Pattern Recognition Letters 170 (2023) 70–76

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Transformer for multiple object tracking: Exploring locality to vision[☆]

Shan Wu[a,*], Amnir Hadachi[a], Chaoru Lu[b], Damien Vivet[c]

[a] ITS Lab, Institute of Computer Science, University of Tartu, Narva mnt. 18, Tartu, 51009, Estonia
[b] Centre of Metropolitan Digitalization and Smartization (MetSmart), Dept. of Built Environment, Oslo Metropolitan University, Pilestredet 46, Oslo, 0167, Norway
[c] ISAE-SUPAERO, Université de Toulouse, 10 Av. Edouard Belin, Toulouse, 31400, France

ARTICLE INFO

Article history:
Received 28 June 2022
Revised 2 January 2023
Accepted 26 April 2023
Available online 30 April 2023

Edited by: Jiwen Lu

MSC:
0000
1111

Keywords:
Multi-object tracking
Transformer
Deep learning
Locality to vision

ABSTRACT

Multi-object tracking (MOT) is a critical task in various domains, such as traffic analysis, surveillance, and autonomous vehicles. The joint-detection-and-tracking paradigm has been extensively researched, which is faster and more convenient for training and deploying over the classic tracking-by-detection paradigm while achieving state-of-the-art performance. This paper explores the possibilities of enhancing the MOT system by leveraging the prevailing convolutional neural network (CNN) and a novel vision transformer technique Locality. There are several deficiencies in the transformer adopted for computer vision tasks. While the transformers are good at modeling global information for a long embedding, the locality mechanism, which learns the local features, is missing. This could lead to negligence of small objects, which may cause security issues. We combine the TransTrack MOT system with the locality mechanism inspired by LocalViT and find that the locality-enhanced system outperforms the baseline TransTrack by 5.3% MOTA on the MOT17 dataset.

1. Introduction

Multi-object tracking (MOT) has been utilized in many applications that require human behavior analysis, traffic analysis, and scene analysis based on video feeds. It is essential that MOT, as a prior task before analysis, provides reliable and consistent tracking results.

Since MOT is a complex task consisting of object detection, classification, and tracking, the tracking-by-detection algorithms are predominant because it is sensible to leverage models that are good at each task and combine them together. Nonetheless, a mega model would prolong the processing overhead. It would also be complicated to train such a model, making the requirement of using a tracking system harsher for those without enough computational resources.

While the joint-detection-and-tracking paradigm will be the future trend of MOT, there emerge new algorithms that utilize the transformer and query-key mechanism. The new architecture showed us great potential to simultaneously produce the tracking bounding boxes and classes. Transformers are good at modeling long embeddings by the attention mechanism. Thus, they find the global dependencies among the image patches. However, there is no place for the intrinsic information interaction, which may cause erratic tracking results because of the movement and the scale of target objects. Moreover, the spatial information is ignored inside the transformer after flattening the feature maps into embeddings. A new mechanism needs to be introduced for the attention module.

Inspired by the novel locality mechanism proposed by [1], we update the transformer-based MOT system TransTrack with depthwise convolutional layers added into the encoders of the transformer. Similar to the original locality block and the MobileNet [2], we substitute the feed-forward network with the inverted residual blocks in all transformer encoder layers to extract both global and local features in encoders. In order to keep the spatial information of the feature maps, the embedding features will be converted to two-dimensional feature maps for convolutional layers, and the output feature map will be restored to embedding features for the next encoder layer. Since TransTrack uses pyramid feature maps from the ResNet backbone, which are concatenated into one embedding for the transformer, we process each feature level separately, so each feature level has a dedicated locality block.

[☆] Editor: Jiwen Lu.
[*] Corresponding author.
E-mail addresses: Shan.Wu@ut.ee (S. Wu), hadachi@ut.ee (A. Hadachi), chaorulu@oslomet.no (C. Lu), damien.vivet@isae-supaero.fr (D. Vivet).

https://doi.org/10.1016/j.patrec.2023.04.016
0167-8655/© 2023 Elsevier B.V. All rights reserved.

Experiments on the MOT17 validation set show that the locality-enhanced architecture achieved a better MOTA score at 72.4%, which is 5.3% higher than the baseline TransTrack [3].

2. Related works

Unfolding recent literature, we notice considerable work has been done in developing and advancing state of the art in multi-object tracking approaches and systems. Among the proposed methods, we can categorize three significant domains (Tracking by detection, joint detection and tracking, and global and local perception in transformer).

