Exploring the Effectiveness of Appearance Descriptor in DeepSORT

Zilin Xiao
College of Computer Science
Sichuan University
China
xiaozilin1@gmail.com

Yanan Sun✉
College of Computer Science
Sichuan University
China
ysun@scu.edu.cn

***Abstract*—Tracking-by-detection approaches have demonstrated their strength in addressing Multiple Object Tracking (MOT) problems. DeepSORT, one of the classical tracking-by-detection MOT methods, relies on a deep appearance descriptor to extract global appearance features of identities. Although the appearance descriptor acts as a key component of such tracking-by-detection methods, which is responsible for modeling appearance information, the relationship between it and tracking performance remains unclear, especially whether further improvements to it will be reflected in the tracking performance. To explore that, extensive experiments are conducted on the appearance descriptor by applying various traditional optimization methods. Furthermore, we propose an Evolutionary Neural Architecture Search (ENAS) strategy for the appearance descriptor named Genetic-SORT to assist exploration. The experimental results demonstrate that tracking performance fails to follow the improvements applied on the appearance descriptor and even shows a negative correlation, which is contrary to our intuition.**

***Index Terms*—Multiple Object Tracking, Evolutionary Neural Architecture Search, Appearance Descriptor.**

I. Introduction

Multiple Object Tracking (MOT) has been a popular task in the research area of computer vision [1]–[3], which aims at tracking the displacement of each object and determining their identities in a continuous series of video frames [4]. MOT has a wide range of real-world applications. For example, MOT-driven intelligent surveillance [5] can provide trajectories of suspects for authorities to track them down. Similarly, robot navigation [6] also benefits a lot from trajectories of multiple surrounding objects. Thanks to significant advances in Deep Neural Networks (DNNs) in recent years [7], the performance of object detection has reached a new level, which leads the tracking-by-detection mechanism to be a popular pipeline in the research scope of MOT.

Commonly, a typical tracking-by-detection MOT system is composed of two parts: a detector and a data association strategy [8]. The detector attempts to precisely localize every object with a bounding box in each frame along with their classifications. The data association refers to the stage of assigning the detection bounding boxes of adjacent frames to specified identities. In those MOT works that favor studying data association, the detector is often replaced with a publicly available detection dataset that corresponds to the training image sequences, making tracking performance dependent only on the data association. For further understanding about data association, considering a case of three detection boxes that wait to be assigned to a specific identity in the $i+1$ frame as shown in Fig. 1: the dotted lines indicate no matching pairs have been determined between these adjacent frames, while the solid lines suggest identities of all detection boxes have been determined by various association approaches, *e.g.* modeling motion and appearance information for each detection box.

Among the enormous amount of tracking-by-detection methods, DeepSORT [9] stands out for its relatively elegant pipeline design and fast inference speed, whose tracking procedure can be illustrated in Fig. 2: frames of the input video sequence along with their corresponding detection information go through the data association stage, where the assignment cost will be calculated between each pair of detection boxes among consecutive frames to assist Hungarian assignment. The assignment result finally contributes to deriving relatively accurate trajectories of tracked objects.

Specifically, at the data association stage of DeepSORT, it introduced a hand-designed Convolutional Neural Network (CNN) [10] called appearance descriptor [11] to extract deep appearance features of possible identities. Afterward, the cosine similarity between each pair of possible matching identities will be calculated in the embedding space as a solid indicator that assists data association. To learn such an appearance descriptor that matches identity pairs better, and thus improving the consistency of tracking trajectories, researchers developed a learning-based scheme for the appearance descriptor based on the training and evaluating policy of re-identification (ReID) [12] since they share a common subtask, that is to construct closer features in the embedding space when dealing with detection boxes from the same identity.

To date, for further performance improvement on classical tracking-by-detection pipelines like DeepSORT, many researchers have devoted themselves to figuring how various components of such pipelines relate to tracking performance. While so far, few researchers have focused on the effectiveness of the appearance descriptor and how it relates to tracking performance. Given this, we would like to adopt some latest optimization methods on the appearance descriptor which covers different aspects of deep learning, *i.e.*, the data

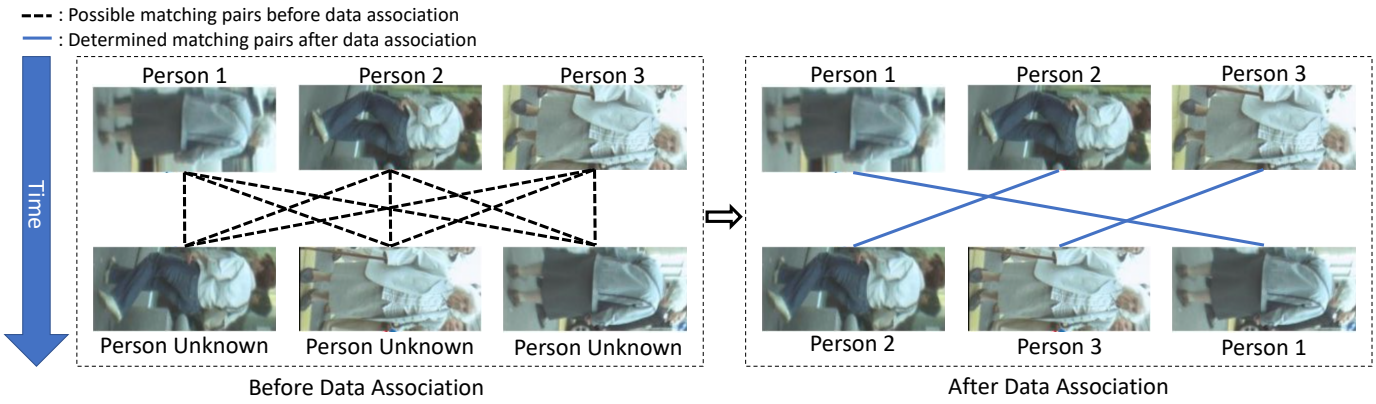

Fig. 1. Illustration of the data association between detection boxes

preprocessing, training, loss function, and model structure, to explore the deeper relation between appearance descriptor and tracking performance. It is worth mentioning that we use the Evolutionary Neural Architecture Search (ENAS) [13] approach when exploring the effect of model structure on the appearance descriptor, and an evolutionary algorithm for searching appearance descriptor named Genetic-SORT is proposed to assist architecture optimization.

Our main contributions are as follows:

- A convenient and efficient experiment pipeline is built in this paper. It incorporates several key components including the ENAS module, ReID training & evaluation, DeepSORT tracking, and MOT metric evaluation. This pipeline greatly helps to obtain results of the subsequent experiments. Code will be available once this paper goes public (https://github.com/MrZilinXiao/Exploring-DeepSORT).
- The extensive experiments demonstrate that the relation of the global appearance descriptor and tracking performance is counter-intuitive: numerous optimization methods on the global feature extractor are not reflected in the tracking metrics, but even show degradation effect.
- Our experimental observations show that the tracking performance of such two-stage tracking pipelines like DeepSORT may suffer from the bottleneck of global features extracted by the appearance model. This may give researchers some insight: attempts to improve the performance of classical two-stage tracking pipelines by improving the global appearance descriptor are likely to be futile at all.

The remaining of this paper are organized as follows. The related works, including the MOT and ENAS, are provided in Section II. In Section III, details of the designed ENAS algorithm and traditional optimizations are documented. Immediately after, the experimental setting and result are given in Sections IV and V. Finally, the conclusions are shown in Section VI.

II. Related Works

To help readers get a better understanding of our work, some related work concerning Multiple Object Tracking (MOT) and Evolutionary Neural Architecture Search (ENAS) are reviewed in this section.

A. Multiple Object Tracking (MOT)

Classical two-stage MOT approaches are usually implemented via detection-and-association pipelines. One of them named SORT [14], proposed by Bewley *et al.* , totally discard appearance features of identities when conducting data association, claiming to keep in line with Occam's Razor. Wojke *et al.* [9] retain almost identical data association strategies of SORT except for replacing assignment cost to cosine distance between deep appearance features. Apart from the Hungarian algorithm, others are also available for the assignment problem, for example, Yu *et al.* [15] adopt the Kuhn-Munkres algorithm for the data association. Some creative ideas for the data association were proposed, such as Kim *et al.* 's success [16] on deploying bilinear LSTM in the appearance model to tackle the long-term dependency of each identity. The recurrent model is also capable of tackling other problems in the detection-association pipeline, for example, Milan *et al.* [17] creatively use LSTMs to solve data association instead of statistical methods. Some researchers want to find an alternative way to improve tracking performance, such as by improving detection: Long *et al.* [18] make some successful attempts to solve unreliable detection and intra-class occlusion through deep neural networks.

