Traffic Video Event Retrieval via Text Query using Vehicle Appearance and Motion Attributes

Tien-Phat Nguyen*[1,2], Ba-Thinh Tran-Le[1,2], Xuan-Dang Thai[1,2],
Tam V. Nguyen[4], Minh N. Do[5], and Minh-Triet Tran†[1,2,3]

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

Abstract

Traffic event retrieval is one of the important tasks for intelligent traffic system management. To find accurate candidate events in traffic videos corresponding to a specific text query, it is necessary to understand the text query's attributes, represent the visual and motion attributes of vehicles in videos, and measure the similarity between them. Thus we propose a promising method for vehicle event retrieval from a natural-language-based specification. We utilize both appearance and motion attributes of a vehicle and adapt the COOT model to evaluate the semantic relationship between a query and a video track. Experiments with the test dataset of Track 5 in AI City Challenge 2021 show that our method is among the top 6 with a score of 0.1560.

1. Introduction

In smart cities, smart traffic systems use data and technology to help people and goods move faster and more efficiently. In fact, smart traffic benefits from insights derived from data captured by sensors. Traffic event retrieval is one of the important tasks for intelligent traffic system management. In fact, natural language description offers another useful way to specify vehicle track queries. As described in a new track of AI City Challenge 2021, vehicle retrieval will be based on single-camera tracks and corresponding natural language descriptions of the targets. Figure 1 illustrates the traffic event retrieval problem. As seen in the figure, only the traffic events related to the input queries are retrieved. In this problem, the retrieval task's performance will be evaluated using standard metrics of retrieval tasks while considering ambiguity caused by similar vehicle types, colors, and motion types.

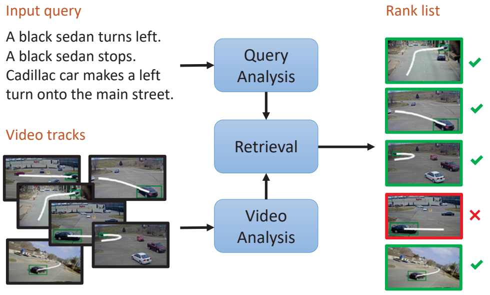


Figure 1. NLP-based traffic event retrieval workflow in the AI City Challenge 2021 Track 5.

In this paper, we propose a novel framework for vehicle event retrieval from a natural-language-based specification. For each text query, we analyse the set of 3 descriptions in natural language format to capture the requirements, including the vehicle type, color, and motion actions. We also group similar semantic terms into clusters to unify the query specification for better consistency, such as *"red"* and *"marron"* should be considered in the same category. For each video track, we create vehicle type and color classifiers, propose a simple yet efficient method to detect actions

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

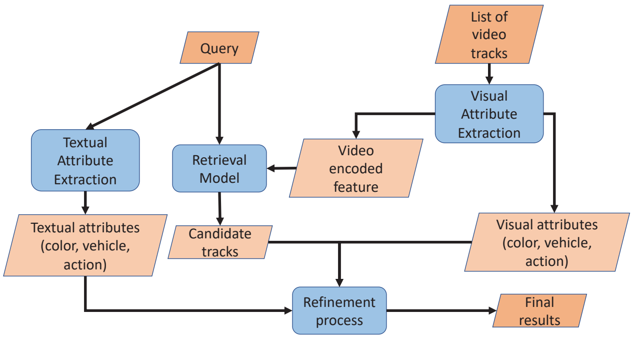

Query
List of video tracks
Visual Attribute Extraction
Textual Attribute Extraction
Retrieval Model
Video encoded feature
Textual attributes (color, vehicle, action)
Candidate tracks
Visual attributes (color, vehicle, action)
Refinement process
Final results

Figure 2. The overview of our proposed method. The dark orange cells denote the raw inputs and final outputs of the system, while the light ones indicate the intermediate results used in the following stages. The other blue blocks named the four main processing modules: textual/visual attribute extraction, retrieval model, and refinement process.

(*“stop“*, *“go straight“*, *“turn left“*, *“turn right“*). Finally, we utilize both appearance and motion attributes of a vehicle and adapt the COOT [6] model to evaluate the semantic relationship between a query and a video track.

We apply our proposed method on the dataset of Track 5 in AI City Challenge 2021. For testing in this track, there are 530 video tracks, each contains the sequence of bounding boxes of a vehicle of interest, and 530 sets of text descriptions. The objective in this track is to return a rank list of video tracks corresponding to each text query. Our method gets promising results in Track 5 of AI City Challenge 2021. In particular, we achieved rank 6 with a score of 0.1560.

