

Project Report

Single-Object Tracking In Images

Name: Keshav Anand

Course: AI and ML

(Batch-4)

Duration: 12 months

Problem Statement: Single-object tracking in images using Mean-Shift .

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/>. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages `pip install -U scikit-learn` `pip install numpy` `pip install scipy` if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages `conda install -c scikit-learn` `conda install -c anaconda numpy` `conda install -c anaconda scipy`

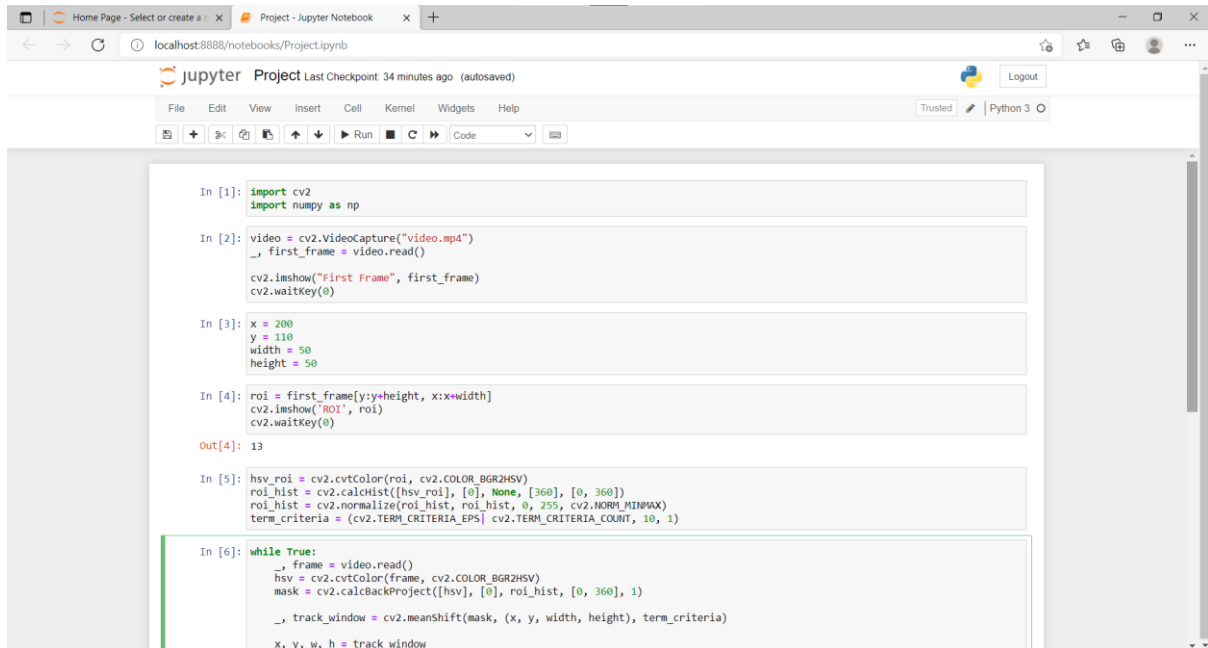
Dataset used:

The dataset used is a random video downloaded from the internet to perform single-object tracking in images obtained from the same video.

Method used for Tracking:

Mean-Shift

Screenshots of Source Code and Output:



```
In [1]: import cv2
import numpy as np

In [2]: video = cv2.VideoCapture("video.mp4")
_, first_frame = video.read()
cv2.imshow("First Frame", first_frame)
cv2.waitKey(0)

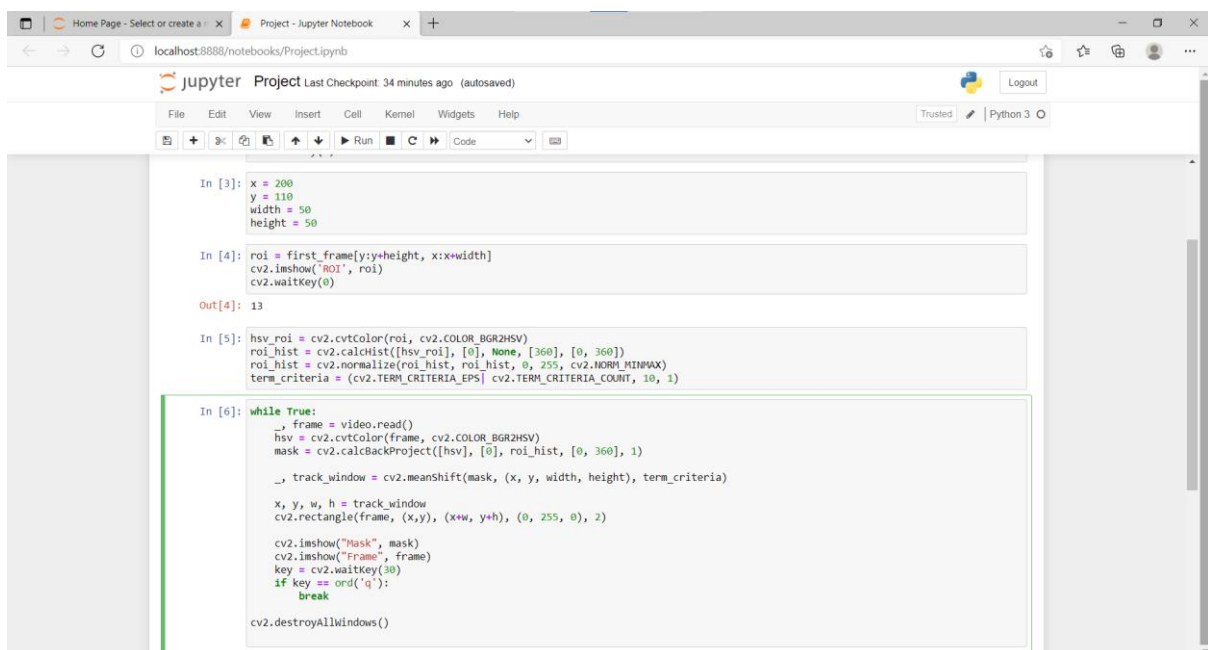
In [3]: x = 200
y = 110
width = 50
height = 50

In [4]: roi = first_frame[y:y+height, x:x+width]
cv2.imshow("ROI", roi)
cv2.waitKey(0)

Out[4]: 13

In [5]: hsv_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
roi_hist = cv2.calcHist([hsv_roi], [0], None, [360], [0, 360])
roi_hist = cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)
term_criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)

In [6]: while True:
_, frame = video.read()
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
mask = cv2.calcBackProject([hsv], [0], roi_hist, [0, 360], 1)
_, track_window = cv2.meanShift(mask, (x, y, width, height), term_criteria)
x, y, w, h = track_window
```



```
In [3]: x = 200
y = 110
width = 50
height = 50

In [4]: roi = first_frame[y:y+height, x:x+width]
cv2.imshow("ROI", roi)
cv2.waitKey(0)

Out[4]: 13

In [5]: hsv_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
roi_hist = cv2.calcHist([hsv_roi], [0], None, [360], [0, 360])
roi_hist = cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)
term_criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 1)

In [6]: while True:
_, frame = video.read()
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
mask = cv2.calcBackProject([hsv], [0], roi_hist, [0, 360], 1)
_, track_window = cv2.meanShift(mask, (x, y, width, height), term_criteria)
x, y, w, h = track_window
cv2.rectangle(frame, (x,y), (x+w, y+h), (0, 255, 0), 2)
cv2.imshow("Mask", mask)
cv2.imshow("Frame", frame)
key = cv2.waitKey(30)
if key == ord('q'):
break
cv2.destroyAllWindows()
```

Output:

