PROJECT REPORT

IMAGE PROCESSING AND OBJECT DETECTION USING PYTHON

Project Submitted to:

Biranchi Panda
Department of Mechanical Engineering
IIT Guwahati

Manali Gogoi
January 2022
NIT Silchar, Assam

DECLARATION

I, Manali Gogoi, do hereby declare that this dissertation titled “Image processing and object detection using python” is a record of bona fide research carried out by me under the supervision and guidance of Sir Biranchi Panda and that it has not been previously submitted for the award of any degree, diploma or other similar title of recognition.

Date:
21-01-2022

Manali Gogoi

TABLE OF CONTENTS

INTRODUCTION

In this project, I tried to experiment the concepts of machine learning and OpenCV, and to explore how machine learning algorithms can be used in real world applications.

What is Image Processing?

First, we need to understand that how to handle images. Images are simply a collection of colors in red, green and blue format. As a human we see an image with some object or shape in it, but for computer it is just an array with color values range from 0 to 255.

The way computer sees anything is different from the way human see an image. But that's the good news for us because if we got an array of the image than it becomes simple for us to implement any algorithm on that array. Images define the world, each image has its own story, it contains a lot of crucial information that can be useful in many ways. This information can be obtained with the help of the technique known as **Image Processing**.

Image processing means processing the image and this may include many different techniques until we reach our goal.

The final output can be either in the form of an image or a corresponding feature of that image. This can be used for further analysis and decision making.

Python is one of the widely used programming languages for this purpose. Its amazing libraries and tools help in achieving the task of image processing very efficiently.

Image processing tools:

- OpenCV
- Scikit-image
- PIL/pillow
- NumPy
- Mahotas

➢ PROBLEM STATEMENT

Use few sequential images and find displacement of a structure.

➢ DESCRIPTION

A project-based image processing on in which we need to find the difference in position of a structure(building/statue/landmark) between 2 photos.
Steps to Perform Image Processing:

- Load images using Python or any other programming you are working on.
- Convert images into array.
- And finally apply some algorithm on that array.

Another good thing is that we have a library known as OpenCV which will help us to read the image and return array of color pixels.

➢ CODE

```
import cv2
import math
print(cv2.__version__)
#__version__is a variable and a property of the package

cap = cv2.VideoCapture(0)
#Tracker Initialization. cv2.VideoCapture() lets you create a video capture object which is helpful to capture videos through webcam and then you may perform desired operations on that video.
tracker = cv2.legacy.TrackerMOSSE_create()
#The MOSSE (Minimum Output Sum of Squared Error) tracker The implementation is based on Visual Object Tracking using Adaptive Correlation Filters.
success, img = cap.read()
#cap.read() returns a bool (True/False).
bbox = cv2.selectROI("Tracking",img,False)
#selectROI() is used to select a range of interest in an image manually by selecting the area on the image.
tracker.init(img,bbox)

c1x, c1y = int(int(bbox[0]) + int(bbox[2])/2),
int(int(bbox[1]) + int(bbox[3])/2)

print(c1x,c1y)

print(bbox)
```

```
def drawBox(img,bbox):
    x, y, w, h = int(bbox[0]), int(bbox[1]), int(bbox[2]),
int(bbox[3])
    cv2.line(img, (c1x, c1y), (int(x+w/2),int(y+h/2)), (0,
255, 0), 3)
   cv2.rectangle(img,(x,y),((x+w),(y+h)),(255,0,255),3,1)
cv2.putText(img,"TRACKING",(75,75),cv2.FONT_HERS
HEY_SIMPLEX,0.7,(0,255,0),2)
    cv2.putText(img,str(int(math.dist([c1x, c1y],
[int(x+w/2),int(y+h/2)]))),(75,100),cv2.FONT_HERSHEY
_SIMPLEX,0.7,(0,255,0),2)
#line(), rectangle(), FONT_HERSHEY_SIMPLEX, putText are
the required drawing functions.

while True:
    timer = cv2.getTickCount()
#The cv.getTickCount function returns the number of clock-
cycles after a reference event to the moment the function is
called.
    success, img = cap.read()

    success, bbox = tracker.update(img)
#Updates the tracker and finds the new most likely bounding
box for the target.

    if success:
       drawBox(img,bbox)
    else:
```

```
            cv2.putText(img,"LOST",(75,75),cv2.FONT_HERSHEY_S
            IMPLEX,0.7,(0,0,255),2)

               fps = cv2.getTickFrequency()/(cv2.getTickCount()-
            timer)
#The cv.getTickFrequency function returns the frequency of
clock-cycles, or the number of clock-cycles per second
            #cv2.putText(img,str(int(fps)),(75,50),cv2.FONT_HERS
            HEY_SIMPLEX,0.7,(0,0,255),2)
               cv2.imshow("Tracking", img)
#cv2.imshow() is used to display an image in a window.

               if cv2.waitKey(1) & 0xff==ord('q'):
#1 is the time in milliseconds and ‘q’ is the escape key
                  break
```

