Tracked-Vehicle Retrieval by Natural Language Descriptions With Domain Adaptive Knowledge

Huy Dinh-Anh Le[1,2,†], Quang Qui-Vinh Nguyen[1,2,†], Vuong Ai Nguyen[1,2], Thong Duy-Minh Nguyen[1,2], Nhat Minh Chung[1,2], Tin-Trung Thai[1,2], Synh Viet-Uyen Ha[1,2,*]

[1] School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper introduces our solution for Track 2 in AI City Challenge 2022. The task is Tracked-Vehicle Retrieval by Natural Language Descriptions with a real-world dataset of various scenarios and cameras. We mainly focus on developing a robust natural language-based vehicle retrieval system to address the domain bias problem due to unseen scenarios and multi-view multi-camera vehicle tracks. Specifically, we apply CLIP [16] to effectively extract both visual and textual representations for contrastive representation learning. Furthermore, for new scenarios in the test set, we propose a novel Domain Adaptive Training method that utilizes information from labeled data and transfers it to the unseen domain by generating pseudo labels. By using this simple and effective strategy, we not only bridge the domain gap between the training set and test set, but also require less computational cost and data compared to previous top performance methods. Finally, we employ a context-sensitive post-processing method to address model's uncertainty and eliminate the wrong retrieved vehicle track. Taking one step further, we also investigate the impact of different text formats and the number of pseudo labels data for the fine-tuning process. Our proposed method has achieved 3rd place in the AI City Challenge 2022, yielding a competitive performance of 47.73% MRR accuracy on the private test set, which verified the effectiveness of the proposed solution.

1. Introduction

Vehicle retrieval is an important asset for the development of intelligent traffic systems in smart cities. In particular, being able to query for vehicles of interest from the pool of large databases is a powerful capability, as it brings along a wide array of useful applications in urban planning, traffic engineering, and security maintenance. While image-based vehicle retrieval systems have been the more prevalent type of approach, text-based vehicle retrieval systems have received noticeably increased attention in research. Unlike image-based retrieval systems which require at least an image of the target of interest, text-based ones can leverage easily obtainable natural descriptions of that target. In comparison with image queries, while text queries are arguably less effective in terms of describing fine-grained appearances, they are more intuitive, user-friendly and can easily provide for more layers of descriptions such as shape, color, position, and relativity to another target.

In the past, the use of natural text descriptions as queries was challenging. However, thanks to the recent development of effective language models, the Natural Language-based Vehicle Retrieval problem has become very promising to solve. Given a textual description of a particular vehicle, state-of-the-art Deep-Learning-based language models can be tokenized and extract useful keywords or phrases pertaining to the appearance, moving direction, scene of that vehicle and map them to an embedding space as a feature vector. Hence, the language-embedded feature vector can be used in a similar manner to the image-embedded feature vector in retrieval.

To accelerate research in the field, The 6th AI City Challenge has especially organized a challenge track to encourage active participation in developing text-to-image retrieval systems. While there have been promising results, it can be observed that a number of technical difficulties are still present:

Firstly, natural textual data can be very diverse.

[†]These authors contributed equally to this work
[*]Corresponding email: hvusynh@hcmiu.edu.vn

Although text data is very intuitive to humans, for machines it is very difficult to distinguish different descriptions of the same vehicle (e.g. "A vehicle is moving straight" - "The vehicle is heading forward"). The small amount of training data only seems to exacerbate the issue in learned models.

Secondly, there is a significant limit of high-quality training data. text-to-image vehicle retrieval is a relatively new domain so unlike the millions of samples in ImageNet [3], COCO [9] datasets for feature training, manual annotations are limited. Effective models are expected to leverage pretrained parameters as much as possible, and use the few labels for fine-tuning.

Finally, although existing state-of-the-arts are useful, their performances are still largely probabilistic in the high-dimensional text-to-image domain. Thus, prediction outputs may lack proper constraints to match a query with its true video.

