Regression-Aware Classification Feature for Pedestrian Detection and Tracking in Video Surveillance Systems

Xuan-Thuy Vo, Tien-Dat Tran, Duy-Linh Nguyen and Kang-Hyun Jo*

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan (44610), Korea
{xthuy,tdat}@islab.ulsan.ac.kr; ndlinh301@mail.ulsan.ac.kr; acejo@ulsan.ac.kr (* Corresponding Author)

Abstract. Pedestrian detection and tracking in video surveillance systems is a complex task in computer vision research, which has widely used in many applications such as abnormal action detection, human pose, crowded scenes, fall detection in elderly humans, social distancing detection in the Covid-19 pandemic. This task is categorized into two sub-tasks: detection, and re-identification task. Previous methods independently treat two sub-tasks, only focusing on the re-identification task without employing re-detection. Since the performance of pedestrian detection directly affects the results of tracking, leveraging the detection task is crucial for improving the re-identification task. The total inference time is computed in both the detection and re-identification process, quite far from real-time speed. This paper joins both sub-tasks in a single end-to-end network based on Convolutional Neural Networks (CNNs). Moreover, the detection includes the classification and regression task. As both tasks have a positive correlation, separately learning classification and regression hurts the overall performance. Hence, this work introduces the Regression-Aware Classification Feature (RACF) module to improve feature representation. The convolutional layer is the core component of CNNs, which extracts local features without modeling global features. Therefore, the Cross-Global Context (CGC) is proposed to form long-range dependencies for learning appearance embedding of re-identification features. The proposed model is conducted on two challenging benchmark datasets, MOT17, MOT17Det, which surpasses the state-of-the-art trackers.

Keywords: Pedestrian Detection, Tracking and Re-identification, Video Surveillance System, Convolution Neural Networks (CNNs).

1 Introduction

Nowadays, surveillance systems have been universally employed in many applications such as intelligent transportation systems, prevention of crime, military supervision systems, prisons, hospitals, industrial applications. The objective of the most surveillance system is to detect and track abnormal pedestrian activities in a video scene. The pedestrians are always walking or running on the street under a supervi-

sion camera. For example, the first pedestrian detection and tracking benchmark [3] is proposed, capturing real human activity on the street by CCTV. Pedestrian detection and tracking in video surveillance systems is a challenging task because of real dynamic environments such as illumination variation, crowded density scene, complicated distractor, shadows, occlusion, object deformation.

Recently, the accelerated development of deep learning, especially for Convolutional Neural Networks (CNNs), has brought a bright future in solving computer vision tasks such as pedestrian detection and tracking.

Pedestrian detection and tracking are one of the core applications of multiple object tracking for understanding visual objects in video. It includes two sub-tasks: detection and data association (re-identification). The pedestrian detection is to determine what objects are presented and where objects are located in each frame. Data association groups the same objects in different frames to output trajectories, assigning and tracking unique identification (ID) to each object across all frames. Previous methods, Sort [4], Deep-Sort [5], Poi [6] treat two sub-tasks independently. Specifically, re-ID is a secondary task in which the performance of it heavily depends on the main detection. Accordingly, leveraging the detection task is important for enhancing re-ID performance. The model complexity is calculated in both the detection and re-ID task, affecting the total inference time. Therefore, this work joins detection and re-ID task in the single end-to-end network based on the single object tracking paradigm, reducing the model complexity.

The generic detection consists of the classification and regression task. However, RetinaNet [19], FCOS [20], and Faster R-CNN [21] only used classification performance for ranking detection during inference without considering regression score. There is inconsistency in object detection. PISA [22] showed that both of tasks have positive correlation. Mean that the detection has high classification quality corresponding to high regression quality, otherwise. Accordingly, this paper introduces a novel module, named Regression-Aware Classification Feature (FACF), to guide regression distribution to classification feature with ignored computational cost. During backward, the gradient is propagated from the classification branch to the regression branch.