The remainder of this paper is organized as follows. Section 2, we briefly review related work. We then present our solutions for traffic event retrieval in Section 3. Experimental results on Track 5 of AI City Challenge 2021 are then reported and discussed in Section 4. Finally, Section 5 draws the conclusion.

2. Related Work

AI City Challenge [10] addressed different problems in a smart traffic management system. Various methods have been proposed to solve many practical problems in smart traffic system, such as vehicle counting [25, 28, 13], velocity estimation[18, 20, 14], behavior analysis, vehicle reidentification [26, 4, 9], anomaly detection [16, 3, 21], etc. In the latest edition, the problem of natural language-based vehicle track retrieval is introduced. There have been numerous related works in literature.

Representation learning has a large range of applications in cross-modal context matching, especially between visual and textual information. It learns to embed the vision and language information into the same latent space for similarity measure. The image-text matching problem is related to various problems such as Image Captioning [27, 24], Image-Text retrieval [23], and Visual Question Answering [1], to name a few. In other scenarios, when one static image cannot provide the full meaning of a concept, which needs the spatio-temporal information from a sequence of images as a video, the video-text matching takes place. Tran et al. [19] obtained deep learned features from C3D. Meanwhile, Nguyen et al. [11, 12] extracted the denser trajectories of interest points in 3D volume and applied action fusion.

The COOT framework [6] concentrates on modeling the long-range video-text relationship, in which the video data is a set of consecutive trimmed video clips displaying a specific event. The text data is a paragraph of multiple sentences describing those events in the respective order. The authors propose a hierarchical architecture, including local and global encoding modules. In each branch, the local modules play the role of encoding the context information from those trimmed segments. The global module encodes the entire paragraph or video to capture the global context of the event. Finally, all those multi-level features is aggregated to produce the final embedding representation for each branch. By efficiently utilizing the transformer architecture integrated with the common attention mechanism [22], the model can capture adequate contents from many parts of the long-range input and perform state-of-the-art on the retrieval task.

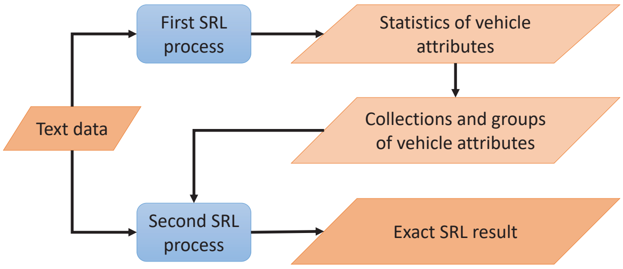

First SRL process
Statistics of vehicle attributes
Text data
Collections and groups of vehicle attributes
Second SRL process
Exact SRL result

Figure 3. The overview of process flow on text branch

3. Proposed Method

3.1. Method Overview

Our proposed method contains four main parts. With a query and list of video tracks, our method first extracts essential features in both branches. In the first step, not only the textual extraction (module 3.2) aims to localize and obtain the potential attributes of each target vehicle in the given descriptions, but the visual extraction (module 3.3) also provides the same attributes and representation feature based on the vehicle visual context and movement trajec-

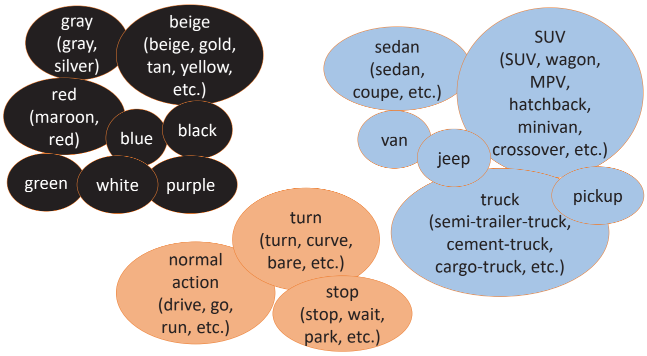

gray (gray, silver)
beige (beige, gold, tan, yellow, etc.)
red (maroon, red)
blue
black
green
white
purple
sedan (sedan, coupe, etc.)
SUV (SUV, wagon, MPV, hatchback, minivan, crossover, etc.)
van
jeep
truck (semi-trailer-truck, cement-truck, cargo-truck, etc.)
pickup
turn (turn, curve, bare, etc.)
normal action (drive, go, run, etc.)
stop (stop, wait, park, etc.)

Figure 4. How we categorize three attribute types. The black, orange, and blue shapes show the group of color, action, and vehicle type, respectively.