Therefore, the main contributions of our paper are stated as follows:

- In order to utilize the training dataset and use it in a more efficient way for the training step, we introduce a pre-processing method for both text and images to maximize the amount of information the model can generalize and leverage for the new domain adaptive training method.
- We propose a new semi-supervised domain adaptive training method to address the domain bias between the training set and test set for the text-image retrieval model. Thus, we enforce the model to adapt new knowledge from the test set domain.
- Finally, to increase the overall performance of the final result by tackling the problem that the retrieval model cannot resolve due to the appearance of different scenarios and multiple camera types and angles, we introduce the context-sensitive post-processing method to address it.

2. Related Work

2.1. Video Retrieval by Natural Language

The recent works in this field are based on mapping the features of multiple different spaces to a common semantic space. Typically, most existing works aim to encode the given text queries using language feature extractor [20], and the vision-based information by video, sparsely sample frames from the video, or even both of them [23]. Besides, attention mechanisms and convolution techniques are commonly used as an encoder to learn the global and local contexts for video retrieval frameworks [12]. From video-language understanding, metric learning plays a main role to learn a function that minimizes the distance between these features. For instance, Bai et al. [1] utilize the InfoNCE loss to deal with the similarity of pairs of samples thus increasing the performance.

2.2. Video-Language Understanding from Multi-modal Features

Text Embeddings: There are many research works that have long shown how to represent words in vector space. For example, traditional text encoders (Word2Vec [13], LSTM [6]) were used to encode the natural language for language representations. However, in recent years, powerful transformer-family architectures are often used for word representations because of their effectiveness. Specifically, Devlin et al. [4] show the importance of bidirectional pre-training for language representations which can outperform the other methods on both sentence-level and token-level tasks.

Visual Features Extraction: Convolutional neural networks are the core of the most image features extractors which not only are used in many city-scale vehicle tracking tasks such as vehicle classification [21], vehicle detection [19] and vehicle re-identification [10] but also can be used as a specific features extractor for vehicle color [7] and vehicle type [18]. While the above vehicle characteristics are important in single-object retrieval, video global features like environment attributes, scenes, and other related objects have an important impact on ranking the accurate candidate videos.

Mapping the aforementioned heterogeneous input embeddings into the same semantic space is the problem that we face in this challenge. On account of the limited text and image data, we propose both to standardize language descriptions for reducing textual embedding variance in model learning, and to augment vehicle track data to improve robustness.

2.3. Contrastive Representation Learning

Contrastive representation learning is a powerful method that is used to teach the model to learn whether the sample pairs are similar or not in an embedding space. CLIP [16] is proposed as a multi-modal network to train both image and text encoders to predict accurate sample pairs.

2.4. Domain Adaptation

Single-object retrieval by new unseen inputs is always a challenging problem of the supervised retrieval pipeline. Due to this problem, in recent years, unsupervised domain adaptation shows significant effectiveness when applied in improving generalization by using

available knowledge from training to apply it to unlabeled data [22]. However, few teams apply this method to address the problem in text-video retrieval.

Therefore, in this paper, we propose a domain adaptive method to bridge the domain gap between the training set and the test set.

3. Methodology

3.1. Data Pre-processing and Augmentation

3.1.1 Natural Language Processing

There are numerous ways to describe anything in natural language and characterize each query and a wealth of information that can be provided. However, most information in queries is redundant and may interfere with other information in the retrieval model. Therefore, NLP pre-processing is essential in order to boost the generalization of the model during the learning stage.

Text Cleaning: With each text description, the cleaning process comprises removing the article, replacing misspelling words, and converting all verbs into their base form. After analyzing the dataset, we realized that many words have different forms but express the same semantic meaning. Therefore, we decide to cluster these words into several clusters based on their semantic similarity and replace them with their cluster name. For example, words like brown, brownish, bay, or beige will become brown while the other cases like mini cooper, couple, or coup will become coupe. We also simplify the vehicle movements into four types: go straight, turn left, turn right and stop.

