Vision

Last year, our robot vision was already accurate and worked well. Our robot could already detect ball and features in a closed room. Our vision weakness was it did not run well outdoors. Its deficient performance can be seen when our robot played drop-in matches. This year, we have two alternatives to handle the problem and perform vision better in our robot.

The first option, to detect the ball, we use Local Binary Pattern (LBP), which is a texture operator popularized by Ojala et al. [1]. LBP calculates local texture representations which are built by comparing each pixel with the surrounding pixels. For the classification process, we use the cascade classifier [2][3]. Cascade classifiers are trained with hundreds of "positive" and "negative" images of the same size. To detect goal post, we use the hough transform described in [4]. This method's weakness is noises outside the field are recognized as a ball. To ignore those noises, the color of the field is segmented based on the color. From the detected contour, the largest contour was selected and then Convex Hull was performed. After that, LBP is used on the object inside convex.

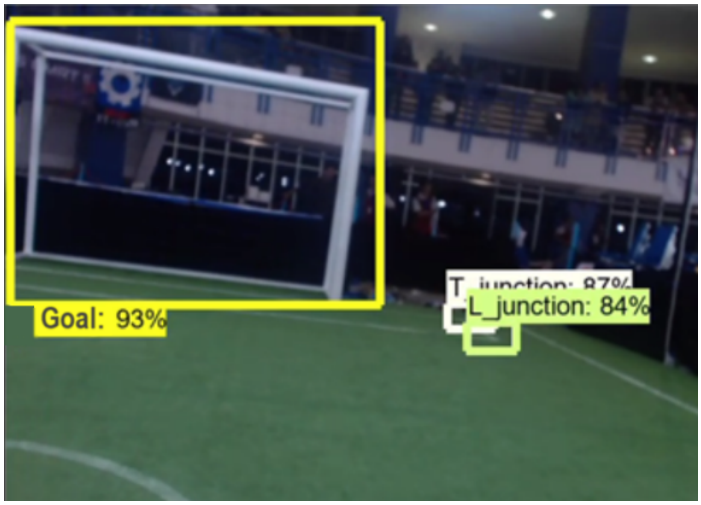

Fig. 1. CNN-Based Object Detection

Another option, MobileNet V1 architecture [4] is used to detect objects like balls, goals, and line features on the field. we plan to recognize another robot this way too. To train our model, we collect datasets manually. Then, we label the desired object and convert the labeled image data from XML to CSV extension. The CSV file is used to generate the TFRecord file. Create a configuration file that is used to train an Artificial Neural Network and a '.pbtxt' file that contains index IDs and names. Finally, we can start to train our model. To Implement a Convolutional Neural Networks (CNN) in our robot, Google's TensorFlow [5] and Single Shot Multibox Detector (SSD) algorithm with our trained model are used to test object detection in CNN models. The results we get from this CNN model shown in Fig. 1. We are sharing our step-by-step guide to

do the learning, fine-tuning, and getting the inference in this repository (https://gitlab.com/rezaarrazi/objectdetection).

1. T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, (2002).
1. Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In IEEE Computer Vision and Pattern Recognition CVPR, (2001)
2. Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. (2002).
3. D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", in: Pattern Recognition, Vol.13, No.2, p.111-122, (1981).
4. Qin, Z., Zhang, Z., Chen, X., et al.: FD-MobileNet: improved MobileNet with a fast down sampling strategy. arXiv preprint arXiv:1802.03750 (2018).
5. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. “Tensorflow: A system for large-scale machine learning,” in 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283, (2016).