2.1. Tracking-by-detection

As its name state, the tracking-by-detection is based on detecting and then predicting the location on the frame of the subsequent detection. A good illustration of this approach is mentioned in work done in [4]. The presented method introduces a Region Proposal Network (RPN) for sharing convolutional features of the entire image with the detection network. In addition, the RPN architecture is a fully convolutional network with the ability to predict the next position of the detected objects simultaneously. However, the approach is more designed for accurate detection, not the tracking aspect, making it unclear to make a precise judgment about tracking performance.

Another approach was presented in [5]. The technique uses a transformer network with a set-based global loss, strengthening unique predictions via bipartite matching. This design allows for eliminating hand-designed components and keeps comparable results to the Faster R-CNN baseline; plus, the implementation is flexible and easily extensible to other applications such as segmentation. However, the approach is not robust in detecting and tracking small objects. To address this weakness, the work proposed by [6] presented a deformable Transformer that relies only on a small set of key sampling points around a reference. The results obtained outperformed DETR.

Furthermore, it is essential to point out that tracking-by-detection methods depend greatly on the detection performance to have a good tracking result. Moreover, the online aspect is crucial, especially for the real-life application of tracking systems. An excellent example in this direction is [7], where the authors explored the usage of a combination of simple techniques to perform tracking, such as the Kalman filter with deep learning method for the detection. The created framework showed the best results in its class even though it had some identity switch cases. To address this aspect, [8] introduced measurement-to-track associations in visual appearance space using the nearest neighbor, while [9] proposed a hybrid track association algorithm. Both turn out to be effective in reducing identity switches. However, there are many other challenges in this section, such as data imbalance, which affects the tracking significantly. To this end, the literature has many applications that tried to address this aspect by introducing a data association approach based on spatial and temporal attention mechanism [10]. This aspect allows good handling of noisy detections and occlusion cases.

2.2. Joint-detection-and-tracking

The second family domain is about joint-detection-and-tracking, the tracking pipeline that performs object detection followed by temporal association. This is thanks to the applications of deep networks, which open new opportunities and challenges. Hence, many methods proposed are getting faster, more accurate, and sometimes even more straightforward. For example, [11] demonstrated the capability of going beyond the classical steps for tracking from detection to data association in an end-to-end fashion using an online model named Chained-Tracker (CTracker). The proposed solution relies on chaining paired bounding boxes regression output to identify-attention of the objects. This has resulted in two novel aspects: the chained structure and pairing attention regression with acceptable results. However, some other approaches outperformed it, for example, in [12], where the authors presented an end-to-end CenterTrack method. The process is point-based, meaning each object is represented as a point for tracking throughout time. In addition, the system works in online mode, and it gets greedy for object association over time.

Furthermore, the use of deep learning and neural networks has opened the possibility of exploring shared objectives of detecting and tracking simultaneously within one network. The work done in [13] is a good example. The MOT model is designed to simultaneously learn target detection and appearance embedding in a shared model. The model incorporates a fast and straightforward association model that works in union with the joint model. The results were comparable to state-of-the-art. However, the recent paper about TransTrack [3] demonstrated interesting and significant outperformance especially concerning MOT17 challenges. The method is built based on a transformer, and its signature uses the learned object query as an input for both detection objects architecture and the track query.

2.3. Global and local perception in transformer

Finally, our last category is about the perception in the transformer from a global and local point of view. It is essential to discuss the first paper addressing global connectivity with a focus on attention. In this paper, [14], the authors demonstrated how a transformer model based only on attention could be trained faster and outperform the state-of-the-art. Indeed, the application of the new approach was made on machine translation, but it opened the door for application in tracking and detection. For example, the latest work done in this direction is [15], which is about a novel attention-based feature fusion network inspired by transformer architecture using a Siamese-like feature extraction backbone (TransT). The outcome proved the approach has potential since it achieved good results in different challenging datasets.

Moreover, adopting a transformer architecture designed for language processing to vision raised several challenges. These challenges are mainly related to the differences between the two domains. For instance, compared to language processing, there are large variations in the scale of visual entities and the impact of the high resolution of pixels in images. That is why not only do we find interest in exploring global points of view but also local ones. Hence, there are many works focused on researching the locality aspect. A good example is proposed in [16]. The method has proven good performance and is based on a hierarchical Transformer augmented with shifted windows for its representation. This aspect makes the process efficient in cross-window connection, thanks to non-overlapping local windows due to the limiting self-attention computation.

