A Unified Multi-modal Structure for Retrieving Tracked Vehicles through Natural Language Descriptions

Dong Xie[1] Linhu Liu[1] Shengjun Zhang[2] Jiang Tian[1]

[1] AI Lab, Lenovo Research, Beijing, China
{xiedong2, liulh7, tianjiang1}@lenovo.com
[2] United Imaging Healthcare Surgical Technology, Wuhan, China
zsjcameron@gmail.com

Abstract

Through the development of multi-modal and contrastive learning, image and video retrieval have made immense progress over the last years. Organically fused text, image, and video knowledge brings huge potential opportunities for multi-dimension, and multi-view retrieval, especially in traffic senses. This paper proposes a novel Multi-modal Language Vehicle Retrieval (MLVR) system, for retrieving the trajectory of tracked vehicles based on natural language descriptions. The MLVR system is mainly combined with an end-to-end text-video contrastive learning model, a CLIP few-shot domain adaption method, and a semi-centralized control optimization system. Through a comprehensive understanding the knowledge from the vehicle type, color, maneuver, and surrounding environment, the MLVR forms a robust method to recognize an effective trajectory with provided natural language descriptions. Under this structure, our approach has achieved 81.79% *Mean Reciprocal Rank (MRR) accuracy on the test dataset, in the 7th AI City Challenge Track 2, Tracked-Vehicle Retrieval by Natural Language Descriptions, rendering the 2nd rank on the public leaderboard. Our code is available at* https://github.com/eadst/MLVR.

1. Introduction

Traffic is a critical aspect of urban infrastructure, intimately connected with urban planning and management. In recent decades, accelerating urbanization and population growth have exerted immense pressure on urban traffic systems. To address these challenges, an increasing number of cities have adopted intelligent transportation and urban monitoring systems. Intelligent transportation systems are traffic management solutions based on advanced information technology and data analysis. Incorporating computer vision and natural language processing techniques into intelligent transportation and urban monitoring systems promises significant advancements in urban traffic management and smart city operations.

Prompted by the demands of improving smart city operations, the AI City Challenge introduces several tracks of traffic-related tasks [12]. The task of Tracked-Vehicle Retrieval by Natural Language Descriptions made significant progress last year, with various teams participating in the challenge and achieving notable results [3, 13]. For instance, the authors of [19] developed a multi-granularity retrieval system and ranked first with an MRR of 56.52%; the authors of [6] proposed a semi-supervised domain adaptation training process and employed a context-sensitive post-processing method to analyze motion and prune retrieval results; the authors of [21] designed a symmetric network model to learn representations between language descriptions and vehicles, and a spatial relationship modeling method to identify relationships between vehicles and their surrounding environment, among other remarkable contributions. However, there remains room for improvement in the performance of these methods, as well as in the development of post-processing and pruning algorithms to achieve better retrieval results [13].

Building upon the successes of prior research, we have developed an innovative deep learning system called Multi-modal Language Vehicle Retrieval (MLVR) for text-vehicle retrieval. The MLVR system primarily comprises three core components: an end-to-end text-video contrastive learning module, a CLIP-based train-free domain adaptation technique, and a semi-centralized control optimization mechanism. The text-video contrastive learning module serves a crucial function in extracting video features by employing a combination of video and text information. The domain adaptation method is integrated to establish vehicle

color and type modules, which facilitate enhanced vehicle attribute matching. The control system is responsible for aggregating vehicle motion data and the surrounding environment, subsequently formulating a robust methodology for improving overall system performance. By synergistically harnessing the capabilities of these core components, the MLVR strategy significantly enhances the accuracy in identifying vehicle trajectories, paving the way for advancements in the field of multi-modal retrieval.

2. Related Work

2.1. Multi-modal and Contrastive Learning

The domains of multi-modal and contrastive learning are indispensable for devising advanced methodologies for the conjoint interpretation of visual and textual features. The Contrastive Language-Image Pre-training (CLIP) model represents an important advancement in this field, employing text-image pairs and a contrastive learning approach to spearhead novel research trajectories in multi-modal and computer vision areas [14]. CLIP is trained on an extensive dataset comprising images and their corresponding textual descriptions, utilizing a contrastive loss function to optimize the similarity between matched image-text pairs while minimizing it for non-matching pairs.

