Driving Scene-Retrieval by Example from Large-Scale Data

Sascha Hornauer* saschaho@berkeley.edu
Baladitya Yellapragada* baladityay23@berkeley.edu
Arian Ranjbar† arian.ranjbar@berkeley.edu
Stella X. Yu* stellayu@berkeley.edu

International Computer Science Institute* / PATH†
University of California, Berkeley

Abstract

Many machine learning approaches train networks with input from large datasets to reach high task performance. Collected datasets, such as Berkeley Deep Drive Video (BDD-V) for autonomous driving, contain a large variety of scenes and hence features. However, depending on the task, subsets, containing certain features more densely, support training better than others. For example, training networks on tasks such as image segmentation, bounding box detection or tracking requires an ample amount of objects in the input data. When training a network to perform optical flow estimation from first-person video, over-proportionally many straight driving scenes in the training data may lower generalization to turns. Even though some scenes of the BDD-V dataset are labeled with scene, weather or time of day information, these may be too coarse to filter the dataset best for a particular training task. Furthermore, even defining an exhaustive list of good label-types is complicated as it requires choosing the most relevant concepts of the natural world for a task. Alternatively, we investigate how to use examples of desired data to retrieve more similar data from a large-scale dataset. Following the paradigm of "I know it when I see it", we present a deep learning approach to use driving examples for retrieving similar scenes from the BDD-V dataset. Our method leverages only automatically collected labels. We show how we can reliably vary time of the day or objects in our query examples and retrieve nearest neighbors from the dataset. Using this method, already collected data can be filtered to remove bias from a dataset, removing scenes regarded too redundant to train on.

1. Introduction

Neural networks need to capture visual, temporal and action aspects of our world to perform well on autonomous driving tasks. To that end, large datasets were created with many examples of expert driving to provide real world references. The BDD-V dataset consists of a very large amount of videos and automatically recorded kinematic information, crowd-sourced from dashcams behind the windshield of many drivers on the West- and East-coast of the United States. However, unlike databases as ImageNet, where images are labeled with their depicting object categories, the BDD data provides similarly rich annotations only for a subset. This motivates a question in driving dataset curation: *How to search through unlabeled data for specific scenes?* In the following, we answer a slightly modified question: *Given exemplary scenes that represent desired features, how to retrieve similar data from a very large, unannotated dataset?*

We compare two variations of example encoding to query for similar data: Single images and sequences of image-action pairs (which we refer to as *scenes*). While the former concept is similar to common image retrieval approaches, the latter includes past actions and camera images and can be thought of as observing bursts of driving behavior within a second.

The retrieval approach is based on work from [4] for unsupervised image classification. In supervised classification, extensive human labeling of data is necessary. However, labeling similar scenes requires the same understanding of *similarity* among human labelers. With our approach we make a network come up with a suitable concept of similarity to rank first-person driving query instances. As shown, retrieved nearest neighbors for query images are indeed similar according to concepts such as the number of objects, street architecture or time of the day. In Figures 1, 2, and 3, we qualitatively show image similarity by comparing queries and retrievals for driving scenes, showing exemplary scene configurations.