As the proposed network follows the Siamese method learning the similarity using correlation filter of the search feature and template feature to emphasize the interest of objects. In this paper, similarity learning is employed with global feature modeling to get informative features from the input. The convolution operation is the main component of CNNs, only extracting local features. As a result, the receptive field is limited inside local neighborhoods. To overcome this problem, many convolution layers can be deeply stacked up to 50 layers or 100 layers. This strategy is not efficient, leading to high computational cost and difficulty to perform back-propagation. Inspired by BNL [18] and GCNet [28], Cross-Global Context (CGC) with an additional computational cost is proposed to model long-range dependencies, i.e., global feature, and additionally learn similarity between features of the current frame and previous frame. Moreover, CGC improves appearance embedding for learning re-ID features. In another aspect, GCNet includes context modeling utilizing the global context pooling and transformation step using two convolutional layers with channel reduction to

learn channel dependencies. Although the channel reduction strategy avoids the high computational cost, it is ineffective because channel reduction can lose important information of input. Therefore, Cross-Global Context avoids channel reduction by using lightweight 1D convolution to excite the importance of each channel without affecting the overall performance.

The proposed method is evaluated on two challenging benchmarks, that are MOT17, and MOT17Det. Compared to previous methods, the performance achieves high multiple object tracking accuracy (MOTA), ID switch, and higher order tracking accuracy (HOTA) with additional computational cost.

2 Related Works

Pedestrian detection and tracking. Pedestrian detection and tracking are grouped into the online method and offline method according to input frame. For the online method, the input employs the current frame and past frame, while the offline method relies on the whole frame. Most of the online methods [4], [5], [18], and offline methods utilize available object detection and only consolidate data association performance. The data association includes the Kalman filter predicting future motions and the Hungarian algorithm for tracking. Several methods such as JDE [10], Tracktor [7], CenterTrack [11], FairMOT [13], and CTracker [14] introduced single end-to-end networks leveraging re-detection to improve appearance features for the re-ID step. Accordingly, this paper inherits “re-detection” method to combine detection and re-ID into one network, inspired by CTracker.

Object Detection. The generic detections such as RetinaNet [19], FCOS [20], and Faster R-CNN [21] are employed in specific categories for pedestrian detection [10], [13], [14], car detection, etc., which achieves the great performance. The object detection is categorized into one-stage detectors [19], [20] and two-stage detectors [21].

Correlation between classification and regression. GA-RPN [23] presented the feature adaptation module between classification and regression branch using deformable convolution to add offset prediction into the rectangular grid sampling locations in regular convolution, thus enhancing feature representation. PISA [22] proposed the positive correlation between classification and regression, improving the overall performance. The classification score is inserted to regression loss to re-weight prime samples, i.e., give more contribution to easy samples. However, the classification score and regressed offsets are computed independently during testing. Mean that there is inconsistent computation during training and testing. Alternatively, this work introduces a simple but effective module performing the correlation between classification and regression during training and testing without relating to the loss function.

Global feature. GCNet [28] introduced global context module modeling long-range dependencies. This module includes the global context pooling and transformation step. The global context pooling squeezes the input tensor to vector to calculate the relationship between a query position and all positions and aggregate features of all positions by taking an averaging. The transformation step using two convolutional layers excites channel dependencies, i.e., whether certain channels are important or

not. BNL [18] proposed the bidirectional non-local network by the dissecting global context module to gather and distribute features between query position and key position, which applied to object detection task.

3 The Proposed Method

This section analyzes the proposed end-to-end architecture, Cross-Global Context (CGC) module, and Regression-Aware Classification Feature (RACF) module.