Following the success of CLIP, an array of models has emerged that build on its foundational principles. The Grounded Language-Image Pre-training (GLIP) model [9], as an extension of CLIP, integrates spatial grounding information during pre-training, thereby augmenting the model's capacity to localize objects and decipher spatial relationships within images. Moreover, a variety of CLIP-modified algorithms, such as CoOp, CLIP-Adapter, and Tip-Adapter, enrich the CLIP model with fine-tuning and supplementary extensions by adapting it to diverse contexts and scenarios [4, 20, 22]. Additionally, algorithms including ViLT, VLMO, and ALBEF employ the concept of contrastive learning to train models, facilitating their functionality in a broad range of tasks and environments [2, 5, 8]. Coca, Flamingo, and BeiT have demonstrated remarkable accuracy surpassing the state-of-the-art in an extensive array of vision and vision-language tasks, exhibiting competitive performance across numerous benchmarks [1, 16, 18].

Motivated by these advancements, our MLVR system fuses video and language information, leveraging contrastive learning to extract vehicle image, frame, and text attributes for the development of a robust and effective vehicle video retrieval approach.

2.2. Video Retrieval through Natural Language Descriptions

In the realm of video retrieval, numerous algorithms have been devised, building upon the foundations of multi-modal and contrastive learning principles. CLIP4CLIP, for instance, adapts the CLIP model for video retrieval tasks by extending the contrastive learning methodology with various frame characteristics combination techniques for video-text pairs [10]. This modification allows the model to acquire rich semantic representations from both video and textual data, subsequently enhancing its retrieval capabilities. X-CLIP is a cross-modal learning algorithm that harnesses the power of the CLIP model and combines it with the versatility of the transformer architecture [11]. By effectively integrating visual, temporal sequence, and textual information, X-CLIP achieves superior performance across a variety of video retrieval tasks. Furthermore, algorithms such as ActionCLIP, CLIPBERT, and InternVideo have made significant advancements in video retrieval tasks [7, 15, 17]. Inspired by the strengths of these algorithms, our MLVR system aspires to develop a robust and efficient vehicle video retrieval approach by merging video and language information and employing contrastive learning to extract meaningful features from vehicle frames and attributes.

3. Methodology

3.1. Method Overview

The core methods and key points of the Multi-modal Language Vehicle Retrieval (MLVR) system are discussed in this section. MLVR is an innovative approach that combines various techniques and strategies to enhance the retrieval process of vehicles by leveraging multi-modal information, including text, images, and videos. The primary focus of MLVR is to seamlessly fuse different types of data to gain a more comprehensive understanding of the vehicles and their trajectories. This is achieved by incorporating state-of-the-art deep learning models, domain adaptation methods, and optimization algorithms to extract valuable features and insights from the available data.

The main structure of MLVR is depicted in Figure 1. Our MLVR system is composed of several key components, including skilled text and image extractors that efficiently extract crucial information from textual and visual data sources. Furthermore, the video recognition model processes video frames and natural language descriptions to generate a sequence of insightful video vectors. Notably, the MLVR system leverages the combined power of images and keywords by employing a series of expertly designed control modules, encompassing vehicle color, vehicle type, vehicle motion, and vehicle surroundings. The furnished modules generate corresponding vector representations that capture the essential characteristics of the vehicle features and sequences. By integrating and weighting generated vectors through an algorithmic match control system, our MLVR system yields a final score matrix that effectively quantifies the relationships between textual and visual ele-