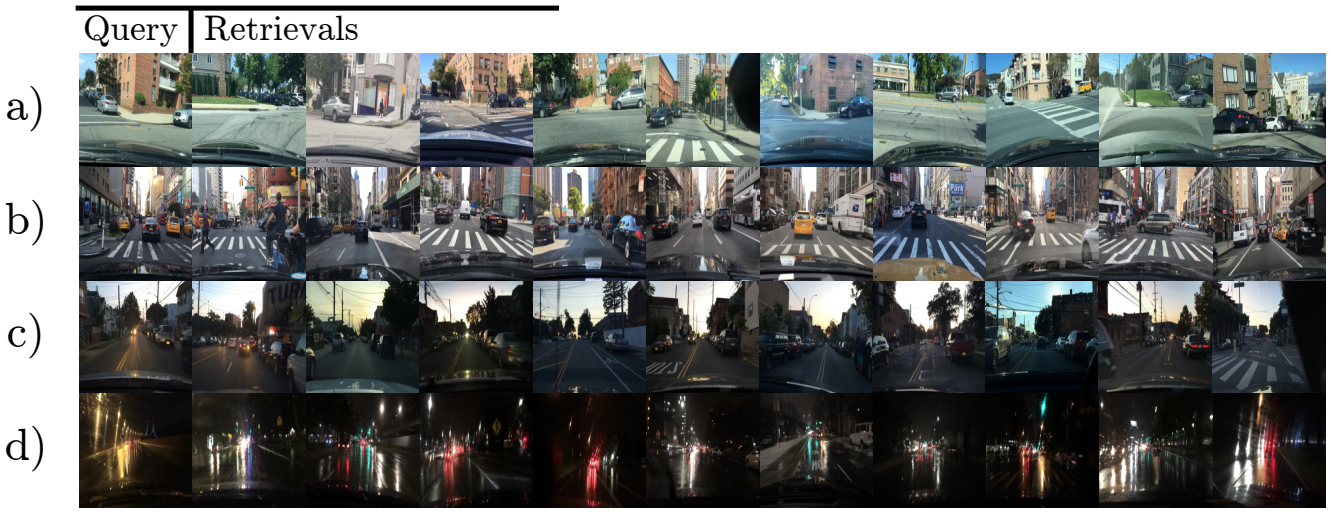

Figure 1: Retrievals of the network for a single query image. In row *a)*, sunny street corners are retrieved even though the street layout is more varied as when using scenes. In *b)* pedestrian crossings with many cars in the scene are reliably found. Also, the time of the day and weather fits in between the query and retrieval as shown in *c)* and *d)*

2. Related Work

Deep metric learning approaches have been used for recognition [3], re-identification and video categorization tasks in the past. Joonseok et al. [7] are related to our approach as they embed video features into a neighbourhood to preserve similarity.

A difference to our method is their metric, which regards videos similar if they were watched in the same session from the same user on YouTube. Other metric learning techniques process pairs or triplets to preserve similarities across samples semi-supervised. For example, parallel neural processing streams with shared weights evaluate pairs, and a final contrastive loss is either pushing pairs together or pulling them apart based on shared labels. This matches pairs from the same class but with different domain features (e.g., different lighting conditions or viewpoints), as shown by Bell and Bala [1].

Another similar technique relies on surrogate patch sampling for supervision, where the network treats the patches as surrogate classes to learn features [2].

Wu et al. [5] showed how an unsupervised instance-based classifier can perform object classification tasks. Their learned feature embedding maps novel images locally close to training images with the same label. They also leveraged some parametric calculations to infer efficiently enough for real-time computation.

3. Method

We train and compare networks to map images and image-action sequences (*scenes*) to nearest neighbors in the dataset. The results are hard to quantify given the unlabeled data, so we evaluate our retrieval approach by inspecting and showing many query-retrieval examples by hand.

3.1. Data

Our used BDD video-dataset [8] contains more than 1.8TB of first-person driving scenes in urban areas. Video sequences are labeled with accelerations, angular velocities and GPS information. These were processed by [6] into action vectors for the task of action prediction. Also, for 100k non-consecutive frames from different videos, images were annotated with labels such as weather, scene and time of day. A subset of 10k images contains further labels such as image segmentations and objects. We do not use any of these labels during training. We used the BDD 100k dataset for single-image queries and the full video dataset for *image-action*-sequences as queries. We adopt the action encoding of [6], which encodes the behavior of the vehicle.

3.2. Driving Scene Definition

Our scenes are defined as a number of past and future frames, relative to a time point t. Similar to Xu et al. [6], we pre-process BDD driving videos into approximately 40 second long chunks. We parse those in a sliding window fashion with window-size of 6 sampled frames, without overlap. To reduce redundant data and action correlation, we *hop* with 4-frame spacing in between consecutive frames. Driving scenes (s) are defined as a number $N = n \cdot 2, n \in \mathbb{N}$ of images x_i and action vectors a_i. Half of the actions lead up to the current point t in time:

$$s_j := (x_i, a_i), j \in \{0 \dots \frac{M}{N}\}$$
$$i \in \{t - (\tfrac{N}{2} - 1), t - (\tfrac{N}{2} - 2), \dots, t, \dots, t + (\tfrac{N}{2} - 1), t + (\tfrac{N}{2})\}$$

