Assignment comments

Person re-ID

I assume that in the timeframe I cannot train anything of my own' but rather try to find suitable pre-trained models.

I am using yolo network for the detections; trying several versions (none give perfect results). I invest some time in tunning the detection and tracking parameters, but there is limited benefit in that. To facilitate for the re-ID task, I assumed I will also need the actual masks, so I use YOLO with its instance segmentation option.

After YOLO is doing its on-line tracking, it is possible to look again at the whole video. In theory it can fix tracking errors, in reality I only removed few very short detections and fixed a single wrong re-ID.

The main focus is re-ID among clips, not inside a clip; this is done by comparing features of identification. I used a pre-trained model that I was advised that it is suitable for the task: it was trained on database that was built for re-ID, and using contrastive learning. The model is osnet_x_1_0 and it was trained in a dataset called msmt17. It yields feature (embedding) vectors of size 512. The input to the network is the detection bounding box, masked with the instance segmentation to avoid “feature contamination”.

The similarity measures the average pairwise distance between features, formally:

If there are N detections for ID1, and M detections for ID2, let $f_i^{ID1},\ 1 \leq i \leq N$ are the features for ID1 and $f_j^{ID2},\ 1 \leq j \leq M$ the feature for ID2

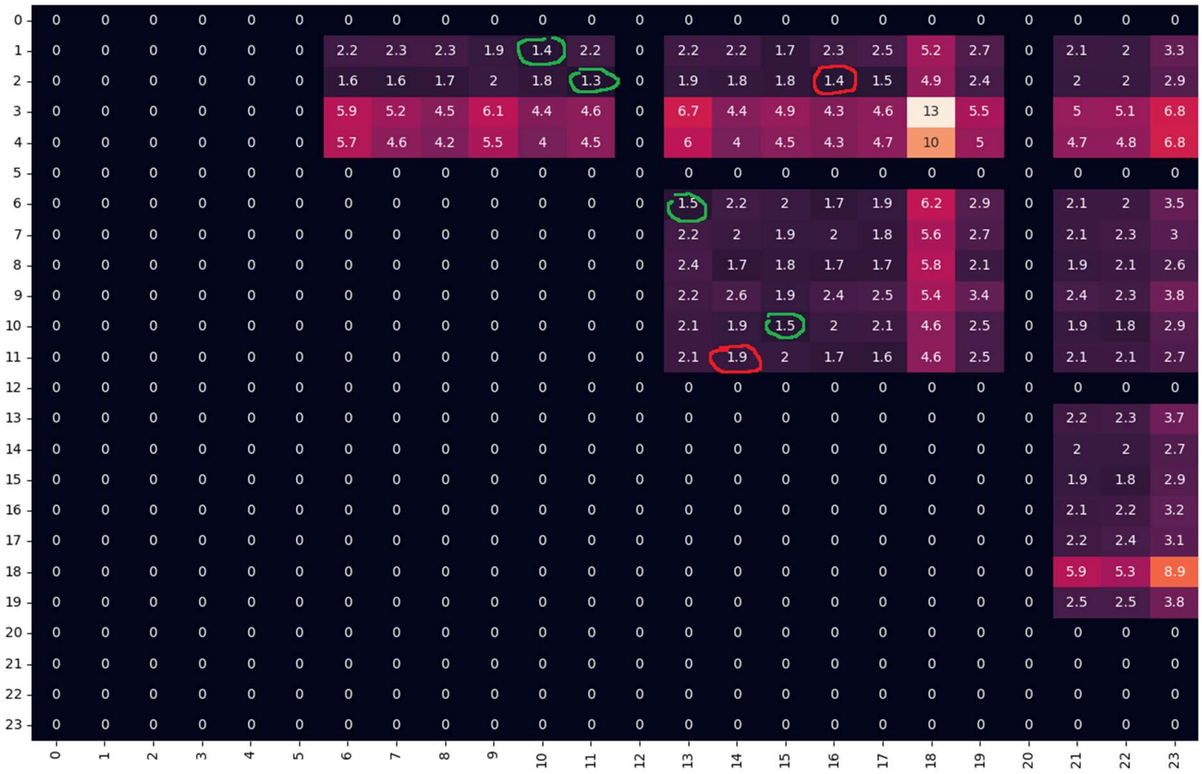
The average distance is $\frac{1}{MN}\sum_i \sum_j \left\| f_i^{ID1} - f_j^{ID2} \right\|$. This distance is further normalized by the (geometric) average of the pairwise distances inside the features of each ID, so the final score is:

$$\frac{\frac{1}{MN}\sum_i \sum_j \left\| f_i^{ID1} - f_j^{ID2} \right\|}{\sqrt{\frac{1}{M^2}\sum_{i\neq j} \left\| f_i^{ID1} - f_j^{ID1} \right\| \cdot \frac{1}{N^2}\sum_{i\neq j} \left\| f_i^{ID2} - f_j^{ID2} \right\| \cdot}}$$

This score is compared to a threshold, and if it falls below the threshold, then ID1 and ID2 are combined.

Here is a visualization of what I am getting:

(the greens are True positives; in red there are one False positive and one False negative).



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	2.2	2.3	2.3	1.9	1.4	2.2	0	2.2	2.2	1.7	2.3	2.5	5.2	2.7	0	2.1	2	3.3
2	0	0	0	0	0	0	1.6	1.6	1.7	2	1.8	1.3	0	1.9	1.8	1.8	1.4	1.5	4.9	2.4	0	2	2	2.9
3	0	0	0	0	0	0	5.9	5.2	4.5	6.1	4.4	4.6	0	6.7	4.4	4.9	4.3	4.6	13	5.5	0	5	5.1	6.8
4	0	0	0	0	0	0	5.7	4.6	4.2	5.5	4	4.5	0	6	4	4.5	4.3	4.7	10	5	0	4.7	4.8	6.8
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1.5	2.2	2	1.7	1.9	6.2	2.9	0	2.1	2	3.5
7	0	0	0	0	0	0	0	0	0	0	0	0	0	2.2	2	1.9	2	1.8	5.6	2.7	0	2.1	2.3	3
8	0	0	0	0	0	0	0	0	0	0	0	0	0	2.4	1.7	1.8	1.7	1.7	5.8	2.1	0	1.9	2.1	2.6
9	0	0	0	0	0	0	0	0	0	0	0	0	0	2.2	2.6	1.9	2.4	2.5	5.4	3.4	0	2.4	2.3	3.8
10	0	0	0	0	0	0	0	0	0	0	0	0	0	2.1	1.9	1.5	2	2.1	4.6	2.5	0	1.9	1.8	2.9
11	0	0	0	0	0	0	0	0	0	0	0	0	0	2.1	1.9	2	1.7	1.6	4.6	2.5	0	2.1	2.1	2.7
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.2	2.3	3.7
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2.7
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.9	1.8	2.9
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.1	2.2	3.2
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.2	2.4	3.1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5.9	5.3	8.9
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.5	2.5	3.8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I have tried few more directions, but so far without success:

- Weighting the feature vectors. I have tried weighting based on the detection mask size, the reasoning behind that is the assumption that features based on larger images are more representative; I did not try to weight according to detection confidence.
- I have tried to estimate average and covariance matrix for each feature set, and use them to calculate several distance measures between (gaussian) distributions, but the results were very bad, probably the distribution of features is far from gaussian.
- It calls for non-parametric estimation of distance between distributions, this is very time consuming (computationally), but maybe it has some potential.

Crime detection

I gave up general crime scene detection. I did not even find datasets for general “crime scene”; there are some datasets for activity detection, but the activities are more concrete.

I only tried to identify weapons, meaning I gave up detecting shop lifting, but do intend to detect the armed robbery.

The dataset I found for weapon detection is https://www.kaggle.com/datasets/kruthisb999/od-weaponsdetection?resource=download ; I have used it to train YOLOv8n (larger models takes more time to train than I had).