

# Large-Scale Object Mining for Object Discovery from Unlabeled Video

Aljoša Ošep\*, Paul Voigtlaender\*, Jonathon Luiten, Stefan Breuers, Bastian Leibe

**Abstract**— This paper addresses the problem of object discovery from unlabeled driving videos captured in a realistic automotive setting. Identifying recurring object categories in such raw video streams is a very challenging problem. Not only do object candidates first have to be localized in the input images, but many interesting object categories occur relatively infrequently. Object discovery will therefore have to deal with the difficulties of operating in the long tail of the object distribution. We demonstrate the feasibility of performing fully automatic object discovery in such a setting by mining object tracks using a generic object tracker. In order to facilitate further research in object discovery, we release a collection of more than 360,000 automatically mined object tracks from 10+ hours of video data (560,000 frames). We use this dataset to evaluate the suitability of different feature representations and clustering strategies for object discovery.

## I. INTRODUCTION

Deep learning has revolutionized the way research is being performed in computer vision, and the success of this development holds great promise for important applications such as autonomous driving [19]. However, deep learning requires huge quantities of annotated training data, which are very costly to obtain. Consequently, progress has so far been limited to areas where such data is available, and community efforts such as PASCAL VOC [8], ImageNet [5], CalTech [6], KITTI [10], COCO [25], or Cityscapes [4] have been instrumental in enabling recent successes. It is largely thanks to those efforts that we nowadays have good object detectors (*e.g.*, [36], [35], [26]) at our disposal for a limited number of 20-80 object categories.

When moving from image interpretation tasks to video understanding problems, however, it becomes clear that the current strategy of using exhaustive human annotation will quickly become infeasible. This problem is of particular relevance in autonomous driving and mobile robotics, where future intelligent agents will have to deal with a large variety of driving scenarios involving a multitude of relevant object classes, many of which are not captured by today’s detectors (see Fig. 1). In this paper, we explore an automatic approach for discovering novel object categories (*i.e.*, categories for which we do not have detectors yet) by mining generic object tracks from large driving video collections.

Object category discovery has attracted a lot of attention from the research community recently, and many approaches have been proposed for discovering object categories from image collections [38], [37], [48], [18], [42], [16] or videos [21], [47]. However, many current discovery approaches are evaluated on carefully pre-processed datasets,

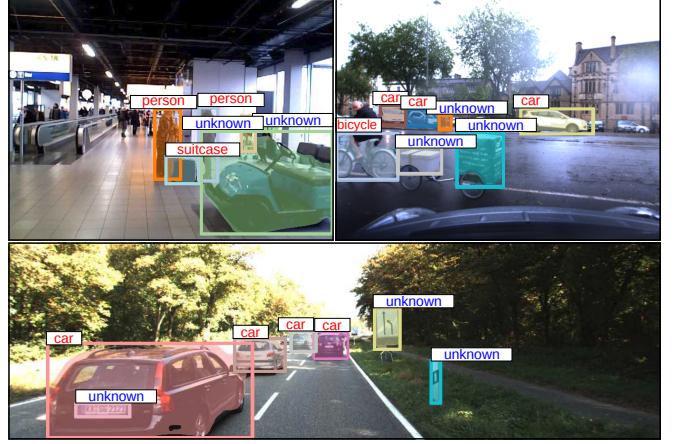


Fig. 1. We propose an approach for automatic discovery of novel and rare object categories from large video corpora. We start by mining generic object tracks (see above) and extract novel object categories by applying clustering using unknown object tracks.

such as MNIST, CIFAR, or subsets of ImageNet [18], [42], where each image contains an object of interest and the number of categories is *a-priori* known. We argue that such a setting is very different from real-world scenarios, where a major aspect of the difficulty of object discovery will be to deal with the long tail of the object distribution. In any practical application scenario, we can expect the frequency of object category observations to follow a power law distribution, with some object categories occurring very frequently and the vast majority being increasingly rare. Thus, even if every training example shows a potential object of interest, many rare object categories will not accumulate enough instances to allow clustering approaches to easily pick them out from the background noise. In order to make progress on this important topic, it therefore becomes important to focus evaluation on more realistic settings.