As we mentioned, the transformer is particularly good at modeling global interaction, but the local one still lacks information exchange within the local region. To address this issue, one of the exciting research works that has been done is trying to bring locality to vision as mentioned in [1]. The proposed design was done by introducing depth-wise convolution into the feed-forward network. This technique has allowed for increasing the accuracy performance within 2% to 3%. Another approach dealing with the same problem was mentioned in [17], where the authors proposed a transformer in transformer architecture. This way, introducing a transformer within another one handles local patches and reinforces the locality interaction. Of course, the method is greedy

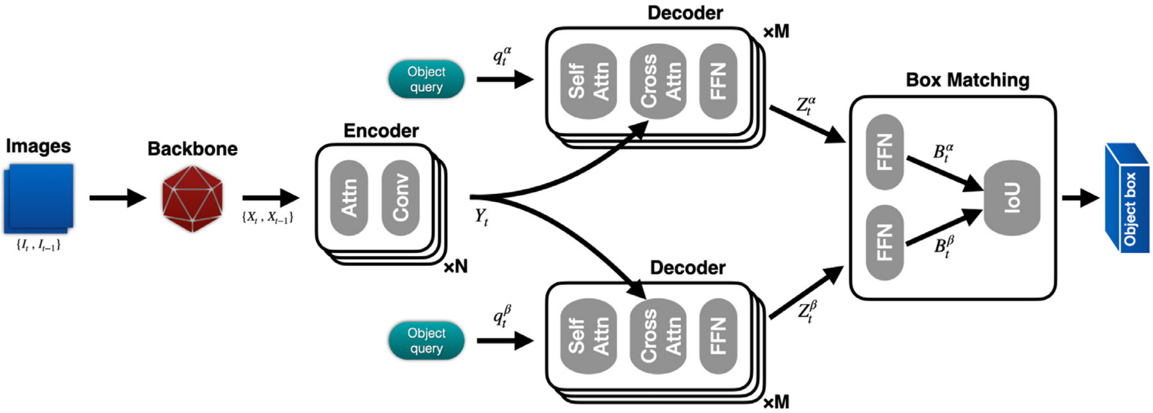

Fig. 1. The pipeline of our locality-enhanced MOT system based on TransTrack. Two video frames are used to extract features. A locality-enhanced encoder will take the combination of two feature maps and find the correlations among the features, followed by two parallel decoders specialized for detection and tracking tasks. Both decoders will generate bounding boxes based on a trainable query and the features from the encoder. All bounding boxes are merged by an IoU matching technique to form the final object bounding boxes.

concerning computational time in training, but it managed to achieve a 1.7% increase in accuracy compared to ImageNet.

The possibility of exploring the locality domain in image transformers is a valuable direction that could further broaden the limit of MOT systems. In this paper, we implement the state-of-the-art locality mechanism on top of the novel transformer-based MOT model. Additionally, our locality module leverages multi-scale feature maps instead of only one feature map from the backbone model. Moreover, a thorough quantitative and qualitative evaluation is conducted comparing with the vanilla model to demonstrate the improvements of the multi-scale locality module.

3. Methodology

Fig. 1 visualizes the experimented MOT pipeline, which is based on the recently proposed TransTrack architecture. Unlike the image classification transformers proposed recently, our MOT pipeline still keeps an encoder-decoder architecture for detecting, classifying, and tracking objects. Thus, we will first describe the pipeline of our system and introduce the locality later based on the semantic meanings of each component in the pipeline.

3.1. MOT pipeline

The original MOT pipeline that we use consists of a convolutional neural network (CNN) backbone, a transformer encoder, two parallel transformer decoders, two learnable object queries, and a bounding box matching component. For the details of the implementation, please refer to the original paper [3].

Firstly, a conventional CNN, namely ResNet50, is used to extract features $\{X_t, X_{t-1}\}$ from two consecutive video frames $\{I_t, I_{t-1}\}$. Next, the features from two frames are concatenated and further processed by the encoder Θ to find the correlations of features (1). In a query-key paradigm, the output embedding χ_θ from the encoder will be projected into keys and values in decoders, where a trainable object query will be used to interact with the keys and the values.

$$\begin{aligned} \chi_t &= Concat(X_t, X_{t-1}) \\ \chi_\theta &= \Theta(\chi_t) \end{aligned} \tag{1}$$

where χ_t is the concatenation of two feature maps X_t and X_{t-1}. Θ is a composite of N encoder layers, where each layer consists of a self-attention module $Attn(\cdot)$ followed by a feed-forward network (FFN) $f_\Theta(\cdot)$. Skip connection and layer normalization are used in both submodules as shown in Eq. (2).