In recent years, some works focus on how to integrate detection and association into a unified task. Wang *et al.* [19] integrate an appearance embedding model into the detector to allow both detection and feature can be inferred from a single model without a significant performance drop. Zhang *et al.* [20] introduce an encoder-decoder network to generate a hires feature map then sends it to two homogeneous branches for yielding detection and appearance features respectively.

B. Evolutionary Neural Architecture Search (ENAS)

Neural Architecture Search (NAS) can be treated as a subfield of automatic machine learning [21]. It attempts to find a proper neural network architecture with better generalization capability on a specific task compared with human-designed

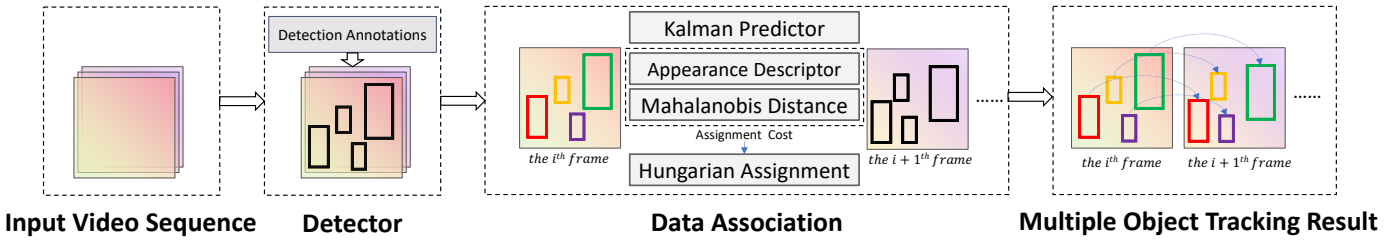

Fig. 2. Tracking procedure of DeepSORT, where colorful boxes indicate identified objects, black boxes indicate unidentified objects and blue arrows indicate object movement between adjacent frames.

ones. Being one of the first to make efforts on NAS, Zoph *et al*. [22], [23] use the Reinforcement Learning (RL) strategy to train recurrent neural networks for generating hyper-parameters of each layer, contributing a network structure that rivaled the best manually-designed network at that time. Liu *et al*. [24] and Chen *et al*. [25] respectively develop a continuous relaxation of the search space, turning the search process from discrete to differentiable, hence bringing the possibilities to apply gradient descent to optimize NAS problems. The above works have laid the foundations for future research on various subfields, for example, Quan *et al*. [26] consider the characteristics of the ReID task and employ DARTS [24] technique into searching part-aware ReID models.

Among a large number of NAS approaches, ENAS stands out for its elegance and relatively modest computational consumption. Unlike other NAS methods, ENAS uses a series of evolutionary computation approaches to optimize model structures instead of RL and gradient-based ones. Genetic Algorithm (GA) [27] and Particle Swarm Optimization (PSO) [28], for example, are used for searching neural architectures of specific tasks like image classification. Genetic CNN [29] designs a fixed-length binary-encoding strategy to represent connections between nodes in a candidate structure. PSO-CNN [30] proposes a searching strategy based on PSO to achieve faster convergence comparing other evolutionary methods. CNN-GA [31] develops a promising GA-based searching algorithm that requires minimal human expert knowledge while still yields outstanding performance.

III. Methodology

In this section, in order to explore the relationship between the appearance model and tracking performance, some optimization methods that are intuitively regarded to improve ReID metrics of the appearance model will be discussed. Specifically, an overview of the appearance model structure for experiments is given first. Then two types of optimization methods: ENAS methods and traditional tricks are illustrated.

A. Method to Exploring Architecture

CNN-GA [31], as one of the state-of-the-art ENAS algorithms, has evolved a great network targeting image classification task. We will make some improvements based on this algorithm to evolving the appearance backbone architecture.