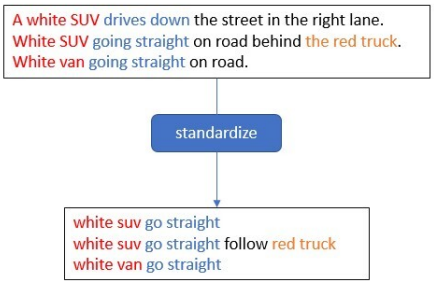

Figure 1. The example of the text standardization

Text Standardization: In order to reduce the variance language embeddings for learning with few data, we decide to standardize the text description into a more consistent format:

Vehicle color + Vehicle type + Vehicle movement + Relative description

Formally, for the i^{th} query q_i, we denote:

$$q_i = [\text{color}_i, \text{type}_i, \text{movement}_i, \text{surrounding}_i]$$

where:

$$\text{surrounding}_i = [\text{color\_others}_i, \text{type\_others}_i]$$

We use English PropBank Semantic Role Labeling (SRL) [17] to transform all statements to the format above, similar to Nguyen et al. [14]. However, there are still numerous sentences in which SRL is unable to extract information due to not finding any verb in the sentence. Therefore, we define a database that stores a dictionary about the valid color, vehicle type, and movement. To convert the sentence to the standard form, we find the first position vehicle type and then separate a sentence into two parts from that position:

For the first part, we locate the vehicle's attributes:

$$q_i = [\text{color}_i, \text{type}_i, \text{movement}_i, \text{surrounding}_i]$$

For the second part, we find the related information about other vehicle's attributes: $[\text{color\_others}_i, \text{type\_others}_i]$. An example of our standardization is shown in Fig. 1.

3.1.2 Data Augmentation

The training set contains only 2155 vehicle tracks, a very small amount of data compared to most video-text retrieval datasets. Thus, to overcome the lack of real-world data, we utilize Yolov5x [8] to detect all the vehicles in the video of each unseen scenario, then, use the tracker from Ha et al. [5] as a tracker to create vehicle track for each detected vehicle. These new annotations will be used for the Domain Adaptation Training and Post-Processing.

3.2. Domain-Adaptive Baseline Model

The baseline model has a big impact on the overall result, therefore selecting pre-trained models with robust visual and text representations is crucial. Following the proposed framework for Contrastive Representation Learning by [2], our baseline model consists of 3 main components: backbone, head, and objective losses. We propose a SSDA training scheme for CLIP to enforce domain adaptation, overall in 2 stages.

3.2.1 Stage 1: Baseline Training

Backbone We use CLIP as our main backbone to leverage its powerful knowledge in creating robust representation for feature extraction tasks with pre-trained model Transformer ViT-B/32 as the visual encoder and Text Transformer as the text encoder.

As also illustrated in Fig. 3, inputs to our backbone are both visual information and textual information:

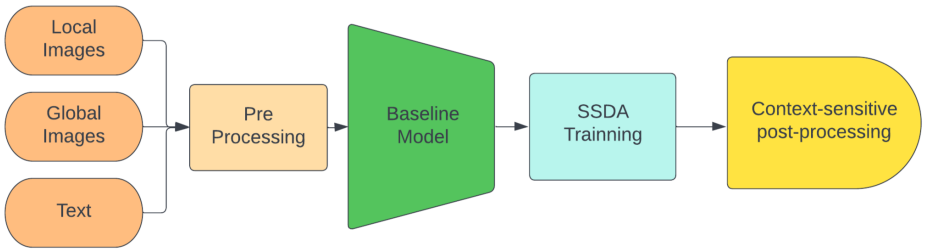

Figure 2. **The proposed Framework**. The proposed method mainly contains two components: Retrieval model and post-processing. Besides, pruning is used in the post-processing.

