A Two-step Model for Multi-object Tracking

Shuaishuai Zhang
School of Automation, Southeast University; Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China
zs2712784665@163.com

Xiaobo Lu
School of Automation, Southeast University; Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China
xblu2013@126.com

Songlin Du
School of Automation, Southeast University; Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, China
sdu@seu.edu.cn

ABSTRACT

Multi-object tracking is widely used in video analysis. However, due to the limitation of detector performance, many multi-object tracking models have the problem of detecting two objects into one object in some occlusion scenes. In this paper, we propose a two-step model for handling this problem. In the first step model, the non-occlusion targets are detected and embeddings are extracted, while the occlusion areas are identified. The second step model processes the occlusion areas to obtain occlusion targets' accurate positions and embeddings. Finally, we integrate and optimize the output results of the two steps models. Experiments show that the number of false positives and missed positives in our model's object detection is significantly reduced. The multi-object tracking performance (MOTA metric) is improved by nearly 3% compared with other models.

CCS CONCEPTS

• **Computing methodologies, Artificial intelligence, Computer vision, Computer vision problems**;

KEYWORDS

Two-step model, Occlusion prediction, Feature clipping, JDE

ACM Reference Format:
Shuaishuai Zhang, Xiaobo Lu, and Songlin Du. 2021. A Two-step Model for Multi-object Tracking. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487075.3487083

Multi-object tracking is an important part of surveillance video analysis. It can not only be directly used for object motion trajectory analysis but also serve as a research basis for many high-level tasks, such as object action recognition, behavior analysis, safety supervision and so on.

To complete the task of multi-object tracking, the strategies of tracking by detection are proposed in many mainstream deep learning algorithms [1][2]. Specifically, these methods divide multi-object tracking into detection and embedding modules. The detection module completes object detection and the embedding module uses relevant algorithms to Extract the features of the object. In this way, there may be many double calculations between the two modules, and the running speed may be affected. For this reason, some scientists have proposed methods to integrate the detection module and the embedding module into a neural network [3][4]. The two modules share the same underlying characteristics, thus avoiding double counting and improving the performance. However, due to the limitations of their own detection framework, these methods are not very effective in the detection and tracking of occlusion objects in some occlusion scenarios. Specifically, the detection framework often detects two occluded objects as one object, which also creates some problems for object tracking. Aiming at the occlusion object detection problem, some scientists have proposed improved non-maximum suppression algorithms, such as soft-NMS [5], softer-NMS [6], adaptive-NMS [7], etc. Some have made some improvements to the loss function, for example repulsion loss function [8]. Some have established links between proposals [9]. But these methods depend on the prediction results of the original detection network. Therefore, for some scenes, the effect of occlusion detection is not good.

In this paper, we designed a two-step neural network for tracking occluded pedestrians based on the JDE [10]. Specifically, targets occlusion score is added to the prediction heads of the JDE (Joint Learning of Detection and Embedding mode) [10] and this model is used as the first step model in our network. When the pedestrians are judged to be occlusion, the feature map areas corresponding to the pedestrians bounding boxes are extracted as the input of the second step network. The second step model carefully and accurately processes the occluded areas and outputs the coordinates and embeddings of the occluded pedestrians. Finally, we integrate the output results of the two steps and do some optimization processing to complete pedestrians tracking.

The rest of the paper is organized as follows: In section 2, we introduce the specific details of our proposed method. In section 3, the training strategy and experimental results are presented. The conclusion is followed in section 4.

1 METHOD

1.1 The Overall Architecture

Our method is a two-step neural network. In the first step, the pedestrian occlusion score is added to the prediction heads of the JDE [10] so that the first step model can use the occlusion

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487083

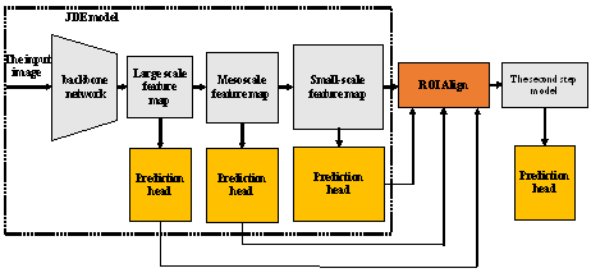

Figure 1: The Overall Architecture Structure of the Two-Step Model.

