Text Query based Traffic Video Event Retrieval with Global-Local Fusion Embedding

Thang-Long Nguyen-Ho[*1,2], Minh-Khoi Pham[*1,2], Tien-Phat Nguyen[*1,2,3], Hai-Dang Nguyen[1,2], Minh N. Do[4], Tam V. Nguyen[5], and Minh-Triet Tran[†1,2,3]

[1]University of Science, Ho Chi Minh City, Vietnam
[2]Viet Nam National University, Ho Chi Minh City, Vietnam
[3]John von Neumann Institute, Ho Chi Minh City, Vietnam
[4]University of Illinois at Urbana-Champaign, U.S.
[5]University of Dayton, U.S.
{nhtlong,pmkhoi,ntphat,nhdang}@selab.hcmus.edu.vn,
minhdo@illinois.edu,tamnguyen@udayton.edu, tmtriet@fit.hcmus.edu.vn

Abstract

Retrieving event videos based on textual description is a promising research topic in the fast-growing data field. However, traffic data increases every day, so it is essential to need intelligent traffic system management in conjunction with humans to speed up the search. We propose a multi-module system that delivers accurate results that meet objectives, including explainability and scalability at the same time. Our solution considers neighbors entities related to the mentioned object to represent an event by rule-based, which can represent an event by the relationship of multiple objects. In our proposed retrieval method, we add our modified model of Alibaba solution with the post-processing techniques from HCMUS method in AI City Challenge 2021 to boost the explainability of the obtained results. As the traffic data is vehicle-centric, we apply two language and image modules to analyze the input data and obtain the global properties of the context and the internal attributes of the vehicle. We introduce a one-on-one dual training strategy for each representation vector to optimize the interior features for the query. Finally, a refinement module gathers previous results to enhance the final retrieval result. We benchmarked our approach on the data of the AI City Challenge 2022 and obtained the competitive results at an MMR of 0.3611. We were ranked in the top 4 on 50% of the test set and in the top 5 on the full set.

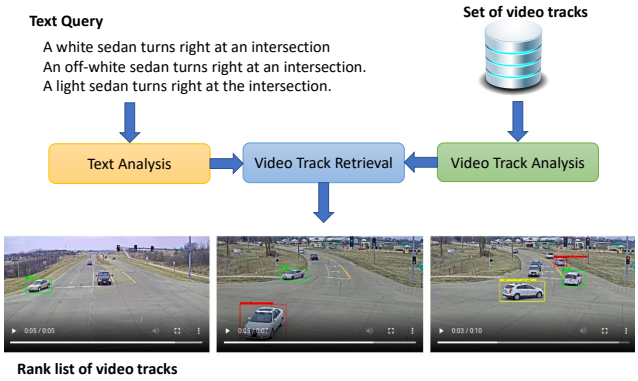

Figure 1. Traffic video event retrieval from text query.

1. Introduction

In urban city planning, analyzing and understanding the traffic patterns from traffic videos are very important. However, the storage of traffic videos is challenging due to the data's exponential growth. Therefore, it is non-trivial to retrieve the traffic video event from the extreme large scale traffic video data. In this paper, we draw our attention to the traffic video event retrieval via the input text query [4]. In practice, natural language description offers a tremendous help to specify vehicle track queries. Figure 1 visualizes the input and the output of the mentioned research task. As can be seen from the figure, the traffic related events are retrieved and ranked according to the input query.

[*]The first three authors share the equal contribution.
[†]Corresponding author. Email: tmtriet@fit.hcmus.edu.vn

In the proposed work, we utilize the global and local features from both the textual and the visual cues. Given the input text query, we extract the global context and the local attributes. Likewise, we extract the global and local features from the video track. We later join both global and local features and feed them into the fusion branch to effectively combine the information.

Given an input text query and video tracks, our retrieval model returns a list of candidates. Then, we refine the results in the post-processing phase by taking the textual and visual attributes into consideration. Furthermore, we also exploit the actions of the main object, such as "*stop*", "*turn left*", "*turn right*", and the inter-relationship between the main object and other vehicles based on their trajectories to boost the final retrieval results.