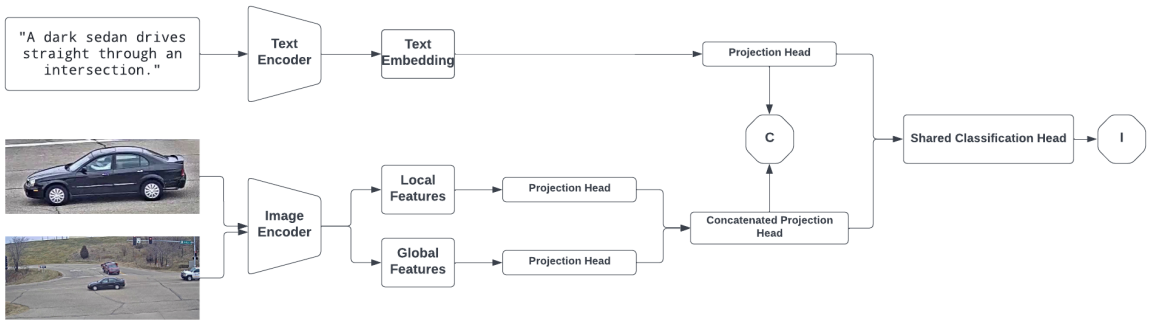

Figure 3. **The baseline model**. Where **C** is **symmetric cross entropy loss** and **I** is **Instance Loss**.

Regarding visual inputs We consider adopting the dual-stream input from Bai et al. [1] for the visual encoder in order to enhance the robustness of the model and capture more information in each vehicle track. Each vehicle track is represented as a single frame extracted randomly. As the dual-stream inputs, each vehicle track consists of a global image and a local image representing the visual global and local features respectively. We denote the global image as the original frame, while the local image is defined by performing cropping on each global image with the ground truth bounding box of the main vehicle. Both streams are then encoded by a share-weights image encoder to get two encoded feature vectors.

Regarding textual inputs We then use the text that is pre-processed in section 3.1.1 as the text input. Due to the inconsistency between provided texts in each vehicle track, encoding all of them jointly will hurt the robustness of the model, therefore we decide to randomly pick one of them and encode it with CLIP text encoder to get the text representation.

The output of each encoder is a feature vector with dimension [Batch Size, 512] (CLIP's encoder dimension) that represents the encoded features of each domain input.

Head Each representation for text and dual-stream image is then fed into independent projection heads with the intention to map each domain space into the space of contrastive representation learning where contrastive losses are applied. Each projection head is a small Multi-Layer Perceptron (MLP) with one hidden layer using a non-linear activation function ReLU and a Normalization Layer with Batch Normalization for visual representation and Layer Normalization for text representation, the visual and textual output's dimensions are 512 and 1024 respectively. In addition, the two streams' outputs are concatenated as the final visual representation. All the vectors are then normalized to be unit vectors.

Additionally, the classification head with the same structure as the projection head but the output dimension is the number of training tracks is introduced to classify the predicted probability of each track.

Contrastive Loss Given a batch B of (frame represents the vehicle track f_i, text t_i) pairs, there are $B \times B$ possible sample pairs. our objective is to maximize the cosine similarity between vehicle track f_i and text t_i:

$$s\left(f_i, t_i\right) = \frac{f_i^\top t_i}{\|f_i\| \, \|t_i\|} \tag{1}$$

Thus, we adopt the symmetric Cross-Entropy Loss [24] due to its ability to alleviate the model to learn multi-modal embedding space by jointly training visual and text embedding to maximize the similarity between

B positive pairs and minimize $B \times (B-1)$ negative pairs simultaneously. The loss consists of two parts: Image-to-Text and Text-to-Image.

Image-to-Text Loss:

$$\mathcal{L}_{f \to t} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp\left(s\left(f_i, t_i\right)\right)}{\sum_{j=1}^{B} \exp\left(s\left(f_i, t_j\right)\right)} \tag{2}$$

Text-to-Image Loss:

$$\mathcal{L}_{t \to f} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp\left(s\left(f_i, t_i\right)\right)}{\sum_{j=1}^{B} \exp\left(s\left(f_j, t_i\right)\right)} \tag{3}$$

Symmetric Cross Entropy Loss:

$$\mathcal{L} = \mathcal{L}_{f \to t} + \mathcal{L}_{t \to f} \tag{4}$$

Instance Loss To capture the global discrepancy in the bi-directional domains in the text-video retrieval task, we adopt instance loss [25] as a common optimization goal. Every vehicle track and its accompanying descriptions are treated as a single category. The purpose of the optimization is to combine visual and textual data into a single categorization space. Thus, we use one single classification head with weight $W_{\text{classification}}$ for both visual and textual embedding to promote the model to learn the mapping function between two domains.

