Supplementary Material for the Paper: Ego-Vehicle Action Recognition based on Semi-Supervised Contrastive Learning

Chihiro Noguchi Toshihiro Tanizawa
Toyota Motor Corporation, Japan
{chihiro_noguchi_aa, toshihiro_tanizawa}@mail.toyota.co.jp

A. GCN architecture

The constructed ST-graphs $G_n, \forall n \in \{1, \ldots, N\}$, where N denotes the number of video clips, are fed into a GCN. Following [1], our GCN model consists of three parts: an encoder, propagation layers, and an aggregator.

Encoder. Node attributes $\boldsymbol{s}_i$ and $\boldsymbol{g}_i$ are separately fed into multilayer perceptrons (MLPs) first:

$$\boldsymbol{s}'_i = \mathrm{MLP}_s(\boldsymbol{s}_i), \quad \forall i \in V_n \tag{1}$$

$$\boldsymbol{g}'_i = \mathrm{MLP}_g(\boldsymbol{g}_i), \quad \forall i \in V_n. \tag{2}$$

Here $\boldsymbol{s}'_i \in \mathbb{R}^{32}$ and $\boldsymbol{g}'_i \in \mathbb{R}^{32}$ are the same 32-dimensional. $\boldsymbol{s}_i$ and $\boldsymbol{g}_i$ can have different properties due to the one-hot encoding of $\boldsymbol{s}_i$. Therefore, it is useful to first map the node attributes at the encoder, rather than feeding them directly to the propagation layer. The encoded attributes are concatenated as $\boldsymbol{x}_i^{(0)} = [\boldsymbol{g}'_i, \boldsymbol{s}'_i]$.

Propagation Layers. In a propagation layer, the features of each node are aggregated according to adjacencies defined by the ST-graphs. Our GCN model adopts the local extrema convolution (LEConv) [2], whose update formula for the lth layer is defined as follows:

$$\boldsymbol{x}_i^{(l+1)} = \sigma\left(\boldsymbol{x}_i^{(l)} W_1^{(l)} + \sum_{j \in \partial i} e_{ij}\left(\boldsymbol{x}_i^{(l)} W_2^{(l)} - \boldsymbol{x}_j^{(l)} W_3^{(l)}\right)\right), \tag{3}$$

for $\forall i \in V_n$, where ∂i denotes a set of indices of adjacent nodes of i, and $W_1^{(l)}$, $W_2^{(l)}$ and $W_3^{(l)}$ denote learnable parameters. $\sigma(\cdot)$ denotes the ReLU activate function.

Aggregator. As outputs of the propagation layers, we obtain node representations $\boldsymbol{x}_i^{(L)}$, $\forall i \in V_n$, where L denotes the number of propagation layers ($L = 3$ in our setting). The aggregator performs a pooling operation to output a graph-level representation. In our GCN model, in order to explicitly learn instance-level features, we introduce an instance-level pooling. As a result, graph-level representation $\boldsymbol{z}_n$ can be obtained as the output of the following aggregator operation:

$$\boldsymbol{z}_n = \mathrm{MLP}_1\left(\sum_{u \in \mathcal{I}_n} \mathrm{MLP}_2\left(\sum_{i \in \mathcal{N}_{nu}} \mathrm{MLP}_3(\boldsymbol{x}_i^{(L)})\right)\right), \tag{4}$$

$\forall n \in \{1, \ldots, N\}$, where $\mathcal{N}_{nu}$ denotes a set of nodes corresponding to object instance u in ST-graph n, and $\mathcal{I}_n$ denotes a set of object instances in graph n.

B. Ablation Studies

To provide a further understanding of the proposed method, we perform two kinds of ablation studies. First, we investigate the effect of removing each of the three types of node attributes: semantic labels, geometric features of bounding boxes, and interaction with lane lines on classification performance. Second, we examine how effective on the classification performance by alleviating the significant imbalance between the number of labeled and unlabeled videos.

B.1. Effect of node attributes.