$$\begin{aligned} \chi_{t,i}^{attn} &= Norm(Attn(\chi_{t,i-1}) + \chi_{t,i-1}), \ 1 < i \le N \\ \chi_{t,i} &= Norm(f_\Theta(\chi_{t,i}^{attn}) + \chi_{t,i}^{attn}) \end{aligned} \tag{2}$$

Once the final encoder embedding is generated, two decoders $\{\Pi_\alpha, \Pi_\beta\}$, each of which has M layers, will work simultaneously for the detection and tracking tasks, respectively. In the detection decoder, detection proposals Z_t^α are produced based on a trainable object query q_t^α along with the projected keys and values derived from χ_θ. At first, a self-attention module is applied to the query, followed by a cross-attention module where key-value pairs are involved. In the end, an FFN adds non-linearity to the model and learns more features. The tracking decoder follows the same process except for the tracking object query q_t^β, which is the detection proposals Z_{t-1}^α from the previous frame. Equation (3) shows the summary of the decoders.

$$\begin{aligned} Z_t^\alpha &= \Pi_\alpha(\chi_\theta, q_t^\alpha) \\ Z_t^\beta &= \Pi_\beta(\chi_\theta, q_t^\beta) \\ q_t^\beta &= Z_{t-1}^\alpha \end{aligned} \tag{3}$$

In the last box matching module, detection and tracking proposals will be further regressed into the detection and tracking bounding boxes $\{B_t^\alpha, B_t^\beta\}$ using an FFN, as shown in (4).

$$\begin{aligned} B_t^\alpha &= f_{\Pi_\alpha}(Z_t^\alpha) \\ B_t^\beta &= f_{\Pi_\beta}(Z_t^\beta) \end{aligned} \tag{4}$$

After obtaining the bounding boxes from both decoders, an IoU matching algorithm [18] is utilized for the final results. In this process, detection boxes are matched to tracking boxes, and the unmatched boxes will be new objects introduced in the current frame.

3.2. Locality

It is common knowledge that the convolution operation utilizes a kernel as a sliding window to aggregate local features on an image. The same mechanism is missing in transformers because the attention operation only attends to the global information within the embedding. Hence, the complementary characteristics of CNNs and transformers lead to a new experiment of adding convolutional layers into the transformer blocks.

In our experimental MOT pipeline, only the encoder can be tweaked for the locality mechanism since it is the only component that works with feature maps ($X_t \in \mathbb{R}^{d \times h \times w}$) as an input source. More specifically, χ_θ is the attended feature embedding containing

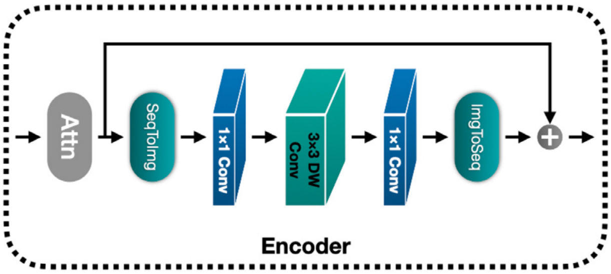

Fig. 2. The figure shows the tweaked encoder of our system, which leverages a locality module (Inverted residual block) instead of a classic feed-forward network. Flattened 1D features are reshaped into a 2D feature map, and the convolutional layers will extract local features from it. 1×1 convolution and 3×3 depth-wise convolution operations are used to control the number of parameters compared with the FFN. The feature map will be flattened before the skip connection to restore the original shape for the next iteration.

the global dependencies of an image. Later, χ_θ will be projected into keys and values in the decoders. Nonetheless, decoders mainly regress on the object queries q_t, which is a learnable parameter containing the information of tracked objects (e.g., features of human bodies). If a key matches the query more, its value will get more weight to form the output. In this way, it is reasonable to introduce the locality mechanism inside the encoder to enrich the feature representations.

To be compatible with convolutional layers, the shape of the original feature map must be restored. Considering a feature embedding χ_t^{attn} after the $Attn(\cdot)$ operation in the encoder Θ, χ_t^{attn} is reshaped to 2D feature map χ_t^{2D}, which fits in the shape of (d, h, w), where d is the number of channels, $h = H/p$ and $w = W/p$, and p is the stride of this feature map while H and W are the shapes of the original image I. After the convolutional layers, the feature map χ_t^{2D} must be converted back to its 1D shape χ_t^{attn} for the following encoder layers. Eq. (5) shows the process of converting a 1D embedding to a 2D feature map and vice versa.