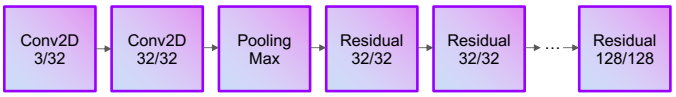

Fig. 3. Corresponding encoding of the DeepSORT original appearance descriptor, where number pairs like 3/32 indicate in/out channel of certain block

By analogy with the genetic algorithm, CNN-GA treats candidate models as individuals, building blocks as genes, and it also proposed corresponding encoding & decoding strategies and genetic operators: crossover and mutation. In this paper, we refer to the original DeepSORT appearance model to design the search space. The original model is a linear combination of Conv2D, Max-Pooling and Residual blocks, which happens to meet the search space of CNN-GA. Different from traditional searching strategies, ENAS methods usually do not restrict the search space by explicit limitations, but rather by a limited number of evolutionary operations. We choose almost the same candidate building blocks as the original model, except for that Mean Pooling block is also allowed. Each block with its hyper-parameter is encoded in the corresponding chromosome. We simply illustrate the encoding strategy in Fig. 3 by turning the original model into an individual chromosome.

In addition to inheriting basic evolving strategy from CNN-GA, we have also adapted some new features to match the characteristics of the ReID task:

1) Extreme evolving situations are thrown away to speed up the evolutionary process and conserve computation resources. This means no architectures that start with a pooling layer and no consecutive pooling layers are allowed.
2) Since pooling blocks and residual blocks with different in/out channel will reduce the spatial size of feature map, it's crucial to set a limit to the number of these blocks to prevent such situations where evolved models produce feature maps with too small spatial size.
3) Crossover and mutation operators are more restricted since the above restrictions still hold for offspring. For example, when performing crossover between two individuals, offsprings that conflict with the above restrictions are not allowed, meaning only those crossover

combinations that meet all restrictions may survive and get into the next generation.

To distinguish it from CNN-GA, we name the proposed evolutionary strategy for the appearance model Genetic-SORT.

B. Traditional Optimization Methods

In this subsection, six promising optimization approaches on the appearance model are introduced and they correspond to many key points of solving a deep learning problem, including data preprocessing, training and loss function. The vast majority of them proved to be effective in later experiments.

1) Random Erasing: Apart from resizing, random horizontal flipping, padding, random cropping and normalization, random erasing from [32] is also a promising data augmentation procedure, especially for small dataset. This procedure puts a rectangle region on image randomly then erases its pixels with random values. It would introduce occlusion in various extents, hence leading to better generalization on a held-out dataset.

2) Warming-up Learning Rate: Warming-up learning rate scheduler proposed in [33] is a strategy to slow the optimization on models for initial epochs rather than using standard multi-step learning rate schedule. Learning rate will be restored to predefined base LR at a linear rate, and milestones of the multi-step scheduler are still allowed.

3) Label Smoothing: Similar to the image classification task, one efficient way of optimizing ReID performance is to add a classifier after the feature extractor. The classifier contributes to making embeddings of different identities separable in feature space with the help of softmax cross-entropy loss. Since this loss here is designed for correctly determining object ID, researchers called it ID loss [34]. To be clear, ID loss can be represented as:

$$\text{Identif}\,(t) = \sum_{i=1}^{K} -p_i \log\left(\hat{p}_i\right)$$

Here t denotes the target class. $\hat{p}_i$ denotes ID prediction logits of class i, and if y denotes ground-truth ID label,

$$p_i = \begin{cases} 0, y \neq i \\ 1, y = i \end{cases}$$

Label smoothing proposed in [35] is a regularization mechanism to prevent overfitting on small dataset. To be simple but specific, label smoothing alters p_i definition above into:

$$p_i = \begin{cases} 1 - \frac{N-1}{N}\varepsilon & \text{if } i = y, \\ \varepsilon/N & \text{otherwise} \end{cases}$$

where N is the number of identities and ε is a small constant, trying to lower the confidence of the model on training ID labels.

