Keyword-based Vehicle Retrieval

Eun-Ju Park Hoyoung Kim Seonghwan Jeong
Byungkon Kang YoungMin Kwon
Dept. of Computer Science, The State University of New York, Korea
{eun-ju.park, hoyoung.kim.1, seonghwan.jeong}@stonybrook.edu
{byungkon.kang, youngmin.kwon}@sunykorea.ac.kr

Abstract

Natural language-based vehicle retrieval system makes controlling a city-scale traffic system easy to maintain and adaptable to changing requirements. It provides a convenient means in managing traffic flows or detecting accidents related to a specific vehicle. Such a system is different from most query-based video retrieval systems because the language for traffic situations and visible objects in traffic video streams are limited. Existing techniques for language-based general video retrieval problems measure the similarity between language representations and video representations. Our system focuses on several features that can distinguish vehicles from others. Particularly, our proposed vehicle retrieval system defines a set of features that can differentiate a vehicle from others and calculates the similarity between queries and video frames based on the features. The proposed technique places our approach in the third place in the 2021 AI City Challenge.

1. Introduction

One of the main functions of smart cities is to manage traffic flows using their widely available resources. To make use of the advanced technologies in smart vehicles, it is desirable to observe and analyze traffic flows so that the collected information is reported to a control center in real time. Vehicle retrieval is a fundamental task in building such a system. To facilitate the task of identifying vehicles in a city-scale traffic system, natural language itself has been foreseen as a desirable form of input to a system. It is not difficult to imagine a various range of applications once such a query based vehicle retrieval system is developed. Some examples are a traffic control system adjusting itself to optimize traffic flows or a system that can trace a vehicle related to crime.

There have been many fruitful results in natural language-based video retrieval techniques [1, 5, 7, 9, 16]. However, the problem of natural language-based *vehicle retrieval* has unique challenges, distinguished from the traditional natural language-based *video retrieval* problem. The challenges are due to the limited context for traffic flow. For example, there are meaningless sound and few kinds of objects in a vehicle video stream. In addition there are a limited number of words used in natural language. As a result, to specify a vehicle track by queries, it is required to analyze video frames along with other information like the visual-location.

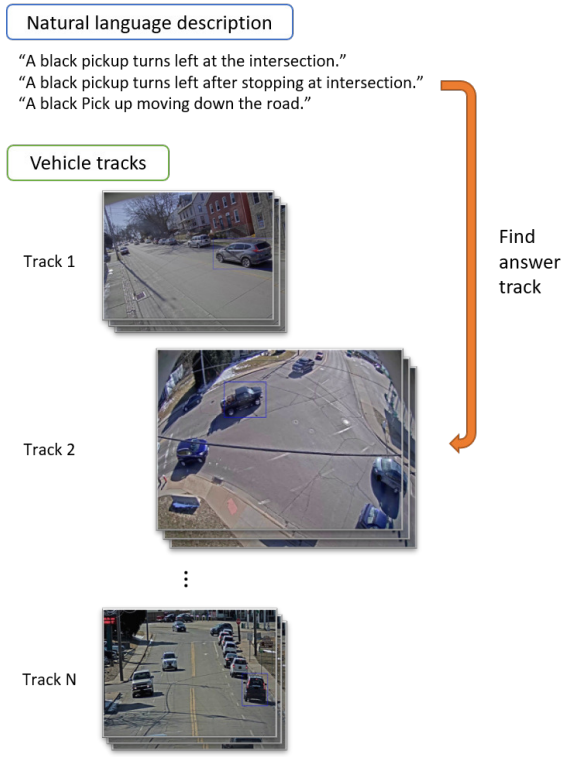

Figure 1. Problem definition

In this paper, we present our solution for the vehicle retrieval problem using queries with the dataset of the 2021 AI City Challenge. In the track 5 dataset of the challenge, each target vehicle has its frame information, a bounding box in the corresponding frame, and three different anno-

tations describing the vehicle in the frames. Given queries of a vehicle, the retrieval model ranks the vehicle tracks as shown in Figure 1. In tackling the challenge, the following are the main contributions of this paper:

- To match a vehicle target and its three different corresponding queries, we proposed a deep learning network that employs a loss function with multiple ground truth.
- To extract vehicle movement features such as making a left/right turn or stopping at an intersection, we devised a movement analysis technique based on obtaining the position of a target vehicle and inspecting its velocity vector.
- The framework can successfully detect preceding and following vehicles by checking the objects on the trajectory.
- The variable weighting technique is a simple, yet powerful method as is demonstrated in the challenge.