Table 1 shows the results of ablation studies for node attributes. These results correspond to the classification performance of SCL for labels of 11 Goal-oriented actions. The top row in table 1 shows the results with three kinds of node attributes. The second, third and forth rows show the results without using semantic labels, geometric features of bounding boxes (bbox features) and interaction with lane lines (lane line features), respectively.

Without using the semantic labels, the mAP values are not significantly different with those of SCL with all kinds of node attributes. This indicates that distinguishing between kinds of object instances has little impact on the classification performance of the goad-oriented actions. However, when recognizing other kinds of scenes including unlabeled scenes, semantic labels in node attribute would be possible to play an importance role. For example, Fig. 5 shows pedestrians crossing a crosswalk. To recognize this

Methods	Individual actions											Overall mAP
	intersection passing	L turn	R turn	L lane change	R lane change	L lane branch	R lane branch	crosswalk passing	railroad passing	merge	u-turn	
SCL	98.3	94.1	95.8	62.6	67.3	53.4	28.4	78.0	1.2	22.2	60.0	60.1
SCL (w/o semantic)	97.9	94.8	95.8	60.5	57.3	53.8	20.6	77.8	3.5	28.7	55.1	58.7
SCL (w/o bbox)	98.2	92.5	94.5	55.0	53.0	55.2	22.9	73.2	1.2	23.5	30.2	54.5
SCL (w/o lane)	95.8	92.8	93.5	50.3	45.6	25.2	13.0	58.1	1.2	10.9	52.0	48.9

Table 1. Comparison of classification performance of SCL when removing each of the three kinds of node attributes. The top row shows the results with three kinds of node attributes. The second, third and forth rows show the results without using semantic labels (w/o semantic), geometric features of bounding boxes (w/o bbox) and interaction with lane lines (w/o lane), respectively.

scenes, it is important to distinguish pedestrians from vehicles.

When the box features are not taken into account, the mAP values largely drop. We consider that this is because without the bounding box information, it cannot accurately track moves of object instances. However, the performance degradation is less than the case without lane line features. We believe that this is because relative positions between object instances, which are given from a ST-graph, can also be used to track moves of object instances. As a result, the performance degradation may have been reduced compared to the case without lane line features.

Finally, when the lane line features were not included in node attributes, the performance degradation was greatest in the three cases. We consider the reason for this is that the lane line features cannot be substituted for other features. In fact, the AP values of individual actions significantly influenced by lane lines (L/R lane change, L/R lane branch and merge) significantly dropped when lane line features were not used.

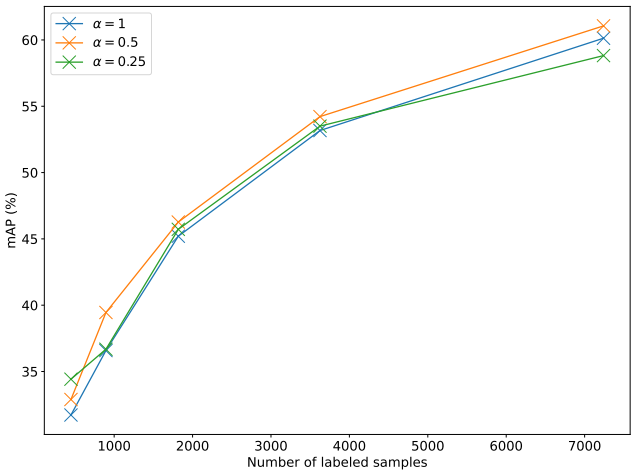

Figure 1. Change in classification performance of the proposed method with SCL when varying α.

B.2. Varying weights in the loss function.