4) Center Loss: Triplet loss proposed in [36] works when optimizing inter-class distance, but it still has some flaws like the inability to provide globally optimal constraint due to random triplet sample strategy from dataset. Center loss proposed in [37] cleverly bypasses the limitations of sampling by learning a center for features of each identity and penalizes the distances between features and their corresponding identity centers. Center loss is formulated as follows:

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

where $\boldsymbol{c}_{y_i} \in \mathbb{R}^d$ denotes the y_i th identity center of features. This formula indicates center loss attempts to minimize euclidean distances between feature and its center, hence increasing intra-class compactness.

5) Last Stride: Last stride means the $stride$ hyper-parameter of the last several layers. By reducing last stride, we may obtain a feature map with a higher spatial size. According to Sun *et al*. [38], reducing last stride helps to decrease the down-sampling rate of the backbone network, which enriches the granularity of feature. Taking ResNet-50 as an example, when the input image size is $(256, 128)$, by reducing last stride from 2 to 1, we get a feature map of size 16×8, which is four times the size of original 8×4 one.

6) BatchNormNeck(BNNeck): BatchNormNeck(BNNeck) refers to the batch normalization layer between backbone and classifier. When training ReID models, the combination of ID loss and triplet loss [36] seems to work as expected while the former one is mainly responsible for optimizing intra-class distance and the latter one is mainly responsible for optimizing inter-class distance. But Luo *et al*. [39] found that these two losses show inconsistency in the same feature space. To relieve this problem, BNNeck is introduced between backbone and classifier. We denote features before the BN layer as f_t, and normalized ones as f_i. Two losses are not going to applied onto f, but be computed on f_t, f_i respectively. BNNeck ensures that f_i are gaussianly distributed near the surface of feature hypersphere, making the ID loss easier to converge. Also, since the hypersphere is almost symmetric about the coordinate axis, we need to freeze the bias of BN layer and disable the bias of fully-connected layer to avoid deviation from the origin of the feature space.

IV. Experiment Design

In this section, we deliver the experimental design to achieve our goal mentioned in Section I, that is discovering the relationship between appearance model and tracking performance. Experiments are present in detail, including dataset, evaluation metrics and settings of extensive experiments.

A. Dataset

We conduct all experiments on the MOT16 dataset [40], a famous benchmark MOT dataset involving mostly pedestrians. But there exists a problem: the data format of MOT16 dataset is not designed for ReID training as the annotation format given in MOT16 dataset only includes coordinates of each identity at every frame, which disagrees with ReID training pipeline. To overcome that, we first segmented all identities based on annotations and split them into training set and test set at a ratio of $9:1$ in a stratified fashion to keep balanced label distribution. Only pedestrians are considered and identities

