Vision:

Image:

As the roadmap says, since RoboCup 2019, natural lighting conditions are now explicitly allowed, which means we have to consider the imaging under uncertain and changeable lighting environments to get satisfying images for later detection.

To have a more robust vision system with sunlight, we try to get a better imaging result with changing exposure among several levels. There will be less overexposure or underexposure with the embedded auto-exposure in the camera driver. However, it will also lead to unwanted exposure changing, which makes it more difficult to detect the ball, goals, field, etc. So we decided to set some exposure levels and tune it by current imaging result. According to the histogram, if it skewed left or right, we will change to the next or previous exposure level. [Figure].

In practice, the lighting is relatively stable for seconds or minutes, calculating the histogram every program cycle will cost too much time. To balance the time cost and effect, we evaluate the cognition result first and decide whether to calculate the histogram.

Unfortunately, this method didn't work as we expected in RoboCup 2018's drop-in game. We'll continue improving and try other methods.

Cognition:

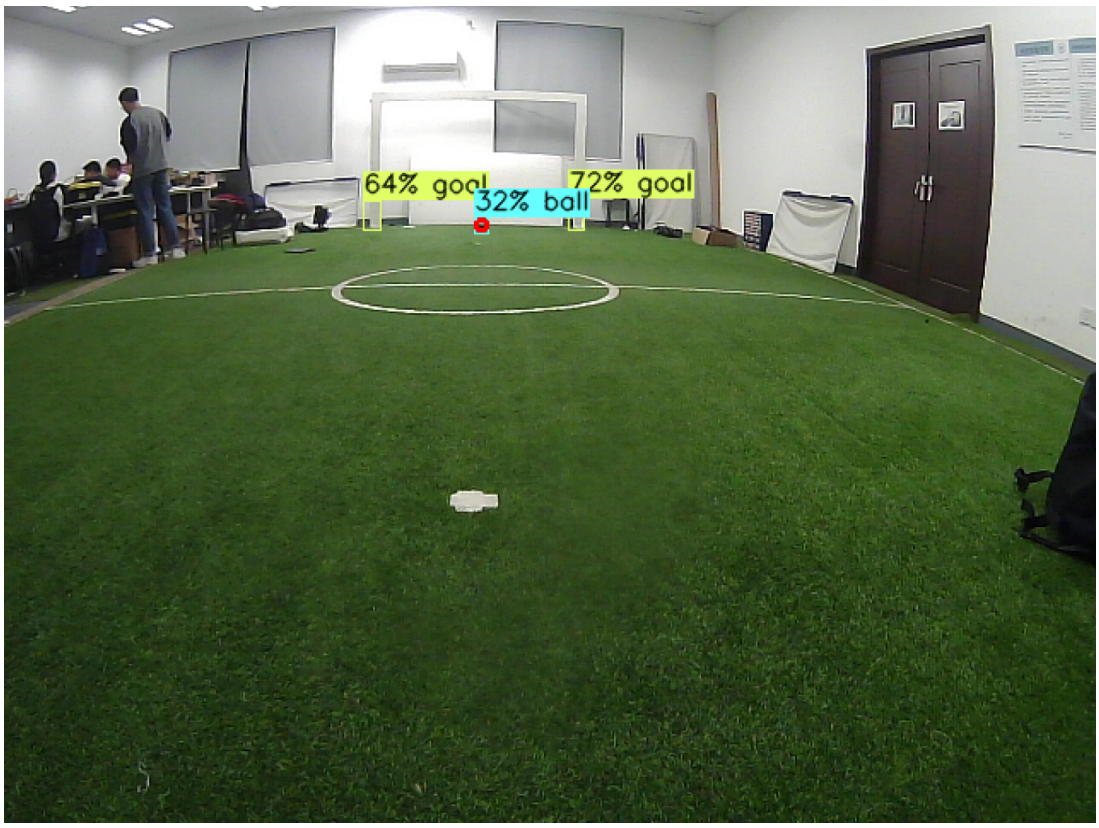
Cognition is a necessary part of the robots to perceive the surroundings. Our detection for the field and the lines are based on traditional computer vision techniques like hough transform, canny or simply using thresholds to get areas. Deep learning method is used for the ball, goalposts and robots.

Until last year, we are using YOLOv1-tiny, which is fast enough to run on our previous hardware TX1. As our hardware is updated to TX2, we change our algorithm to a newer version YOLOv3-tiny and change the implementation to TensorRT. Compared to YOLOv1-tiny, YOLOv3-tiny runs a little slower, but has higher accuracy, especially for small objects, like a ball 10 meters away.

For higher inference performance, we choose TensorRT to optimize our network model. NVIDIA TensorRT is a platform for high-performance deep learning inference. It can apply optimization to trained models and selects platform-specific kernels to maximize performance on NVIDIA GPUs.

Besides, we undertake to recognize the corner using its rich line features. When the robots are around the field corner, where they can see almost nothing except the corner lines. Such additional reliable landmarks will improve the localization. The location is given by the voting result of the white lines. With stable image, this method can locate the corner quickly and accurately.

The figure shows the detection result. The picture is taken on our robot standing at the center of goal area line. The two goal posts on the images are about 8m away in the real world, while the ball is on the center of two goal posts, also 8m away.


64% goal
72% goal
32% ball