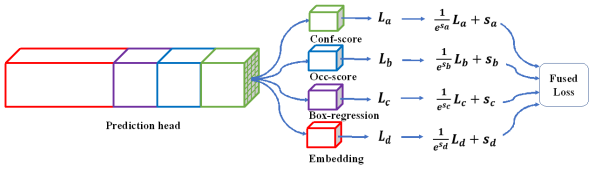

Figure 2: Prediction Head and Loss Function.

score to determine whether there is an occlusion in the predicted pedestrians bounding boxes. Then, the results of the occluded pedestrian bounding boxes after non-maximum suppression are mapped to the small-scale feature map in JDE [10] and are cropped by ROI Align [11] as the input of the second step model. The second step model processes the clipped feature maps and completes the task of occlusion pedestrian detection while embeddings are extracted. Finally, we synthesized the output results of the two steps models and optimized them to get the final results. The overall architecture of our model is shown in Figure 1.

1.2 JDE Model and Occlusion Prediction

The full name of JDE [10] is Joint Learning of Detection and Embedding mode which combines object detection with embedding extraction. Specifically, based on the framework of yolo-v3 [12], it adds an embedding extraction branch to the output branch of the model, so that it can simultaneously detect the objects and obtain the embeddings corresponding to the objects. It uses the cosine distance between embeddings as the matching basis, and the Hungarian algorithm is used to match the objects between the before and after frames. In our model, we still use this matching rule.

In our first step model, occlusion score is added to the JDE [10] prediction heads to determine whether the pedestrians in the corresponding bounding box have occlusion. We define occlusion and non-occlusion as two categories and adopt cross entropy loss for training. Unlike other models, our first step model encourages two or three occluded pedestrians to be placed in the same bounding box, but this bounding box is marked occlusion. Its bounding box and the bounding box of a single pedestrian are regression simultaneously. The loss function weighting strategy is similar to JDE [10]. Specifically, using the concept of task-independent uncertainty, an automatic learning reduction scheme [13] is adopted. The total loss function of the first step model can be described as follows:

$$L_{the\ first\ step} = \sum_{i}^{M} \sum_{j=a,b,c,d} \frac{1}{2}\left(\frac{1}{e^{s_j^i}} L_j^i + s_j^i\right) \tag{1}$$

Where M is the number of prediction heads and L_j^i ($j = a, b, c, d$) represents confidence loss, occlusion classification loss, bounding box regression loss, and embedding loss respectively, s_j^i is the task-dependent uncertainty of each component loss, which is modeled as a learnable parameter. The modified prediction head and loss functions are shown in Figure 2.

1.3 Feature Map Mapping and Clipping

Since the small-scale feature map retains more information of the objects, we choose to crop the mapping areas of occlusion bounding boxes on the small-scale feature map [14]. Our model uses the non-maximum suppression algorithm to process the bounding boxes marked as occlusion and eliminate the bounding boxes that are not mapped to the small-scale feature map.

ROI Align algorithm is proposed to complete the fine clipping of feature map in [11]. It overcomes the precision loss of ROI pooling [14] algorithm due to decimal rounding in feature map clipping process. Specifically, it uses bilinear interpolation to replace the decimal rounding operation, so that the value in the clipped feature map is more accurate. Since we need to accurately locate the occlusion pedestrians, which has a high requirement for the accuracy of the clipped feature map, we choose the ROI Align algorithm to complete this operation. In view of the requirement of the targets size in our scene, we resize the clipped feature maps to 15*35(width*height).