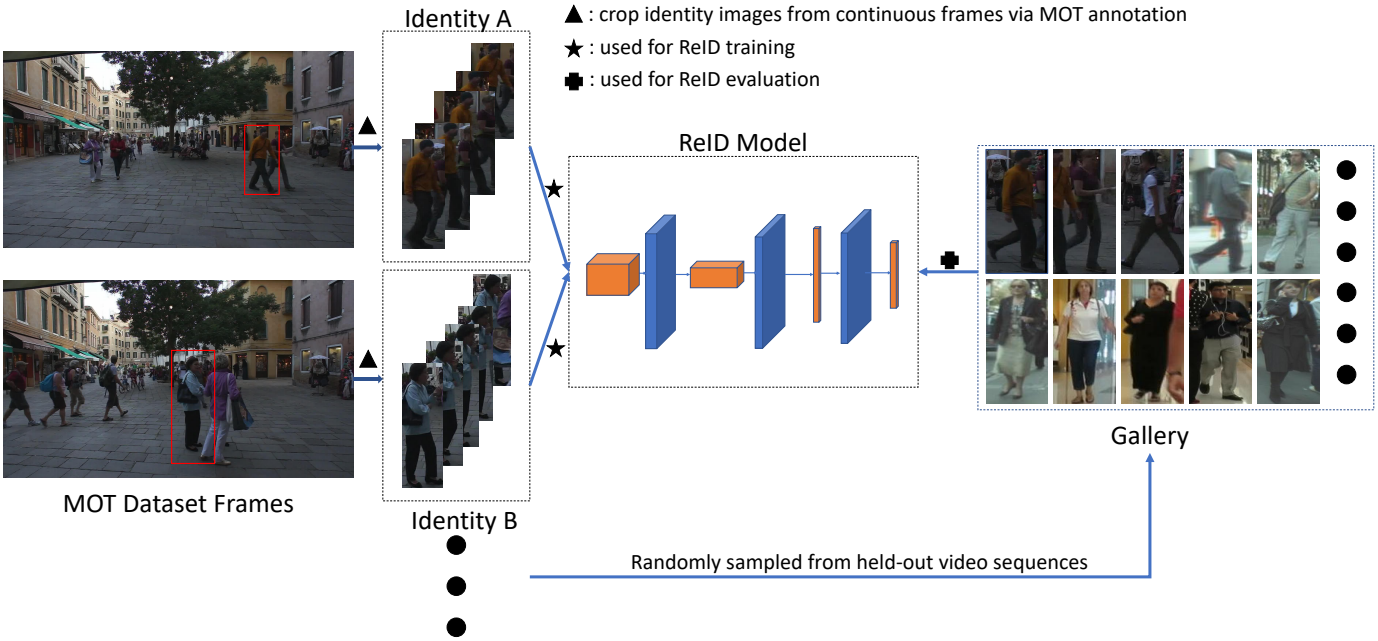

Fig. 4. Proposed MOT16 dataset pre-processing pipeline for ReID training & evaluation

with less than 10 frames are removed from dataset. Given the imprecise detection from MOT16, all experiments conducted below used public detection from [15]. The preprocessing pipeline is illustrated in Fig. 4.

For fairness and ease of evaluating tracking metrics, the sequences MOT16-09 and MOT16-10 are designated as a held-out validation set.

All images are first resized into $(256, 128)$, then flipped horizontally with a 50% probability, and randomly cropped to $(256, 128)$ after being padded with 10 pixels. If random erasing is enabled, images will be erased at randomly distributed rectangle areas.

B. Metric

To measure the performance of tracking and Re-ID accurately, metrics for each should be determined. For MOT task, it's hard to use a single metric to evaluate its performance since object tracking is never a single-objective task, so we borrowed the most frequently used metrics from [41], [42], defined as follows:

- **MOTA**($\Uparrow$): Multi-object tracking accuracy, which takes false negatives, false positives and identity switches into consideration.
- **IDsw**($\Downarrow$): Identity Switch. The number of cases where the identity of a tracked object is altered incorrectly.
- **FP**($\Downarrow$): False Positives. The number of false alarms.
- **FN**($\Downarrow$): False Negatives. The number of missed targets.

We compute all metrics mentioned in [41] when conducting experiments but focus on MOTA, FP, FN and IDsw, where $\Uparrow$ denotes that a higher metric indicates better performance, and $\Downarrow$ denotes a lower metric indicates better performance.

Besides those, CMC and mAP, two crucial metrics for ReID evaluation are calculated after each epoch for determining whether ReID performance has a huge impact on tracking metrics.

- CMC, or cumulative matching characteristics, can be illustrated as a curve on the 2-D plane, where the y-axis stands for a rank and the x-axis stands for identification accuracy at the specific rank. At our single-gallery-shot scenario, where no camera identities are provided, identification accuracy for a single query is represented as $TopK$ accuracy, defined as follows:

$$TopK = \begin{cases} 1, \text{where the first K results contain one with the same identity of query} \\ 0, \text{otherwise} \end{cases}$$

- mAP, or mean average precision, is nothing different with traditional classification mAP, except that mAP for ReID calculates AP for each query instead of for each class.

C. Experimental Settings

In this subsection, settings for traditional optimizations are first introduced. We choose ResNet-50 [43] with softmax loss and triplet loss as initial baselines. Each model has gone through 30 epochs training with Adam optimizer. Specifications of these improvements are as follows: If triplet loss is enabled, then the margin of it will be set to 0.3, and the training sampler will be a triplet sampler that samples 4 instances in every mini-batch instead of a default shuffler. If the center loss is enabled, the weight of it will be set to 0.005, and the optimizer for center loss is SGD with $lr = 0.5$. If label smoothing is enabled, ϵ will be set to 0.1. Base LR is set to

TABLE I
EVOLVING HYPER-PARAMETERS

Population Size	10
Predefined Block List	Conv2D Block(3×3) Residual Block Max/Mean Pooling Block
Predefined Channel List	[64, 128, 256]
Max Generation Number	20
Conv2D Limit	[1, 3]
Residual Limit	[3, 6]
Pool Limit	[1, 2]
Reduction Blocks Limit	3
Genetic Operation Probabilities	[0.9, 0.2]
Mutation Probabilities	[0.7, 0.1, 0.1, 0.1, 0]

3.5×10^{-4} and the learning rate holds constant if warming-up is disabled. The initial warming-up factor is $\frac{1}{3}$ and warming-up ends at epoch 10. A milestone of multi-step scheduler is set at epoch 20 if warming-up is enabled.

