Project Report

Project B: Object recognition (difficult)

ZHAO HAIQI

201436201H

Computer Vision

CSC 179

methods:

I used Point Feature Matching methods for my Object recognition.

I detect a specific object based on finding point correspondences between the reference and the target image. It can detect objects despite a scale change or in-plane rotation.

This method of object detection works best for objects that exhibit non-repeating texture patterns, which give rise to unique feature matches. This technique is not good at detect uniformly-colored objects, or for objects containing repeating patterns.This algorithm is aim to detect a specific object.

It find strongest feature points of training image and testing image to see whether two image are similar with each other or not. It depend on the number of matching pair of feature points.

The weakness of this method is how to tell the training object is the testing object. and the number of matching pair of feature points is always changing in different image. what is the boundary number of matching pair. This program always cost a lot of time to detect and recognize the object.

Use estimateGeometricTransform matlab toolbox to locate the object in the image, Get the bounding polygon of the testing image.

I detect the testing object with training image of different object one by one. Each object have many training image. find out the maximum matching point pair of each object. and once the pair number is bigger than boundary, it could be considered that training object is in the testing image. and so on. Finally calculate the object number and display the location of object with line.

This program also referred to the example of matlab official website

Ref: https://www.mathworks.com/help/vision/examples/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

codes:

clear all

close all

%%

%initialization

% turn off the image size warming or not

imageSizeWarnOff = true;

if imageSizeWarnOff

warning('off','images:initSize:adjustingMag');

end

% object number

objNum = 7;

% object name

objName = {'scissors';'sugar';'stapler';'bottle';'clip';'box';'book'};

%color of each object

color = {'y';'r';'g';'m';'c';'blue';'w'};

%for showing the procedure image: true ; otherwise false

show\_procedure = false;

%good condition (considered to be detected) for minimum MatchPair number of

%each object

goodCon = [38;41;50;39;29;47;85];

%pair number of each successful object

pairNum=zeros(1,7);

%result figure

resultFig = figure('name','result');

%project folder

projectFolder = 'E:\Learn\Computer Vision & Applications\project\';

%train folder

trainFolder = strcat(projectFolder,'training\_image\');

%test folder

testFolder = strcat(projectFolder,'testing\_image\');

%result folder

resultFolder = strcat(projectFolder,'result\');

%load scean image

testFileName = 'test007';

sceneImage = imread([testFolder testFileName '.jpg']);

result = sceneImage;

sceneImage = rgb2gray(sceneImage);

%%

% i: each object

% j: each train image in object

%testObj=7;%test used

%for i=testObj:testObj %test used

for i=1:objNum

%step 1

trainFolderEach = strcat(trainFolder,objName{i}) ;

if ~isdir(trainFolderEach)

errorMessage = sprintf('Error: The following folder does not exist:\n%s', trainFolderEach);

uiwait(warndlg(errorMessage));

return;

end

filePattern = fullfile(trainFolderEach, '\*.jpg');

jpegFiles = dir(filePattern);

curCon = 0; % set current pair number

howWell(i) = 0;

for j = 1:length(jpegFiles)

baseFileName = jpegFiles(j).name;

fullFileName = fullfile(trainFolderEach, baseFileName);

trainImage = rgb2gray(imread(fullFileName));

if show\_procedure

figure('name',baseFileName);

showImage = imresize(trainImage,0.3);

imshow(showImage);

title('Image of a Train');

end

%-----------------

if show\_procedure

figure('name',baseFileName);

showImage = imresize(sceneImage,0.3);

imshow(showImage);

title('Image of a Cluttered Scene');

end

%step 2

trainPoints = detectSURFFeatures(trainImage);

scenePoints = detectSURFFeatures(sceneImage);

if show\_procedure

figure('name',baseFileName);

imshow(trainImage);

title('100 Strongest Feature Points from Train Image');

hold on;

plot(selectStrongest(trainPoints, 100));

figure('name',baseFileName);

imshow(sceneImage);

title('300 Strongest Feature Points from Scene Image');

hold on;

plot(selectStrongest(scenePoints, 300));

end

%step 3

[trainFeatures, trainPoints] = extractFeatures(trainImage, trainPoints);

[sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);

%step 4

trainPairs = matchFeatures(trainFeatures, sceneFeatures);

matchedTrainPoints = trainPoints(trainPairs(:, 1), :);

matchedScenePoints = scenePoints(trainPairs(:, 2), :);

if show\_procedure

figure('name',baseFileName);

showMatchedFeatures(trainImage, sceneImage, matchedTrainPoints, ...

matchedScenePoints, 'montage');

title('Putatively Matched Points (Including Outliers)');

end

%step 5

if (size(trainPairs,1)>goodCon(i)) && (size(trainPairs,1)>curCon)

howWell(i) = 1; %good detect

curCon = size(trainPairs,1); % current pair number

[tform, inlierTrainPoints, inlierScenePoints] = ...

estimateGeometricTransform(matchedTrainPoints, matchedScenePoints, 'affine');

if show\_procedure

figure('name',baseFileName);

showMatchedFeatures(trainImage, sceneImage, inlierTrainPoints, ...

inlierScenePoints, 'montage');

title('Matched Points (Inliers Only)');

end

trainPolygon = [1, 1;... % top-left

size(trainImage, 2), 1;... % top-right

size(trainImage, 2), size(trainImage, 1);... % bottom-right

1, size(trainImage, 1);... % bottom-left

1, 1]; % top-left again to close the polygon

newTrainPolygon(i).array = transformPointsForward(tform, trainPolygon);

else

%bad detect

if show\_procedure

disp(strcat('bad detect in :',baseFileName));

end

end

if (size(trainPairs,1)>curCon)

curCon = size(trainPairs,1); % current pair number

end

if (howWell(i) == 1)

if show\_procedure

disp(strcat('good detect in :',baseFileName,';MatchPairNumber:',num2str(size(trainPairs,1))));

end

end

end

pairNum(i)=curCon;

if (howWell(i) ==1)

figure(resultFig);

hold on;

%show the successful dection object name

result = insertText(result, [newTrainPolygon(i).array(2, 1) newTrainPolygon(i).array(2, 2)], objName{i},'FontSize',50,'TextColor',color{i},'BoxColor','w');

imshow(result);

for k=1:i

if (howWell(k) ==1)

%show the bounding line

line(newTrainPolygon(k).array(:, 1), newTrainPolygon(k).array(:, 2), 'Color', color{k});

end

end

hold off;

title('Detected result');

end

if (i == objNum)

figure(resultFig);

result = insertText(result, [10 10], strcat('object number:',num2str(sum(howWell))),'FontSize',50,'TextColor','black','BoxColor','w');

imshow(result);

hold on;

for k=1:i

if (howWell(k) ==1)

%show the bounding line

line(newTrainPolygon(k).array(:, 1), newTrainPolygon(k).array(:, 2), 'Color', color{k});

end

end

hold off;

title('Detected result');

end

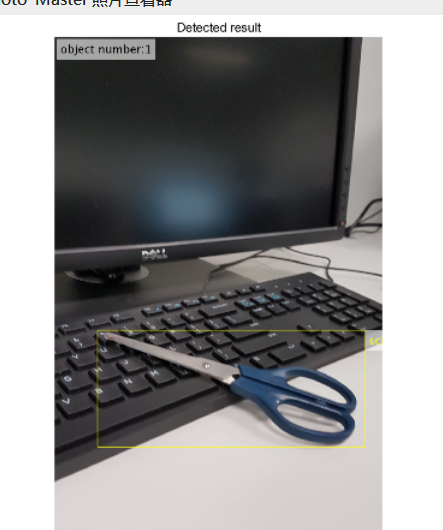
end

%save result

print(resultFig,[resultFolder testFileName '\_result'],'-dpng')

