Github link of project (contain training dataset that didn't fit the file size limit on ISIS):
https://github.com/noranMoh/RoboCup2020

Orange Ball detection

My first attempt to detect the ball was to try to detect a uni color ball similar to the ball used in RoboCup competitions before. Detecting a uni color orange ball is easier than detecting the normal black and white ball because we can detect it using its unique color. Since I did not have access to the NAO humanoid robot I used the laptop's webcam to detect the ball using openCV. First, the image is pre-processed by reducing the frame size to be able to process the frame faster, blurring it to reduce high frequency noise and finally converting the frame to HSV color space. Then the parts of the frame that belongs to the preset range of the color orange are detected using cv2.inrange. A series of erosions and dilations are then done to remove any small blobs that may be left on the mask. We then check to ensure at least one contour was found in the mask . Provided that at least one contour was found, we find the largest contour and compute the minimum enclosing circle of the blob, and then compute the center (x, y)-coordinates. I then draw a circle around the ball tracking it. The video orange_ball_detection shows the tracking of the orange ball.

Black and white ball detection

As people are attempting to make robots behave more and more as humans, the normal black and white ball was introduced to the RoboCup competition. This made detecting the ball more challenging as we can no more search for a specific color in our frame. This is why a classifier should be made and trained to be able to detect the ball. First I had to have both positive and negative data sets. Since I did not have access to a NAO robot and pictures taken by its cameras, I downloaded pictures of photos taken by the NAO robot top cameras[1]. For the training I used the openCV cascade classifier training. The idea of cascade classifiers is to concatenate weak classifiers to build a strong classifier. Setting the number of stages to 20, the minimum hit rate to 0.999, maximum alarm false rate to 0.5 and setting the nonsymetrical parameter I began the training process. I then used the trained classifier in detecting the ball. Since again I do not have access to a NAO's camera, I downloaded images taken by NAO robots[1] or random photos taken in NAO soccer games[3] to test my classifier. The photos and the results after detection are found in the folder test. The video black_and_white_ball_tracking shows detection of ball using webcam.

Resources

1. http://www.dis.uniroma1.it/~labrococo/
2. http://spqr.diag.uniroma1.it
3. https://commons.wikimedia.org/wiki/File:SPL_Team_B-Human,_RoboCup_2016.jpg