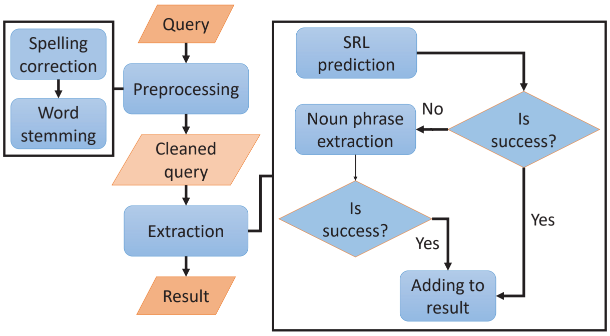


Figure 5. An overview of the heuristic method to extract a query into SRL parts.

Query	A whit semi truck	drives	straight up a hill	behind so many cars
Preprocessed query	a white semi-trailer-truck	drove	straight up a hill	behind so many cars
Extraction result	subject	action	argm_dir	argm_loc
Subject extraction	subject: semi-trailer-truck color: white			

Figure 6. An example of a query transformed and extracted through 2 stages of the heuristics method.

tory. Then we utilize a representation learning-based retrieval model 3.4 to produce a list of candidate tracks for each query. The final result is re-ranked by a refinement block 3.5, using the extracted attributes from previous steps. The overview of our method is explained in Figure 2.

3.2. Text Query Semantic Extraction

On such retrieval problems, the text data usually contains rich information about the objects and their activities. Also, this dataset mainly focuses on traffic and vehicle, narrowing the scope of vocabularies but still providing essential information. Therefore, besides feeding the text query into the next-step model, we employ a method to obtain specific query keywords to construct a special collection of vehicle attributes. This task helps us label each query on some categories, useful for later tasks about classification, detector, or re-ranking.

It should be noticed that most of the queries are structured as follow:

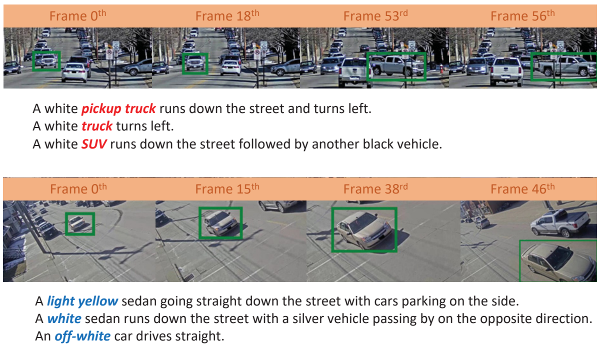
Vehicle + Action + Optional object(s) + Other information.

That is the reason we consider English PropBank Semantic Role Labeling (SRL), via the method proposed in [15], as a possibly efficient means to parse verb and noun phrases. A two-phase flow describing the text-branch process is shown in Figure 3.

First, to analyze the data, we take an SRL extraction on raw data to make statistics on the vehicle's attributes. This process helps us confirm that it is necessary to define potential values about the vehicle types, actions, colors. After filtering the top most frequent and suitable words on each attribute and observing the relation among queries in a track, we create certain groups for three types of the attribute. Figure 4 illustrates the result of categorizing.

The second phase of this flow described the heuristic method we build to extract each query on mentioned categories exactly. There are two main stages in this phase, shown in Figure 5:

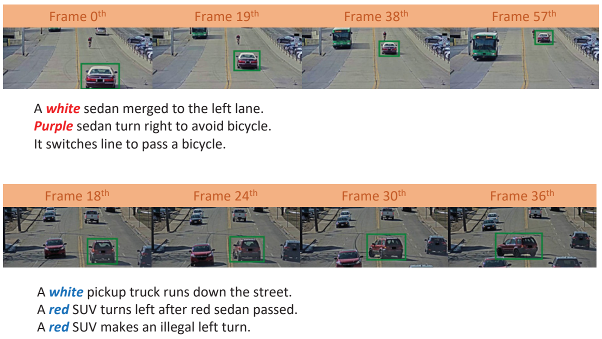
1. **Preprocessing stage:**
   To make the SRL predictor work efficiently, the query needs to be corrected if it has wrong spelling. We use the Levenshtein distance metric to convert a target word to a source word in our defined vocabulary collection of vehicle attributes. To avoid the false convert, we set the condition of correcting a distance equal to 1 between the target word and the source word. We also add a rule to skip certain words which are not misspelled but have a distance of 1. For example, in the preprocessed query provided in Figure 6, we have changed the wrong word *"whit"* to *"white"* but kept the word *"so"* although it has the Levenshtein distance equal to 1 with the word *"go"*. After this step, there is another collection containing rules to convert inconsistent words or phrases into a common one and/or verbs that are easily confused with nouns into past tense. As seen in the mentioned example, we have also made the word *"semi truck"* (in the same group with *"semi-truck"*, *"tractor-trailer"*) become *"semi-trailer-truck"*, and the word *"drives"* become *"drove"*. When finishing this stage, all of the essential terms related to traffic have been clear enough to be extracted.
2. **Extraction stage:**
   For each query, a predictor toolkit is used to obtain parts of the SRL result. If the query is a complete sentence, the function will extract successfully. Otherwise, the query can be a noun phrase, and hence we



