HW3 Report

Student: Min-Hao (Mike) Huang

Experiment Results of baseline implementation

```
[ ]  1 # Evaluate your tracking performance
     2 !python eval.py dataset/test/gt/gt.txt results.txt

          IDF1   IDP   IDR   idtp   idfp   idfn  MOTA  MOTP  Rcll  Prcn IDs MT PT ML
MultiCam 51.71 52.27 51.16 8302.0 7580.0 7927.0 93.95 11.63 96.28 98.38 121 22  0  0
```

IDF1	MOTA	IDF1*MOTA
51.71	93.95	4858.1545

I trained yolov8n model on SportsMOT dataset for 20 epochs and I used IoU_tracker() as tracker to implement the tracking experiment. The conf threshold is set to 0.5 and IoU threshold is set to 0.3. The result shown above turns out that it is not quite well because of the poor IDF1 score, which indicates that the model struggles with tracking association accuracy.

I tried several methods to enhance the tracking performance, including adjusting threshold values and replace baseline tracklet with Kalman filter tracklet. However, these adjustments led to only minimal improvements.

Therefore, I tried an open-source method given in the reference, which is Deep_EIoU, and it achieved huge improvement. Here is how I approached it:

1. Git clone the repo and follow the README to set up the environment (also fix some np.float bugs, stupid google colab)

```
1 !git clone https://github.com/hsiangwei0903/Deep-EIoU.git
2 import sys
3 sys.path.append('/content/drive/MyDrive/HW3/Deep-EIoU/')

Cloning into 'Deep-EIoU'...
remote: Enumerating objects: 455, done.
remote: Counting objects: 100% (55/55), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 455 (delta 12), reused 2 (delta 1), pack-reused 400 (from 1)
Receiving objects: 100% (455/455), 36.87 MiB | 11.65 MiB/s, done.
Resolving deltas: 100% (64/64), done.
Updating files: 100% (377/377), done.

1 %cd /content/drive/MyDrive/HW3/Deep-EIoU/Deep-EIoU/reid
2 !pip install -r requirements.txt
3 !pip install cython_bbox
4 !python setup.py develop
5 !cd ..
```

2. Modify the path to dataset/test/ in demo.py
3. Run demo.py

```
# run demo.py
%cd /content/drive/MyDrive/HW3/Deep-EIoU/Deep-EIoU
!python tools/demo.py
```

4. Post process the output file (result_deep_eiou.txt) to match the ground truth's data format

```
# post-process the output file
input_file_path = '/content/drive/MyDrive/HW3/Deep-EIoU/Deep-EIoU/YOLOX_outputs/yolox_x_ch_sportsmot/track_vis/2024_11_10_17_32_28.txt' # modify when needed
output_file_path = '/content/drive/MyDrive/HW3/results_deep_eiou.txt'

with open(input_file_path, 'r') as file:
    lines = file.readlines()

modified_lines = []
for line in lines:
    parts = line.strip().split(',')
    parts[0] = str(int(parts[0]) + 1)  # frame+1
    # Set the last three columns to 1
    parts[-1] = '1'
    parts[-2] = '1'
    parts[-3] = '1'
    del parts[-4] # remove score
    modified_lines.append(','.join(parts))

# write to new file
with open(output_file_path, 'w') as file:
    for modified_line in modified_lines:
        file.write(modified_line + '\n')

print(f"Modified data saved to {output_file_path}.")
```

5. Evaluate the tracking performance

```
# Evaluate your tracking performance
%cd /content/drive/MyDrive/HW3
!python eval.py dataset/test/gt/gt.txt results_deep_eiou.txt
```

```
/content/drive/MyDrive/HW3
          IDF1   IDP   IDR    idtp   idfp   idfn  MOTA MOTP  Rcll  Prcn IDs MT PT ML
MultiCam 81.13 81.50 80.75 13105.0 2974.0 3124.0 98.68 5.98 98.96 99.88  27 22  0  0
```

IDF1	MOTA	IDF1*MOTA
81.13	98.68	8005.9084

As we can see from the tracking result, the tracking performance enhanced significantly. This is likely because of the pretrained detector and tracker being trained on the same type of images and videos, using large dataset and computational resources.

Reference:

Huang, Hsiang-Wei, et al. "Iterative scale-up expansioniou and deep features association for multi-object tracking in sports." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2024.

GitHub - hsiangwei0903/Deep-EIoU