Image-to-Classification Loss:

$$\mathcal{L}_i^f = -\log\left(W_{\text{classification}} f_i\right) \tag{5}$$

Text-to-Classification Loss:

$$\mathcal{L}_i^t = -\log\left(W_{\text{classification}} t_i\right) \tag{6}$$

Instance Loss:

$$\mathcal{L}_i instance = \mathcal{L}_i^f + \mathcal{L}_i^t \tag{7}$$

3.2.2 Stage 2: Semi-supervised Domain Adaptation (SSDA) Training

To tackle the unseen scenarios that appear in the test set, which lead to domain bias between the training and test sets, we propose a semi-supervised domain-adaptive method and training strategy to address the problem. The method consists of two main parts: generate pseudo labels and fine-tune the baseline model.

For the pseudo label part, due to the difference between the two domain features, we cannot use regular feature clustering methods like typical Unsupervised Domain Adaptation ReID tasks. Thus, we develop a new method to generate pseudo labels using the current knowledge of the baseline model and training set. For text, due to the diversity of the content, we cannot re-create any pseudo label near that content level. Hence we decided to re-format the description text of each vehicle track into a much clearer form using the technique in section 3.1.1:

Vehicle color + Vehicle type + Vehicle movement

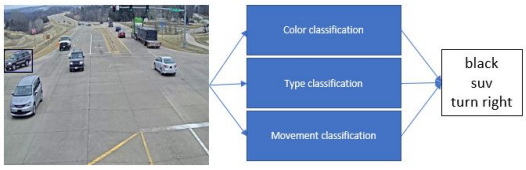

Figure 4. The process for generating pseudo labels.

In addition, we also convert the test queries into the new format in order to maximize the effectiveness of Domain Adaptation Training during the inference process.

With this new format, we can easily generate pseudo labels for any vehicle track using three main modules: color, type, and movement classification.

For color and type classification, we observe that the baseline model can retrieve the vehicle track that has the color and type that match closely to the text description. Thus, we utilize the knowledge of the model trained in **Stage 1** (only the local image stream is considered in this model). In addition, the classification head is replaced with a new one that has the output dimension equal to the number of type/color classes. We also re-use type and color clusters in section 3.1.1 as the label for classification models.

The trajectory analysis method in section 3.1.2 is applied for vehicle movement classification. However, to avoid the variety in the description of the vehicle's trajectory, we split trajectory descriptions into six movements: "turn left and go straight", "turn right and go straight", "stop and go straight", "turn left", "turn right" and "go straight".

Then, by using the generated pseudo labels, we fine-tune the baseline model trained in **Stage 1** for 4 epochs.

3.3. Context-sensitive Post-processing

To tackle the problem that occurs due to different camera types and angles, we propose several heuristic-driven algorithms in order to analyze the right vehicle movement.

Figure 5. The example of the motion analysis method

3.3.1 Motion Analysis

There are many recent works analyzing the vehicle trajectory to re-rank the retrieval result. Park et al. [15] use GPS to get the velocity vector, then compute the angle by using the cross-product of the two vectors. Nguyen et al. [14] determine vehicle direction by calculating the area of the vehicle's trajectory. Because each of the solutions above has its own problems, we come up with an improved solution.

Firstly, our method needs to handle the stop case, thus, we count the number of frames where the bounding box location of the vehicle is unchanged. When the counter reaches the threshold, we consider that the car stops. We also remove all the points that consider stopped points. To handle the turning case, we will use three heuristics to consider going straight or turning:

- In the first heuristic, we compute the vectors at the start and end point of motion. With the start point is $A(x_1, y_1)$, end point is $B(x_{|P|}, y_{|P|})$ and $|P|$ is total number of points on the trajectory. We define $M(x_M, y_M)$ as the point at one fifth of $|P|$ from the starting trajectory and $N(x_N, y_N)$ as the point at one fifth of $|P|$ from the ending trajectory. Then we calculate the angle $\theta = \cos(\vec{AM}, \vec{NB})$. If the θ is in the range from θ_1 to θ_2, itemwe will consider the vehicle is turning. (Fig. 5a).
- In the second heuristic, we find the distance d so that $d = \max_{i \to |P|} (\text{Dis}(p_i, AB))$ where Dis is the minimum distance between the p_i and the segment AB. If d is in the range from d_1 to d_2, this trajectory will consider as turn. (Fig. 5b)
- In the final heuristic, we find the difference between the max and min ratio change of the vehicle bounding box $r = \frac{\max_{i \to |P|}(\frac{w_i}{h_i})}{\min_{i \to |P|}(\frac{w_i}{h_i})}$. If r is larger than the threshold, the vehicle is considered to be turning. (Fig. 5c and Fig. 5d)