1.4 The Second Step Model

The purpose of the second step model is to obtain the bounding boxes of the occlusion pedestrians and the corresponding embeddings more accurately. We designed the network structure as shown in Figure 3. The clipped feature maps have two branches after passing through a convolution block, one for confidence prediction and the regression of the pedestrian bounding boxes, and the other is used to generate the embeddings corresponding to the occlusion pedestrians. The bounding boxes still choose to be regression by the anchors and we select three anchors covering different scales (11*28,7*18,3*8) on the clipped feature maps. Both branches are processed by a number of convolution blocks. The convolution block consists of a convolution layer, a Batch Normalization layer and an activation layer that uses the Leaky ReLU function.

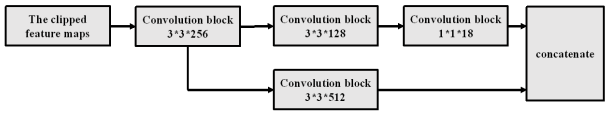

Figure 3: The Second Step Model Structure Diagram. The Number under the Convolution Block Represents the Size and Number of Convolution Kernel.

1.5 Loss Function of The Second Step Model

Our second step model has only one prediction head including confidence prediction, bounding boxes regression and embedding. The weighted strategy of the loss function is the same as that of the first step model. The calculation formula of loss function is shown below:

$$L_{the\ second\ step} = \sum_{i=1,2,3} \frac{1}{2}\left(\frac{1}{e^{s_i}}L_i + s_i\right) \tag{2}$$

where L_i $(i = 1, 2, 3)$represent confidence loss, bounding box regression loss, and embedding loss respectively. s_i Is a weighted parameter for each task. However, in addition to the original SmoothL1 loss, we have added two items to calculate the bounding box regression loss. The total bounding boxes regression loss is calculated as shown below:

$$L_{box} = L_{smooth_{L_1}} + \alpha * L_{RepGT} + \beta * L_{RepBox} \tag{3}$$

where $L_{smooth_{L_1}}$ represents the loss calculated by the SmoothL1 function, while L_{RepGT} and L_{RepBox} are the two penalty terms of the bounding boxes regression mentioned in [8]. The coefficients α and β are weight parameters. The purpose of the SmoothL1 loss is to make the predicted bounding boxes approximate to the real object's bounding boxes. Its calculation formula is shown below:

$$L_{smooth_{L_1}}(x) = \begin{cases} 0.5x^2 & if \quad |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{4}$$

where x is the difference between the true value and the predicted value. The RepGT loss is designed to make the predicted bounding boxes and its neighboring ground-truth objects which are not its target mutually exclusive. Its calculation formula is shown as follows:

$$L_{RepGT} = \frac{\sum_{P \in P_+} Smooth_{L_n}\left(IoG\left(B^P, G^P_{Rep}\right)\right)}{count\left(P_+\right)} \tag{5}$$

$$Smooth_{Ln}(x) = \begin{cases} -\ln(1-x) & x \le \sigma \\ \frac{x-\sigma}{1-\sigma} - \ln(1-\sigma) & x > \sigma \end{cases} \tag{6}$$

$$IoG(B, G) = \frac{area(B \cap G)}{area(G)} \tag{7}$$

where P_+ is the set of all positive proposals(anchors) which used to regress to the targets' bounding boxes. B^P is the predicted box. G^P_{Rep} is defined as a proposal P' which has the largest IoU region with its target bounding box except the positive proposal used to regress to this target bounding box. $\sigma \in [0, 1)$.

The RepBox loss is aimed at keeping predicted bounding boxes that regression to different objects as far apart as possible. Its calculation formula is as follows:

$$L_{RepBox} = \frac{\sum_{i \neq j} Smooth_{L_n}\left(IoU\left(B^{p_i}, B^{p_j}\right)\right)}{\sum_{i \neq j} count\left(IoU\left(B^{p_i}, B^{p_j}\right) > 0\right) + \varepsilon} \tag{8}$$

where B^{p_i}, B^{p_j} is the predicted box regressed from proposal p_i, p_j respectively. ε is a small constant used to prevent division by zero.

These three loss functions coordinate with each other and have a positive effect on the regression of the occlusion pedestrian bounding boxes.