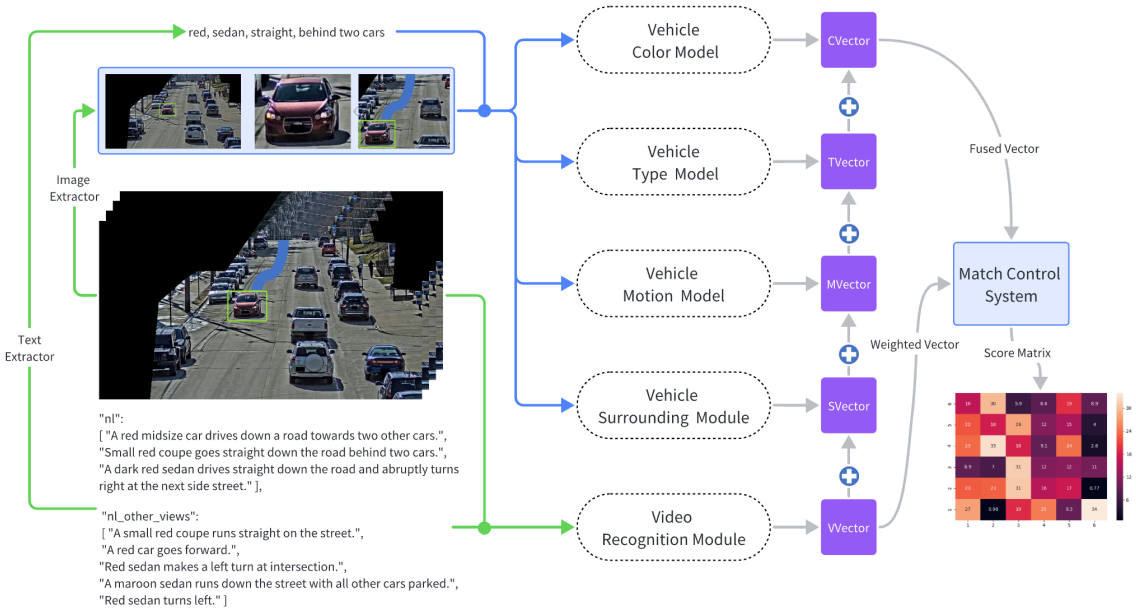

Figure 1. The structure of our MLVR system. The text extractor and image extractor extract the text and image effective information, respectively. Then, the video frames and NL description are fed into the video recognition module to generate the video vector sequence. The bundle information of images and keywords is delivered to the vehicle color module, vehicle type module, vehicle motion module, and vehicle surrounding module to create the corresponding vectors. The fused and weighted vectors are finalized in the match control system to produce the final score matrix.

ments.

3.2. Data Cleaning and Processing

3.2.1 Natural Language Analysis

In the context of text-based vehicle retrieval, textual information serves as a crucial component, offering abundant language details. By employing statistical analysis and natural language processing techniques on descriptive text, key attributes such as vehicle color, type, and motion can be accurately identified and extracted. Furthermore, a keyword parser is produced to categorize vehicle information based on a predefined set of keywords. Notably, color classifications include black, white, red, blue, and green, among others. Vehicle types contain sedan, SUV, pickup, van, bus, and truck, and motion directions consist of straight, stop, left, and right. A comprehensive summary of the keywords pertaining to each category is presented in Table 1.

Moreover, an analysis of Natural Language (NL) descriptions and corresponding descriptions from alternative perspectives (NL other view descriptions) reveals a connection, suggesting that NL other view descriptions are supposed to be transferred from descriptions in other scenarios. As illustrated in Figure 2, the relationship between NL descriptions and NL other view descriptions can be observed. The diagram indicates a potential weak connection,

Class	Label List
Color	blue, brown, gray, orange, black, purple, silver, green, white, yellow, red
Type	sedan, SUV, pickup, van, bus, truck
Motion	straight, stop, left, right

Table 1. The keywords information of each category. Word lists are created from the CityFlow-NL training dataset [3].

whereby the current scenario 1 could be projected onto scenario 2 or scenario 3. Consequently, when producing text-video pairs, partial penalty weights are considered for NL other view descriptions in comparison to standard NL descriptions, accounting for the observed relationships among the main scenario and additional scenarios.

3.2.2 Frame Analysis

The video frame information serves as an essential component in the text-video module, significantly influencing the outcome of the retrieval process. Each camera video is partitioned into multiple video clips, assigned unique track IDs, and accompanied by corresponding bounding boxes. Within the video recognition model, a pristine local road background is pictured by calculating the median value of

Figure 2. The different video frames and NL descriptions of the same vehicle in the CityFlow-NL train dataset. In the NL descriptions, tracked-vehicle color, type, motion, and other surrounding vehicle information are provided. Some NL annotations from other camera views are arranged in the NL other views.

each pixel across the video frames. Subsequently, a short clip is produced, incorporating the given region of interest (ROI) mask and background, along with the vehicle associated with the designated track ID. Additionally, a random number is selected to determine the frame interval for each iteration, facilitating image augmentation and enhancing the overall robustness of the model. A frame example of the processed video is displayed in the left corner region of Figure 3.