A white ***pickup truck*** runs down the street and turns left.
A white ***truck*** turns left.
A white ***SUV*** runs down the street followed by another black vehicle.

A ***light yellow*** sedan going straight down the street with cars parking on the side.
A ***white*** sedan runs down the street with a silver vehicle passing by on the opposite direction.
An ***off-white*** car drives straight.

Figure 7. Examples of ambiguous color/vehicle type labelling affected by different viewpoints or external influences.



A ***white*** sedan merged to the left lane.
Purple sedan turn right to avoid bicycle.
It switches line to pass a bicycle.

A ***white*** pickup truck runs down the street.
A ***red*** SUV turns left after red sedan passed.
A ***red*** SUV makes an illegal left turn.

Figure 8. Examples of multi-color vehicles.

must use another method to handle this. In this situation, we will collect the result if we can find the subject and confirm it as a vehicle type (i.e., *a white truck*, *a typical jeep*), or it will be skipped (i.e., *straight on the main road*, *light short to the right*).

3.3. Video Track Attribute Extraction

3.3.1 Vehicle/Color Classifier

We build two modules, color encoder E_{col} and vehicle encoder E_{veh}, with EfficientNet [17] backbone to learn the visual features representation for each target vehicle from the video tracks. The encoders are trained as a classification task, which takes the target bounding boxes as input and learns to classify them to the most reasonable groups extracted from the previous query processing step 3.2.

However, in practice, the vehicle's visual attributes are not consistent between different viewpoints or could be easily affected by external conditions (sunlight, dust), as pointed in Figure 7. Also, there are some cases the vehicle itself has multi-color, as shown in Figure 8. Therefore, instead of labeling each main subject with a specific class, we gather all textual attributes provided by the three captions as the multi-label ground-truth for each target. We train the classifiers with multi-label approaches, where a sigmoid function replaces the softmax activation in the classification layer. About the training images, for a given track with T frames and T temporal bounding boxes, we sample four boxes: $[B_0, B_{T/3}, B_{2T/3}, B_{T-1}]$ to handle this problem.

3.3.2 Action Detector

From the description of a query, we can obtain multiple actions of a vehicle of interest, such as *"go straight"*, *"stop"*, *"turn right"*, *"turn left"*, and so forth. In this section, we present our method to detect two important action types: stop and turn (left or right). Our method analyzes the trajectory of a vehicle, a sequence of points p_1, p_2, ..., p_n, where p_i is the center of the vehicle's bounding box at the i^{th} frame.

1. **Stop detector:**
   To detect a stop event, we first calculate a sequence of motion speed $v_i = dist(p_{i+k}, p_i)$ where $dist$ is the Euclidean distance and $k = 5$ in our implementation. To remove noise of a temporary slow down in speed, we apply a moving average filter with the window size $delta = 10$ on the sequence of v_i. We consider a stop event occurs when the speed is significant small, comparing to the average speed. We choose a simple yet efficient formula to detect a stop event if the motion speed $v_i < \alpha \times mean\{v_i\}$ where $\alpha = 0.15$.
2. **Turn detector:**
   We use algebraic area to classify the motion direction of a vehicle's trajectory into 3 categories: *"go straight"*, *"turn right"*, and *"turn left"*. Let A be the algebraic area corresponding to the polygon with n vertices p_1, p_2, ..., p_n.

$$A = \frac{\sum_{i=3}^{n} \overrightarrow{p_1 p_{i-1}} \times \overrightarrow{p_1 p_i}}{2}$$

   The vehicle is going in a straight line if $|A|$ is small. If $|A|$ is considerably large, the vehicle is like to turn left or right. The vehicle turns left when $A > 0$, and turns right when $A < 0$. We observe that the error due to possible distortion in cameras is directly proportional to the square value of the distance between the start and end points in the trajectory $|p_1 p_n|^2$. We define $B = A/|p_1 p_n|^2$. As illustrated in Figure 9, B is the (signed) ratio between the area of the red polygon and the blue square. If $|B|$ is greater than a certain threshold ϵ, we conclude that the vehicle turns (left or right), and goes straight otherwise.