2. Related Work

Diverse research on video-language retrieval has appeared rapidly as the neural networks have been developed actively. The latest studies on the video retrieval with linguistic description are reviewed briefly in Section 2.1.

The AI City Challenge [17, 18, 19, 20] has promoted numerous studies related to deep neural networks for intelligent transportation system since 2017. Our approach for vehicle retrieval is highly relevant to several algorithms and techniques from these studies. Section 2.2 introduces the previous works.

2.1. Language-Video Retrieval

A Video includes various features such as motion, audio, appearance, face, speech, etc. Such features could be extracted by well-known, pre-trained models. Liu et al. [16] retrieves videos based on the similarity between text representation and the feature representations from the various pre-trained models. MMT (Multi-modal Transformer) [9] employs transformers to understand contextual information. Two different BERT architectures [5] embed the captions and the sequences of the features individually. MDMMT model (Multidomain Multimodal Transformer) [7], a successor of the MMT, improves the performance with deeper BERT and promising motion extraction from videos.

Bain et al. [1] proposes an encoder for both images and videos with ViT [6] and Timesformers [2]. The joint learning with both images and videos enables video-language retrieval with a 'frozen' snapshot of a video. ClipBERT [14] sparsely samples a few clips in a video. This method brings the effect of data augmentation in training step. Due to the sparsity, it is possible to apply cross-modal models on each sample and results in well-performed text-to-video retrieval.

2.2. City-Scale Vehicle Tracking

The AI City Challenge has presented various city-scale computer-vision problems for intelligent transportation system. To solve these problems, one of the fundamental approaches is to track vehicles. Several techniques and algorithms for vehicle tracking in common use are shortly covered in this Section.

Feature extraction Prominent convolutional neural networks are commonly used to extract features for vehicle detection, classification or re-identification. VGGNet [23] shows deep networks made of simple convolution layers perform well for feature extraction. The improvement, however, might not be significant as the depth of the networks because of gradient vanishing/exploding and degradation. ResNet [12] has introduced the residual architecture, solved the problems and made it possible to build efficient, deeper models. DenseNet [13] adds novel concept, dense connectivity. Still, the ResNet has an issue of information vanishing over passing through many layers. The DenseNet directly connects all layers to all subsequent layers. Due to the densely connected structure, the number of parameters decreases and back-propagation is more effective.

Object detection Existing object detection algorithms provide object class and information of bounding box with high accuracy. YOLO [22] is one of popular real-time object detection algorithms. It divides an image into multiple grids. Each grid generates bounding boxes with confidence score based on the intersection over union and class-specific confidence. SSD [15] takes advantage of VGG-16 networks to extract feature map. To detect different size of objects with a single-shot, it adopts multi-scale feature maps. Mask R-CNN [11] has been proposed especially for instant segmentation. It consists of region proposal networks, which reduces inference time, and feature pyramid networks, which makes it perform better likewise ResNet.

Tracking algorithms For analysis of vehicles movement, each object should be accurately tracked above frames in a video. SORT [3] algorithm tracks multiple objects by using Kalman filter and Hungarian method in real-time. Deep SORT [31] is the advanced version of SORT. Due to additional appearance descriptors comprising convolutional networks and Mahalanobis distance, it enhances the tracking accuracy. Another tracking algorithm, the tracklet-clustering based tracker [27], precisely tracks multiple vehicles. Tracklet is a chuck of the moving object's trace. To

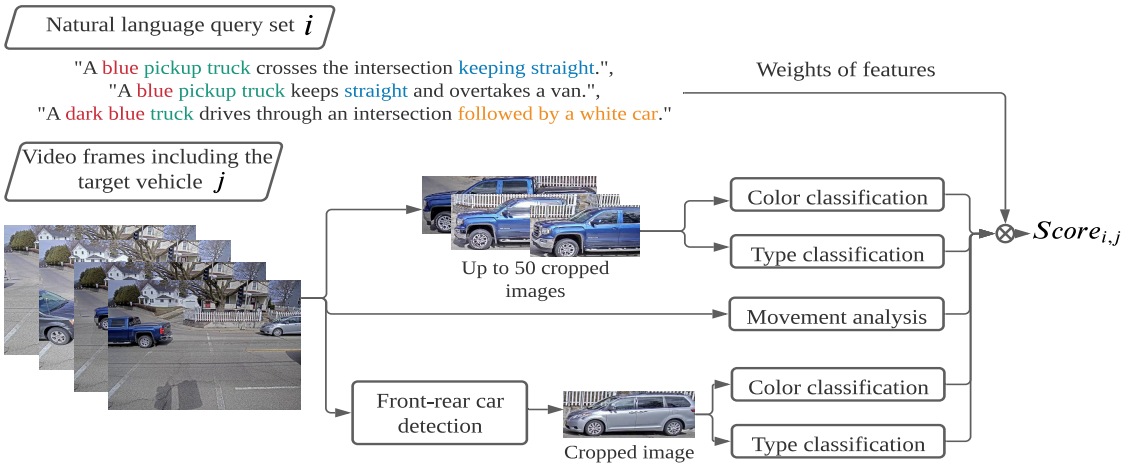

Figure 2. The overall design of our text-to-vehicle retrieval system. The system extracts features and decides the weights on the features.

associate the tracklets into longer trajectories, the clustering method is applied. The next version, TrackletNet tracker [30], applies epipolar geometry [10] and convolutional networks to associate different tracklets. This approach results in remarkable performance for re-identification and even can handle occlusion. The other tracking model, MOANA [24], has adaptive structure for online learning. The appearance of the object might not be useful in long-term tracking since the appearance changes along the sequences. Due to the adaptive model, it demonstrates robust tracking results in the case that similar objects are closely located.

3. Methodology

Figure 2 shows an overall design of our text-to-vehicle retrieval system: it extracts various features from video frames and decides weight values assigned to the features based on natural language descriptions. The details of the proposed system are given as follows.

3.1. Vehicle Features

In natural language-based vehicle retrieval system, the queries consist of words that people use to describe vehicles. Vehicles have various features such as color, size, and design, but people identify cars using relatively simple descriptions; e.g. *a red car*, *a midsize SUV*, or *a black sedan*. In addition appearances, vehicles run at various speed, (e.g. 20mph or 50mph), but words used in queries are far from being specific; e.g. *at a fast speed*, *slowly*, or *stops at the intersection*. Using such characteristics of the queries, we decompose a vehicle into three categories: color, type and motion.

On the other hand, the queries are made up of descriptions of the subject car and its surroundings, i.e. *A white hatchback goes straight at the street followed by another white vehicle* or *A blue hatchback is one of five vehicles driving straight on a small highway*. Since we are focusing on the vehicle itself, we combine the information of the subject car with that of the vehicles around it; the color, the type, the movement of the subject car and the color, the type of the front or the rear car of that. We use some words in queries when they are matched pre-defined labels for vehicle features. For example, when a query is *A gray sedan turns right at the intersection followed by a green bus*, we extract *gray, sedan, turns right* for a subject car and *green, bus* for the rear car of it. The extracted words as ground truth are utilized to find the corresponding vehicle using its image frames and location.

As is shown in Figure 3, the proposed system evaluates the similarity between queries and video frames with respect to the defined vehicle features. In the remainder of this section, we illustrate (1) how we classify color and type of vehicles, (2) how we calculate movement of vehicles, (3) how we detect front and rear cars of a subject car, and (4) how we answer to a query in details.

3.2. Vehicle Classification

We classify vehicles in video frames into two kinds of classes: color and type. The methods for classifying the vehicle color and type are the exactly same, so we only describe how to classify a vehicle color below.

To train the color classifier, we employ transfer learning. Cropped images extracted from frames with given bounding box are input to a ResNet50 pre-trained on ImageNet dataset [4]. We replace the original softmax layer with the one tailored to the color labels in our problem.

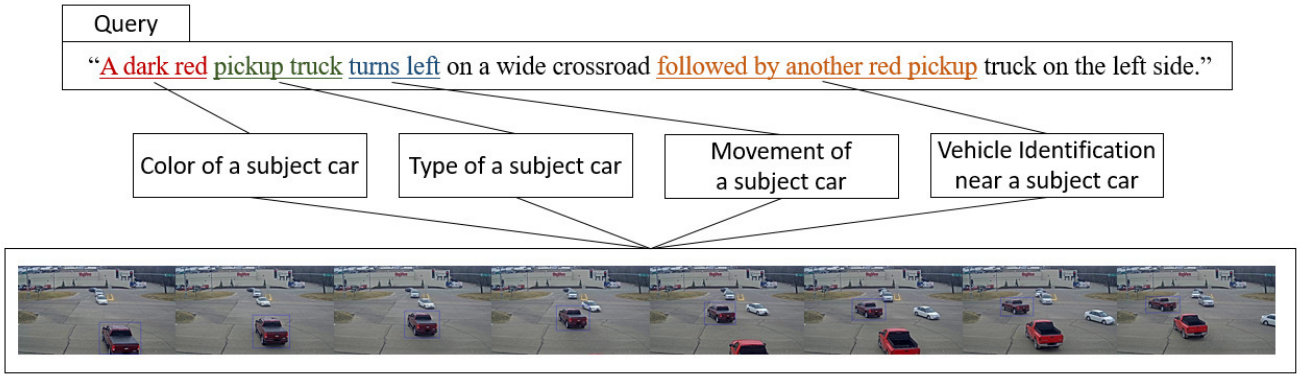

Figure 3. An example of matching a query to video frames of a vehicle. For specifying a vehicle, we use four criteria. The order of the video frames is left to right.

Loss Function Each vehicle of the training/test dataset has three different annotations, so a vehicle can have a maximum of three different ground truth for colors. In order to deal with the issue of multiple ground truth labels, we employ the loss function similar to that proposed by Yun et al. [32]. When C is the size of color labels, let x_c be the softmax output, and y_i the color label of a query as ground truth. S is the number of the unique color labels mentioned in the queries, and $P(y_i)$ is the relative frequency of each color in the queries. Using $f_{ce}()$ as cross entropy function, we optimize Equation (1).

$$Loss = \sum_{i=1}^{S} P(y_i) f_{ce}(x_c, y_i). \quad (1)$$

3.3. Movement Analysis

The movement of a vehicle is one of the key features to characterize the vehicle. For example, fractions of the query dataset like *SUV turns right*, *sedan takes a left*, or *car waits at the intersection* describe whether the car turns right or left, whether it slows down or stops, etc. We analyzed the movement of vehicles and ensures that it matches the query sentence describing the movement. We computed the GPS coordinates from the locations in the image and decided whether the vehicle was turning left, turning right or stopping.

Figure 4 shows an example of a right-turn track and its trajectory. The top figure shows a car making a right-turn. The blue bounding boxes are the beginning, a middle, and the end positions of the track, and the dotted red line shows the movement between the points. When the track positions are converted to GPS values, the graph of the bottom figure can be obtained. It shows the trajectory of the right-turn in the GPS coordinate.

Because the location of a car in an image depends on the location and the direction of the camera that captured the image, analyzing the image is not a straightforward task.

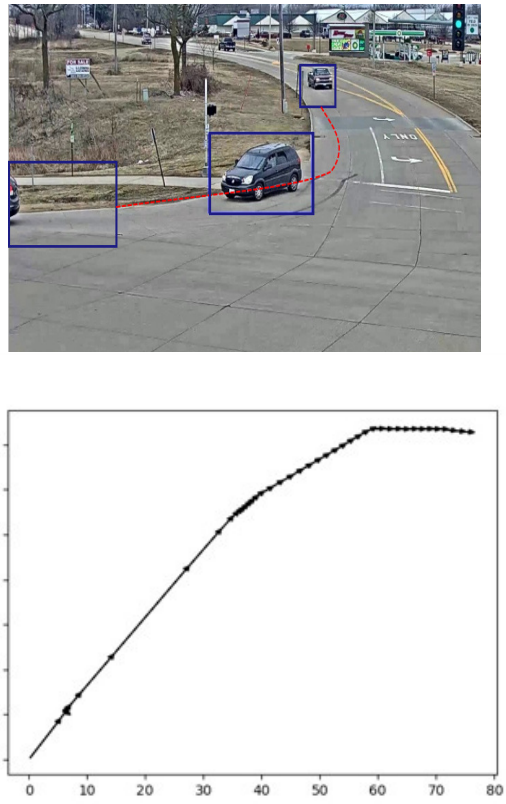

Figure 4. An example for turning right vehicle and its trajectory.

For instance, compared to the case when a car is captured near a camera, a car captured afar from the camera makes a smaller change in the images even though it moves at the same speed or even when it made a turn. To cope with the difficulty, we utilized the camera calibration data provided in track 3 in converting the positions in an image to the GPS coordinates [26, 28]. In addition, to reduce the effect of noise, we applied a Kalman filter in estimating the vehicle's position and velocity vectors from the GPS positions.

When a car made a turn, the velocity vectors before the turn ($\vec{V}_{in}$) and after the turn ($\vec{V}_{out}$) will show about 90° difference. We computed the angle change θ using the following formula derived from the cross-product of the two vectors.

$$\theta = \arcsin\left(\frac{\left\|\vec{V}_{in} \times \vec{V}_{out}\right\|}{\left\|\vec{V}_{in}\right\| \cdot \left\|\vec{V}_{out}\right\|} \cdot n\right), n \in \{1, -1\} \quad (2)$$

Here, n depends on the cross product of two vectors $\vec{V}_{in}$ and $\vec{V}_{out}$. Because $\vec{V}_{in}$ and $\vec{V}_{out}$ are two-dimensional vector, the cross product of two vectors has only z coordinate value. If the value of the z coordinate is positive, it means left-turn by the right-hand rule, n should be 1. In the opposite case, n is -1. We decide that the vehicle made a turn if the magnitude of angle θ is larger than a threshold value. Also, by using n, we decide whether the turn was a right-turn or a left-turn.

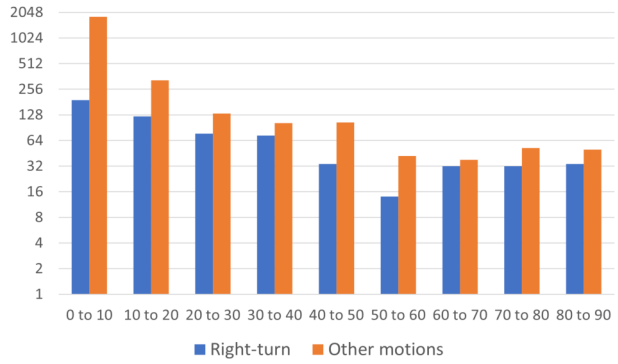

Figure 5. Angle histogram of turning right and other cases

To find a good threshold value for θ in Equation (2), we used statistics obtained from the training data. Figure 5 shows a histogram of the angle changes of vehicles making a right-turn. In the figure, blue bars show the cases when the query sentence includes a right-turn and the orange bars show the cases when the query does not have a right-turn. Some false positives are unavoidable because there is no clear separation in the angle changes when there is a right-turn and when there is not. To reduce the false positives, we set the threshold value to 20°: when the angle is in between 0° to 10°, the cases that did not turn right were about 8.51 times larger than the cases that did and when the angle is in the range of 10° to 20°, 1.65 times more cases did not made the turn. Hence, including these ranges will only hamper the accuracy. Similarly, decision on whether a vehicle is stopped or not is based on a threshold for the speed. We set the threshold value to $3.6Km/h$, the average walking speed of a human.

3.4. Front-Rear Car Detection

Preceding and following vehicles are strong evidences for retrieving scenes including the target car. A few sentences include description on the front or rear cars, such as *after a black vehicle*, *in front of a pickup truck* or *followed by another yellow vehicle*. Therefore, we devised a system for detecting the front-rear vehicles in Figure 6.

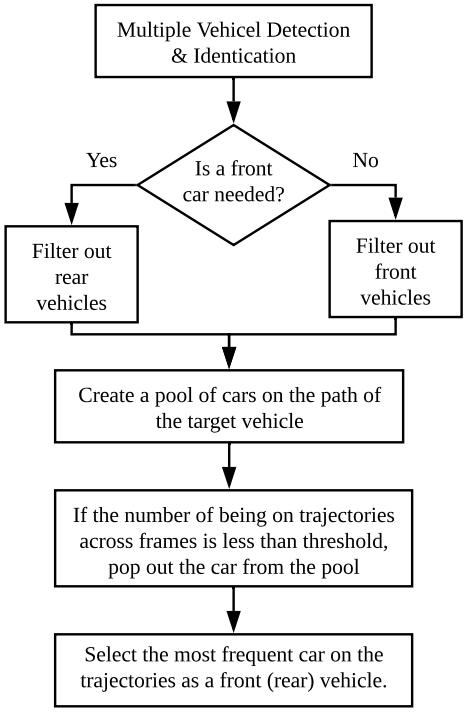

Figure 6. A flow chart for front-rear vehicle detection.

In order to detect the front and rear vehicles in the frames, detecting multiple cars and getting bounding boxes of them are necessary as the first step. Mask R-CNN [11] is applied to recognize cars, after which unique IDs are assigned to each car [27]. The position of each vehicle on Cartesian coordinate systems is induced by the method given in Section 3.3.

When rear cars are wanted, the cars on the opposite side are excluded based on the direction of the target vehicle as shown in Figure 7. The rear cars are filtered out in the same way when we want the front cars.

In the next step, we create a pool of cars on trajectories of the target. When the distance between each vehicle and a tracklet of the target car is less than one meter, it is considered as the candidate on the target's trajectories. To get the distance, two closest footprints of the target from each candidate should be figured out. With a triangle comprising the footprints and the coordinate of the candidate, we can estimate the distance by using Heron's formula [21].

In the pool, the right front or back car is decided based on the number of being on the trajectories across frames. If

method	weighting strat.	angular thld. [°]	turning wt.	stopping wt.	front-rear wt.	MRR	Recall@5	Recall@10
CL	-	-	0	0	0	0.1215	0.1585	0.2906
CL+MVT+FR	Fixed	60	0.5	0.3	0.3	0.1479	0.2132	0.3264
CL+MVT+FR	Fixed	20	0.5	0.3	0.3	0.1583	0.2340	**0.3491**
CL+MVT	Fixed	20	0.5	0.3	0	0.1533	0.2283	0.3453
CL+MVT+FR	Fixed	20	0.8	0.6	0.3	0.1566	0.2283	0.3358
CL+MVT+FR	Variable	20	-	-	-	**0.1594**	**0.2396**	0.3472

Table 1. The performances with different settings.

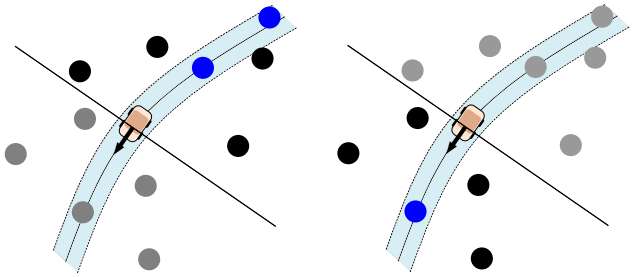

Figure 7. Unnecessary cars for finding the front and rear cars are filtered out. If the rear cars are needed for vehicle retrieval, the front cars are excluded based on the diving line decided with the vector of the target vehicle (left). The rear cars are excepted if the front objects are required (right). Only cars on trajectories of the target vehicle are chosen (blue circles).

a car appears on the trajectories over certain threshold and mostly, the algorithm regards the car as right front or rear car.

3.5. Weighting Strategy

In order to rank association between a vehicle and three language queries among others, a sophisticated scoring system is required. The final rank depends on the way weights are assigned to the four vehicle features, because each feature has different error rate and correlation with the target vehicle. The simplest and the most effective way is to use variable weights based on the number of feature-related expression appearance in the queries. The variable weight w_i of the feature i is defined as:

$$w_i = \frac{n_i}{\sum_{j=1}^{N} n_j}, \tag{3}$$

where n_i is the number of descriptions relevant to the feature i in the three sentences, and N is the number of feature types. The association score s between the language query set and the vehicle is presented in the Equation (4).

$$s = p_{color} + p_{type} + \sum_{i=1}^{N} w_i p_i, \tag{4}$$

where p_x denotes the predicted probability of feature x. In case of our system, we use a fixed weight of 1 for the color and type features, which makes $N = 2$ despite the four features we use.

4. Experiments

In this section, we describe the dataset, comparison of different approaches, and the results of the proposed system.

4.1. Dataset

We use CityFlowV2 dataset [25] and natural language description sets [8] provided in the 2021 AI City Challenge. The datasets comprise of real-world traffic scenes, three linguistic descriptions per vehicle, and metadata including homography matrix for camera calibration, the results of baseline vehicle tracking, the segmentation results from Mask R-CNN [11]. The proposed system is evaluated with the standard retrieval metrics: Mean Reciprocal Rank (MRR) [29], Recall@5 and Recall@10.

4.2. Evaluation Results

4.2.1 Comparison of different settings

The performance highly gets affected by the angular threshold setting and weighting strategy. With the settings, we could discuss which features are more crucial.

As can be seen in Table 1, the method using only color and vehicle type classification (CL) shows relatively lower performance than others. When the threshold of the turning angle is 20°, all results are higher than the one of 60° threshold. The performance has improved by applying the front-rear vehicle detection compared to the one without the feature (CL+MVT). With the higher weights on movement factors, the performance has rather decreased. The suggested variable-weighting strategy has achieved the best result among the tested settings.

As for the main competition, we ranked in the third place with a MRR score of 0.1594 (Table 2). It can be interpreted that our framework predicted the correct vehicle for the queries in the top 6.2 rank on average.

Rank	Team ID	Team Name	MRR
1	132	Alibaba-UTS	0.1869
2	17	TimeLab	0.1613
3	**36**	**SBUK**	**0.1594**
4	20	SNLP	0.1571

Table 2. The ranking in the 2021 AI City Challenge.

5. Conclusion

In this paper, we demonstrated that natural language-based vehicle retrieval can be tackled by leveraging vehicle features extracted from the queries and images. Although end-to-end training is the preferred trend nowadays, we showed that a system composed of individual components can score on par with sophisticated deep learning models.

Furthermore, our result is achieved by using rudimentary feature extractors (e.g., simple text matching). This indicates that our method has potential to improve more by individually fine-tuning the components. For example, we could perform semantic parsing to extract relevant information from the queries more accurately.

6. Acknowlegment

This work was supported by NRF of MSIT, Korea (2019R1F1A1058770). The authors contributed equally to this work.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *arXiv preprint arXiv:2104.00650*, 2021. 1, 2

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 2

[4] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[7] Maksim Dzabraev, Maksim Kalashnikov, Stepan Komkov, and Aleksandr Petiushko. MDMMT: Multidomain multimodal transformer for video retrieval. *arXiv preprint arXiv:2103.10699*, 2021. 1, 2

[8] Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. CityFlow-NL: Tracking and retrieval of vehicles at city scale by natural language descriptions. *arXiv preprint arXiv:2101.04741*, 2021. 6

[9] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal Transformer for Video Retrieval. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 5, 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[14] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. *arXiv preprint arXiv:2102.06183*, 2021. 2

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[16] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 1, 2

[17] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M. Chang, S. Lyu, and Z. Gao. The NVIDIA AI City Challenge. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–6, 2017. 2

[18] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C. Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, Jenq-Neng Hwang, and Siwei Lyu. The 2018 NVIDIA AI City Challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[19] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng

Hwang, and Siwei Lyu. The 2019 AI City Challenge. In *Proc. CVPR Workshops*, pages 452–460, Long Beach, CA, USA, 2019. 2

[20] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th AI City Challenge. In *Proc. CVPR Workshops*, Virtual, 2020. 2

[21] Münevve Ozcan and Rüstem Kaya. Area of a triangle in terms of the taxicab distance. *Missouri Journal of Mathematical Sciences*, 15, 10 2003. 5

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[24] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019. 3

[25] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proc. CVPR*, pages 8797–8806, Long Beach, CA, USA, 2019. 6

[26] Zheng Tang, Gaoang Wang, Tao Liu, Young-Gun Lee, Adwin Jahn, Xu Liu, Xiaodong He, and Jenq-Neng Hwang. Multiple-kernel based vehicle tracking using 3D deformable model and camera self-calibration. arXiv:1708.06831, 2017. 4

[27] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 108–115, 2018. 2, 5

[28] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In *Proc. CVPR Workshops*, pages 108–115, 2018. 4

[29] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82. Citeseer, 1999. 6

[30] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019. 3

[31] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2

[32] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision (ICCV)*, 2019. 4