We submit our proposed method to Track 2 in AI City Challenge 2022. The CityFlow-NL dataset [4] was first used for the natural language-based vehicle retrieval task in AI City Challenge 2021 and 2022. For testing in this track, there are 184 video tracks, each contains the sequence of bounding boxes of a vehicle of interest, and 184 sets of corresponding text queries. The output here is a list of related video tracks corresponding to the input query. Our method achieves good results in Track 2 of AI City Challenge 2022. In particular, we achieve rank 5 with the MRR = 0.3611, Recall@5 = 0.5489, and Recall@10 = 0.6467.

The rest of this paper is organized as follows. In Section 2, we briefly review existing work related to this task. Our proposed method for video track retrieval from text queries is presented in Section 3. Experimental results from Track 2 of AI City Challenge 2022 are then reported and discussed in Section 4. Finally, Section 5 draws the conclusion and open problems for future work.

2. Related Work

Intelligent transportation systems take advantage of the ever-increasing amount of data to automate traffic flow management in smart cities. Image data has qualified to be utilized by automatic learning models. AI City Challenge series[7, 8, 9, 11, 10] provides various extensive video datasets that capture real-world traffic with a variety of scenarios available in the real world: including tracking vehicles through multiple cameras at intersections across the city, retrieving a target-based track vehicle that describes both its shape and motion characteristics and its relationships to other vehicles or the environment, detecting abnomal events in traffic, recognize distracted behaviors of drivers; etc.

We perform language-based retrieval of video containing media features in the tracked-vehicle retrieval by natural language descriptions challenge. To keep up to date with the trending solutions, some of the insight methods in 2021 at this challenge ([1, 16, 12, 14]) are applying a representation learning-based model to descriptive and textual query inputs to perform the retrieval task. Experiments show that simple networks with different representations provide competitive results for this approach.

For text representation tasks, many popular methods use models based on pretrained transformers (BERT [3], Roberta[6]) to embed the input query to perform representation semantic modeling. However, the limitation of this relative of transformers is that it takes much time to converge and is difficult to train from scratch. Bai *et al.* [1] use back-translation to generate various sentences. Park *et al.* [13] and Nguyen *et al.* [12] apply conventional natural language tools to analyze queries. Extract helpful cues such as vehicle type, color, motion attributes, or relationship to nearby vehicles.

For the video modeling task, the best performing team, Bai et al [1] propose a dual-flow architecture. One flow aims to extract vehicle background information, and the other for the trajectory of the target vehicle. However, the main limitation is that the model cannot learn correlated vehicles and relations such as following or being followed by.

Based on the reported results from the top teams, we hypothesize that in this vehicle-centric retrieval problem, the attributes of the target provided by the input descriptions are the critical cues for determining which object is mentioned. In other words, during the video encoding phase, the model should focus on the appearance of the target vehicle and the vehicle's trajectory relative to the query. From this point of view, in this work, we propose to make the model pay attention to the vehicle characteristics by describing the primary vehicle represented by the corresponding subject sentence instead of complex sentences containing the information about related objects. However, we use most of the techniques from Nguyen *et al.* [12] to perform relationship matching with surrounding vehicles to match all aspects described in input queries, eliminating the weakness of using only the deep learning model.

3. Proposed Method

To effectively apply the representation-based approach, we follow the idea from Siamese network [2], which is a popular method to solve the image-text matching problem. We build a dual flow that forces the model to learn symmetrically between the track query and the mentioned action, and the specific object mentioned in the query and the vehicle.

Our main contribution is to explore the use of input attribute-aware encoding concepts in Siamese architecture inspired by the work of Bai *et al.* [1] to integrate specific knowledge of the targeted attributes into a vehicle representation. Furthermore, we also improve the global and local features proposed in our previous work [12] in AI City Challenge 2021.

3.1. Method Overview

Overall, our proposed method contains four main components: Retrieval Model, (Text) Query Analysis, Tracking and Visual Analysis, and Refinement.