3.4. Retrieval Method

We apply the idea from the state-of-the-art COOT model with some modification for the CityFlow-NL benchmark

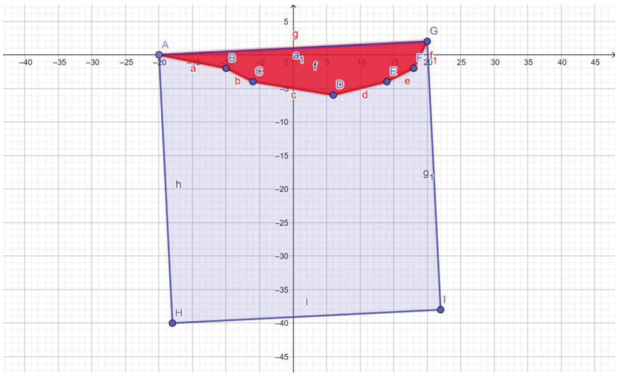


Figure 9. Illustration for using algebraic area to evaluate the direction of a motion trajectory.

[5]. The original COOT aims to tackle the long-range temporal problem in video-text representation learning. It inputs the text descriptions which are paragraphs with multiple sentences, and the long videos which are constructed from many trimmed clips. The model focuses on learning the mapping between local-level context (clip/sentence), global-level context (video/paragraph), and the relationship between them via a hierarchical architecture. In summary, the video and the paragraph are first encoded in words/frames level by pretrained models, then fed forward the embedding modules to produce the corresponding video/paragraph embedding vectors.

In the CityFlow-NL dataset, each query contains three different captions in text and may provide information from various aspects for a specific track. And the video is not segmented as the COOT setting. Therefore, in this work, we only utilize the global-level branch of the COOT framework. Our modification is shown in Figure 10.

In text branch, we can naively model the paragraph input in many different ways by choosing one from the combinations of three different sentences for each query. However, we observe that the three descriptions can contain mutual meaning. For instance, the later caption may have reference to the former one.

- *Yellow car keeps straight.*
- *A yellow coupe keeps straight before a diverging road.*
- *There was a black pickup behind the vehicle.*

Thus, in the encoding step, we aim to concatenate those descriptions as a whole paragraph, which provides the complete information for a specific target track. For a query $q_i = [s_{i=1:3}]$, the preparing process is setup as follow:

1. Splitting each sentence s_i into a list of tokens.
2. Encoding each token to a d_{word}-dimension vector. Consequently, the sentence vector is therefore a list of word vectors.
3. Concatenating the three-sentence vectors as a final representation $\mathbf{v_{query}}$ for the query.

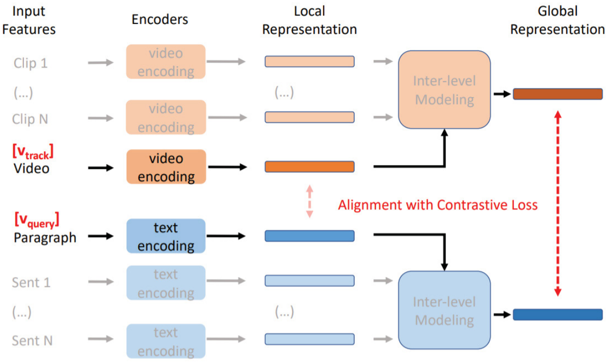


Figure 10. Comparison between the original COOT model and our modification for the CityFlow-NL benchmark. We do not utilize the clip/sentence-level encoding process, which is denoted as the blurred components. The pair of paragraph and video track encoded feature are fed into the corresponding encoder block to obtain the final embedding.

In video branch, the video track also contains the local information of the target vehicle, which is usually the main subject and provides potential visual attributes described in the given query. For that reason, different from the original COOT method, we also include the target's attribute features in the video encoding vector.