$$\begin{aligned} &\chi_t^{2D} = SeqToImg(\chi_t^{attn}), \ \ \chi_t^{2D} \in \mathbb{R}^{d\times h\times w} \\ &\chi_t^{attn} = ImgToSeq(\chi_t^{2D}), \ \ \chi_t^{attn} \in \mathbb{R}^{j\times d} \\ &\text{where } j = h \times w \end{aligned} \tag{5}$$

Similar to Li et al. [1], inverted residual blocks are used as our locality module to substitute the feed-forward network $f_\Theta(\cdot)$ inside the encoder because of its efficiency and lightweight in computation. As shown in Fig. 2, the locality module Φ_Θ includes a 1×1 convolutional layer to expand the number of channels to γd, a 3×3 depth-wise convolutional layer that greatly reduces the number of parameters compared with the classic one, and another 1×1 convolutional layer to restore the original number of channels to d, which allows the skip connection across the whole block. Hence, the tweaked layer of an encoder Θ' is described in Eq. (6).

$$\begin{aligned} \chi_{t,i}^{attn} &= Norm(Attn(\chi_{t,i-1}) + \chi_{t,i-1}), \ \ 1 < i \le N \\ \chi_{t,i} &= Norm(ImgToSeq(\Phi_\Theta(SeqToImg(\chi_{t,i}^{attn}))) + \chi_{t,i}^{attn}) \end{aligned} \tag{6}$$

where Φ_Θ is the convolutional layers of the locality module. Since the aim of using FFN in the original encoder is to enhance the latent features and create local dependencies [14], a convolutional network could do this task better. The combination of global attention and local convolution leads to more comprehensive feature representations.

3.3. Multi-scale features

In addition to the tweaked encoder with a locality module, the experimented MOT system also features a multi-scale attention module, which leverages feature maps in different scales instead of the output of the last layer from the backbone. Let $\{X_t^l\}^L$ represents the multi-scale output at time t from the backbone, which contains L levels. $\{X_t^l\}^L$ are flattened and concatenated into χ_t for the attention module (review [6] for attention module details).

Similarly, we deploy L locality modules for each level of feature maps. The attended feature χ_t^{attn} is split by the spatial shapes and levels into $\{\chi_t^l\}^L$, and passed to a dedicated locality module Φ_Θ^l separately to find the local correlations in different scales. In the end, the output features are flattened and concatenated again for the next layer in the encoder Θ'. Equation (7) summarizes the multi-scale locality module.

$$\begin{aligned} \chi_{t,i}^{attn} &= Norm(Attn(\chi_{t,i-1}) + \chi_{t,i-1}), \ \ 1 < i \le N \\ \{\chi_{t,i}^l\}^L &= Split(\chi_{t,i}^{attn}) \\ \chi_{t,i}^l &= ImgToSeq(\Phi_\Theta^l(SeqToImg(\chi_{t,i}^l))), \ \ \forall\, l \in L \\ \chi_{t,i}^L &= Concat(\chi_{t,i}^l, \ldots) \\ \chi_{t,i} &= Norm(\chi_{t,i}^L + \chi_{t,i}^{attn}) \end{aligned} \tag{7}$$

Because we tweak the module in the encoder while keeping the same pipeline and training target, the same loss function (8) could be applied to this architecture.

$$\mathcal{L} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{L1} \cdot \mathcal{L}_{L1} + \lambda_{giou} \cdot \mathcal{L}_{giou} \tag{8}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{L1}$, and $\mathcal{L}_{giou}$ are focal loss [19] of predicted object classes, L1 loss of predicted object bounding boxes, and generalized IoU loss [20] of the bounding boxes, respectively. λ is the coefficient of the corresponding loss value.

4. Experiments

This section will discuss the experiment details about the datasets, training schemes, and different architecture variants, as well as their comparisons and evaluations. With the help of the locality mechanism, we are able to push the MOT task boundary further. Both advantages and disadvantages of the transformer in the system will be covered and analyzed to give a comprehensive insight into its capacities.

4.1. Datasets

MOT17 dataset [21], which includes 7 training sequences and 7 testing sequences with 3 sets of detected bounding boxes, is used in our experiments. The dataset focuses on pedestrians only, and only pedestrians are annotated in the ground truths. Due to that there is no official train-validation split, and it is suggested to fine-tune the model using the training set. Thus, we divide it into two halves for training and validation similar to Sun et al. [3]. The validation set is used for model comparison during the evaluation process for three reasons. Firstly, it is suggested to use the validation set for fine-tuning. Secondly, the testing set does not provide any ground-truth label. Thirdly, provided detections are recommended to be leveraged for evaluating a tracking system online.