Since it is much easier to collect unlabeled videos than to collect labeled videos, the number of unlabeled data is much more than that of labeled data in many cases. If the focus is solely on improving the classification performance, this imbalance can have a negative impact on the performance. A straightforward approach to alleviate the imbalance is to introduce weights into the loss function. Therefore, in this section, we investigate how much the classification performance is improved when reducing the weight of unlabeled video. The loss function with explicitly introduced weights is

$$\mathcal{L} = \sum_{n=1}^{N} \mathcal{L}_n = -\sum_{n=1}^{N} \alpha_n \sum_{\boldsymbol{z}_+ \in \mathcal{P}_n} \log \left(\frac{e^{\boldsymbol{z}_+ \cdot \boldsymbol{z}_n}}{\sum_{\boldsymbol{z}_k \in \mathcal{A}_n} e^{\boldsymbol{z}_k \cdot \boldsymbol{z}_n}} \right), \tag{5}$$

where

$$\alpha_n = \begin{cases} 1 & (\text{if } n \text{ is a labeled data}), \\ \alpha & (\text{if } n \text{ is an unlabeled data}). \end{cases} \tag{6}$$

Here, α controls the strength of the effect of unlabeled videos. When $\alpha = 1$, Eq. 5 is equivalent to Eq. 8 in the main text.

Figure 1 shows overall mAP values for labels of 11 Goal-oriented actions when $\alpha = 1, 0.5$ and 0.25. As can be seen, although the mAP values are highest at $\alpha = 0.5$ in most cases, the difference is slight. The smaller the number of labeled videos, the larger the difference between the number of labeled and unlabeled videos. However, even the number of labeled videos was very small, the effect of the introduction of weights was not able to be confirmed. Therefore, in the other experiments in this paper, the value of α was fixed at 1.

C. Query-Retrieval Examples of Unlabeled Videos

In this Appendix, we present query-retrieval examples to qualitatively evaluate video-to-video distances learned by the proposed methods. As described in Sec. 4.3 of the main text, we chose a query video from unlabeled videos in the validation set and searched the nearest neighbor video on the query video in the embedding space learned by each

method. The nearest neighbor video was found from all videos including both labeled and unlabeled videos in the train set. The distances were measured using cosine similarities between feature vectors output from the GCN. In Figs 2-13, the remaining samples, which could not be included in the main text due to space limitations, are presented.

In addition, we present average SOIA distances between query videos and corresponding top-1 retrieved videos in Table 2. As can been seen, retrieved videos by using SCL have the smallest average SOIA distance to the corresponding query videos.

Methods	Average SOIA distances ($\times 10^4$)
SCL	8.68
GCL	9.83
FSL	11.69

Table 2. Average SOIA distance between query videos and corresponding top-1 retrieved videos.