In the COOT framework, given a video V with F frames, the video encoding is constructed by concatenating all F frame-level feature d_F-dimension vectors, which are extracted by pretrained backbones (ResNet-152 [8], ResNext101 [7], etc.). For each frame, the feature vectors, enriched by deep neural networks pretrained on large benchmark datasets, contain helpful global information but lack local ones, which could be the target vehicles we need to focus on in the CityFlow-NL setup. From this point of view, we modify the frame-level encoded vectors with the following strategy. Let $\mathbf{v_{frame}}$ denotes the feature vector for each frames in a video track, $\mathbf{v_{frame}} \in \mathbb{R}^{d_f}$. We define this feature as a combination of three sources:

1. **Global context information** ($\mathbf{v_{global}}$). The elements that play the same role as the original COOT framework's extracted visual features, aim to provide general information of a specific video frame.
2. **Attributes representation** ($\mathbf{v_{veh}}$). The compact feature produced by the attribute classifiers (section 3.3.1) provides the important details of the target subject that the model needs to focus on during the retrieval process.
3. **Target vehicle location** ($\mathbf{v_{loc}}$). The relative location and size of the target in a given frame provide the main subject's movement trajectory information.

For the $\mathbf{v_{global}}$, we apply the same approach as in COOT, which is the global feature extracted by the ResNet-152 pretrained on ImageNet [8]. The $\mathbf{v_{veh}}$ is constructed as a concatenation of color and vehicle-type encoded vectors ($\mathbf{v_{veh-col}}$, $\mathbf{v_{veh-type}}$) provided by the corresponding encoders E_{col} and E_{veh}, details in 3.3.1.

$$\mathbf{v_{veh}} = concat(\mathbf{v_{veh-col}}, \mathbf{v_{veh-type}})$$

And the vehicle location component is constructed from the bounding box coordinates at a given frame.

$$\mathbf{v_{loc}} = [x/W, y/H, w/W, h/H]$$

where (x, y, w, h) denotes the box top-left coordinate, width, height. (W, H) are respectively the width, height of the video frame.
In total, the frame encoded feature is modelled as:

$$\mathbf{v_{frame}} = concat(\mathbf{v_{global}}, \mathbf{v_{veh}}, \mathbf{v_{loc}})$$

which contains both global context and target's descriptive informations. And the final encoding for each F-length video track $\mathbf{v_{track}}$ is the combination of F frame-level features, resulting in $F \times d_f$-dimension vector.

The track/query representation feature ($\mathbf{v_{track}}$, $\mathbf{v_{query}}$) is then fed forward through the corresponding encoding blocks to obtain the final embedding vectors. We then train the model with the same strategy as the original COOT method (Figure 10).

3.5. Ranking and Refining

In the retrieval method, we have encoded the visual feature using information from color, vehicle types, and location of the target object. However, we did not exploit the motion features of the tracked objects. Therefore, we build another module to further refine the previously obtained results. In particular, this module focuses on the assessment of similar motion extracted from the query and the video track. The annotation action results of the text are retrieved from the second phase of SRL extraction (section 3.2). Meanwhile, the annotation results of the video action are obtained from the stop and turn detector (section 3.3.2). The retrieval results are refined with the following priorities, i.e., motion, color, and vehicle types. The details of the refinement process are as follows:

1. **Action refinement**. The module takes a query, a list of candidate tracks as input and outputs several tracks placed in multi-level priority groups. The detail of this process is demonstrated in Figure 11.
2. **Vehicle/Color refinement**. In each group, we then sort the tracks with regards to the similarity between their color, vehicle types, and the input query's extracted attributes.

The final result is the combination of these filtered groups from the highest to the lowest priority order.

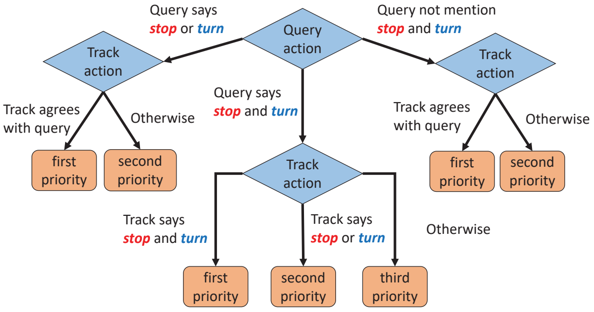


Figure 11. The action refinement module. Query action denotes the motion annotation obtained from the query extractor, while track action represents the motion annotation achieved by the stop and turn detector.

4. Experiments

In this section we discuss our experimental setup, configuration and pretrained models used in each step. We also present our submission results of many different versions as a ablation experiment to evaluate the performance of each processing modules and details about our best submission at the AI City Challenge Track 5 (Table 2).