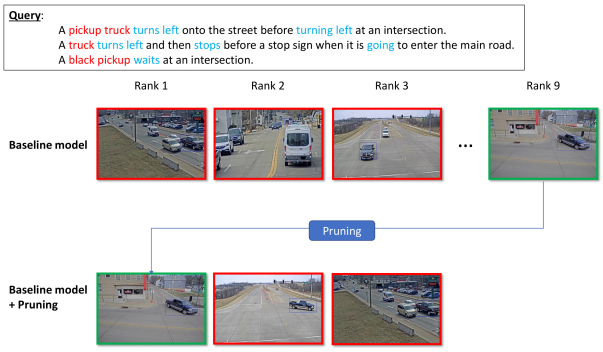

Figure 6. The result of our pipeline after pruning.

To determine if the vehicle turn left or turn right, our team determine based on the sign of the counterclockwise $CCW(A, M, B)$. If the sign of CCW is positive, the vehicle turn left; otherwise turn right.

3.3.2 Pruning

Pruning based on surrounding vehicles In the query, several sentences state the relationship between the target vehicle and the vehicle around, which is our baseline model's limitation.

Hence, our team's approach to taking into account surrounding vehicles is based on the relationship between the query and the contextual information of the candidate vehicle in the scene. With each text query q_i, we will get the vehicle's information around it, like $\text{color\_others}_i, \text{type\_others}_i$.

After that, with the result vehicle tracks sorted by the distance in the previous step, we will choose the top first K candidate vehicles to prune. For each candidate's vehicle, we denote this vehicle is:

$$\text{candidate\_vehicle}_k, \text{ where } 0 \leq k \leq K$$

If the $\text{candidate\_vehicle}_k$ has at least one vehicle

near it with the color_others$_i$ and type_others$_i$, we will append it to the keeplist; otherwise, append it to the skiplist. The output in this module is the result of concatenating keeplist, skiplist, and the remainder result vehicle tracks.

Pruning based on main vehicle direction Our model is trained with a randomly chosen frame to best learn its appearance and pose via color, type, and direction description, but it is tough for the model to learn the vehicle's direction due to this randomness. Hence, similarly to how neighbor vehicles are used to constrain context information, we further propose to prune vehicles by their trajectories directly. For this effort, we only consider the vehicles' direction movement$_i$.

Specifically, we select the first K cars candidates for each query. We utilize the vehicle direction method described previously in section 3.3.1 to determine the direction of each vehicle's trajectory. Following that, we may use the same pruning idea in section 3.3.2.

4. Experiments

4.1. Dataset

The data used in this work is built upon the CityFlow Benchmark, which contains 3.25 hours (215.03 minutes) of footage from 40 cameras spanning 10 junctions in a mid-sized US metropolis with a distance of 2.5 kilometers between the two furthest existing benchmarks. Each of the dataset's five scenarios represents a different type of location but only two scenarios are utilized for training in this challenge. The dataset contains 2155 tracks of vehicles with three unique natural language descriptions each. Specifically, there are 2155 vehicle tracks in the dataset, each with three distinct natural language descriptors. For this challenge, 184 unique vehicle tracks were curated, along with 184 query sets, each with three descriptions.

4.2. Validation Data

Since the evaluation server only has 20 submissions, we have to evaluate our method offline by re-creating a validation set that is close to the evaluation server. Therefore, we use a portion of the training dataset from the Track 5 AI City Challenge 2021 since the scenarios appearing on the training dataset are the same as the 2022 test set. The data contains 161 tracks with 91 tracks of scene 1 and 70 tracks of scene 4.

