Assignment Project Report

Mean-Shift: Single Object Tracking in Image

Name: Pavan Tiwari
Course: AI and ML
(Batch 4)

- **Problem Statement**

  Using OpenCV implement a single object tracker. Steps to be implemented:
  a) Use a pre-recorded video or your webcam to have a video Capture object.
  b) Mark the region of interest (ROI or the object you want to track) using it coordinates in the first frame.
  c) Calculate the histogram of the ROI.
  d) Iteratively calculate the histogram at each location (using cv2,calcBackProject) and then apply mean shift to get the updated location of the ROI.

  Prerequisites

  - Software:
    - Python 3 (Use anaconda as your python distributor as well)
  - Tools:
    - Pandas
    - Numpy
    - Matplotlib
    - Seaborn
    - OpenCv
  - Dataset: : Video_ball.avi

- **Method Used**

  Object tracking is one of the most popular areas of video processing. The main purpose of object tracking is to estimate the position of the object in images continuously and reliably against dynamic scenes. This can be achieved by using the mean shift object tracking algorithm. The mean shift algorithm is an efficient approach to tracking objects whose appearance is defined by histograms. Thus, it can be used to track non-rigid objects by discovering clusters in a smooth density of samples

- **Implementation:**

1. Load all required libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import os
import warnings
warnings.filterwarnings('ignore')
```

2. Preprocessing data And Reading Dataset

```
from os.path import isfile, join

def convert_frames_to_video(pathIn,pathOut,fps):
    frame_array = []
    files = [f for f in os.listdir(pathIn) if isfile(join(pathIn, f))]

    for i in range(len(files)-1):
        filename = pathIn + files[i]
        img = cv2.imread(filename)
        height, width, layers = img.shape
        size = (width,height)
        print(filename)
        frame_array.append(img)

    out = cv2.VideoWriter(pathOut,cv2.VideoWriter_fourcc(*'DIVX'),fps, size)

    for i in range(len(frame_array)):
        out.write(frame_array[i])
```

```
pathIn = "./Ball_Dataset/"
pathOut = "video_ball.avi"
fps = 30.0
convert_frames_to_video(pathIn, pathOut, fps)
```

3. Reading the Feed and Applying Mean Shift Clustering

```
pathIn = "./Ball_Dataset/"
pathOut = "video_ball.avi"
fps = 30.0
convert_frames_to_video(pathIn, pathOut, fps)
```

```
video = cv2.VideoCapture("./video_ball.avi")

_,first_frame = video.read()
# cv2.imshow("First_Frame",first_frame)
# cv2.waitKey(0)
print(first_frame.shape)
```

```python
x = 200
y = 110
width = 50
height = 50
roi = first_frame[y:y+height, x:x+width]
# cv2.imshow('ROI',roi)
# cv2.waitKey(0)
```

```python
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)
```

```python
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 == 10:
        break
```

4. Output