In addition to MOT Challenge Dataset, CrowdHuman dataset [22] is also used for pre-training the model. There are 470K human instances in the dataset, including 15000 training images, 4370 evaluation images, and 5000 testing images. The human instance density and diversity of the images make CrowdHuman a suitable dataset to pre-train a pedestrian tracking system.

4.2. Metrics

Various tracking metrics [21,23] are used while evaluating and testing the model. MOTA (Multiple Object Tracking Accuracy, in equation 9) is the main metric among all other metrics, which describes the overall accuracy between true positives (TP) and all

Table 1
The table shows a full comparison of all available models. Three locality-enhanced variants are listed as well as three models trained with TransTrack architecture. Our best locality-enhanced model outperforms all other models in many metrics, such as MOTA, MOTP, MT, ML, and FN, while the other metrics are also comparable with other models.

Model	MOTA	MOTP	MT	PT	ML	FP	FN	IDs
TransTrack	66.5%	83.4%	39.5%	42.5%	18.0%	2.9%	30.1%	0.6%
TransTrack*	67.1%	83.5%	41.9%	39.8%	18.3%	3.1%	29.4%	0.5%
TransTrack-mix	72.0%	85.2%	49.3%	37.8%	13.0%	2.0%	25.5%	0.4%
Locality	68.5%	85.2%	45.1%	37.8%	17.1%	1.5%	29.6%	0.5%
Locality+	**72.4%**	**85.4%**	**54.0%**	**36.6%**	**9.4%**	**3.8%**	**23.1%**	**0.7%**
Locality+	72.1%	85.5%	54.3%	36.6%	9.1%	4.4%	23.0%	0.6%

other false detections, including false positives (FP), false negatives (FN), and ID switches (IDSW).

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDSW_t)}{\sum_t GT_t} \tag{9}$$

where TPs are the detected instances which are also ground truths, FPs are the invalid instances detected, FNs are the missed instances that should be detected, IDSWs record the tracking ID switches, and GTs are the ground truths.

Other MOT metrics are assessed, including MOTP (Multiple Object Tracking Precision, in Eq. (10)), mostly tracked (MT), mostly lost (ML), and partially tracked (PT). MT counts the trajectories tracked over 80% of the ground truth, while ML counts those tracked less than 20%, and all trajectories in between are PT. The final results of all metrics are averaged for all video sequences that have been tested.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{10}$$

x=1, where $d_{t,i}$ are the overlapped bounding boxes between detected ones and the ground truths in time frame t for all instances, and c_t is the number of those matched bounding boxes in time frame t. This metric only shows the precision of detected boxes, ignoring whether the detection is correct.

4.3. Training details

During our experiments, feature maps with strides ×8, ×16, ×32 are fetched from our ResNet50 backbone for each time frame t. The locality module is deployed in the encoder with the expansion ratio set to 4. Batch normalization is added after every convolution. Dropout layers are removed, which were initially used in the FFN. H-swish activation [24] is selected for all convolutions, and squeeze-and-excitation (SE) module [25] is added between depth-wise convolution and the last convolution. We investigate three variants of locality modules – shared-weight locality, multi-scale locality, and locality with more layers. Other models are evaluated and compared, including the original TransTrack system, both shared publicly and self-trained, using 4 Tesla-V100 GPUs.

The other details are set to match [3] regarding the optimizer, batch size, learning rates, as well as image augmentation techniques. In addition to the classic training with two consecutive frames or two randomly selected frames within a short time difference, it is also possible to train the MOT system by randomly cropping and scaling a single image from the sequence.

4.4. Evaluation

When we modify the components of a network, we barely increase the computational cost because the tracking system is already quite complex and large compared with other single-task models. Hence, the inverted residual block is a good candidate for this purpose.

Table 2
The table compares computational complexity and the number of parameters for the original FFN and the locality module in the encoder.

Module	#param	Mac
Locality	1.06M	0.42G
FFN	0.53M	0.41G

Table 3
The table compares the performance of the models trained by a mixed dataset and only the MOT17 training partition.

Dataset	MOTA	MOTP	MT	PT	ML
Mixed data	68.5%	85.2%	45.1%	37.8%	17.1%
MOT17 half	60.8%	83.5%	33.0%	44.5%	22.4%

Table 2 shows the computational complexity and number of parameters for both the locality module and the FFN inside the encoder. We use MultiplyAccumulate operation (Mac) to demonstrate the calculations needed for such a module since a convolutional layer contains a matrix multiplication operation (op) and an addition op. However, some metrics count it as two ops, while some hardware can combine two ops into one, so we only count the combination of multiplications and additions.

