object distinguishing and arranging system

Microcontroller Lab Report

Pages 1–3
2019

Md. Ahasanul Alam(10) , Mustafizur Rahman(22)

Microcontroller Lab(CSE-3116)

University Of Dhaka

27/4/2019

Keywords: Servo Motor,Sonar Sensor,DC Motor,Conveyor belt,Camera,Arduino Board,Motor Shield Driver,Breadboard,Power Supplier,Image Processing,Keypoint,Descriptor etc

ABSTRACT

In this project we made an autonomous system which can differentiate between different objects and pass them through different gates according to their similarity. In this paper we will discuss about the theory needed behind the project. Besides, how we have done our project, result analysis, difficulties we have faced, accuracy etc .

1 INTRODUCTION

Object differentiating and arranging plays big role in industry. So, an automated system in that field will be a breakthrough to the industry. Though, there are some automation system working in the industry e.g color detection etc. But, if we can use AI or Image Processing to differentiate object then it will speed up the whole system and can provide diversified use in industry and in other fields too such as an automated security system. Therefore, we tried to build an automated system using image processing which can differentiate objects and also put them in different baskets based on their likeness. We emphasized on color and shape as differentiating parameter.

2 THEORY

2.1 Image Processing Part

We took help of SIFT (Scale Invariant Feature Transform) algorithm.[1] to distinguish different objects.This algorithm was proposed by D.Lowe, in 2004. This algorithm actually extracts keypoints and computes descriptors. This algorithm is scale invariant in that sense is the features extracted will remain invariant in terms of zoom in or zoom out.
What is Keypoint and descriptor?
A selected image region which is detected as SIFT feature is called Keypoint. A 16 X 16(not constant) neighbourhood points around keypoint is called keypoint descriptor.
A SIFT algorithm mainly involves in four steps:

1. Scale-space Extrema Detection:
   This step finds potential keypoint at (x,y) at different scaling parameters. After doing different operations on image a point will be called probable keypoint if that point is local extrema i.e the pixel has extreme value among it's 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scale.
2. Keypoint Localization:
   Actually it is a filtering process. it filters all the probable keypoints and keeps only the strong interest points.
3. Orientation Assignment:
   Now an orientation is assigned to each keypoint to achieve invariance to image rotation.
4. Keypoint Descriptor:
   In this phase keypoint descriptor is assigned with each keypoints.

Keypoint matching:
Keypoints between two images are matched using the keypoint descriptor of the two images. If two keypoints matches well within a certain threshold value in our case 0.9 then they can be called good points.

2.2 Algorithm for object distinguishing

```
1. Pick images of 1st object and save them
2. Pick images of 2nd object and save them
3. Run SIFT algorithm for object 1 and get keypoint1 and descriptor1
4. Run SIFT algorithm for object 2 and get keypoint2 and descriptor2
5. Pick image of a random object between 1st and 2nd object and save that
6. Run SIFT algorithm for that random object and get keypoint and descriptor
7. Find good matches between keypoint1 and keypoint and derive percentage1 = no.of good points/min(no.of keypoint1 , no.of keypoint)
8. Find good matches between keypoint2 and keypoint and derive percentage2 = no.of good points/min(no.of keypoint2 , no.of keypoint)
9. if percentage1 > percentage2 :
10.         then the random object is object1
11. else :
12.         the random object is object2
```

3 METHODS

3.1 Apparatus

3.1.1 Hardware: Servo Motor,Sonar Sensor,DC Motor,Conveyor belt,Camera,Arduino(mega) Board,Motor Shield Driver,Breadboard,Power Supplier(DC 10 volt),Partex board,wires,USB Cables.

3.1.2 Software: Arduino(IDE), Processing, Py-charm, Droid-Cam.

3.1.3 Connections: All the connections are described in fig-1.

3.2 Procedure

Procedure is described in fig-2.

4 RESULTS

We experimented our system with lot's of different objects with different color and shapes. We saw that around 65-70 % of time we get correct result and other times it provides wrong result. So, we can say that our system is 65-70 % accurate on object distinguishing.

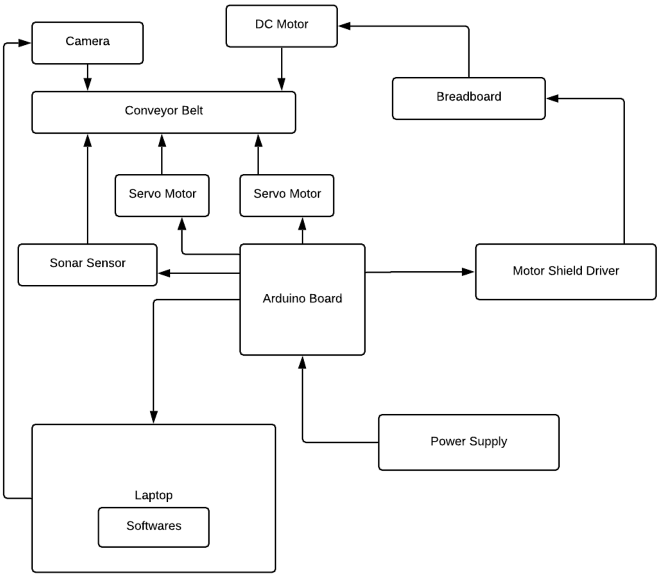

Fig. 1. Connection between different apparatus

5 DISCUSSION

As we have said that our system is 65-70 % accurate, but our expectation was more than that. But, the system works well when it doesn't have any effect of background or without conveyor belt that time it distinguishes object with 85 % accuracy. There are some limitation when the system works with conveyor belt, Firstly, when camera captures image on the belt there comes some unwanted region of the structure on which conveyor belt is situated. Secondly, the position of the object is not always fixed during image capture because when object comes with speed and sonar detects the object and the motor gets stopped. This time period is not always fixed and also the position of the object during motor stopping moment also depends on the size of the object.