1.6 Some Optimization Measures

We remove the bounding boxes in the prediction head of the second step model where the area is too small and the coordinates are not in the clipped feature maps. Soft-NMS [5] algorithm is first used to process the bounding boxes of the second step model's prediction head and then all the bounding boxes of the two-step model's prediction heads. In the process of merging the output results of the two models, we also did some optimization. When no target is detected in the clipped feature maps, there is no need to discard the bounding boxes judged as occlusion in the first step model's prediction heads. At the same time, for each clipped feature map, the maximum number of effective bounding boxes in the corresponding prediction head of the second step model is set to 3. For target matching and trajectory generation, our model still adopts the matching rules of JDE [10].

2 TRAINING STRATEGY AND EXPERIMENTAL RESULTS

2.1 Training Strategy

Our model training is divided into two stages. The first stage is to complete the training of the first step model, and the second stage is to complete the training of the second step model. We used our own private dataset and the MOT16-04 [2] dataset as the training set. Our private training data set consists of four scenes, each of which is a sequence of 1,000 frames extracted from surveillance video taken from a public place. These scenes all have some occluding targets. Before the image is input into the model, the image has undergone color dithering, random scale transformation, rotation and other data enhancement operations to suppress overfitting. During the training of the first step model, we set the batch size to 4 and trained a total of 200 epochs. The initial learning rate was 0.01 and the learning rate decreased by 10 times in the 60th, 120th and 160th epochs. Only the loss function of the first step model is used to calculate the loss, while the second step model did not carry out training. The change of losses in the training process is shown in Figure 4. On the basis of the completion of the first step

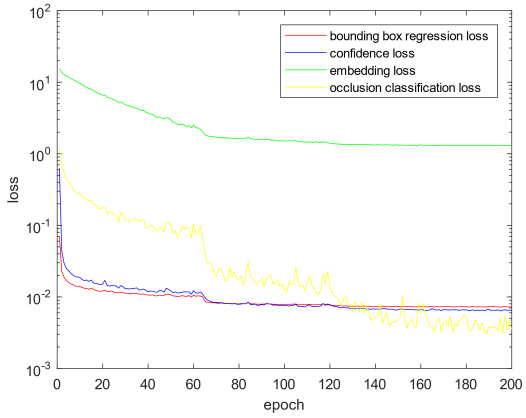

Figure 4: The Change of Losses of the First Step Model's Training.

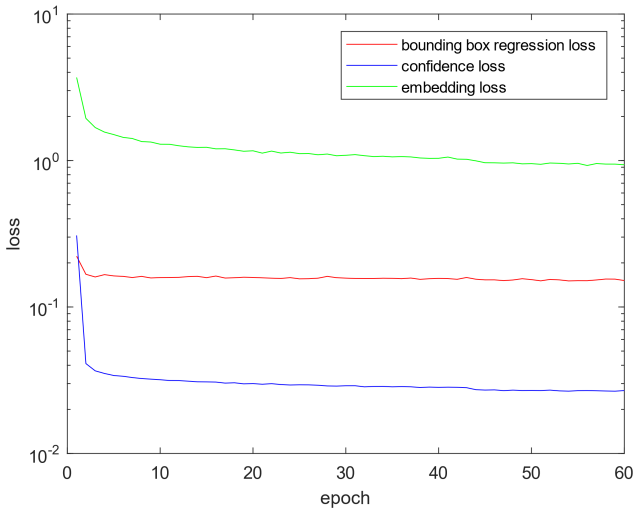

Figure 5: The Change of Losses of the Second Step Model's Training.

Figure 6: The Detection Effect of the Two-Step Model. The Left Figure is the Target Marked as Occlusion Detected by the First Step Model and the Right Figure is the Target Identified after the Second Step Model Processing.

model training, the second step of model training was carried out. The training sample of the second step model is the clipped areas of the small-scale feature map in the first step model. We froze the gradient of the first step model and only used the first step model to complete the task of generating the small-scale feature map. The batch size of the second step model is still set to 4, but the initial learning rate was set to 0.001, for a total of 60 epochs of training. The learning rate dropped by a factor of 10 in epochs 30 and 45. We also only use the loss function of the second step model to calculate the loss. The loss curve of the second-step model is shown in Figure 5.