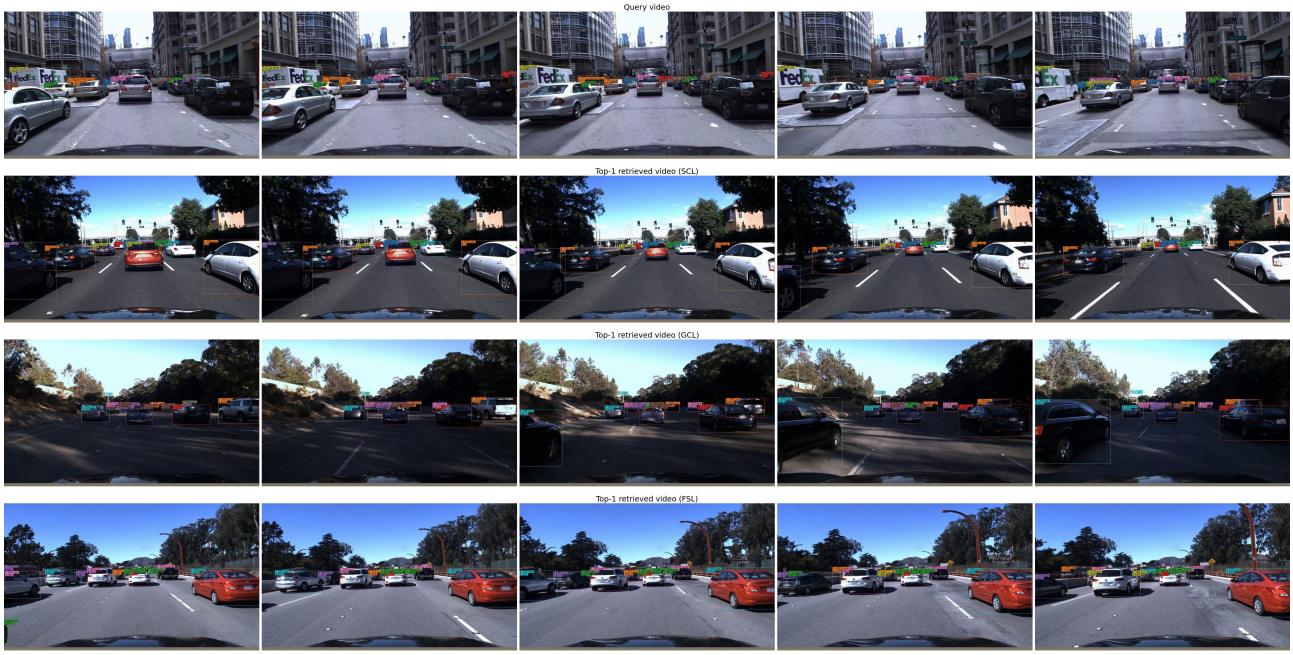

Figure 2. Five frames extracted at equal interval from query and retrieved videos. The top row shows a query video, and the second, third and forth rows show top-1 retrieved videos obtained from the proposed methods with SCL, GCL and FSL, respectively. In the query video, the ego-vehicle is on a busy road. The second row of video similarly shows a crowded driving scene. Note that the proposed methods primarily focus on the relationship between object instances detected in video and do not consider environmental conditions such as road conditions, surrounding buildings and nature.

Figure 3.

Figure 4.

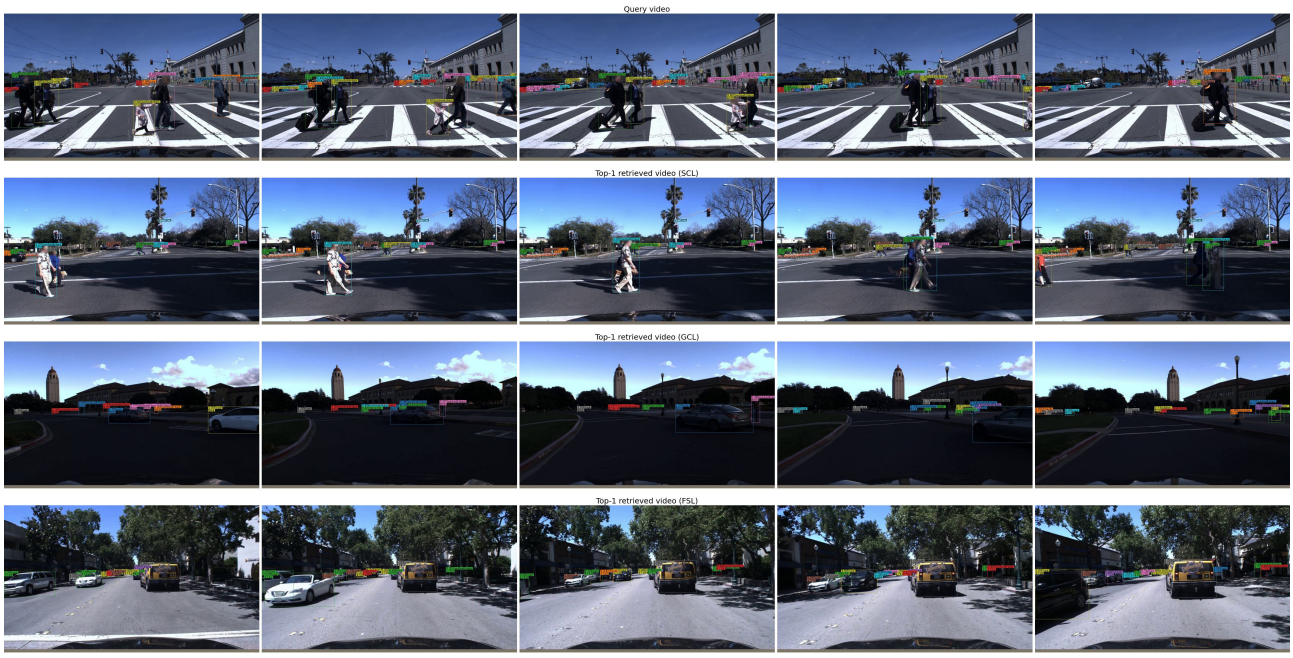

Figure 5.