3.3. Model Architecture and Components

The architecture of the MLVR model has three different parts, comprising five interconnected modules, namely the video recognition module, vehicle color module, vehicle type module, vehicle motion module, and vehicle surrounding module. These modules synergistically ensure the accurate matching and fusion of textual and frame information, resulting in a more robust and generalized MLVR system. By integrating the above components, the MLVR model demonstrates enhanced performance in producing reliable retrieval results, thus contributing to the advancement of vehicle retrieval methodologies in practical traffic applications.

3.3.1 Video Recognition Module

The video recognition module, which serves as the foundation of our MLVR model, is adapted from the X-CLIP algorithm to effectively discern the association between video clips and their corresponding text sentences [11]. Multiple natural language descriptions are linked to a single track ID, so an equivalent number of text-video pairs are generated based on the number of descriptions to facilitate model training. Additionally, the weights of NL other view descriptions' text-video pairs are adjusted to mitigate the influence of other views on matching outcomes.

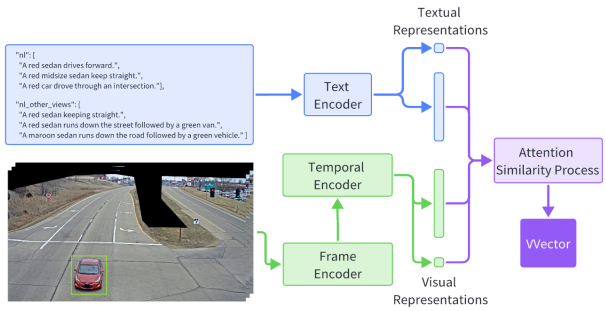

Figure 3. The primary architecture of the video recognition module. Frames are input into the frame and temporal encoders to generate visual representations, while text information is processed by the text encoder to yield textual representations. The attention similarity process combines visual and textual representations to predict a score for model evaluation and updating.

Figure 3 presents the primary architecture of the video recognition module. Frames are input into the frame encoder to extract visual features, which are subsequently processed by the temporal encoder to establish time series information. As a result, visual representations and their corresponding mean-pooled vectors are generated. Concurrently, the text encoder processes textual information to produce textual representations, encompassing both sentence and word-level data. The attention similarity process integrates visual and textual representations to compute the video vector for model evaluation and optimization. The video vector is displayed as follow:

$$V(v_i, t_j) = [s(v_i, t_{j1}), s(v_i, t_{j2}), ..., s(v_i, t_{jk})] \ , \quad (1)$$

where $V(v_i, t_j)$ represents the video vector corresponding

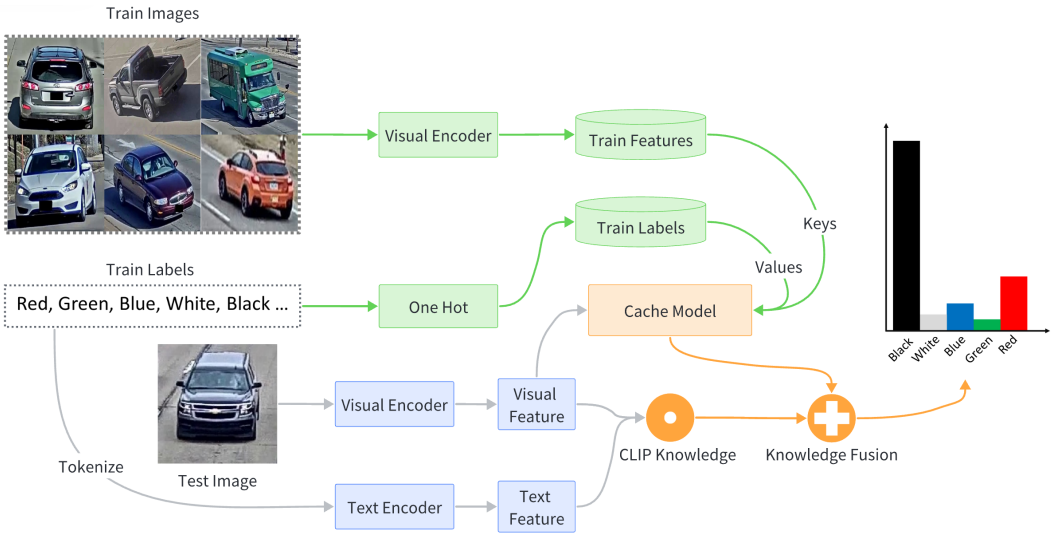

Figure 4. The architecture of the vehicle color module, which employs a CLIP-based few-shot learning model, consists of several distinct segments. The blue part collects general-purpose CLIP knowledge from the pre-trained model. After combining these knowledge sources in the orange section, the final logits are calculated for class predictions.