The main retrieval module recommends a list of video track candidates matching a given text query. We present our design for this main component in Section 3.2. Simultaneously, two analysis modules are used to get helpful information, including attributes and relationships, from text queries and video tracks. For the analysis of query sentences, we reuse the SRL module as same as in [12] to extract textual attributes, whereas two convolution neural networks are introduced for the visual analysis to identify local attributes of the tracks. Finally, all information, namely vehicle types, colors, actions, and relationships, are exploited for the refinement stage to re-rank the retrieval result.

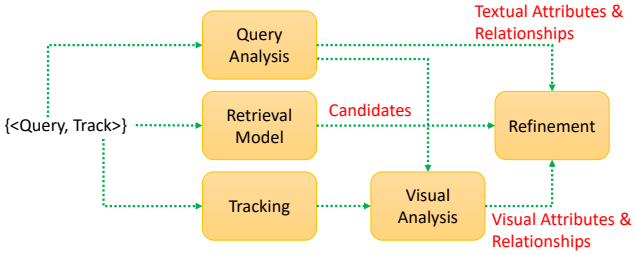

Figure 2. Overview of our proposed training phase.

Figures 2 and 3 show the overview of our method in both training and inference phases, respectively. In the training phase, we use the set of tuples <Query, Video Track> to train the retrieval model and finetune the refinement techniques. We only use the training data provided in Track 2 of AI City Challenge; thus, we train the Visual Analysis module classifiers using the labels extracted from text description of the main vehicle in each track.

In the inference phase, we extract attributes of the main vehicle of interest and its relationship with nearby ones, both from text and visual data of the query and video track, respectively. We feed both the text query and video track data into our trained retrieval model to get the candidate list for refinement. The refinement module utilizes visual and textual attributes and relationships to re-rank the candidate list to generate the final retrieval result.

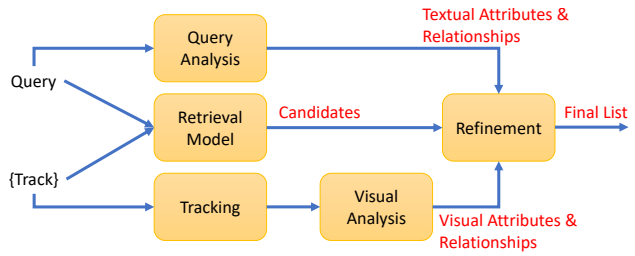

Figure 3. Overview of our proposed inference phase.

3.2. Retrieval Model

As illustrated in Figure 4, our model consists of a query encode branch and a track encode branch. The former describes the event in natural language while the latter processes the event through a list of frames, respectively. To match a query and a video track, we use the global and local concepts at each branch.

- Global context information with the task of summarizing the information of the video.
- Local property representation. The analysis of the object mentioned in the query, representing the object generically, provides important identifying details of the target vehicle that the model needs to focus on to aid in the retrieval process.

Following the joint global and local features, we feed them into the fusion branch to combine the information, which projects to the latent space. Our loss function minimizes the cosine bias at multi-tasks and learn how to simultaneously make local, global, and fusion embedding perform as best as possible. In the inference stage, we only use the output of the fusion branch for the retrieval.

We are motivated by the symmetry architecture and the drawbacks of the original model[1], which we will discuss in more detail in section 3.2.2 and 3.2.1. Specifically, the main idea of our model is that we propose the corresponding local object representation in both branches to learn the similarity between the character of the vehicle and the description in the query to be suitable for searching.

We adopt InfoNCE loss[18] to ensure that the training image matches the described sentence without other noise information. The metric learning approach is associating objects represented in the latent space by training an image encoder and a text encoder parallel to maximize the cosine similarity of the two feature types.

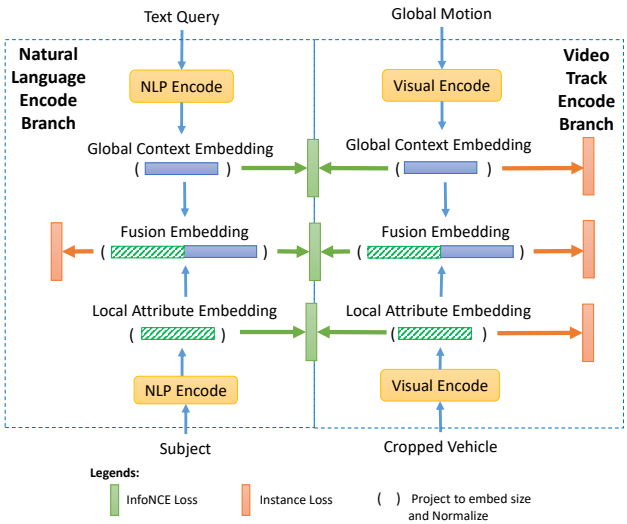

Figure 4. Detailed structure of the retrieval model.