4.3. Implementation Details

Baseline Models in Stage 1 All the training images are resized to 224×224 and normalized. In both training stages, we use AdamW [11] as the optimizer with the initial learning rate set to $1e^{-2}$. We train the model with 5 epochs with a batch size of 64. We totally train 2 baseline models on the training dataset with the difference in the text format, with **Type 1 (original format)** and **Type 2 (new format)** respectively to evaluate the performance of the new format.

SSDA Training on Stage 2 All the training settings are the same as the baseline model, then we fine-tune each baseline model for 4 epochs with the dataset that is augmented from section 3.1.2. However, to examine the performance of each unseen scenario, we split unseen scenarios into two datasets where dataset 1 contains scene 1, dataset 2 contains scenes 3 and 4 and fine-tuned them separately. All models are trained on GPU Tesla P100

Inference text format During the inference stage, we use the text format similar to the format in which the model is trained/fine-tuned to maximize the retrieval performance.

Classification Models All the training settings are the same as the baseline model, except the batch size is changed to 128. Each classification model is trained for 5 epochs and used to automatically generate type and color for vehicle tracks by inference through all frames of each track and then choose the class with the highest occurrence.

4.4. Ablation Study

4.4.1 Two-Stage Training

To evaluate the effectiveness of the SSDA Training method, we first conduct experiments on the new text format on the baseline model. From the table 1, the result of the new text format on the baseline model has smaller results compared to the original format. However, after SSDA Training, we can observe that the performance difference between the two format types is significant. We believe that the difference in performance after fine-tuning is due to the inconsistency between the new and original format. Therefore, fine-tuning with data that is similar to baseline model knowledge is crucial in improving the performance of SSDA Training.

4.4.2 Dataset for SSDA Training

To investigate the effectiveness between augmented data from section 3.1.2 and the test set, we use two datasets for SSDA Training and train them separately, one is augmented data on Scene 1 and one is from the test set. The result is shown in table 2 that our augmented data yields better performance. The reason

Methods	MRR	Recall@5	Recall@10
Baseline (Type1)	0.3242	0.4658	0.6522
+SSDA Training	0.3635	0.5342	0.7019
Baseline (Type2)	0.2384	0.3043	0.5155
+SSDA Training	0.3965	0.5342	0.7329

Table 1. The ablation study of two-stage training using CLIP as backbone. Where **type 1** is the **original text format** and **type 2** is the **new text format**

is due to the number of fine-tuning data, where augmented data has 398 vehicle tracks on Scene 1 while the test set only has 184 vehicle tracks on three scenes and the distribution between three scenes is imbalanced.

Methods	MRR	Recall@5	Recall@10
SSDA Training (S1)	0.4531	0.7640	0.5776
SSDA Training (Testset)	0.3965	0.5342	0.7329

Table 2. The ablation study of datasets for SSDA Training

4.4.3 Post-processing

From the table 3, we can see that each pruning method can help increase the overall performance of the result. Using post-processing, the increase in performance depends on the retrieval model. We believe the cause is due to the domain bias between the test set and training set, thus the baseline model will retrieve the vehicle track that is closer to seen scenarios than unseen scenarios. Hence, it leads to certain retrieved vehicle tracks having the wrong direction and surrounding vehicles.

Methods	MRR	Recall@5	Recall@10
Baseline	0.3242	0.4658	0.6522
+Pruning	0.3879	0.5528	0.7143
SSDA Training (S1)	0.4531	0.7640	0.5776
+Pruning	0.4829	0.5839	0.7640

Table 3. The ablation study of post-processing

4.4.4 Ensemble

To test the robustness between two models and boost the final performance, we ensemble two fine-tuning models on two datasets mentioned in section 4.4.2. From the ensemble result on table 4, we can see that the MRR Score increased drastically to 0.5029 even when the score of the model trained on scene 1 and test set is only 0.4531 and 0.3965 respectively. Thus, proving that each scene contributes independently to the knowledge of the model and enriches the contrastive representation space separately.