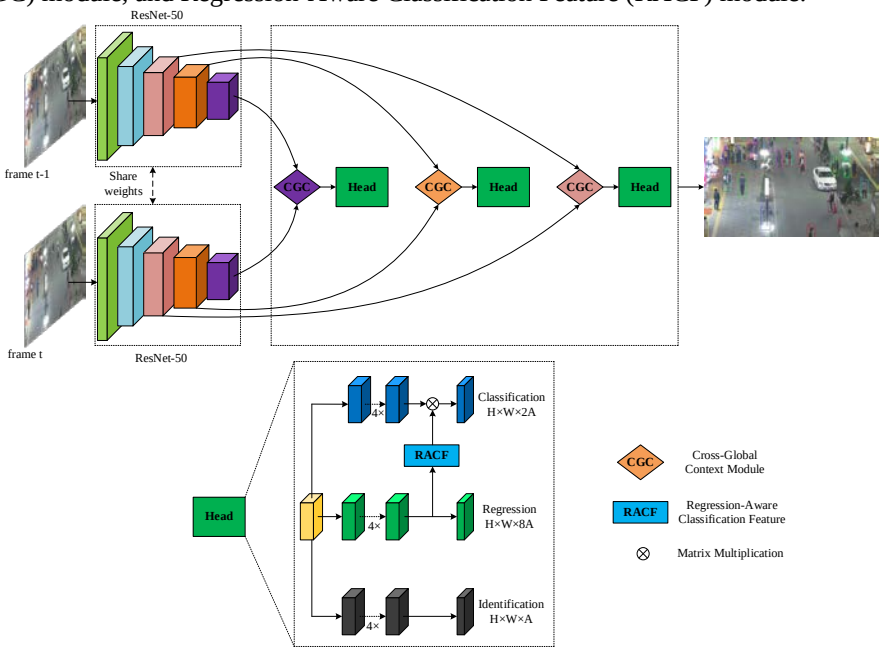

Fig. 1. The overall architecture of the proposed method. The two adjacent frames, frame t – 1, and frame t, are the input of the single end-to-end network. The backbone ResNet-50 extracts feature from two input frames. The shared weights are used to reduce model complexity because two adjacent frames are similar. Then, the feature pyramid is constructed from stage 3, stage 4, and stage 5 of the backbone structure. CGC is the cross-global context module to model long-range dependencies and perform similarity learning between the feature at frame t – 1 and feature at frame t. Each head predicts classification score, regressed offset of paired bounding boxes of the same target, and Re-id score according to IoU score between paired bounding boxes with the same ID. 4× denotes four convolutional layers, each convolutional layer includes 3×3 convolution following by group normalization and ReLU activation function. A indicates the number of anchor boxes per location.

The overall architecture is shown in Fig. 1. The input continuous video of this task is captured by CCTV, separated into discrete frames at a certain frame rate. Following the online method, the input only takes the current frame and last frame. Inspired by CTracker, two adjacent frames are used as input. The shared backbone extracting feature is ResNet-50 pre-trained on ImageNet [27]. Similar to EFPN [17], and FPN

[26], a feature pyramid is constructed to detect the objects with different scales, i.e., solve scale imbalance problem. For example, the large objects, medium objects, and small objects are assigned to a small feature, medium feature, and large feature, respectively. The CGC module will be discussed in subsection 3.1. Note that there feature maps corresponding to three heads are selected as a pyramid. Each head includes the classification, regression, and identification branch. The classification branch outputs objectness scores of each anchor box (pre-defined box) because the network only contains the pedestrian class. The regression branch predicts eight offset values corresponding to paired bounding boxes of the same target (the first four values for a target in the previous frame and last four values for a target in the current frame). The RACF module will be described in subsection 3.2. The identification branch predicts Re-score computed IoU (Intersection of Union) between paired bounding boxes with the same ID. It means that the data association tracks IoU matching between paired bounding boxes of two adjacent frames without applying the Hungarian algorithm for tracking. Therefore, the proposed network is a one-shot tracker, which reduces inference time.

3.1 Cross-Global Context

The cross-global context models long-range dependencies avoiding channel reduction and performs correlation learning between two adjacent frames, shown in Fig. 2.

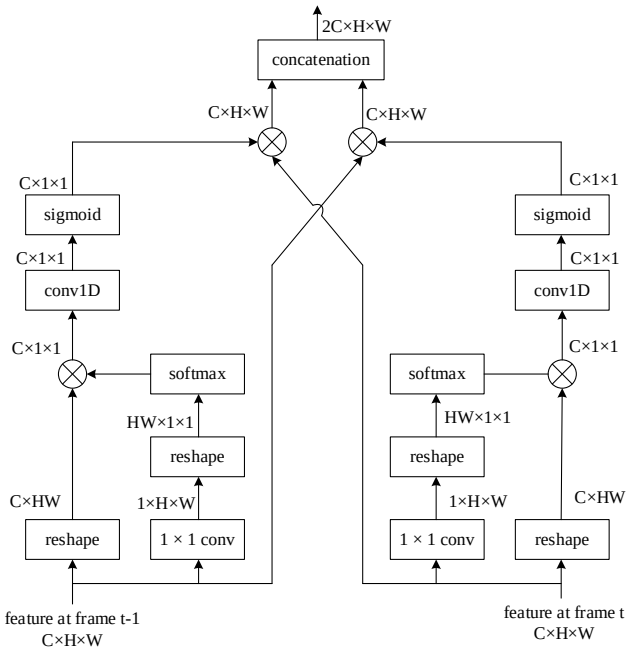

Fig. 2. The Cross-Global Context (CGC) takes two adjacent features as input. For each input, the CGC consists of the global context pooling and transformation step. The global context pooling learns the correlation between query position and all positions, which models long-range dependencies. The transformation step using light-weight convolution computes channel dependencies.