to video ID i and text ID j. The term $s(v_i, t_{jk})$ denotes the score associated with video ID i and text ID j for the k-th natural language description. Furthermore, the $s(v_i, t_{jk})$ can be represents as follow:

$$s(v_i, t_{jk}) = w_{jk} \times (s_{vs} + s_{vw} + s_{fs} + s_{fw})/4 \ , \quad (2)$$

where w_{jk} denotes the weight assigned to the k-the NL description of text ID j. The weight is higher when the description is part of the NL description section and lower in NL other views. The variables s_{vs}, s_{vw}, s_{fs}, and s_{fw} represent the video sentence score, video word score, frame sentence score, and frame word score, respectively, which are generated from the video recognition module. Additionally, the symmetric InfoNCE loss is incorporated to optimize the video recognition module with the mean value of the video vector, as illustrated by the following equations:

$$L_{v2t} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(mean(V(v_i, t_j)))}{\sum_{j=1}^{N} exp(mean(V(v_i, t_j)))} \ , \quad (3)$$

$$L_{t2v} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(mean(V(v_j, t_i)))}{\sum_{j=1}^{N} exp(mean(V(v_j, t_i)))} \ , \quad (4)$$

$$L_{vrm} = L_{v2t} + L_{t2v} \ , \quad (5)$$

where the L_{vrm} is the video recognition module loss consisting of video-to-text loss L_{v2t} and text-to-video loss L_{t2v} [11]. This dynamic approach not only enhances the overall performance of the MLVR system but also ensures its adaptability and generalizability across various text-video retrieval applications.

3.3.2 Vehicle Color and Type Module

Both the vehicle color module and the vehicle type module are based on the Tip-Adapter model, which effectively integrates the strengths of visual and textual few-shot information to enhance classification accuracy [20]. Figure 4 presents the architecture of the vehicle color module. In this module, vehicle images are cropped using bounding boxes from the trajectory, and corresponding labels are extracted using the keyword parser. These images are then input into the CLIP visual encoder to generate training features, while the labels are encoded using a one-hot encoder to produce training labels. The feature-key and label-value pairs are utilized to train the cache model.

During the testing phase, the input image is fed into the visual encoder to generate a visual feature, while the label list is tokenized and input into the text encoder to produce a text feature, as the CLIP framework. The CLIP knowledge, which incorporates both textual and visual information, is then combined with few-shot learning knowledge acquired from the trained cache model, using the visual feature as input. This fused knowledge is employed to generate the final logits, leading to more accurate and meaningful classification results. The logits calculation equations can be formatted as below:

$$C = exp(-w_{cm}(1 - f_{test}F_{train}) \ , \quad (6)$$

$$logits = w_{kf}C \odot L_{train} + f_{test} \odot L_{token} \ , \quad (7)$$

where the cache model output C is derived by employing

Figure 5. The example of vehicle motion module. The starting vector and the ending vector are represented by a green arrow and a pink arrow, respectively. The angle between two vectors is illustrated by an orange arc.

the weight w_{cm}, the CLIP visual feature of the test image f_{test}, and the few-shot training feature F_{train}. Moreover, the logits result from a combination of the cache model $C \odot L_{\text{train}}$ and the CLIP base model $f_{\text{test}} \odot L_{\text{token}}$. In equation 7, w_{kf}, L_{train}, and L_{token} denote the knowledge fusing ratio weight, the few-shot training one-hot label vectors, and the weight of CLIP classifier from the textual encoder, respectively. Considering the different pairs of multiple track frames and their associated descriptions, the vehicle color and type modules employ the mode value from a series of logits as the output of the vector.

3.3.3 Vehicle Motion Module

Through an in-depth analysis of vehicle maneuver trajectories, the vehicle motion module has been developed as a cultured direction control system. Utilizing the provided trajectories (comprising a list of bounding boxes), the vehicle starting vector is obtained by fitting the first partition of the trajectory center points using linear regression, while the vehicle ending vector is generated by fitting the last partition center points with an alternate linear regression function. Subsequently, the angle between the two vectors is computed to establish a baseline. Additionally, a set of threshold parameters is hypothesized to ascertain the final direction. The process for calculating the motion degree of a pickup vehicle is visually depicted in Figure 5, providing a clear representation of the methodology employed in the vehicle motion model.

3.3.4 Vehicle Surrounding Module

The vehicle surrounding module aims to extract information about neighboring vehicles in the vicinity of the tracked vehicle. Initially, textual representations of neighboring vehicle data are derived from natural language descriptions using the keyword parser. Simultaneously, other tracked vehicles within the same frame are identified and marked with bounding boxes, akin to the Region Proposal Network (RPN) generating multiple region candidates. Subsequently, the vehicle surrounding module branches into two distinct pathways.

The first branch processes the original frame image, feeding into the GLIP model [9] to establish a connection between NL descriptions and image feature information. In the second branch, different track ID vehicles in the given frame are merged into the relevant vehicle regions. These vehicle proposals are then inputted into the vehicle color and vehicle type models to determine the best-matching surrounding vehicle in accordance with the NL descriptions' adjacent vehicle textual representations.

The outputs from both branches are combined to generate the final vehicle surrounding SVector. The architecture of the vehicle surrounding module is visually represented in Figure 6, providing a comprehensive overview of the process. This intricate structure effectively leverages multiple sources of information to generate accurate predictions.

3.4. Model Postprocessing

3.4.1 Data Fusion

In this subsection, we focus on the post-processing and result fusion of the MLVR model. Utilizing the vector outputs from the five modules described earlier, weighted and fused vectors are computed. The weighted vector, denoted as V^w, is determined using the following equation:

$$V^w = (w_1^w \times \sum_{i=1}^{3} V_i^v + w_2^w \times \max_{i=1}^{n} V_i^v) \times \frac{1}{2} \quad , \tag{8}$$

where V_i^v represents the i-th text-video pair vector, while w_1^w and w_2^w denote the weights associated with NL descriptions and all descriptions, respectively. To capture the primary information from NL descriptions, the first three NL text-video pairs are averaged. Moreover, the maximum vector from the entire set of text-video pairs is obtained to consider the information from NL other view descriptions.

The fused vector, denoted as V^f, integrates the vector information from the five modules, and is calculated using the following equation:

$$V^f = w_c^f \times V^c + w_t^f \times V^t + w_m^f \times V^m + w_s^f \times V^s + w_v^f \times V^v, \tag{9}$$

where V^c, V^t, V^m, V^s, and V^v represent the CVector, TVector, MVector, SVector, and VVector, respectively, as illustrated in Figure 1. These vectors are generated from the five distinct modules. Corresponding weights, w_c^f, w_t^f, w_m^f, w_s^f, and w_v^f, are applied during the computation. This systematic approach to vector calculation ensures that the MLVR model effectively incorporates information from various sources, thereby enhancing the overall rigor and robustness of the retrieval process.

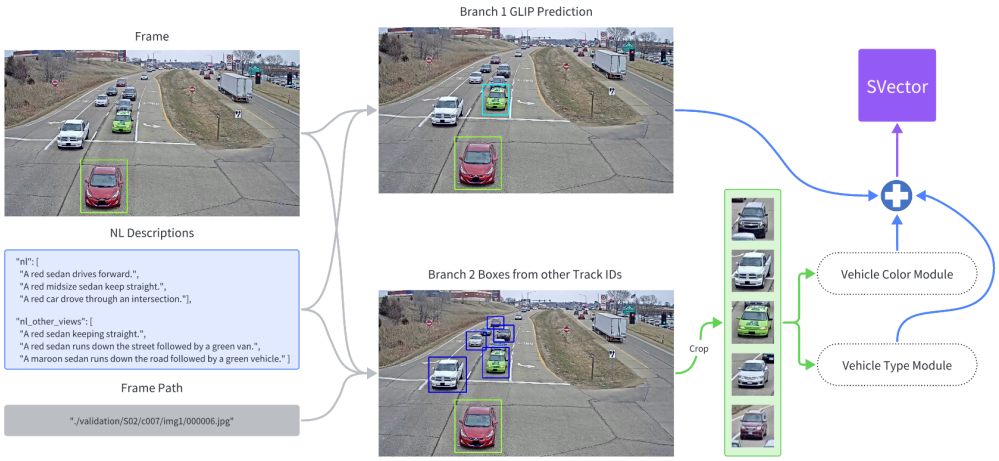

Figure 6. The structure of vehicle surrounding module. The frame and text are fed into the GLIP model to predict the track vehicle and adjacent vehicle positions and attributes. Concurrently, other vehicles mentioned in the frame with different track IDs, are marked as proposals, which are subsequently filtered by the vehicle color model and vehicle type model. The final SVector output is a product of the combined results from the GLIP model, vehicle color model, and vehicle type model.