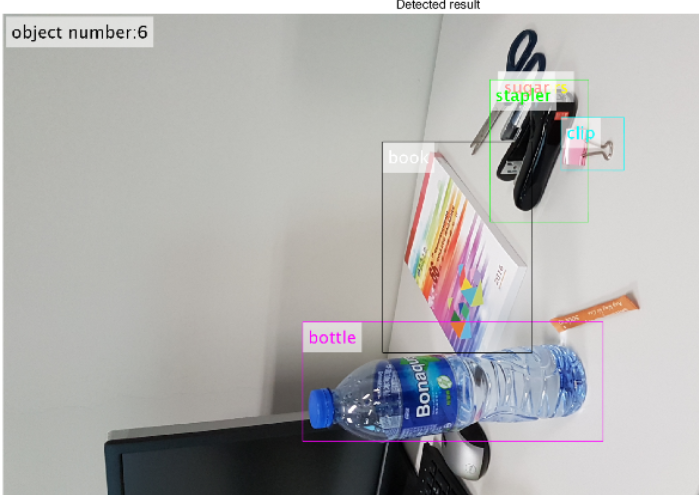
results:

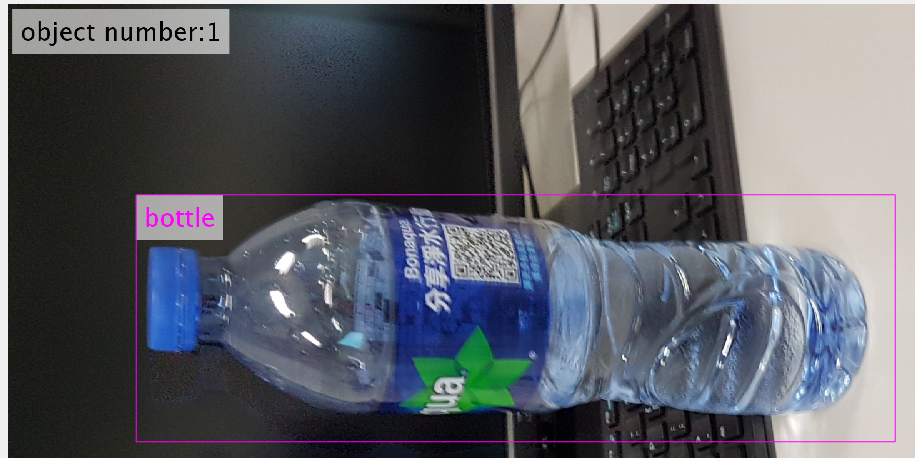


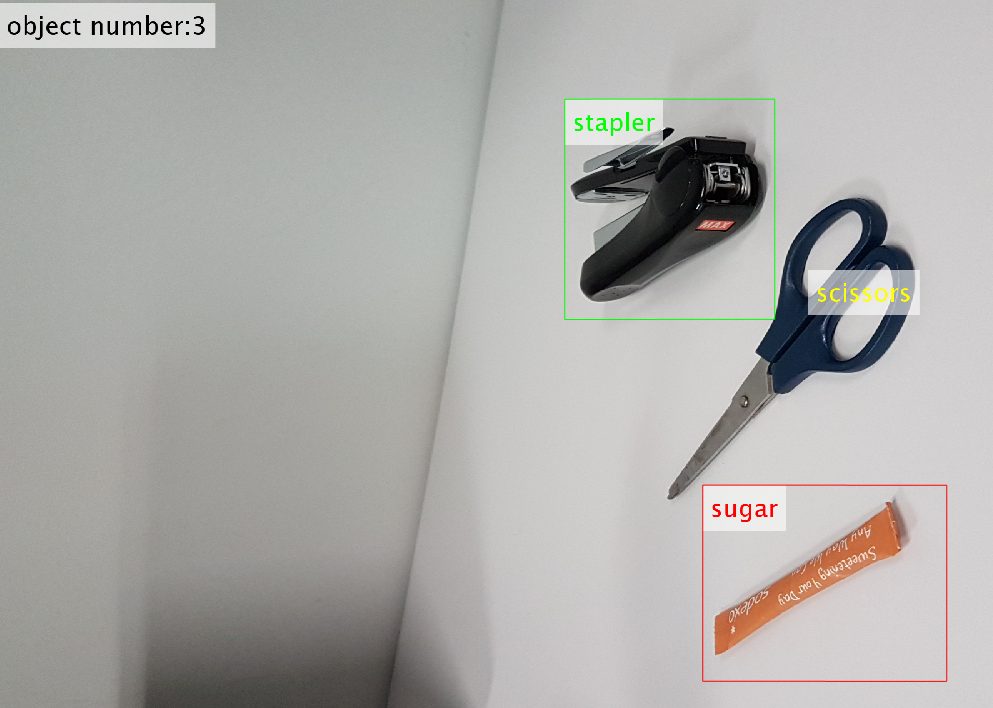


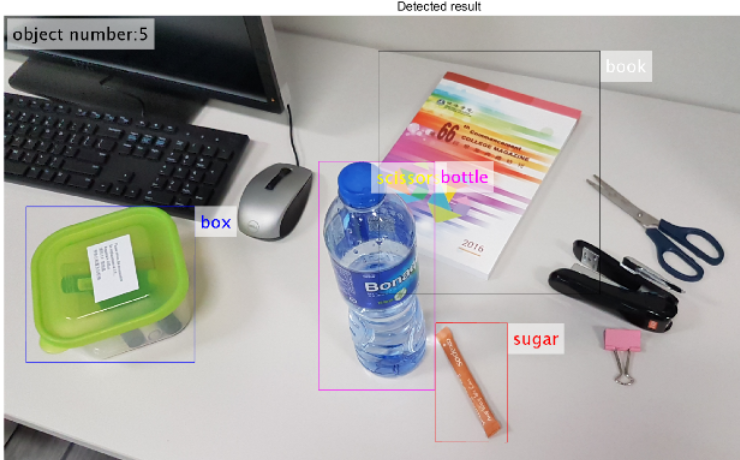


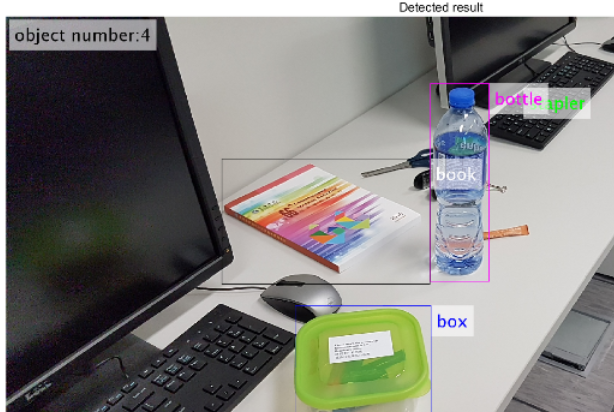


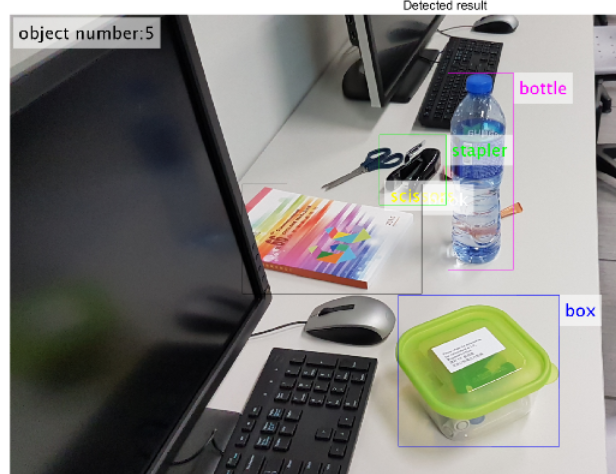














comparisons:

object comparisons: compare with different object. I found that the bigger object would be easy to detect. In this project. book and bottle are easy to detect. sugar and clip is diffcult to detect.

stapler is a unique and speicial scale. so staper is easy to detect.

if the object is behind the keyboard. it would be hard to detect because the background make the program confuse a lot.

method comparisons: compare to the other method. Point Feature Matching is not too good but it can be used in some simple object detection.

this method contains : object detection ; object recognization; object orientation； multi-object dection;

discussions:

Some testing image are hard to detect. For example, some object is hide from other object. It is really hard to detect evan I (human vision) can not tell what is it immediately.

The program is still need to improve. I did not leave too much comment in the source code because I did it in two days, so it would be a liitle bit hard to understand my program.