3.2.1 Natural Language Encode Branch

To search for a video track, we are provided with a query in the form of a set of description sentences, each represents a view on a traffic event. To efficiently exploit information from the description sentences of a text query, we propose to have two inputs for the Natural Language Encode Branch, namely Text Query and Subject. Text Query input is actually the concatenation of multiple sentences in a query , while the Subject input is the sequence of subjects extracted from these sentences.

The original model [1] takes a single sentence as the input and represents it in a latent space before matching it with visual tracks. However, it may not always be sufficient to identify a video track, *i.e.* to differentiate a video track from others based on only a single short sentence. This observation motivates us to propose an improved approach to combine multiple views through the concatenation of multiple query texts into a prolonged and information-rich paragraph, which is expected to describe an individual track more comprehensively and therefore helps the deep models to look for more distinctive aspects between tracks. It should be noticed that the "Text Query" input of the Natural Language Encode Branch in Figure 4 is not a single short sentence but a long paragraph with multiple sentences selected from the query.

There are multiple descriptions of the same target vehicle for each query to be retrieved from the video track collection. These can make our solution inconsistent in focusing on the same target object. Hence we propose to add a sub-branch to represent the main object mentioned in these sentences, denoted as the "Subject" in Figure 4. Rather than letting the model, by itself, learn to understand that from full sentences, we help our model to focus more on the subject by taking out the subject's description parts from query sentences, then concatenate and encode them, with the expectation that the information about the vehicle type, color and other identifying features can be learned more efficiently.

3.2.2 Video Track Encode Branch

Our model proposed for AI City Challenge 2021 [12] uses a list of frames and feed it into a sequence model to encode video into latent space embedding. However, we simplify this year version of the video encoder to look at only one image but a summarized representation of the entire video content. To recap, background maps are generated by taking the average of frames from the same video, then the consecutive locations of the tracklet are cropped and pasted on the background to form the Global Motion input in Figure 4.

For local feature, we only take one single cropped version of the target vehicle into consideration, thus enable the model to understand the type, color and appearance of it.

Although Bai *et al.* [1] propose a novel yet efficient method about creation of motion maps as a video summarization technique, it might not be effective in some scenarios, or even worsens the model's performance than other commonly used summarization methods. Specifically, in many cases involving the relationship between vehicles such as "*A followed B*" or "*B followed by A*", these motion maps will undoubtedly fail owing to the fact that it only concentrates on the main tracks and overlooks all other ones. This weakness motivates us to come up with a simple solution in Section 3.3, which utilizes the vital relation information in post-processing stage.

3.2.3 Implementation Details

All of the implementations are based on the Pytorch framework. Moreover, we adopt the Pytorch Lightning framework for our entire training process.

First of all, we follow Alibaba's [1] processing stages to produce the motion maps for each track video. As for the query texts, the SRL module [15] is used to extract the main subject from each query.

At the beginning of the training process, the track's motion map and the instance image are resized to the same 224×224 image size. The instance image, in this case, is the cropped version of the track at a random timestamp, given bounding boxes from the dataset are utilized for this step. Meanwhile, random and shuffle sampling are performed on the corresponding queries and subject queries as an augmentation method. After that, these queries are concatenated together before being encoded by the tokenizer. We inherit the word-piece and positional embedding from the original BERT [3], provided by HuggingFace, to encode the texts into numeric vectors, then longest padding strategy is applied to group vectors into batches.