Figure 6.

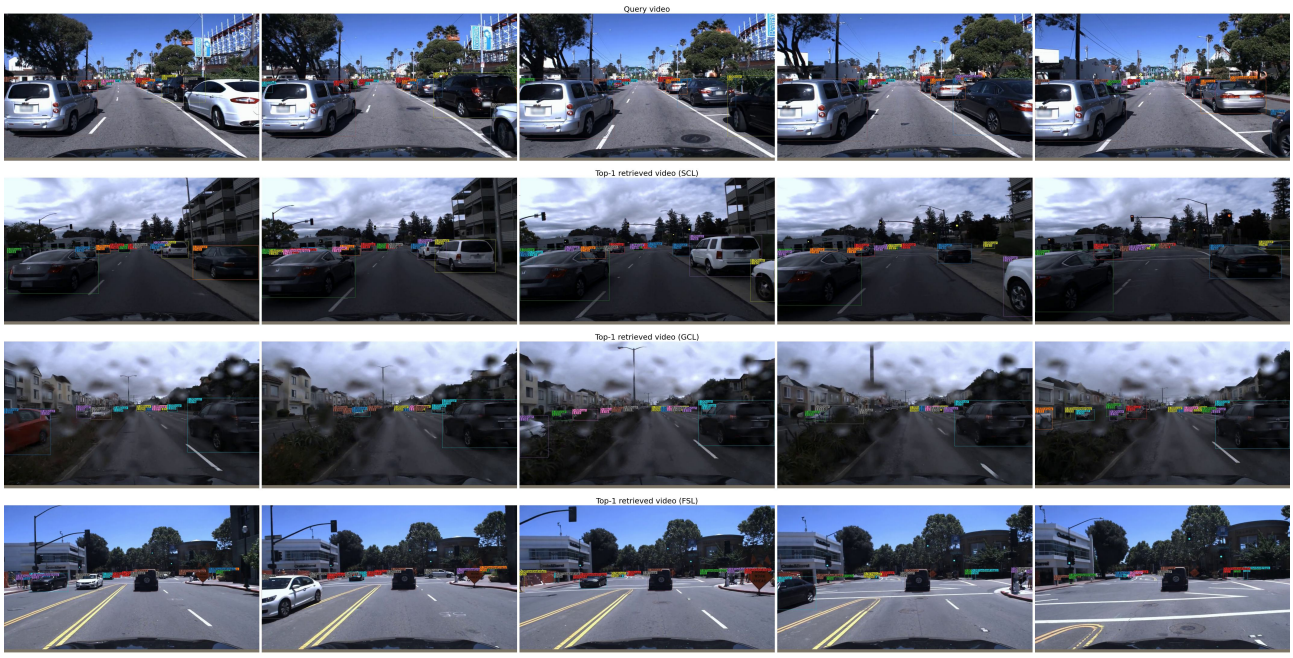

Figure 7.

Figure 8.

Figure 9.