4.1. Experimental setup

1. **Vehicle/Color classifier**. We adopt EfficientNet-b5 as the backbone and train it with multi-label classification strategy. The number of label categories contains six groups of vehicles and eight groups of color, as shown in Figure 4.
2. **Action detector**. In the implementation, we use algebraic area of the polygon formed by n points in the motion trajectory of a tracked vehicle to classify the motion into *"go straight"*, *"turn right"*, and *"turn left"*.
3. **Video encoding**. For the global context , we use ResNet152 [8] pretrained on ImageNet [8] to extract 2048-d feature vector for each frame. Vehicle and color feature vector is extracted from the classifier backbones with 2048 dimensions also.
4. **Text encoding**. Following the same approach of the COOT model, paragraphs are encoded by a pretrained BERT-based model [2] and extract the last 2 layers to construct the final representation for each token as 1536-d features.
5. **Evaluation metrics**. In addition to MRR, R@5, R@10 metrics as evaluated in the leaderboard, we also consider R@1, Mean Rank and Median Rank score to choose the best version when training.

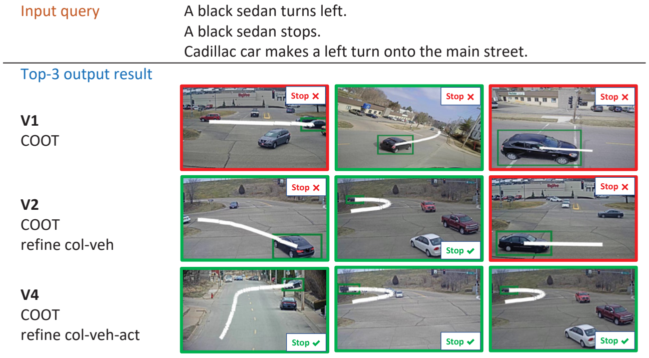


Figure 12. Visualization of an input query and the top-3 result of each version. We draw a white curve on each represented photo to demonstrate the bounding-boxed vehicle's trajectory through the video. The green border line indicates the track video satisfies the turn motion condition of the query, while the red one does the opposite. We also add a stop tag into each photo to denotes whether the target vehicle has stop motion.

Table 1. Ablation experiment on different configuration strategies

No.	Configuration	MRR	R@5	R@10
V1	COOT	0.1010	0.1453	0.1869
V2	COOT + refine col-veh	0.1433	0.2000	0.3226
V3	COOT + refine col-veh w/o loc	0.1368	0.2113	0.3170
V4	**COOT + refine col-veh-act**	**0.1560**	**0.2321**	**0.3302**

4.2. Experimental result

4.2.1 Ablation experiment

In this session, we present several configuration versions submitted to the evaluation system and analyze different modules' effects. Visualization results of these versions are illustrated in Figure 12 and their scoring results are shown in Table 1. In V1, we use the raw COOT model with default configurations with the feature encoding strategy discussed in 3.4 and without any re-ranking methods. Then we refine the V1 result on color and vehicle type attribute (mention in 3.5) to obtain V2. In V3, we also try to eliminate the location information from the frame-level encoding process to evaluate its effect on the final result. Based on the previous experiments' performance, we choose the best retrieval model with the highest validation score and apply all the refinement methods to produce the final version, V4.

4.2.2 Result analysis

According to the scoring results (Table 1 and Figure 12), we observe that the raw COOT model can pay attention to the right vehicle type and color as the target object in the query during the retrieval process. As shown on the first line of the visualization results in Figure 12, the V1 version has returned tracks matching correctly those attributes.

However, the retrieval model still lacks local features and does not focus on the target vehicle. Thus the color/vehicle-type refinement module (version V2) helps boost those related tracks to higher ranks and obtain a higher scoring result in V2. In version V3, without the location feature, the model performance reduces significantly. Thus we conclude that the motivation information plays an essential role in the retrieval process. In addition, the visualization of V1 and V2 in Figure 12 indicates the current retrieval model's lack of action-understanding performance. The special action types as "stop" and "turn" of each track are not considered to gain optimal results. Thus, we apply both three filtering methods as an attribute-based re-ranking strategy (discussed in 3.5) to refine retrieval results and re-produce the prediction lists (version V4). The visualization result of V4 in Figure 12 shows the effect of our refinement methods, where the top tracks have the same color, vehicle type, and action described in the query. The V4 version is our best result for Track 5 of AI City Challenge 2021, which gains a 0.156 MRR score, 0.2321, and 0.3302 for R@5, R@10 respectively.