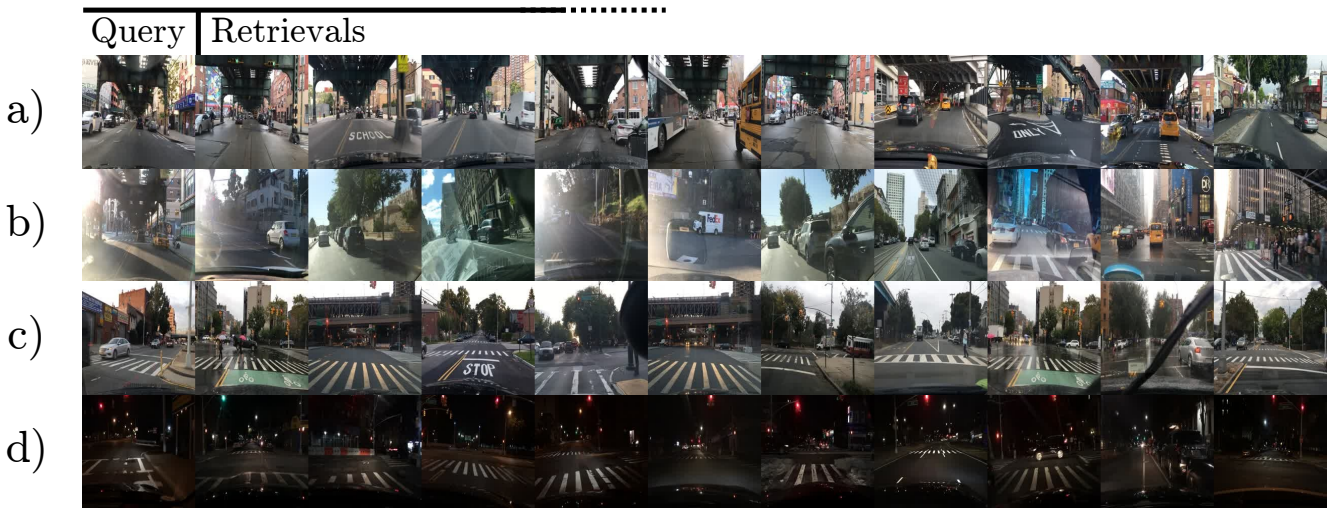

Figure 2: Retrievals of the network for visual-action scenes. *a)* Instances of overhead bridges are retrieved from the training set using query instances from the test set. *b)* Scenes with strong sunlight, creating glare effects, dominate the retrievals. *c)* Pedestrian crossings are retrieved at similar perceived angles and similar in *d)* at night.

Where M is the amount of samples retrieved from our sliding window sampling.

We train only on the past part of the a scene $s_j^p := (x_i, a_i), i \in \{t - (\frac{N}{2} - 1), \ldots, t\}$. During training, we observe the difference of actions of the query and the retrievals to monitor the training status.

3.3. Neighborhood Metric Learning

In the work of [4], image features are stored in a memory bank. From there, cosine distances between features on this high-dimensional sphere are computed efficiently to measure similarity. Query images (e.g., *lions*) are mapped to visually similar instances of related or same classes (e.g., big cats).

Similarly, we adopted this approach to train on the BDD 100k dataset of individual driving images. Furthermore, we extended the approach to work with our defined scenes (i.e., sequences of image-action pairs) on BDD-V. During training, to map every sample to its own ID, the network has to derive filters matching useful discriminatory features in the input space. This creates a network which maps samples into local neighborhoods with similar visual-action features, both spatially and temporally. An example of similarity would be all scenes driving on a straight road at night or on a highway in broad daylight.

Using a Resnet18 architecture on scenes s_j^p, we perform instance-based learning for every sample in the our training dataset (i.e., the network is trained to correctly predict the numerical sample ID of input s_j^p, according to a fixed enumeration). We use the same 128 dimensional feature vector and perform non-parametric Softmax classification as described in [5].

In order to match the standard input size of ResNet models, the input video frames within our scenes are resized to 224 $\times$ 224. For validation, we parse query scenes from a validation set and compare the retrieved top-K scenes from the training set. Due to our lack of ImageNet-like labels, we compare the ground truth future actions of all test scenes to the future action labels of the retrieved top-K training scenes. That way, we keep track of the best epoch, with respect to minimizing the difference of future actions. This indicator allows us to see convergence along both dimensions, action and visual. It improves training for visual similarity as actions and visual information during driving are heavily correlated. We also chose action labels as our basic performance metric because they are automatically generated while driving and allow us to avoid relying on any manually generated label. Correlation of future actions in scenes shows how future scene progression is similar. Since during driving, visual elements, e.g. a traffic light, a pedestrian crossing or an obstacle, determines the course of the future scene we see future action correlation as an indication of how the network learned to detect and map these concepts.