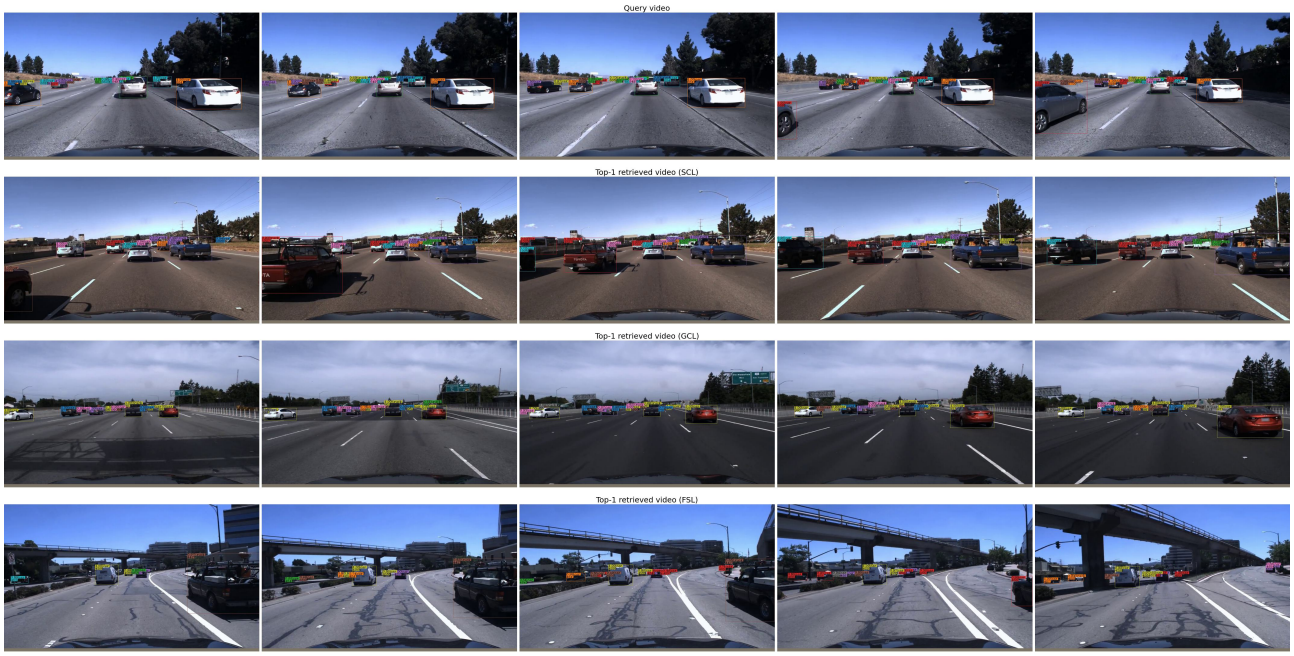

Figure 10.