Algorithm 1 Matching Elimination System

1: Input the text-video matrix tv
2: **for** start row = 1, length **do**
3: Get the highest score column index hci in $tv[row, :]$
4: Get the highest score row index hri in $tv[:, hci]$
5: **if** $row == hri$ **then**
6: For every element in column hci except $tv[hri, hci]$, minus a threshold mt
7: **end if**
8: **end for**

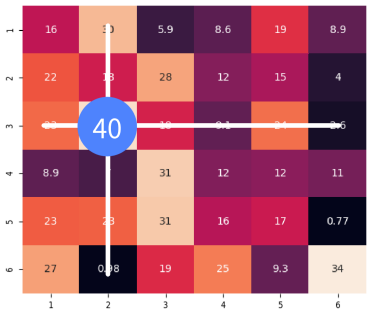

Figure 7. The matrix example of matching elimination system.

3.4.2 Match Control System

Following data fusion, the match control system is employed to identify the optimal text-video match. A matching elimination system is devised to re-rank the score matrix, thereby enhancing the final MRR score. Algorithm 1 outlines the core steps of the matching elimination system. The input text-video matrix is generated with m rows corresponding to text IDs and m columns corresponding to video IDs. In this algorithm, the highest score column index (hci) and the highest score row index (hri) are obtained for each row in the input text-video matrix. If the current row is equal to the highest score row index, a predefined threshold (mt) is subtracted from every element in the column hci, except for the element at position $tv[hri, hci]$.

Figure 7 illustrates a 6×6 text-video matrix example. In this example, the highest score is found at the intersection of row 3 and column 2, indicating a match between text ID 3 and video ID 2. Consequently, a predetermined threshold bias is subtracted from all other elements in column 2. This procedure is iteratively applied to all rows, resulting in a more robust and effectively sound matching process.

4. Experiments

4.1. Dataset and Evaluation Metric

The dataset employed for the evaluation of the MLVR model is CityFlow-NL, a comprehensive dataset that consists of 2,155 distinct vehicle trajectories and associated track IDs, as well as corresponding natural language descriptions [3]. These descriptions encompass both the current view and additional views with the same vehicle color and type information. In addition to the primary dataset, a separate test set comprising 184 distinct vehicle trajectories is utilized to assess the MLVR model's final performance. This test set follows a similar format to the training dataset, ensuring a fair and accurate evaluation of the model's capabilities in handling various scenarios and vehicle attributes.

The mean reciprocal rank (MRR) serves as the primary evaluation metric for assessing the performance of the MLVR model using the CityFlow-NL dataset. MRR is a widely recognized and effective measure for quantifying the quality of text-video retrieval systems, taking into account the ranks of correct matches. The MRR is mathematically defined as follows:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \quad , \tag{10}$$

where N represents the total number of text query IDs, and $rank_i$ denotes the rank index of the correct match track ID for the i-th query. By calculating the average of the reciprocal ranks for all queries, the MRR offers a comprehensive and reliable criterion for evaluating the performance of the MLVR model in the context of text-video retrieval tasks. Utilizing this evaluation metric allows for systematic comparisons among different models.

4.2. Implementation Details

In the experiment, we modified the state-of-the-art video-text retrieval model, X-CLIP, as the baseline for our video recognition module. The module incorporates the CLIP ViT base pre-trained model with patch 16 as its underlying architecture. The training process is executed with a learning rate of 1e-4 and is configured to accept a maximum of 32 words for each text query and a maximum of 20 frames for each video clip. The model is trained over 50 epochs, utilizing a batch size of 40 samples per batch. Our experimental setup employs a distributed training approach, harnessing the power of four NVIDIA V100 GPUs for efficient parallel computation. For the vehicle color and type modules, we employ the Tip-Adapter model to achieve robust and accurate few-shot classification performance. The model utilizes the ResNet-50 architecture as its backbone. The training process is conducted with a learning rate of 0.001 and spans a total of 100 epochs, leveraging a few-shot setting with 1024 shots.