➢ METHODOLOGY

❖ ***STEP 1: Loading OpenCV***.

So first we have to load OpenCV package, after importing OpenCV first let's read an image by capturing images from the webcam.

(*OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. The library has more than 2000 optimized algorithms. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Will help us to load images in Python and convert them into array. Each index of array represents (red, green, blue) color pixel which ranges from 0 to 255*.)

❖ ***STEP 2: Tracker Initialization (type – MOSSE).***

Then we create a tracker object of type MOSSE This tracker requires a bounding box to initialize it with the elements it needs to track.

❖ ***STEP 3: Starting and Ending the loop.***

Then we enter the while loop that we can end by pressing 'q'.

(The image will open in same file so to open in new window we have written a separate code, i.e. first we started a while loop then end using cv2.waitKey. {Now what cv2.waitKey is doing} 1 is the time in milliseconds and 'q' is the escape key. So it means if we press escape key 'q', then loop is going to break and in last line it will

detroy or close all the windows that are opened by OpenCV.)

- ***STEP 3: Selecting the required structure.***
  Inside the while loop we use the tracker to draw the next frame from the webcam along with the updated bounding box.

- ***STEP 4: Measurement of the displacement.***
  A line is drawn from the center of the initial bounding box to current bounding box, and its length is calculated using the Euclidean formula.