In our network, two EfficientNet-b3s [17] and a single shared-weight BERT [3] are used as feature extractors for the images and texts, respectively. As can be seen in the Figure 4, motion images and instance images are put through the EfficientNet backbones to get the representation vectors while encoded queries and subject queries are inputted to BERT to get similar outputs. As a result, four embeddings have been generated as we call these global context embeddings and local attribute embeddings . Furthermore, each of these embeddings is projected into a different 768-dim latent space by using linear layers. Subsequently, as mentioned above, InfoNCE [18] is used as a contrastive loss to project these latent spaces into one so that similar tracklets and queries are near to each other. Specifically, the similarity error is computed between the three pairs after the projection and normalization: global *(motion, query)*, local *(instance, subject query)* and fusion *(motion + instance, query + subject query)*, where + indicates the concatenation

(as visualized in Figure 4). At the same time, instance loss, which is just simply cross-entropy loss, is used to enforce the assignment of the motion, instance, and fusion embeddings to one of the suitable track id so that the network is encouraged to find the fine-grained details in images and texts to discriminate between different tracklets.

In terms of the testing phase, we use the fusion of language embeddings to search for the fusion of visual embeddings. These are the same-size 768-dim output vectors from the projected spaces trained by the contrastive loss. In our experiment, we utilize the Faiss library [5] to perform fast distance calculation and searching process. We use cosine similarity as the distance for our retrieval process since the InfoNCE loss[18] aims to optimize this function.

3.3. Post-processing for Refinement

From the retrieval model, we have a list of candidate video tracks as the input for the re-ranking step. To further improve the retrieval performance and to enhance the explainability of final results, we apply several refinement methods to enhance the appearance attributes that were not modeled well in the previous stage: vehicle turning, stopping actions, and the straight-following relationship between the target and surrounding vehicles in the video modality. These predictions are then used to score each track's similarity and the query to re-rank the retrieval result.

For the action detection task, we inherit our proposed methods in previous work [12]. We utilize the box locations in each video to compute the sequence of motion speeds of the target vehicle. The object is considered to perform a stop action if its speed is much lower than the average speed. While the movement trajectory constructed from the list of box centers is utilized to detect the turning action. Sign and magnitude of the algebraic area of the polygon created from these points describes the vehicle moving behaviors (turning left, right or going straight).

We measure the relative positions of a target object A and a neighboring vehicle B using their velocity vectors (V_A, V_B) and the distance vector directed from A to B (D_{AB}). In the same lane, if the angle between V_A and D_{AB} is small enough, vehicle A moves behind B, and B is behind A if the angle is close to 180 degrees. Then, the angle formed by two velocity vectors is utilized to determine if the two vehicles are heading in the same direction or not (the smaller it is, the more confident we have). When A and B are moving in the same direction, we expect that A *follows* B if A is behind B, and A *is followed by* B if B is behind A. Finally, the vehicle type and color predictions of the related objects are compared to textual cues resulting from the query analysis stage to score the relation similarity of each track to the input query.

4. Experiments and Evaluation

In this section, we present the experimental results and evaluation for our proposed method, both in quantitative and qualitative ways. In Section 4.1, we introduce the settings in different modules of our retrieval and refinement methods, then the pre-liminary ablation study on different configuration strategies, and finally the leaderboard in AI City Challenge 2022 of Track 2 on Tracked-Vehicle Retrieval by Natural Language Descriptions. For qualitative evaluation, we present three examples in Section 4.2.

4.1. Experimental Results

In Track 2 of AI City Challenge 2022, we strictly follow the requirements and only use training data provided by the organizers.

To recognize vehicle types and colors, as we do not have extra labeled visual data to train, we extract the color and type of the main vehicle in each video track from its corresponding text descriptions. Using this limited resource, we train classifiers for vehicle's colors, and types using EfficientNet-b0s with six vehicle types and eight groups of colors, similar to our proposed solution in [12].

To detect the actions of a vehicle, we use the algebraic area of the polygon formed by n points in the motion trajectory [12] of a tracked vehicle to classify the motion into *"go straight"*, *"turn right"*, and *"turn left"*.

For the visual encoding branch, we use two separate EfficientNet-b3s [17] to obtain the 1536-d global and local features from the motion maps and cropped images of the vehicle instances. For the language branch, query and subject query paragraphs are encoded by a pretrained BERT-based model [3] and extract the last two layers to construct the final representation for each token as 1536-d features.