The input takes two adjacent features at frame t – 1, and frame t. For feature $\mathbf{F}^{t-1}$ with dimension C×H×W, CGC includes the global context pooling and transformation step. The global context pooling according to GCNet [28] gathers features from a query position and all position by computing average, as follows:

$$\omega_{ij}^{t-1} = \sum_{j=1}^{H*W} \frac{\exp\left(\mathbf{W}_k \mathbf{F}_j^{t-1}\right)}{\sum_m \exp\left(\mathbf{W}_k \mathbf{F}_m^{t-1}\right)} \mathbf{F}_j^{t-1}, \quad (1)$$

where H, W, C is the height, width, and number of channels of the input feature map. ω_{ij}^{t-1} is a correlation function between the query position $\mathbf{F}_i^{t-1}$ and key position $\mathbf{F}_j^{t-1}$, in which the input tensor is squeezed to vector C×1×1. $\mathbf{W}_k$ is a 1×1 convolution operation to gather feature of all positions. The exp is the exponential function. In CNNs, this function is softmax operation to output the attention map of each position, i.e., which positions contain the informative feature. Then, the matrix operation is performed between attention map and the reshaped input $\mathbf{F}_j^{t-1}$ to create a channel vector.

The transformation step learns channel dependencies by using excitation operation. In GCNet [28], the two 1×1 convolution layers with the channel reduction excite channel relationship, leading to losing information. To avoid channel reduction, CGC only utilizes lightweight 1D convolution with a kernel size of 5, learning cross-feature interaction. This computation is defined as:

$$e^{t-1} = \delta\left(\mathbf{W}_z \omega_{ij}^{t-1}\right), \quad (2)$$

where e^{t-1} is a re-scale function. δ is the sigmoid function to output the probability of each channel. $\mathbf{W}_z$ is a 1D convolution with a kernel size of 5 and padding of 2.

Similarly, the global context pooling function ω_{ij}^{t} and transformation function e^t of current feature at frame t are computed as Equation 1, and Equation 2, respectively. The rescaled features of two transformed features are crossed to learn similarity between the current and previous features, defined as:

$$\mathbf{R}^{t-1} = e^{t-1} \odot \mathbf{F}^t, \quad (3)$$

$$\mathbf{R}^t = e^t \odot \mathbf{F}^{t-1}, \quad (4)$$

where $\mathbf{R}^{t-1}$ and $\mathbf{R}^t$ are crossed features at frame t and t – 1 by using broadcast element-wise multiplication. e^{t-1} and e^t are re-scale functions computed as Equation 2. $\mathbf{F}^t$ and $\mathbf{F}^{t-1}$ are input features at frame t and frame t-1.

Finally, two crossed features are concatenated as input of head part:

$$\mathbf{CF} = [\mathbf{R}^t, \mathbf{R}^{t-1}], \quad (5)$$

where **CF** is concatenated feature with dimension 2C×H×W.

3.2 Regression-Aware Classification Feature

The head consists of classification, regression, and identification branch, shown in Fig. 1. The detection quality depends on classification quality and regression quality. Independently learning the classification and regression branch is a straightforward way to improve detection quality. This paper introduces Regression-Aware Classifi-

cation Feature (RACF) to positively correlate two branches, shown in Fig. 3. RACF module is computed consistently during training and testing. The easy samples and hard samples are measured by regression quality (IoU score). Specifically, the easy samples normally have a high IoU score corresponding to high regression quality. It means that the regression branch gives more contribution (more signals) to easy samples through the FACF module. Alternative speaking, the network focuses on easy samples. Otherwise, the hard samples have low regression quality. The regression branch gives less contribution to hard samples. During Non-maximum Suppression (NMS), the low-quality samples are filtered out.