2.2 Experimental Results

We selected 2000 frames of images from two scenes in our private dataset as the test set. Figure 6 shows the detection effect of the two-step model. The visualization results of multi-object tracking are shown in Figure 7. Our model uses MOTA index to evaluate multi-target tracking performance and is compared with the JDE [10] and the yolo v3 [12] +deep-sort [15]. The comparison results are shown in Table 1. FP and FN represent the number of false positives and false negatives in object detection, respectively. FPS stands for the number of frames per second processed by the model. It can be seen that compared with other models, our model has better detection and tracking performance.

3 CONCLUSION

In this paper, we propose a two-step model for multi-object tracking. The first step model is used to detect the non-occluded pedestrians and generates the corresponding embeddings, while the second step model is used to accurately locate the occluded pedestrians and generates its embeddings. We integrate the output results of the two models and carry out related steps after optimization to finally complete the task of multi-object tracking. The experimental results show that our model has superior detection and tracking performance compared with other models. However, our model also has many limitations in extremely crowded pedestrian scenarios, which is also a problem for us to solve in the future.

ACKNOWLEDGMENTS

This work is supported by National Key R&D Program of China under Grant 2020YFB1600700, Major scientific research projects of China Railway Group under Grant K2019G046, the National Natural Science Foundation of China under grant 62001110, and the Natural Science Foundation of Jiangsu Province under grant SBK2020041044.

Table 1: Comparison of Detection and Tracking Performance of Different Models

Model	FP	FN	MOTA	FPS
Yolo v3+Deep-sort	1373	6930	73.1%	7.8
JDE	1258	6953	73.7%	16.5
ours	1033	6439	76.3%	13.4