Table 1 shows the preliminary ablation study on different configuration strategies to evaluate the contribution of each component in our proposed method. In Version V1, we use a single retrieval model without any refinement, and the MRR score is 0.3137. When we employ the refinement stage after the single retrieval model (Version V2), the MRR score increases to 0.3445. We illustrate several scenarios for the enhancement of retrieval results with post-processing techniques in Section 4.2.

We expect that we can boost the results by using multiple retrieval models. Thus we consider Version V3 and V4, in which we ensemble seven retrieval models trained independently following our proposed scheme. For Version

Table 1. Ablation experiment on different configuration strategies

No.	Configuration	MRR	R@5	R@10
V1	Single + No Refinement	0.3137	0.4674	0.7011
V2	Single + Refinement	0.3445	0.5000	0.6304
V3	Ensemble + No Refinement	0.3483	0.5761	0.7500
V4	**Ensemble + Refinement**	**0.3611**	**0.5489**	**0.6467**

Table 2. Official ranking result (Public Leaderboard) on Track 2

Rank	Team ID	Team Name	Score
1	176	Must Win	0.6606
2	6	Thursday	0.5251
3	4	HCMIU-CVIP	0.4773
4	183	MegVideo	0.4392
5	**91**	**HCMUS**	**0.3611**
6	44	P & L	0.3338
7	10	Terminus-AI	0.3320
8	41	MARS_WHU	0.3205
9	24	BUPT_MCPRL_T2	0.3012
10	56	folklore	0.2832

V3 (without refinement), the MRR score is 0.3483, slightly higher than the results with the single retrieval model with the refinement phase. Finally, we obtain our best result in Version V4 with the MRR score of 0.3611, Recall@5 and Recall@10 of 0.5489 and 0.6467, respectively. In this version, we use multiple retrieval models (7 models), and employ all refinement techniques in the post-processing phase.

Table 2 shows the ranking results in the public leaderboard of Track 2 in AI City Challenge 2022. Our team (ID 91)'s method is ranked in the top 4 on 50% of the test set and the top 5 on the full dataset.

4.2. Case Study

In Figure 5, we illustrate three scenarios of video track retrieval taking advantages of inter-vehicle relationship and action refinement in the post-processing phase. Each row represents one example case with frames extracted in chronological order from a prominent candidate video track corresponding to a text query. In each frame, the main target vehicle is in the green box, following the vehicle in red box, and followed by the vehicle in yellow box.

Example 1: In row 1 of Figure 5, we show a candidate video track corresponding to the query with 3 sentences, "*Red sedan straight*.", "*A red sedan goes straight followed by another car*.", and "*A red sedan keeps straight*." Based on the trajectory of each vehicle, we build the list of vehicles in a flow of traffic. In the first frame, the main car (a red sedan in green box) appears far away and is running toward the car in red box. In the second frame, we can see a car in yellow box appears and follows the main vehicle. We can also see the next following vehicle in the third frame. As we can confirm that the main vehicle (red sedan) is going straight and followed by some vehicle, we select it into the result list for this query.

Example 2: This example aims to demonstrate the usage of vehicle's actions in our retrieval process. The given query text consists of the follow statements: "*A red sedan makes a right turn as it is followed by a gray sedan*.", "*Sedan (4 Door) goes in front of a gray car and stops at the intersection and then continues*.", and "*A maroon sedan stops at the intersection and then turns right followed by another grey vehicle*." In this scenario, we can identify a potential video track matching the query by filtering the main vehicle in red color, then exploiting the sequence of actions of the main vehicle. In the first frame, the main vehicle stops at the intersection before turning right (the second frame) and go straight in the main street (the third frame). We can also confirm the appropriateness of this video track by looking for the gray vehicle following the main one. The gray car is visualized in the yellow box of this example.

Example 3: The third row in Figure 5 shows the scenario described by the following statements: "*A black wagon turns left*.", "*A large black car turns left*.", and "*A black SUV takes a left at the intersection with a white truck in front of it*." This candidate video track is selected by filtering with the main vehicle ("black wagon", "black car", "black SUV") and highlighted in the green bounding box. Using the trajectory matching, our method can link the main vehicle to follow a white truck (visualized in a red box) in turning left at an intersection.

