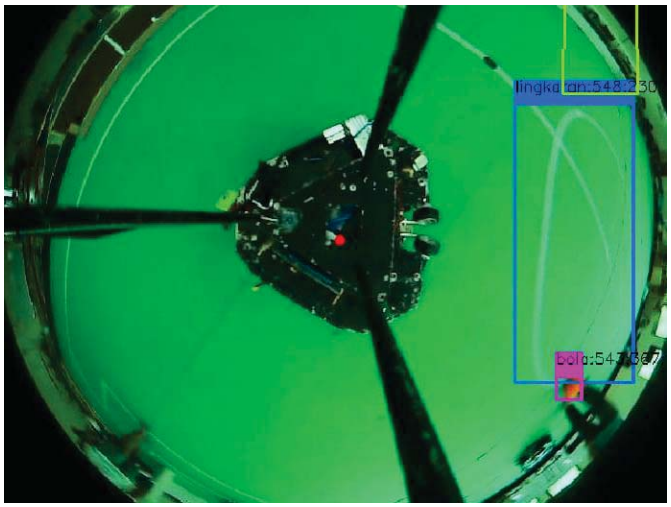

Fig. 16. Circle line object detected

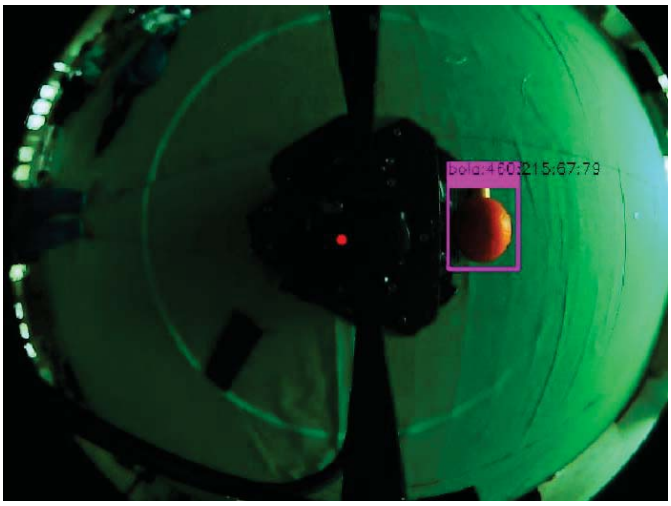

Fig. 17. Ball detection in less lighting condition

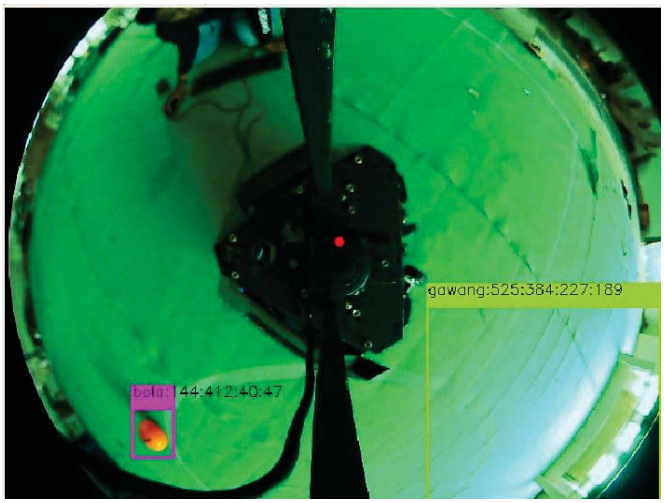

Fig. 18. Goal detection in max distance

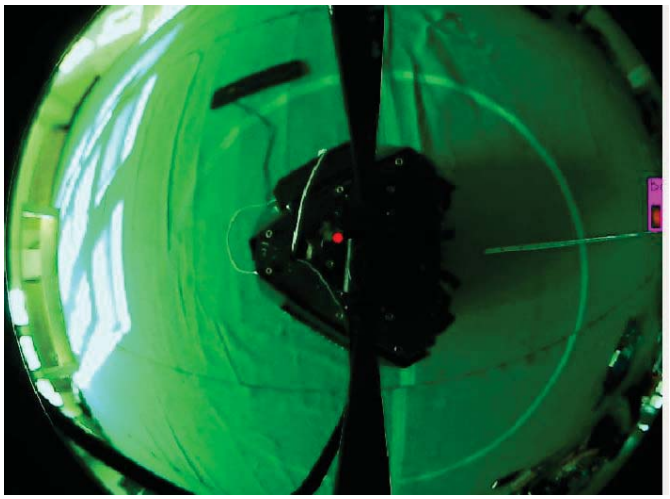

Fig. 19. Ball detection in max distance

The maximum distance from the object YOLO v3 can detect is 8 meters for the goal (Fig. 18) and 3 meters for the ball (Fig. 19). The smaller the size of the object, the YOLO v3 will be difficult to detect.

V. CONCLUSION

This paper presents an implementation of YOLO v3 on Barelang mobile robot soccer team to detect orange ball, goal, other robot and circle landmark in center field. From the experimental results it can be concluded that the YOLO v3 can detect object in a different situation and in real-time condition. By using the proposed method, it also can show the coordinate of object which already detected so that it can be used to take further action, such as chasing the ball, kicking the ball towards the goal and avoiding opponents. In future work, we will use the method to detect other landmark on field such as edge lines, intersection of lines, corner lines and goal lines for input of particle filter localization.

REFERENCES

[1] B. Sugandi, S. Prayoga, I. A. Riandi and D. G. Tinambunan, "Goal Detection and Opponent Avoidance Algorithm for Wheeled Robot Soccer using Color Filtering and Contour Extraction," 2018 International Conference on Applied Engineering (ICAE), Batam, 2018, pp. 1-5.

[2] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Handwritten digit recognition with a back-propagation network. In Advances in neural information processing systems, 1990.

[3] Redmon J, Divvala S, Girshick R, et al. “You only look once: unified, real time object detection”. Computer Vision and Pattern Recognition. 2016:779-786.

[4] Redmon J,Farhadi A. “YOLO9000: Better, Faster, Stronger” IEEE Conference on Computer Vision and Pattern Recognition. 2017:6517-6525

[5] Redmon J, Farhadi A. “YOLOv3: An Incremental Improvement”. IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[6] Girshick R. “Fast R-CNN”. Proc of IEEE International Conference on Computer Vision. 2015:1440-1448.

[7] Ren S, He K, Girshick R, et al. “Faster R-CNN: towards real-time object detection with region proposal networks”. Proceedings of the 2015 advances in Neural Information Processing Systems. Palais des Congrès de Montréal, Montréal CANADA. 2015:91-99.

[8] W. Yang and Z. Jiachun, "Real-time face detection based on YOLO," 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju, 2018, pp. 221-224.

[9] Rothe, R., Guillaumin, M., & Van Gool, L., ”Non-maximum suppression for object detection by passing messages between windows”, In Asian Conference on Computer Vision (pp. 290-306), Springer, Cham, November 2014.

[10] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition” arXiv preprint arXiv:1409.1556, 2014

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, “The pascal visual object classes (voc) challenge”, International journal of computer vision, 88(2):303– 338, 201