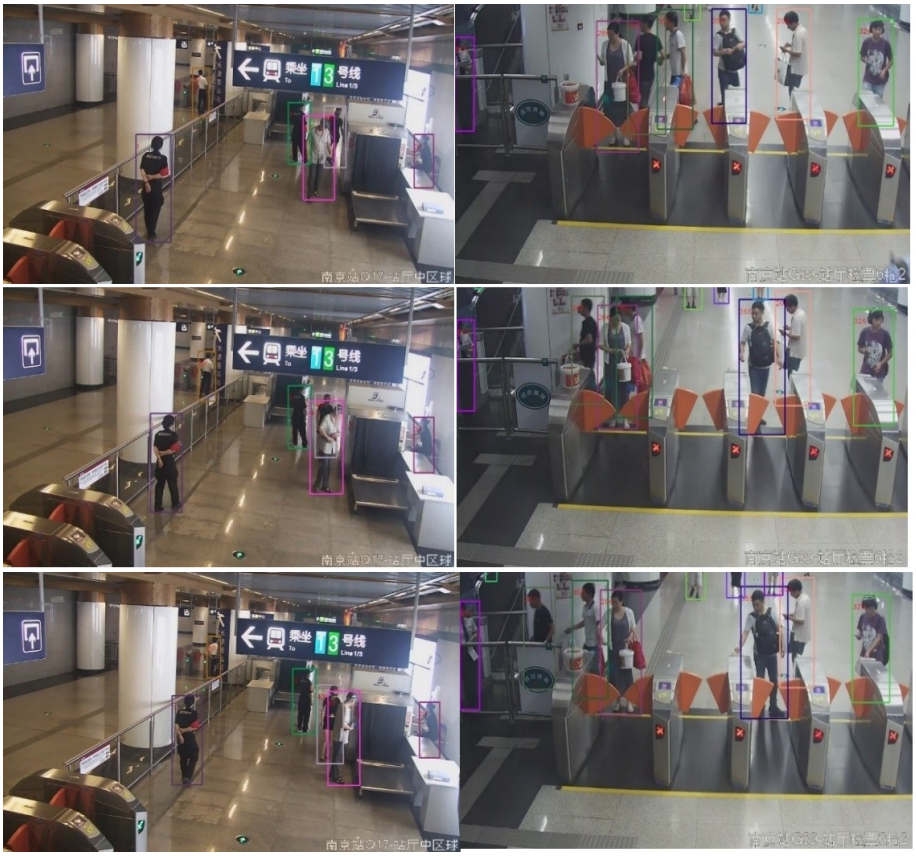

Figure 7: Visual Display of Multi-object Tracking Effects. The Left Column Is Scene 1 and the Right Column Is Scene 2. There Are 25 Frames between the Two Adjacent Images of Each Scene. The Bounding Box Marked with the Same ID and the Same Color Represents the Same Target.

REFERENCES

[1] Yu F, Li W, Li Q, Liu Y, Shi X and Yan J (2016). Multiple objects tracking with high performance detection and appearance feature arXiv:1610.06136 [cs.CV].

[2] Milan A, Leal-Taixe L, Reid I, Roth S and Schindler K (2016). Mot16: A benchmark for multi-object tracking arXiv:1603.0083 [cs.CV].

[3] Voigtlaender P, Krause M, Osep A, Luiten J, Sekar, B. B. G, Geiger A and Leibe B (2019). Mots: Multi-object tracking and segmentation IEEE Conf.on Computer Vision and pattern recognition (Los Angeles) pp 7942-7951.

[4] Xiao T, Li S, Wang B, Lin L and Wang X (2017). Joint detection and identification feature learning for person search IEEE Conf.on Computer Vision and pattern recognition (Hawaii) pp 3415-3423.

[5] Bodla Navaneeth, Singh Bharat, Chellappa Rama and Davis Larry S (2017). Soft-NMS – Improving Object Detection with One Line of Code arXiv:1704.04503 [cs.CV].

[6] He Yihui, Zhu Chenchen, Wang Jianren, Savvides Marios and Zhang Xiangyu (2019). Bounding Box Regression with Uncertainty for Accurate Object Detection IEEE Conf.on Computer Vision and pattern recognition (Los Angeles) pp 2888-2897.

[7] Liu Songtao, Huang Di and Wang Yunhong (2019). Adaptive NMS: Refining Pedestrian Detection in a Crowd IEEE Conf.on Computer Vision and pattern recognition (Los Angeles) pp 6459-6468.

[8] Wang Xinlong, Xiao Tete, Jiang Yuning, Shao Shuai, Sun Jian and Shen Chunhua (2018). Repulsion Loss: Detecting Pedestrians in a Crowd IEEE Conf.on Computer Vision and pattern recognition (Salt Lake City) pp 7774-7783.

[9] Hu Han, Gu Jiayuan, Zhang Zheng, Dai Jifeng and Wei Yichen (2017). Relation Networks for Object Detection arXiv:1711.11575 [cs.CV].

[10] Wang Zhongdao, Zheng Liang, Liu Yixuan, Li Yali and Wang Shengjin (2020). Towards Real-Time Multi-Object Tracking European Conf.on Computer Vision (Amsterdam) pp 107-122.

[11] He Kaiming, Gkioxari Georgia, Dollar Piotr and Girshick Ross (2018). Mask R-CNN arXiv:1703.06870v3 [cs.CV].

[12] Redmon Joseph, Farhadi Ali (2018). YOLOv3: An Incremental Improvement arXiv: 1804.02767v1 [cs.CV].

[13] A Kendall, Y Gal and R Cipolla (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics arXiv:1705.07115 [cs.CV].

[14] Ren Shaoqing, He Kaiming, Girshick Ross and Sun Jian (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks arXiv: 1506.01497 [cs.CV].

[15] Wojke Nicolai, Bewley Alex and Paulus Dietrich. Simple Online and Realtime Tracking with a Deep Association Metric arXiv:1703.07402 [cs.CV].