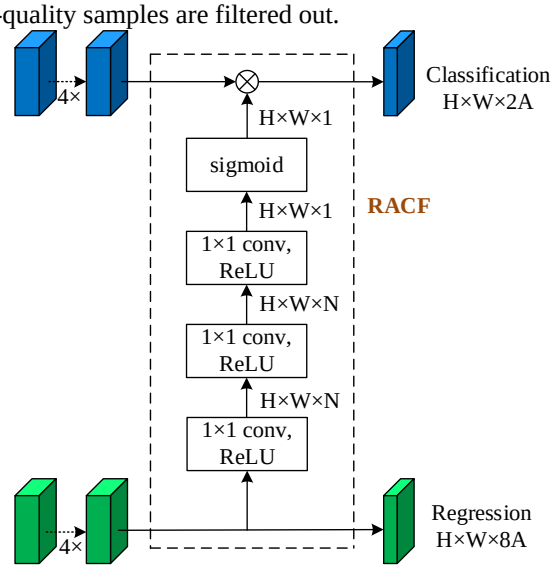

Fig. 3. The Regression-Aware Classification Feature (RACF) takes regressed offsets of paired bounding boxes as input. H, W, N is the height, width, and number of hidden channels, respectively. A denotes the number of anchor boxes placed per location. 4× indicates four convolutional layers, each convolutional layer uses a 3×3 convolution operation following by group normalization and ReLU function.

To measure regression quality, the *topk* function is applied to select two maximum values from the regression distribution of each sample. Regularly, the regression distribution is Gaussian distribution. The three convolutional layers learn the correlation between regression and classification quality to improve classification feature which is aware of regression feature. Each convolutional layer includes a 1×1 convolution operation following by a ReLU activation function. Since regression quality and classification score are different ranges, the sigmoid function normalizes the regression quality between 0 and 1. Finally, the matrix multiplication is performed between learned regression quality and classification to enhance the classification feature.

During optimization, the gradient from the classification branch is propagated to the regression branch. The samples with higher classification loss will bring a larger gradient for regression quality, which means higher suppression on the regression quality.

4 Experiment Setup

The proposed method conducts the experiments on two challenging benchmarks, that are MOT17 [2], and MOT17Det [2]. Each dataset includes 7 training videos and 7 testing videos. The difference between MOT17 and MOT17Det is annotation definition in detection, ID number, and different evaluation procedures. Note that MOT17Det does not provides detection ground truth. Pedestrian tracking is a complex task depending on classification, localization, and re-ID task. To measure whole aspects of performance, all results are evaluated by three primary metrics for tracking, detection, and ID assignment such as multiple object tracking accuracy (MOTA), and ID switch (IDF1) proposed by CLEAR MOT [15]; higher order tracking accuracy (HOTA) proposed by [16]; and Average Precision (AP) for the detection task.

For implementation details, all experiments are implemented by the deep learning Pytorch framework. The parameters of the ResNet-50 are pre-trained on ImageNet [27] as initialization. The added convolutional layers in FPN, three branches in the head part, CGC module, and RACF module are initialized by selecting values from the normal distribution. The model is trained for 100 epochs with a batch size of 8. The Tesla V100 SXM2 device with Cuda 10.2, CuDNN 7.6.5 is used for implementation. The learning rate is set to $3\times e^{-5}$ for all experiments. Adam optimizer is employed to optimize objective function defined as:

$$L = \alpha L_{reg}(b_i^{t-1}, g_i^{t-1}; b_i^t, g_i^t) + \beta L_{cls}(s, \hat{s}) + \gamma L_{id}(d, \hat{d}), \quad (6)$$

where L_{reg} is regression loss for paired bounding boxes of the same target at frame t-1 and frame t, which uses smooth-L1 loss. $b_i \in \{x, y, w, h\}$ denotes bounding box prediction with four offsets such as center coordinates (x, y), width w, and height h. $g_i \in \{x, y, w, h\}$ indicates ground truth bounding box. L_{cls} is the classification loss using Focal loss [19] in which s, $\hat{s}$ is classification score and target label. L_{id} is defined by CTracker [14] for identification loss, utilizing Focal loss [19] to predict ID according to IoU matching. α, β, γ is a balancing term, set to 1.0 during training. For anchor setting, we only place one square anchor box per location to reduce model complexity.