Actions in our approach are action probability vectors, which are calculated like in [6] from ground truth speed and course information at each time point. We have chosen the same possible actions, **go straight**, **stop or slow**, **turn left** or **turn right**.

Action probability vectors are concatenated with the second layer output of the Resnet-backbone, after the vectors are expanded to full output feature maps with a size of 28 $\times$ 28. This fusion adds only about 10K parameters to the model and overall, the backbone ResNet18 model, with action fusion, has about 11M parameters.

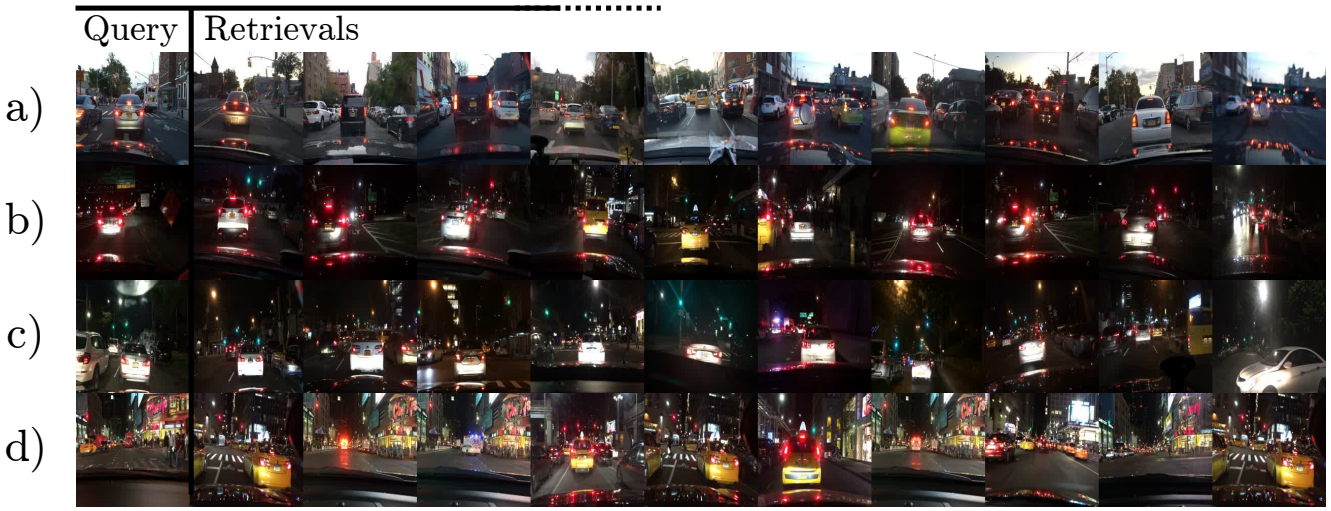

Figure 3: Further retrieval instances from the visual-action network. *a)* shows a combination of *cars* with illuminated *brake lights*, in the early *afternoon*. Some of these conditions can change independently such as in *b)*, where illuminated brake lights are retrieved at night and in *c)* where the reflection of the car paint is visually more salient than the weak taillights. In *d)* the same crossing, with the same sign of an American casual dining restaurant is retrieved several times. From the amount of cars in the scene it can be observed that retrievals come from a different point in time than the query but are most likely from the same video clip as an emergency vehicle is leaving the scene.

4. Results and Conclusion

The single-image network retrieves images, similar in time of the day, weather and matching in many objects, as can be seen in Figure 1. Nevertheless, some details such as pedestrian crossings, angle to the street or colors of head- or taillights do not match. Since single images are used for training and inference, no temporal information is available.

However, the information contained in a sequence of images can help separate features such as turns. When using a sequence of image-action pairs as input we see matching pedestrian crossings even at the right angle (figure 2) and in figure 3 it is possible to distinguish in between illuminated brake and tail lights.

In summary we show how our approach can query a large scale driving dataset for data, similar to a query sample. Fine grained control over the content of the retrieval is possible, without defined labels, by choosing a fitting query. This can be used to filter datasets in order to create training curriculums, better suited to a particular machine learning task.

References

[1] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *Siggraph*, 2015.

[2] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014.

[3] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[4] Zhirong Wu, Alexei A. Efros, and Stella X. Yu. Improving Generalization via Scalable Neighborhood Component Analysis. 2018.

[5] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. *CVPR*, 2018.

[6] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end Learning of Driving Models from Large-scale Video Datasets. pages 2174–2182, 2016.

[7] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286. IEEE, 6 2015.

[8] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. 2018.