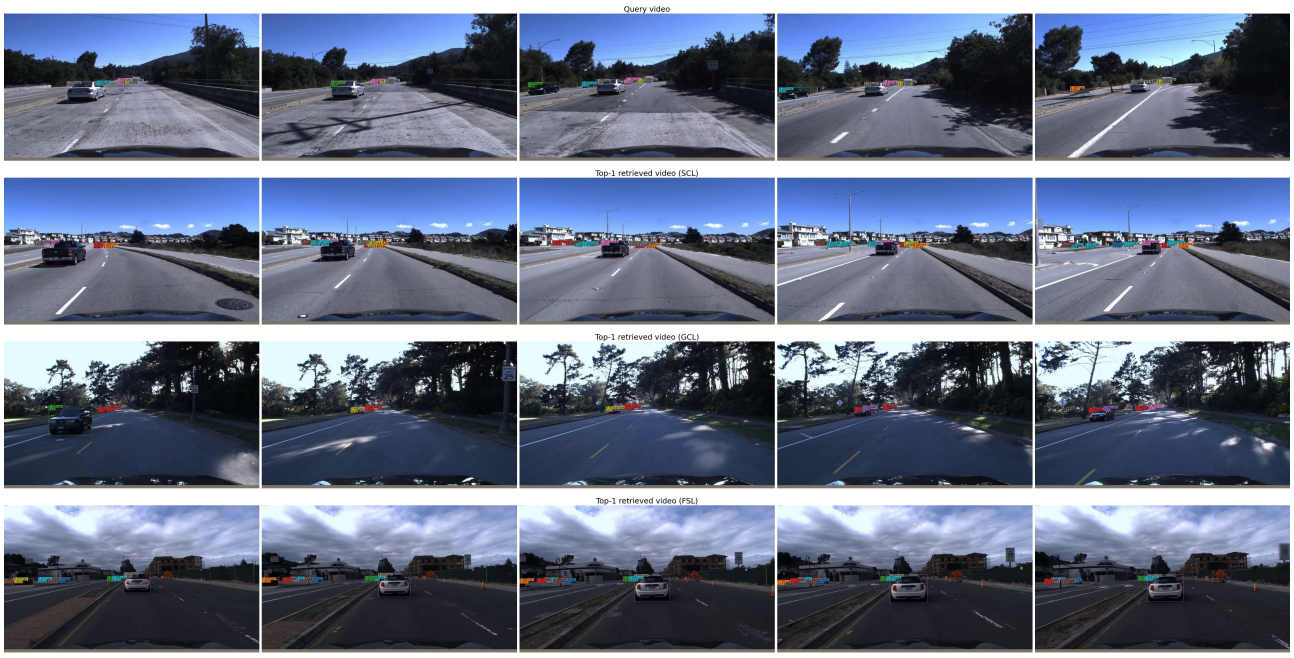

Figure 11.

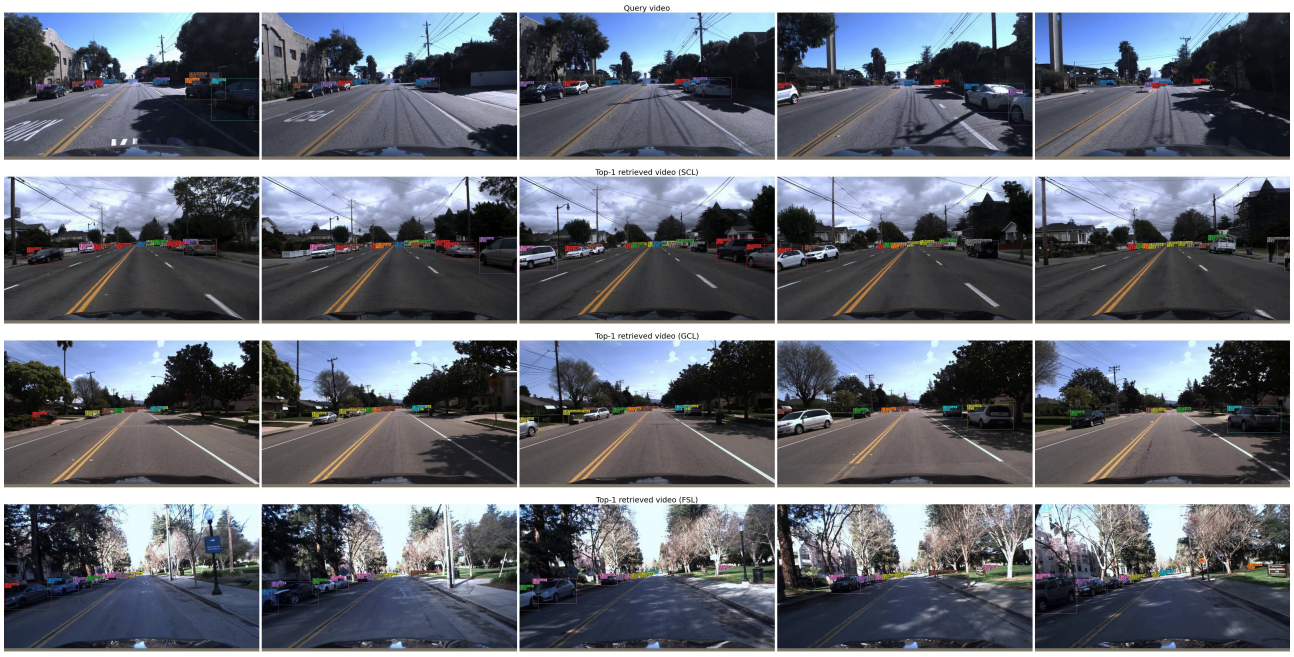

Figure 12.

Figure 13.

References

[1] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3835–3845. PMLR, 09–15 Jun 2019.

[2] Ekagra Ranjan, Soumya Sanyal, and Partha Pratim Talukdar. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. *ArXiv*, abs/1911.07979, 2020.