5 Results

This section analyzes how to select the hyperparameters in the CGC, RACF module and the importance of each component in subsection 5.1. The results of the ablation study are measured on the sub-set of the training set since the MOT benchmark did not provide ground truth annotation for the testing set. The main results tested on the MOT17 testing set are shown in subsection 5.2, which are submitted to the evaluation protocol system[1]. Because the testing set and training set are very different, the performance evaluated on them is also different.

[1] https://motchallenge.net/

5.1 Ablation Study

The number of hidden channels in RAFA. Some implementations are conducted to select the hyperparameter N in the RAFA module. We select N ∈ {16, 32, 64, 128, 256} to train the single-end-end model. The results are shown in Table 1, measured by the evaluation tool[2].

Table 1. The effects of the number of hidden channels on the performance

N	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	FP↓	FN↓	IDS↓	#par
16	74.3	65.8	85.1	259	60	1954	25927	1015	+0.4k
32	74.6	66.6	85.2	270	53	2060	25413	1032	+1.3k
64	75.7	66.3	**85.7**	270	55	**1493**	24807	998	+4.7k
128	**76.1**	**67.3**	85.6	**278**	**51**	1668	**24190**	**977**	+17.5k
256	74.8	66.2	85.2	266	57	2115	25158	1033	+67.8k

where MOTP is Multiple Object Tracking Precision. MT, ML is Mostly Tracked Trajectories, Mostly Lost Trajectories. FP, FN is the number of False Positives and False Positive. IDS is the number of Identity Switches. #par is the additional parameter of the RAFA module to the total parameters.

The performance of the whole network is insensitive to the number of hidden channels N. The small N will output coarse features for the classification branch. Alternatively speaking, the 1×1 convolution with output channels, 16 or 32, does not satisfy to learn variables of regression distribution. The model with a large N means that RAFA can learn rich information of regression quality. Specifically, the proposed method achieves the best results at N=128 with 76.1% of MOTA. Hence, we select N=128 for all experiments. In another aspect, the RAFA only takes 17.5k (thousands) parameters while the total parameters of the whole network are up to million parameters. Therefore, it demonstrates the RAFA module is simple but effective.

Avoiding channel reduction in CGC. This experiment analyzes the effect of channel reduction on the CGC performance, which is shown in Table 2.

Table 2. The effect of channel reduction on the results

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	FP↓	IDS↓	#par
CGC w/ CR	74.9	65.3	85.1	273	1836	1038	$2\times C^2/r$
CGC w.o/ CR	75.6	65.7	85.1	283	2280	989	k=5

As expected, the CGC module with channel reduction (use two 1×1 convs with a reduction ratio of r) decreases the MOTA score by 0.7% when compared CGC module without channel reduction (use 1D convolution with the kernel size of 5). Moreover, the lightweight operation only takes 5 parameters while using 2 consistent convolutions is still high computational cost. Usually, the number of channels C is 256 and r

[2] https://github.com/dendorferpatrick/MOTChallengeEvalKit

= 8, the number of parameters is 16.4k parameters in which larger than our method by 3280 times.

The effect of each component. We investigate the importance of individual components on the sub-training set. The results are shown in Table 3.

Table 3. The importance of each component

Baseline	CGC	RAFA	MOTA↑	IDF1↑	MOTP↑	MT↑	FP↓	IDS↓
✓			73.1	63.3	85.3	245	2229	2385
✓	✓		75.5	66.0	84.9	277	1747	983
✓		✓	74.9	65.3	85.1	273	1836	1038
✓	✓	✓	76.1	67.3	85.6	278	1668	977

The baseline is the simple version of the proposed method, which outputs 73.1% MOTA. When CGC is added to the baseline, the results gain the MOTA score of 2.4%. Similarly, the RAFA boosts the baseline performance by 1.8%. Remarkably, the full version includes the CGC and RAFA module, which increases the baseline results by a large margin, 3.0% MOTA score. It is easy to understand that the proposed model can learn finer features from global features and regression quality, effectively.

Error Tracking Decomposition. To measure the error of the proposed method, we decompose the tracking performance into three components: detection errors, association errors, and localization errors, shown in Fig. 4.