The locality module has a doubled amount of parameters compared with the original FFN mainly because of the SE layer, which has nearly half of the total parameters. Nonetheless, the SE layer is very efficient that only takes 0.001 GMac because it only pools the input to size 1 to calculate a scaling factor and scale the original input. Two convolutional layers (the first and the last) take up almost all Mac due to the change of channels for the feature maps. The computational cost of the locality module mainly depends on the expansion ratio between the convolutions. Considering our purpose of not increasing the model's complexity, we set the ratio to 4, which costs almost the same as the original FFN.

During the experiments, we verified that training with a mixed dataset of Crowdhuman and MOT17 training partition could improve overall performance. As shown in Table 3, the MOTA performance of our system boosted drastically after mixing the training datasets. MT increased to 45.1% from 33.0%, showing that the Crowdhuman dataset does help the system recognize pedestrians better. There are larger volumes, more crowded, and higher quality pictures of pedestrians in the Crowdhuman dataset, providing good quality learning features. When using only the MOT17 training partition, more trajectories are partially tracked. Due to the lack of pedestrian variety and sample quantity, a system with a MOT17 dataset can hardly fully track pedestrians. Nevertheless, MOTP improved by 1.7% means that the detected bounding boxes' quality is almost the same. However, more objects are detected and tracked, which indicates that the variety and volume of new samples helped.

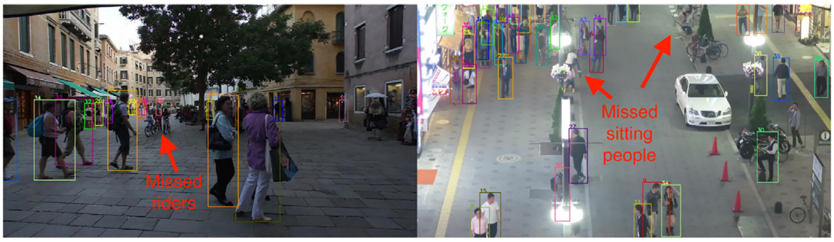

Fig. 3. Screenshots show the missed objects. The scene on the top is captured from the MOT17-02 sequence, and the bottom is captured from the MOT17-04 sequence.

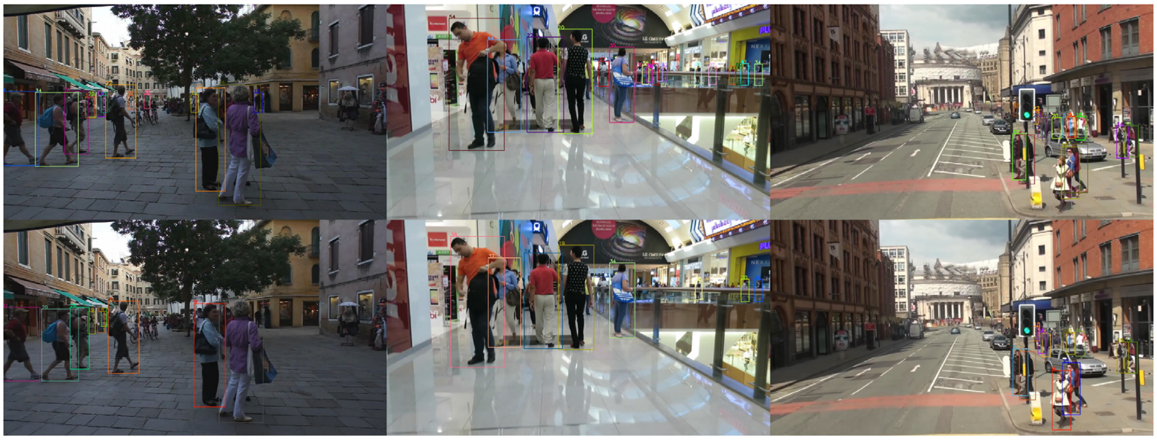

Fig. 4. A visual comparison between the Locality+ model (upper row) and the TransTrack* model (lower row).

As mentioned in Section 4.3, three MOT models with locality modules are trained. All models leverage multi-scale feature maps since the output resolution in the last layer of the backbone would be too small to detect small objects. The difference comes from the locality module, where the first one shares the weights of the locality module for the multi-scale feature maps; the second one adopts dedicated locality modules for the multi-scale feature maps; and the third one, which is based on the previous one but added more layers for the ablation study (namely Locality, Locality+, and Locality++ in Table 1 respectively).

Besides, three models from TransTrack architecture (listed in Table 1) have been tested to compare with our models. TransTrack is the default model trained with suggested training procedures and parameters. TransTrack* is the model trained and provided publicly by the author, which is also our baseline model. TransTrack-mix is the model trained with a mixed training set like Locality+.