➢ OUTPUT

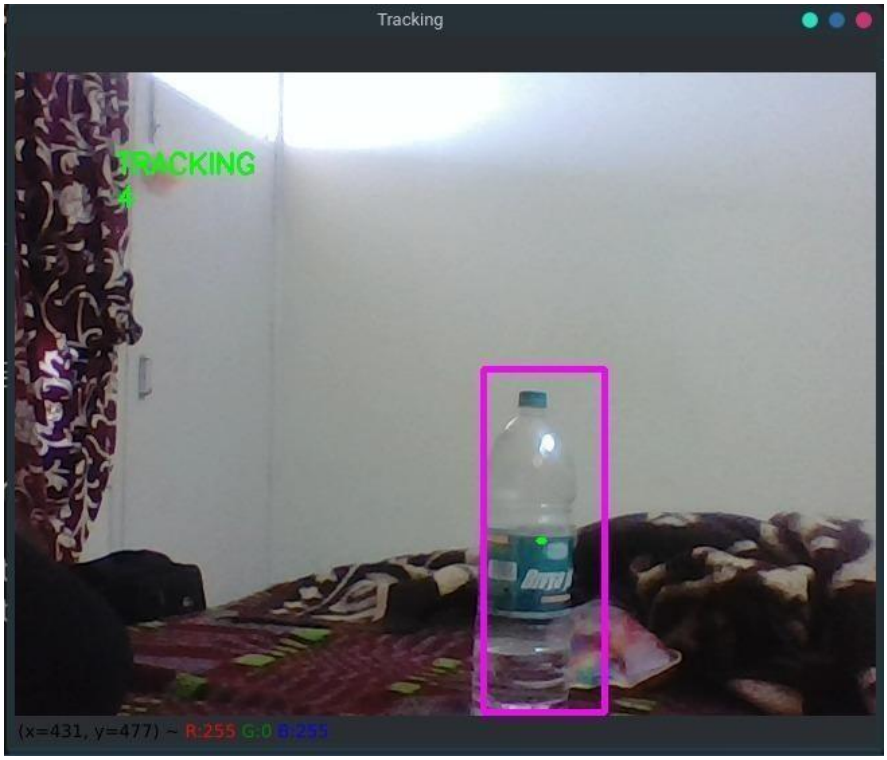

Fig 1: Picture of selected structure(bottle) slightly displaced

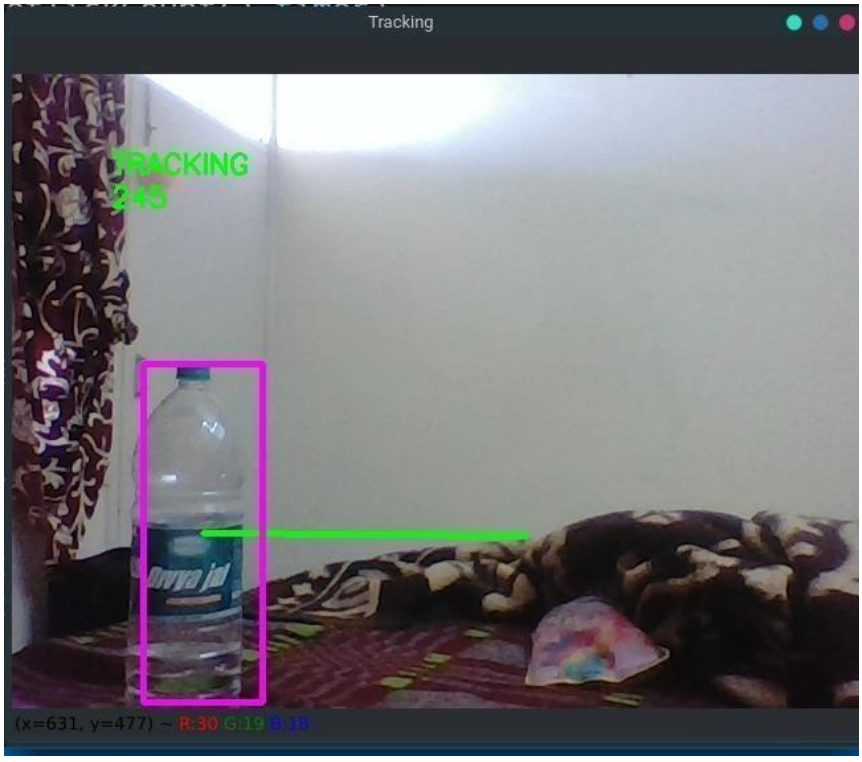

Fig 2: Picture of selected structure(bottle) further displaced

RESULTS

In the above images we can see that, at first the object is selected by dragging the mouse around the object.
Now, we move the object and a line is drawn from the center of the box tracking its distance from its original placed where it was selected.
In the first image (Fig 1), it is slightly moved by a distance of 4 pixels and then in the second image (Fig 2), we are moving it further to a distance of 245 pixels.

➢ REFERENCES

❖ Image Processing
Retrieved December 21, 2021,
from https://www.geeksforgeeks.org/image-processing/

❖ Open Source Computer Vision
Retrieved December 25, 2021,
from https://docs.opencv.org/3.4/

❖ Object Tracking Using OpenCV (MOSSE)
Retrieved December 29, 2021,
from https://learnopencv.com/tag/mosse/

❖ Visual Object Tracking using Adaptive Correlation Filters
Retrieved January 9, 2022, from
https://www.cs.colostate.edu/~draper/papers/bolme_cvpr10.pdf