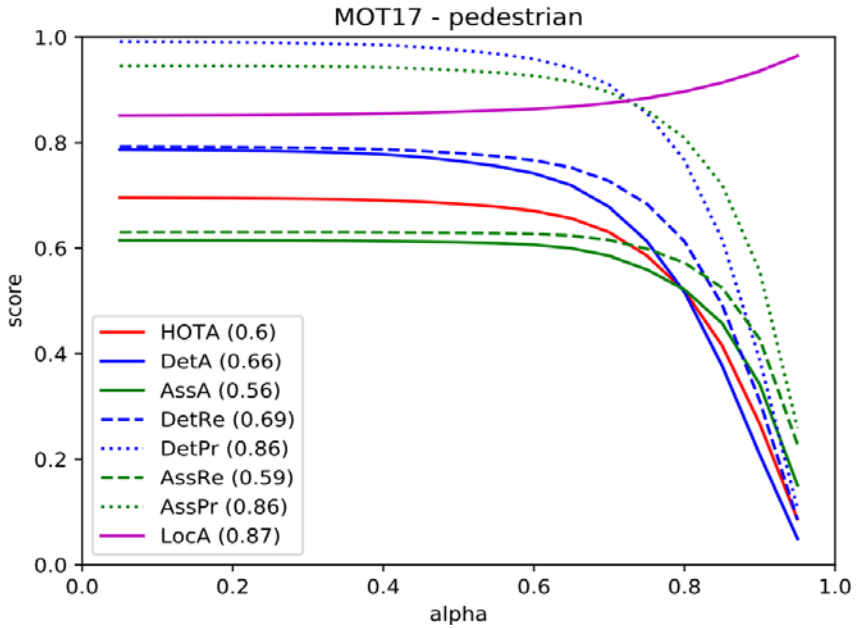

Fig. 4. HOTA and decomposed components

where alpha indicates the performance at different localization score thresholds from 0.05 to 0.95 with step size 0.05. DetA is the detection accuracy score achieving the average score of 0.66, which decomposing to DetRe (Detection Recall) and DetPr (Detection Precision). For localization error, we compute LocA (localization accuracy score) at various alpha and average LocA_{α}. Accordingly, LocA achieves a score of 0.87, high localization quality because of that the RAFA learns localization quality to guide classification features in which the detection works well at the high alpha threshold. For association errors, the AssA (association accuracy) is employed for assigning predicted IDs to ground truth trajectories, which decomposes to AssRe (association recall) and AssPr (association precision).

5.2 Comparison with State-of-the-Art Methods

This subsection shows the main results of the proposed method tested on MOT17 and MOT17Det. The performance of the whole network is compared with other methods, listed in Table 4. The bold font indicates the best results among trackers.

Table 4. Comparison with state-of-the-art online methods on MOT17 test set

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	FP↓	IDS↓
DMAN [9]	48.2	55.7	75.9	19.3	26281	2194
MOTDT [8]	50.9	52.7	76.6	17.5	24069	2474
Tracktor [7]	53.5	52.3	78.0	19.5	**12201**	**2072**
Tracktor+CTDet	54.4	56.1	78.1	25.7	44109	2574
DeepSORT [5]	60.3	61.2	**79.1**	31.5	36111	2442
CTracker [13]	66.6	**57.4**	78.2	32.2	22284	5529
Ours	**67.3**	54.7	78.6	**32.9**	18771	5910

Table 5. The detection and tracking results on MOT17Det test set with 7 sequences

Sequence	AP	MODA	MODP	F1	Rcll	Prcn	FAR
MOT17Det-01	65	62.6	80.2	79.5	72.7	87.8	0.9
MOT17Det-03	77	91.5	80.5	95.7	97.9	93.9	2.9
MOT17Det-06	69	70.1	78.1	83.6	76.1	92.6	0.5
MOT17Det-07	78	79.9	80.3	89.3	83.9	95.5	0.7
MOT17Det-08	87	85.2	84.1	92.4	90.2	94.7	0.5
MOT17Det-12	51	65.8	79.4	80.1	68.6	96.3	0.2
MOT17Det-14	51	57.1	78.2	74.1	61.3	93.6	0.7
Overall	68	82.4	80.3	90.9	88.2	93.8	1.1

The proposed method achieves 67.3% MOTA, outperforming all trackers by a large margin. Specifically, our tracker surpasses the DMAN [9] at 48.2% MOTA, MOTDT [8] at 50.9% MOTA, Tracktor [7] at 53.5% MODA, Tracktor with CTDet [7] at 54.4% MOTA, DeepSORT [5] at 60.3% MOTA, and strong tracker CTracker [13] at 66.6% MOTA.