From the experiment results in Table 1, the Locality+ achieved the best overall MOTA performance, which bolsters the effect of the locality mechanism. Compared with the locality models, the one with dedicated locality modules advances nearly all metrics. It is because the shared weights locality module could not robustly extract features from different scales. When using dedicated locality modules for multi-scale feature maps, local features from all levels are extracted and combined to enrich the original features. Locality+ gets the best mostly-tracked result and the least mostly-lost result among all competitors, showing the effectiveness of leveraging local features to track objects.

Compared with the TransTrack-mix model, Locality+ outperforms it by 0.4% in MOTA, but Locality+ is much better in tracking objects regarding the MT and ML results. False negative is also an essential metric in traffic, which determines whether an object will be detected or not. Locality+ achieves a much lower FN rate by trading off the FP rate, which is still comparable with the baseline model. However, the ID switch rate is above the average due to the higher number of detections, which also increases the FP rate. MOTP defines how accurate the detected bounding boxes are, and both locality-enhanced models got an over 85% MOTP accuracy, while only one TransTrack model achieved this. Besides the help of the mixed dataset, convolutions also contributed to the detection accuracy.

In the Locality++ model, we added one more 3×3 convolutional layer in the locality module. However, there is no significant improvement using the same dataset. The additional convolutional layer contributes to many metrics, such as MOTP, MT, ML, FN, and IDs, which show better tracking precision but not overall accuracy. In other words, the inverted residual block is optimal in this structure.

Figure 4 shows a visual comparison between the Locality+ model and the TransTrack* model. In general, Locality+ tracks more pedestrians compared to the original TransTrack, especially those who are far away and occluded. It is crucial to detect pedestrians in advance if we deploy the system on an unmanned vehicle. Additionally, Locality+ works well for pedestrians in different scales.

In our experiments, there are still some occasions people are not detected, such as sitting people in the MOT17-04 sequence and riding children in the MOT17-02 sequence (shown in Fig. 3). Those undetected people usually are in different poses other than standing or walking. The training process should pay more attention to rare cases, which will be our future work for investigation.

5. Conclusion

In this paper, we experimented with the possibilities of combining novel transformers with classic convolutional networks. This way, local features could be learned from locality modules (convolutions) while the global features are exchanged in the multi-head attentions. The inverted residual block module is selected for the locality module and deployed in every encoder layer. Our tweaked model with multi-scale locality modules achieved 72.4% in MOTA and 85.4% in MOTP on the MOT17 validation set, which outperforms the baseline TransTrack. The results show the potential of leveraging convolutions to extend the limit of transformers in MOT.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Li, K. Zhang, J. Cao, R. Timofte, L. Van Gool, LocalViT: Bringing locality to vision transformers, arXiv preprint arXiv:2104.05707(2021).
[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861(2017).
[3] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, P. Luo, TransTrack: multiple object tracking with transformer, arXiv preprint arXiv:2012.15460(2020).
[4] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.
[6] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable transformers for end-to-end object detection, arXiv preprint arXiv:2010.04159(2020).
[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), IEEE, 2016, pp. 3464–3468.
[8] N. Wojke, A. Bewley, 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.
[9] X. Lin, C.-T. Li, V. Sanchez, C. Maple, On the detection-to-track association for online multi-object tracking, Pattern Recognit. Lett. 146 (2021) 200–207.
[10] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, 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.
[11] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Fu, Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 145–161.
[12] X. Zhou, V. Koltun, P. Krähenbühl, Tracking objects as points, in: European Conference on Computer Vision, Springer, 2020, pp. 474–490.
[13] Z. Wang, L. Zheng, Y. Liu, Y. Li, S. Wang, Towards real-time multi-object tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 107–122.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[15] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, H. Lu, Transformer tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8126–8135.
[16] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.
[17] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, Y. Wang, Transformer in transformer, Adv. Neural Inf. Process. Syst. 34 (2021).
[18] H.W. Kuhn, The hungarian method for the assignment problem, Naval Res. Logist. Q. 2 (1–2) (1955) 83–97.
[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[20] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.
[21] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler, MOT16: a benchmark for multi-object tracking, arXiv preprint arXiv:1603.00831(2016).
[22] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, J. Sun, CrowdHuman: a benchmark for detecting human in a crowd, arXiv preprint arXiv:1805.00123(2018).
[23] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the clear MOT metrics, EURASIP J. Image Video Process. 2008 (2008) 1–10.
[24] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.
[25] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.