Table 2. Ranking result on Track 5

Rank	Team ID	Team Name	Score
1	132	Alibaba-UTS	0.1869
2	17	TimeLab	0.1613
3	36	SBUK	0.1594
4	20	SNLP	0.1571
5	147	HUST	0.1564
6	**13**	**HCMUS**	**0.1560**
7	53	VCA	0.1548
8	71	aiem2021	0.1364
9	87	Enablers	0.1314
10	6	Modulabs	0.1195
11	51	AIPERT	0.1078
12	11	CE_UIT	0.0852
13	30	Fiberhome-YYDS	0.0850
14	82	CMU INF	0.0184
15	146	UFL_nlp	0.0172

5. Conclusion

In this paper, we propose a novel framework for traffic event retrieval. Through experiments, our proposed framework obtains competitive results with the rank 6 in Track 5 of in AI City Challenge. In our method, we utilize both appearance and motion information of a vehicle to match the appropriateness of a video track with a given query in text. We propose a modified solution based on COOT model for CityFlow-NL benchmark. Currently, we continue to enhance our solution to further exploit spatial and temporal relationship for better understanding video tracks.

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.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] K. Doshi and Y. Yilmaz. Fast unsupervised anomaly detection in traffic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 624–625, 2020.

[4] V. Eckstein, A. Schumann, and A. Specker. Large scale vehicle re-identification by knowledge transfer from simulated data and temporal attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 616–617, 2020.

[5] 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.

[6] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox. Coot: Cooperative hierarchical transformer for video-text representation learning. *arXiv preprint arXiv:2011.00597*, 2020.

[7] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] S. He, H. Luo, W. Chen, M. Zhang, Y. Zhang, F. Wang, H. Li, and W. Jiang. Multi-domain learning and identity mining for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 582–583, 2020.

[10] 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 *IEEE CVPR Workshops*, page 2665–2674, June 2020.

[11] T. V. Nguyen, J. Feng, and K. Nguyen. Denser trajectories of anchor points for action recognition. In *Proceedings of International Conference on Ubiquitous Information Management and Communication*, pages 1:1–1:8. ACM, 2018.

[12] T. V. Nguyen and B. Mirza. Dual-layer kernel extreme learning machine for action recognition. *Neurocomputing*, 260:123–130, 2017.

[13] A. Ospina and F. Torres. Countor: Count without bells and whistles. In *Proceedings of CVPR Workshops*, pages 600–601, 2020.

[14] H. Shi. Geometry-aware traffic flow analysis by detection and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 116–120, 2018.

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

[16] L. Shine et al. Fractional data distillation model for anomaly detection in traffic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 606–607, 2020.

[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] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *CVPR Workshops*, pages 108–115, 2018.

[19] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision*, pages 4489–4497, 2015.

[20] M.-T. Tran, T. Dinh-Duy, T.-D. Truong, V. Ton-That, T.-N. Do, Q.-A. Luong, T.-A. Nguyen, V.-T. Nguyen, and M. N. Do. Traffic flow analysis with multiple adaptive vehicle detectors and velocity estimation with landmark-based scanlines. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 100–107, 2018.

[21] M.-T. Tran, T. V. Nguyen, T.-H. Hoang, T.-N. Le, K.-T. Nguyen, D.-T. Dinh, T.-A. Nguyen, H.-D. Nguyen, X.-N. Hoang, T.-T. Nguyen, et al. itask-intelligent traffic analysis software kit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 612–613, 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[23] N. Vo, L. Jiang, C. Sun, K. Murphy, L. Li, L. Fei-Fei, and J. Hays. Composing text and image for image retrieval - an empirical odyssey. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6439–6448, 2019.

[24] V. Vo-Ho, Q. Luong, D. Nguyen, M. Tran, and M. Tran. Personal diary generation from wearable cameras with concept augmented image captioning and wide trail strategy. In *Proceedings of International Symposium on Information and Communication Technology*, pages 367–374. ACM, 2018.

[25] Z. Wang, B. Bai, Y. Xie, T. Xing, B. Zhong, Q. Zhou, Y. Meng, B. Xu, Z. Song, P. Xu, et al. Robust and fast vehicle turn-counts at intersections via an integrated solution from detection, tracking and trajectory modeling. In *IEEE/CVF CVPR Workshops*, pages 610–611, 2020.

[26] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, and S.-Y. Chien. Vehicle re-identification with the space-time prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 121–128, 2018.

[27] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In F. R. Bach and D. M. Blei, editors, *Proceedings of International Conference on Machine Learning*, volume 37, pages 2048–2057, 2015.

[28] L. Yu, Q. Feng, Y. Qian, W. Liu, and A. G. Hauptmann. Zero-virus: Zero-shot vehicle route understanding system for intelligent transportation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 594–595, 2020.