
Objects Recognition Based on On-line Machine Learning

Project Status Report for CIS 519

Abstract

To recognize objects (various kinds of ball in our project) in the given images, the model we proposed in this report will take three steps. The first step is using Hough transform to detect the most possible region that contains the ball. The second step contains techniques to extract feature from the selected region. In the third step, neural network is applied on the extracted features to classify the given instances. Several training/testing experiments have been carried out to show the performance of this model.

1. Introduction

This work presents an idea of objects recognition. We decided to use images of four kinds of balls from Caltech 256(Griffin, G. Holub, AD. Perona, P.) as our dataset, which contains 98 images of golf in the folder “088.golf-ball”, 174 in “193.soccer-ball”, 104 in “017.bowling-ball” and 98 in “216.tennis-ball”.

The task is challenging in the following aspects:

1. The objects with the same label can have very different colors, sizes and textures.
2. The objects we want to classify may not be the main parts in the image, sometimes with even other spherical objects such as a bowling-ball with a head in “017.bowling-ball\017.0006.jpg”.
3. The objects may not have high contrast against the surroundings, such as a white soccer-ball in a bright background in “193.soccer-ball \ 193.0070.jpg”.
4. It’s possible that several objects with the same label are in one image (such as “193.soccer-ball \ 193.0171.jpg”).
5. Some images are even hard for humans to classify, such as “017.bowling-ball \ 193.0155.jpg”, which has the color of a typical tennis ball but the pattern of pentagon and hexagon on a soccer ball.

2. Model Details

2.1. Overview of the model

The model for ball recognition is shown in Figure.1. Ball recognition can be considered as the problem of detect the circular region in the image, then fetch features inside the circle as input features for next step. In the final step, use neural network to classify the processed data into four categories.

2.2. Circle Detection

At the beginning, we considered using corner detection directly on the original image to extract features. But the result is not accuracy and robust enough for ball detection. From Figure.2, we can see that the detected corner features are distributed on the entire image. The complicated background could generate large amount of features, which are irrelevant to the ball. Thus, we think it’s necessary to first determine the circular region in each image.

During building the training set, the area can be located by either circle detection algorithm or manually selection. However, when classifying a ball in a new image, the circle area has to be detected automatically. Several methods for circle detection are studied in the last few years,(?),(?),(?). One of the robustest approach is Circle Hough Transform(CHT), which takes the image M and desired radius range $[r_{min}, r_{max}]$ as inputs and return the positions of circle centers. The algorithm could be presented in Algorithm 1. In our case, we use the log phase for the accumulator which is

$$\phi_{m,n}^{logr} = 2\pi \left(\frac{\log [\sqrt{(m-i)^2 + (n-j)^2}]}{\log r_{max} - \log r_{min}} - \log r_{min} \right) \quad (1)$$

While applying CHT method, the sensitivity factor is often introduced, which is a scaler in $[0, 1]$. A higher sensitivity factor helps in detecting weaker and partially obscured circles, however, increases the risk of false detection. In our case, we increased the sensitivity gradually until the first circle is detected to decrease the risk of false detection and miss detection.

Algorithm 1 Circle Hough Transform

Input: image I , radius range $[r_{min}, r_{max}]$
Initialize accumulator matrix $M = 0$.
for each edge point (i, j) in image I **do**
 for each point (m, n) in image I **do**
 if $r_{min}^2 < (m - i)^2 + (n - j)^2 < r_{max}^2$ **then**
 $M(m, n) = M(m, n) + \phi_{m,n}$
 end if
 end for
end for
Return: positions of local maximum points in accumulator matrix M

2.3. Results

Several results of applying our circle detection algorithm on Caltech256 dataset are shown in Figure.3.
The statics of the performance on four catagories is summarized in Table.1. Note: -1 means the detected circles are all wrong; 0 means it fails to detect any circle; 1 means only circular region that contains the ball is detected; 2 means result contains other false circles;

3. Work for Next Stage**3.1. Reconstruct the dataset**

In order to introduce neural network into this problem, the training date set need to be normalized to a $n - d$ matrix, where n is the number of training instance and d is the number of pixels. The idea is to resize the circle area into a particular size, convert the resized image into grey scale and reshape the pixel information into a vector for each circle we found in the circle detection step. Then construct a binary label vector manually according to which kind of ball each instance belongs to.

3.2. Neural Network learning and Deep learning

Given all the dataset and the labels, train the classification model by Neural Network learning and Deep learning with cross validation and try to figure out which method is better for ball classification problem. To evaluate the performance of each learning methods, we plans to compute the overall accuracy and precision, recall for each class and plot the ROC curve.

Acknowledgments

None.