As for ENAS approach, we first give all hyper-parameters of Genetic-SORT in Table I, of which predefined channel list gives all possible out_channel of a block, and Conv2D/Residual/Pool Limit tells lower and upper bound of their numbers during the evolutionary progress. Two elements of genetic operation probabilities designate the chance of crossover and mutation respectively. Five elements of mutation probabilities stand for the chances of adding a residual block, adding a convolutional block, adding a pooling block, removing block and altering parameters of a certain block successively.

V. EXPERIMENT RESULT

In this section, we illustrate our experiment result. Both the traditional approach to optimizing and ENAS one yield a similar conclusion: no significant increases in tracking metrics when improving ReID metrics via introducing various methods mentioned in Section III. At the end of this section, we offer some discussion related to such a conclusion.

A. Result of Traditional Optimization

The best metrics of traditional optimizations are recorded in Table II. In Table II, metrics of two baselines are at the top line of each sub-table, while metrics of other improved models are reported in relative numbers(+/-).

Each line in Table II tells evaluation metrics of certain model on both tasks. We can see clearly that, with additional improvements, ReID performance improves while tracking deteriorates at most scenes. Label smooth, center loss, last stride all show a positive effect on ReID metrics, but no trace of a similar effect on tracking metrics are found. Random erasing worsens metrics on both tasks. BNNeck works as expected on relieving the inconsistency between triplet loss and ID loss, but still fails to perform better than its corresponding baseline on tracking task.

TABLE II
COMPARISONS ON METRICS OF DIFFERENT IMPROVEMENTS DEPLOYED ON THE BASELINES

Model \ Metric	mAP	CMC@Rank1	MOTA
resnet50_softmax	93.63%	97.13%	0.5861
+labelsmooth	+2.69%	+0.32%	-0.0013
+warmup	+3.85%	+0.95%	-0.0012
+center	+4.00%	+0.64%	-0.0006
+last_stride	+0.64%	-0.32%	-0.0021
+random_erasing	-1.06%	-0.32%	-0.0034
resnet50_triplet	86.56%	93.31%	0.5881
+softmax	+6.88%	+4.14%	-0.0052
+softmax_bnneck	+9.58%	+4.46%	-0.0016
+softmax_last_stride	+5.63%	+3.50%	-0.0071

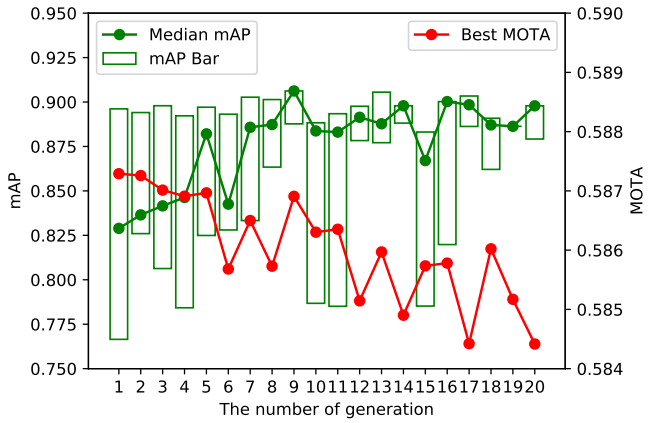

Fig. 5. The evolutionary trajectory of Genetic-SORT attempting to search architectures on MOT16 dataset.