For detection in MOT17Det, we investigate the performance in 7 testing sequences, describe in Table 5. MODA denotes multiple object detection accuracy combining false positives and misses ground truth. MODP indicates multiple object tracking

precision, shows misalignment between bounding box prediction and ground truth bounding boxes. F1 is the harmonic mean of precision and recall. Rcll, and Prcn mean recall and precision. FAR denotes the average number of false alarms per frame. The visualization of the tracking performance in each sequence is shown in Fig. 5.

Fig. 5. The qualitative results on some sequences of MOT17 benchmark.

6 Conclusion

This paper introduced the online single end-to-end network joining detection and data association in the one-shot tracker for pedestrian detection and tracking in video surveillance systems, which reducing computation cost. Moreover, the RACF module with simple but effective learned regression distribution to guide classification feature, leveraging the positive correlation between classification and regression task. The CGC module with lightweight operation is proposed by investigating channel reduction in transform step to model long-range dependencies and similarity learning between two adjacent frames (current frame and previous frame). The performance is evaluated on the challenging benchmark MOT17, and MOT17Det, which outperformed all online trackers by a large margin. In the future, we will test the proposed method on various datasets such as MOT20, PETS, and KITTI benchmark.

References

1. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., & Schindler, K. (2015). Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942.
2. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.
3. Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2009). Pedestrian detection: A benchmark. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 304–311).

4. Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP) (pp. 3464–3468).
5. Wojke, N., Bewley, A., & Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645–3649).
6. Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., & Yan, J. (2016). Poi: Multiple object tracking with high performance detection and appearance feature. In European Conference on Computer Vision (pp. 36–42).
7. Bergmann, P., Meinhardt, T., & Leal-Taixe, L. (2019). Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 941–951).
8. Chen, L., Ai, H., Zhuang, Z., & Shang, C. (2018). Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In 2018 IEEE International Conference on Multimedia and Expo (ICME) (pp. 1–6).
9. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., & Yang, M.H. (2018). Online multi-object tracking with dual matching attention networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 366–382).
10. Wang, Z., Zheng, L., Liu, Y., & Wang, S. (2019). Towards real-time multi-object tracking. arXiv preprint arXiv:1909.12605, 2(3), 4.
11. Zhou, X., Koltun, V., & Krähenbühl, P. (2020). Tracking objects as points. In European Conference on Computer Vision (pp. 474–490).
12. Zhang, Y., Wang, C., Wang, X., Zeng, W., & Liu, W. (2020). FairMOT: On the fairness of detection and re-identification in multiple object tracking. arXiv e-prints, arXiv–2004.
13. Peng, J., Wang, C., Wan, F., Wu, Y., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., & Fu, Y. (2020). Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In European Conference on Computer Vision (pp. 145–161).
14. Bernardin, K., & Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing, 2008, 1–10.
15. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. (2020). HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking. International Journal of Computer Vision, 1–31.
16. Vo, X.T., & Jo, K.H. (2020). Enhanced Feature Pyramid Networks by Feature Aggregation Module and Refinement Module. In 2020 13th International Conference on Human System Interaction (HSI) (pp. 63–67).
17. Vo, X.T., Wen, L., Tran, T.D., & Jo, K.H. (2020). Bidirectional Non-local Networks for Object Detection. In International Conference on Computational Collective Intelligence (pp. 491–501).
18. Ross, T.Y., & Dollár, G. (2017). Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2980–2988).
19. Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9627–9636).
20. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497.

21. Cao, Y., Chen, K., Loy, C., & Lin, D. (2020). Prime sample attention in object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11583–11591).
22. Wang, J., Chen, K., Yang, S., Loy, C., & Lin, D. (2019). Region proposal by guided anchoring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2965–2974).
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. (2014). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740–755).
24. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117–2125).
26. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).
27. Cao, Y., Xu, J., Lin, S., Wei, F., & Hu, H. (2019). Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (pp. 0–0)