Methods	MRR	Recall@5	Recall@10
SSDA Training (S1)	0.4531	0.7640	0.5776
SSDA Training (Testset)	0.3965	0.5342	0.7329
Ensemble	0.5029	0.6522	0.8261
+Pruning	0.5175	0.6646	0.8261

Table 4. The ablation study of ensemble

4.5. Challenge Results

As shown in table 5, the final score of our team (Team ID 4) final mean reciprocal rank for the test set is 0.4773. We achieved the rank #3 among 15 teams on Track 2 Natural Language-Based Vehicle Retrieval of AI City Challenge 2022.

Rank	Team Name	MRR
1	Must Win	0.6606
2	Thursday	0.5251
3	**HCMIU-CVIP (Ours)**	**0.4773**
4	MegVideo	0.4392
5	HCMUS	0.3611

Table 5. The overall ranking on MRR score of AI City Challenge 2022 - Track 2: The Natural language based retrieval

5. Conclusions

In this paper, we proposed a robust natural language-based vehicle retrieval system with a new domain adaptive training method which can enhance the model's knowledge and tackle the domain bias problem. In addition, the proposed framework requires less computation capability and data compared to previous top performance methods but still yields a competitive result in AI City Challenge 2022 at Top-3 on the private test set.

Acknowledgment

We would like to express our heartfelt appreciation to Ho Chi Minh City International University—Vietnam National University (HCMIU-VNU) for facilitating our efforts. Additionally, we would like to express our heartfelt appreciation to all of our colleagues for their contributions, which considerably aided in the revision of the manuscript.

References

[1] Shuai Bai, Zhedong Zheng, Xiaohan Wang, Junyang Lin, Zhu Zhang, Chang Zhou, Hongxia Yang, and Yi Yang. Connecting language and vision for natural language-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4034–4043, June 2021.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia 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. Ieee, 2009.

[4] 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.

[5] Synh Viet-Uyen Ha, Nhat Minh Chung, Tien-Cuong Nguyen, and Hung Ngoc Phan. Tinypirate: A tiny model with parallelized intelligence for real-time analysis as a traffic counter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4119–4128, June 2021.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[7] Chuanping Hu, Xiang Bai, Li Qi, Pan Chen, Gengjian Xue, and Lin Mei. Vehicle color recognition with spatial pyramid deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2925–2934, 2015.

[8] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014.

[10] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2016.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017.

[12] 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. *arXiv preprint arXiv:2104.08860*, 2021.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[14] Tien-Phat Nguyen, Ba-Thinh Tran-Le, Xuan-Dang Thai, Tam V. Nguyen, Minh N. Do, and Minh-Triet Tran. Traffic video event retrieval via text query using vehicle appearance and motion attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4165–4172, June 2021.

[15] Eun-Ju Park, Hoyoung Kim, Seonghwan Jeong, Byungkon Kang, and YoungMin Kwon. Keyword-based vehicle retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4220–4227, June 2021.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[17] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling, 2019.

[18] Pranjay Shyam, Kuk-Jin Yoon, and Kyung-Soo Kim. Adversarially-trained hierarchical feature extractor for vehicle re-identification. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13400–13407. IEEE, 2021.

[19] Zehang Sun, George Bebis, and Ronald Miller. On-road vehicle detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 28(5):694–711, 2006.

[20] Dongyang Wang, Junli Su, and Hongbin Yu. Feature extraction and analysis of natural language processing for deep learning english language. *IEEE Access*, 8:46335–46345, 2020.

[21] Wei Wu, Zhang QiSen, and Wang Mingjun. A method of vehicle classification using models and neural networks. In *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202)*, volume 4, pages 3022–3026. IEEE, 2001.

[22] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[23] Hongkai Xiong, Zhiming Pan, Xinwei Ye, and Chang Wen Chen. Sparse spatio-temporal representation with adaptive regularized dictionary learning for low bit-rate video coding. *IEEE transactions on circuits and systems for video technology*, 23(4):710–728, 2012.

[24] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

[25] Zhedong Zheng, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, and Yi-Dong Shen. Dual-path convolutional image-text embeddings with instance loss. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2):1–23, 2020.