B. Result of Genetic-SORT

In the evolutionary process which takes mAP as fitness, the numerical trends among the number of generation, ReID metrics and tracking metrics can be easily seen from Fig. 5. The upper and lower edges of each green bar indicate the highest and lowest mAP reached by a certain generation while the red curve tells the best MOTA achieved by each generation. This evolving strategy is proven to work in optimizing mAP for ReID task heuristically from Fig. 5, but it still suffers the similar fact as we discussed in Part V-A: as the number of generation grows and ReID metrics improves, tracking metrics are not showing any positive relevance with ReID metrics but oscillate and show overall negative correlation trend.

C. Discussion

Such a counter-intuitive conclusion leads one to ponder the reasons for it. Here we offer some possible explanations:

1) For detection-association two-stage tracking pipeline, there exists an upper bound to the expressiveness of global features extracted directly from detection bounding boxes.

2) There is a bottleneck for the two-stage tracking pipeline where more discriminating features will not contribute to improvement in tracking performance.

In fact, in present days numerous feature representation learning methods are gradually being discovered, and lots of them can be applied in multi-object tracking. Local feature approach, for example, learns part-aggregated features instead of global ones. Auxiliary feature approach attempts to integrate semantic information, *e.g*. pose and dressing, with the original image. Sequence feature approach learns consecutive frames in a video rather than a single image, trying to grab temporal information to assist tracking. These approaches are more promising in tracking tasks than simply learning global features.

VI. Conclusion

The goal of this paper is to explore the effectiveness of the appearance descriptor in one of conventional tracking-by-detection methods, *i.e*., DeepSORT. To achieve this, we conduct extensive experiments on it and yield the following conclusion: various optimization strategies on the global feature extractor are not reflected in the tracking metrics, and may even lead to degradation. This implies that global feature is not the best choice for data association of multi-object tracking, and researchers need to seek alternatives to further improve the performance of tracking-by-detection methods.

Acknowledgment

This work was sponsored in part by CCF-Baidu Open Fund (NO.2021PP15002000) and in part by CAAI-Huawei MindSpore Open Fund.

References

[1] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 941–951.

[2] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 366–382.

[3] L. Wen, D. Du, S. Li, X. Bian, and S. Lyu, "Learning non-uniform hypergraph for multi-object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8981–8988.

[4] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[5] H. Qian, X. Wu, and Y. Xu, *Intelligent surveillance systems*. Springer Science & Business Media, 2011, vol. 51.

[6] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[9] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[10] Y. Li, Z. Hao, and H. Lei, "Survey of convolutional neural network," *Journal of Computer Applications*, vol. 36, no. 9, pp. 2508–2515, 2016.

[11] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 748–756.

[12] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and vision computing*, vol. 32, no. 4, pp. 270–286, 2014.

[13] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[15] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *European Conference on Computer Vision*. Springer, 2016, pp. 36–42.

[16] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear lstm," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 200–215.

[17] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," *arXiv preprint arXiv:1604.03635*, 2016.

[18] C. Long, A. Haizhou, Z. Zijie, and S. Chong, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *ICME*, 2018.

[19] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[20] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *arXiv preprint arXiv:2004.01888*, 2020.

[21] T. Elsken, J. H. Metzen, F. Hutter *et al.*, "Neural architecture search: A survey." *J. Mach. Learn. Res.*, vol. 20, no. 55, pp. 1–21, 2019.

[22] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[23] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[24] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[25] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive differentiable architecture search: Bridging the depth gap between search and evaluation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1294–1303.

[26] R. Quan, X. Dong, Y. Wu, L. Zhu, and Y. Yang, "Auto-reid: Searching for a part-aware convnet for person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3750–3759.

[27] S. Mirjalili, "Genetic algorithm," in *Evolutionary algorithms and neural networks*. Springer, 2019, pp. 43–55.

[28] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[29] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379–1388.

[30] F. E. F. Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm and Evolutionary Computation*, vol. 49, pp. 62–74, 2019.

[31] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing cnn architectures using the genetic algorithm for image classification," *IEEE Transactions on Cybernetics*, 2020.

[32] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation." in *AAAI*, 2020, pp. 13 001–13 008.

[33] X. Fan, W. Jiang, H. Luo, and M. Fei, "Spherereid: Deep hypersphere manifold embedding for person re-identification," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 51–58, 2019.

[34] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–20, 2017.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[36] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[37] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.
[38] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline)," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 480–496.
[39] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
[40] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.
[41] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.
[42] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 951–958.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.