4.3. Result and Analysis

A comprehensive analysis of our MLVR model's performance is presented, comparing it with other participating teams and conducting an ablation study to assess the contributions of each individual module towards the overall performance. Table 2 depicts the public leaderboard for the task of tracked-vehicle retrieval using natural language descriptions. The table illustrates the top 5 teams ranked by their MRR scores. Our MLVR model achieves a second-place ranking with an MRR score of 0.8179, marginally trailing the top-performing team, which attained an MRR score of 0.8263. These results underscore the competitive performance of our MLVR model in the context of tracked vehicle retrieval using natural language descriptions.

Rank	Team ID	Team Name	MRR
1	9	HCMIU-CVIP	0.8263
2	**28**	**IOV**	**0.8179**
3	85	AIO-NLRetrieve	0.4795
4	151	AIO2022	0.4659
5	76	DUT_ReID	0.4392

Table 2. The public leaderboard of tracked-vehicle retrieval by natural language descriptions.

Baseline	VCT	VM	VS1	VS12	MCS	MRR
✓						0.2761
✓	✓					0.4191
✓	✓	✓				0.5885
✓	✓	✓			✓	0.6714
✓	✓	✓	✓		✓	0.7160
✓	✓	✓	✓	✓	✓	0.8179

Table 3. Ablation study analysis of our MLVR method.

Table 3 presents an ablation study of our MLVR approach, elucidating the influence of each module on the MRR score. Due to submission limits, only six distinct ablation experiment results are displayed. Employing the baseline video recognition module alone results in an MRR score of 0.2761. The integration of the vehicle color and vehicle type modules (VCT) elevates the MRR score to 0.4191. Further incorporating the vehicle motion module (VM) enhances the MRR score to 0.5885. The inclusion of the match control system (MCS) yields an MRR score of 0.6714. By combining the branch 1 (VS1) GLIP prediction of vehicle surrounding module and the previous modules, the MRR score reaches 0.7160. Adding the vehicle surrounding module with branches 1 and 2 (VS12) culminates in the highest MRR score of 0.8179. This ablation study emphasizes the efficacy of each module in augmenting the overall performance of our MLVR model.

5. Conclusion

This research paper introduces an innovative multi-modal technique, called the Multi-modal Language Vehicle Retrieval (MLVR) system, for retrieving the trajectory of tracked vehicles based on natural language descriptions. By seamlessly integrating text, image, and video knowledge, we unlock immense potential for multi-perspective retrieval, particularly locating the best vehicle candidate from the multi-camera with multiple natural language descriptions. Our proposed approach demonstrates its effectiveness by achieving an impressive score on Track 2 at the 7th AI City Challenge, which highlighted the promising potential of MLVR in the traffic field of multi-modal retrieval.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[2] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022. 2

[3] 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. 1, 3, 7

[4] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 2

[5] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 2

[6] Huy Dinh-Anh Le, Quang Qui-Vinh Nguyen, Vuong Ai Nguyen, Thong Duy-Minh Nguyen, Nhat Minh Chung, Tin-Trung Thai, and Synh Viet-Uyen Ha. Tracked-vehicle retrieval by natural language descriptions with domain adaptive knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3300–3309, 2022. 1

[7] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 2

[8] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 2

[9] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 2, 6

[10] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 2

[11] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647, 2022. 2, 4, 5

[12] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, et al. The 7th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1

[13] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Archana Venkatachalapathy, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, et al. The 6th AI City Challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022. 1

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[15] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2

[16] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022. 2

[17] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 2

[18] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[19] Jiacheng Zhang, Xiangru Lin, Minyue Jiang, Yue Yu, Chenting Gong, Wei Zhang, Xiao Tan, Yingying Li, Errui Ding, and Guanbin Li. A multi-granularity retrieval system for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3216–3225, 2022. 1

[20] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 2, 5

[21] Chuyang Zhao, Haobo Chen, Wenyuan Zhang, Junru Chen, Sipeng Zhang, Yadong Li, and Boxun Li. Symmetric network with spatial relationship modeling for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3226–3233, 2022. 1

[22] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2