Machine Learning: Video Object Detection

- **Objective:** Detect sports balls in video footage.
- **Instructions:**
    - Generate pseudo-annotations for video data.
    - Build an object detection pipeline that can train on your annotations.
    - Create an evaluation procedure with visualizations of predictions and annotations.
    - Design a process to iteratively improve your dataset.
- **Dataset:** Sports-1M
- This task is perfect for showcasing your computer vision and machine learning skills, especially if you're creative with your annotations and improvement pipeline!

Preface

I chose this project because it fits my scope. Sports ball can have many types, but I only chose football to be my example target object. However, the method can be extended to any other ball games.

Introduction

Pseudo-labelling uses trained models to predict labels from our custom data. The models have not been trained on our data so they are weak labellers; in other word, they do not produce ground truth labels, but the label quality can be "close to ground truth". Thus, we can use them to produce semi-decent, massive amount of data, at very low cost (since there is no human labour involved).

In this project, I use Segment Anything Model 2 (Ravi et al., 2024) to produce labels. It has powerful image encoder that the model can take in prompts from the users and produce corresponding masks in the proceeding frames. The prompts can be pixel locations, bounding boxes, or masks in the first frame of the videos.

The object detector I used is Yolov7 since it is fast and light for real-time detection. My object detection pipeline inspired by (Ferreira et al., 2023) is as followed:

- Step 1. Use SAM2 to produce two groups of data,

(1) The first group is train and validation set; their labels are verified by human eyes.
(2) The second group is pseudo-annotated set. Although SAM2 has produced its labels, they are not verified by human eyes. (I currently don't do anything with these labels but in the future the data can be added to the train set if human verifies its quality)

- Step 2. Train and Validate Yolov7
- Step 3. Use trained Yolo to predict confidence for (2)
- Step 4. Add high-confidence samples from (2) to the train set in (1). Retrain model on the bigger train set and check if validation results are improved.
- Step 5. Iterative Step 3 and 4 to get an improved model.

Note that SAM2 can fail to predict correct labels. We tell SAM2 the coordinates of the ball in the first frame, and it detects it in all subsequent frames. However, some of the videos fade between different shots, and we would need to re-input the coordinates of the ball in the new shot or SAM2 will make poor predictions.

Experiment setup and results

I installed two software from the following repositories:

- SAM2: https://github.com/facebookresearch/sam2/tree/main
- Yolov7: https://github.com/WongKinYiu/yolov7

Firstly, I used SAM2 to create [a custom dataset] (total: 579 samples) which holds train/validation/test and pseudo-annotated folders. Labels in the train_val_test/ dataset are verified by human eye by using this [GUI] which is custom-made. Therefore, they are treated as ground truth. Then, I moved the custom dataset to Yolov7 folder. There are few files need to modify to train the model:

(1) Download a pretrained [weight]
(2) Create [dataset.yaml] and put it to custom dataset folder if it is not there
(3) In the terminal, we train 20 epochs with a pretrained weight yolov7.pt:

```
python train.py --img-size 640 --cfg cfg/training/yolov7.yaml --hyp
data/hyp.scratch.custom.yaml --batch 16 --epochs 20 --data
custom_data/dataset.yaml --weights yolov7.pt --workers 24 --name yolo_ball_det
```

After training, we can visit this [notebook] for iteratively add more data to train set and improve our benchmark through the next-run, or future-run trainings. For reproducing without training, download [first run best.pt] and [second run best.pt]. Then, we can run the following command line to visualise our object detector performance:

```
python detect.py --weights runs/train/yolo_ball_det_2/weights/best.pt --conf 0.25 --
img-size 640 --source association_football_4.mp4
```

There are few tasks to think about for future improvement.

1. Because SAM2 can fail to predict correct labels when background drastically change in the frame, it took me a while to identify and [group images] from similar video shots. We can consider retrain SAM2 to get better mask/bounding-box detection ability so it can deal with background shift better.
2. Since val/test sets are small, it cannot represent real-world data distribution. Therefore, we can always add more val/test data whenever new ground truth data is accessible. By doing so, our model can produce more convincing validation and testing results.
3. We can write a script to automate training sessions after each run of data collection

Bibliography

Ferreira, R. E. P., Lee, Y. J., & Dórea, J. R. R. (2023). Using pseudo-labeling to improve performance of deep neural networks for animal identification. *Scientific Reports*, *13*(1), 13875. https://doi.org/10.1038/s41598-023-40977-x

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). *SAM 2: Segment Anything in Images and Videos*.