We have also faced some difficulty during the project and came up with solution too.

1. Adding motor shield driver with arduino requires more than 5 volt. So, to solve that we used two 5 volt dc supply in series to achieve 10 volt dc supply.
2. Which algorithm should we use to differentiate different objects was another problem. So, after a week of study we understood SIFT algorithm is best to use in our case. Besides, it also has package in python opencv. As a result, it reduces the toughness of coding.
3. The most interesting problem was to decide when camera will trigger to capture image as the camera was an android phone. After a day of brainstorming we found that after detecting the object we need to write a value in arduino's serial output and processing software can detect what is written in serial monitor it reads that and writes to a file that value and python program reads that value in the file and decides whether to trigger camera or not. Exactly, reverse process is used to tell arduino board which gate it should open.

The result of our project can be used in other similar projects too. Besides, using machine learning algorithm we can improve the accuracy of the system.

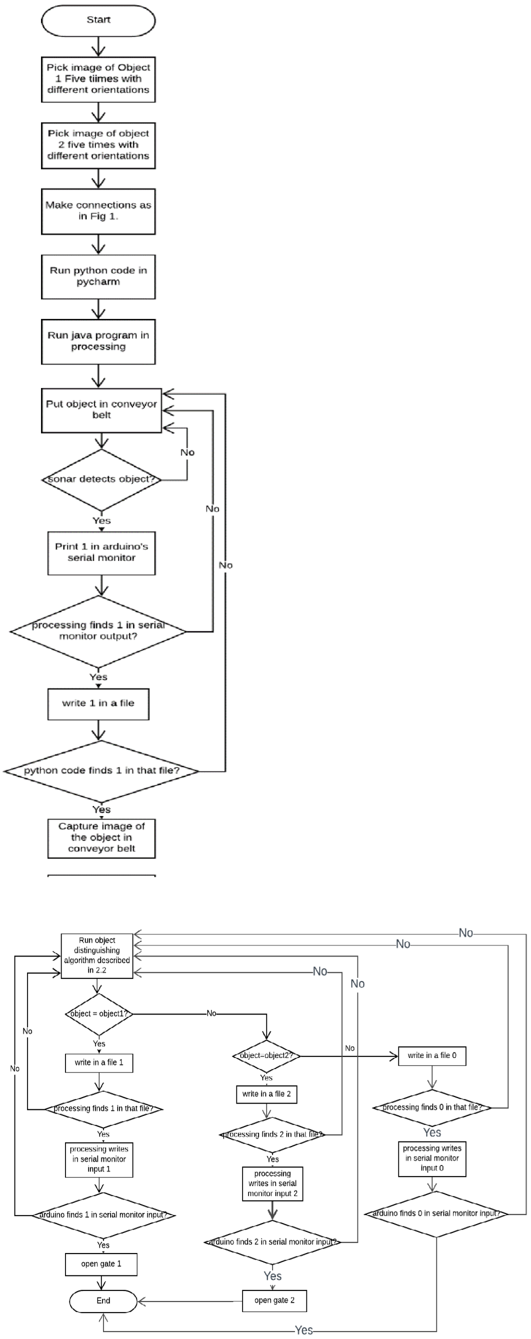
Start
Pick image of Object 1 Five tiimes with different orientations
Pick image of object 2 five times with different orientations
Make connections as in Fig 1.
Run python code in pycharm
Run java program in processing
Put object in conveyor belt
No
sonar detects object?
No
No
Yes
Print 1 in arduino's serial monitor
processing finds 1 in serial monitor output?
Yes
write 1 in a file
python code finds 1 in that file?
Yes
Capture image of the object in conveyor belt
Run object distinguishing algorithm described in 2.2
No
No
No
No
object = object1?
No
No
No
Yes
write in a file 1
object=object2?
No
write in a file 0
Yes
write in a file 2
processing finds 1 in that file?
processing finds 2 in that file?
processing finds 0 in that file?
Yes
Yes
Yes
processing writes in serial monitor input 1
processing writes in serial monitor input 2
processing writes in serial monitor input 0
arduino finds 1 in serial monitor input?
arduino finds 2 in serial monitor input?
arduino finds 0 in serial monitor input?
Yes
Yes
open gate 1
End
open gate 2
Yes

Fig. 2. Flowchart of the procedure

6 CONCLUSION

As we have told that object distinguishable and arranging autonomous system has demand in industry, So, our system is applicable in industry. Besides, more work can be done to improve the accuracy of the system.
It was a great pleasure for us to work in this project. We learned lot from this project. This learning experience will help in future to work on more projects.

ACKNOWLEDGEMENTS

We thank Dr. shugata ahmed and Dr. md. mosaddek khan sir to instruct us throughout the project. Also, we want to thank all the author of the references we have used throughout our project.

REFERENCES

[1]https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html
[2]https://pysource.com/2018/07/20/find-similarities-between-two-images-with-opencv-and-python/
[3]https://github.com/tishat-ahasan/10_22_microcontroller_project