5. Conclusion

In this paper, we introduce a novel framework for traffic event retrieval. In particular, we utilize the global and local features from both the textual and the visual cues. We later combine global and local features and feed them into the fusion branch to effectively combine the information. Given an input text query and video tracks, our retrieval model returns a list of candidates. Then, we refine the results in the post-processing phase by considering the textual and visual attributes. Our proposed framework achieves competitive results through experiments with the rank 5 in Track 2 at AI City Challenge 2022.

To further enhance the accuracy of our proposed solution, we aim to synthesize more data to train better visual classifiers for vehicle types and colors. Currently, as we only use limited samples from the training data, these classifiers still need improvements. In addition, we plan to use the sequence of actions to better select appropriate video track candidates based on the chronological behaviors of the main vehicle of interest.

Acknowledgements

This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA19 and National Science Foundation (NSF) under Grant No. 2025234. Tien-Phat Nguyen was funded by Vingroup JSC and supported by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.ThS.JVN.04.

Query Q_1 = {"Red sedan straight", "A red sedan goes straight followed by another car", "A red sedan keeps straight"}

Query Q_2 = {"A red sedan makes a right turn as it is followed by a gray sedan", "Sedan (4 Door) goes in front of a gray car and stops at the intersection and then continues", "A maroon sedan stops at the intersection and then turns right followed by another grey vehicle"}

Query Q_3 = {"A black wagon turns left", "A large black car turns left", "A black SUV takes a left at the intersection with a white truck in front of it"}

Figure 5. Examples of video track retrieval with inter-vehicle relationship and action refinement. In each frame, the main target vehicle is in the green box, following the vehicle in red box, and followed by the vehicle in yellow box.

References

[1] S. Bai, Z. Zheng, X. Wang, J. Lin, Z. Zhang, C. Zhou, H. Yang, and Y. Yang. Connecting language and vision for natural language-based vehicle retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, pages 4034–4043, 2021.

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[4] Q. Feng, V. Ablavsky, and S. Sclaroff. Cityflow-nl: Tracking and retrieval of vehicles at city scaleby natural language descriptions. *arXiv preprint arXiv:2101.04741*, 2021.

[5] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[7] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M.-C. Chang, S. Lyu, and Z. Gao. The nvidia ai city challenge. In *Prof. SmartWorld*, Santa Clara, CA, USA, 2017.

[8] M. Naphade, M.-C. Chang, A. Sharma, D. C. Anastasiu, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M.-Y. Liu, R. Chellappa, J.-N. Hwang, and S. Lyu. The 2018 nvidia ai city challenge. In *Proc. CVPR Workshops*, pages 53—60, 2018.

[9] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu. The 2019 ai city challenge. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2019.

[10] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M.-C. Chang, X. Yang, Y. Yao, L. Zheng, P. Chakraborty, C. E. Lopez, A. Sharma, Q. Feng, V. Ablavsky, and S. Sclaroff. The 5th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021.

[11] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M.-C. Chang, X. Yang, L. Zheng, A. Sharma, R. Chellappa, and P. Chakraborty. The 4th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, page 2665–2674, June 2020.

[12] T. Nguyen, B. Tran-Le, X. Thai, T. V. Nguyen, M. N. Do, and M. Tran. Traffic video event retrieval via text query using vehicle appearance and motion attributes. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 4165–4172. Computer Vision Foundation / IEEE, 2021.

[13] E.-J. Park, H. Kim, S. Jeong, B. Kang, and Y. Kwon. Keyword-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4227, 2021.

[14] C. Sebastian, R. Imbriaco, P. Meletis, G. Dubbelman, E. Bondarev, et al. Tied: A cycle consistent encoder-decoder model for text-to-image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4138–4146, 2021.

[15] P. Shi and J. Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

[16] Z. Sun, X. Liu, X. Bi, X. Nie, and Y. Yin. Dun: Dual-path temporal matching network for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4061–4067, 2021.

[17] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[18] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.