At the core of any object discovery approach is the question what constitutes an object. A common approach is to define object regions by a consistent appearance that separates the region from the surrounding background [1]. In the literature, this definition has been adopted by region proposal networks [36] that have been used successfully for object detection [36], [13] and object discovery in internet images [37]. However, in our experience, such region proposals are not stable and not distinctive enough to permit generic object tracking in real street scenes. We therefore adopt a farther-reaching definition of generic objects as *regions that have well defined and temporally consistent boundaries in 3D space*. Further, as *known* objects, we consider those for which we have a pre-trained detector available (*i.e.*, the 80 annotated object categories in COCO [25]), all the rest we

\* Equal contribution. The authors are with the Visual Computing Institute, RWTH Aachen University. E-mail: lastname@vision.rwth-aachen.de

consider to be *unknown* objects.

In summary, we present a large-scale study for object mining and category discovery on two large datasets (KITTI Raw [9] and Oxford RobotCar [27]) for autonomous driving, comprising altogether roughly 10 hours of video data consisting of more than 560,000 frames. From this data, we extract more than 360,000 object tracks using a fully automatic generic object tracking pipeline. As verified in our experiments, although the object tracks are extracted without human supervision and comprise both *known* and *unknown* object categories, less than 10% of them are affected by tracking errors. Thus, they can serve as a stable basis for object discovery experiments. We use this dataset to evaluate the suitability of different feature representations and clustering strategies for object discovery. To the best of our knowledge, this is the first time such a large-scale generic object mining effort has been undertaken in an automotive scenario. We make our code, datasets, and annotations publicly available to serve as a benchmark for the research community<sup>1</sup>.

## II. RELATED WORK

**Object Discovery.** Object discovery denotes the problem of identifying previously unseen object categories without human supervision. Russell *et al.* [38] propose a vision-based method that uses multiple object segmentations in order to group visually similar objects and their segmentations. Sivic *et al.* [40] propose a method for discovery of hierarchical structure of objects from unlabeled images. Lee and Grauman [23] propose an iterative procedure that starts with easy-to-discover instances and progressively expands to more challenging cases and demonstrates that recognition in the form of a region classifier helps with the discovery by narrowing down object candidates [22]. We similarly utilize multiple object hypotheses in the form of tracklet proposals and utilize a classifier that assigns semantic information to tracklets. Rubinstein *et al.* [37] propose to identify potential objects in Internet images using saliency to find reoccurring patterns between images using dense correspondences. For a more detailed overview of existing image-based methods we refer to [48]. Kwak *et al.* [21] propose a method for joint tracking and object discovery in videos. Their method localizes and tracks the dominant object based on motion and saliency cues in each video. A similar idea is applied in Tsai *et al.* [47] for semantic co-segmentation in videos. Both methods demonstrate excellent results on the YouTube-Objects dataset [34]. However, these video sequences are usually dominated by a single object and cover only a limited number (10) of categories.

In the field of mobile robotics, [7], [46], [14] propose methods in which RGB-D scans are segmented into object candidates. These candidates are then grouped using either clustering methods or based on probabilistic inference. However, all of these methods were only applied to simple indoor scenarios, containing well-separated objects such as boxes

and chairs. In [55], [30] object discovery in traffic scenarios using LiDAR sensors is addressed. While for clean LiDAR data even simple methods can be used to segment scans into meaningful regions, obtaining object candidates from image data is far more challenging [17], [56], [1], [3], [31].

**Clustering and Embedding Learning.** Clustering is typically used to find patterns in unlabeled data by grouping data points by their similarity. Here the main challenge is defining similarities or distance measures between the data points. Recent methods approach this problem by learning distance metrics [52], [39], [41]. In order to adapt learned embeddings to a specific domain, [53] proposes to iteratively cluster data and re-learn embeddings. Hsu *et al.* [18] propose a method that use a separate, labeled dataset to learn a Similarity Prediction Network (SPN), which is then used to produce binary labels for each pair of objects of an unlabeled dataset. These labels are used to train ClusterNet, which directly predicts cluster labels. In contrast to the above-mentioned methods, we work with raw image data, where object localization is not given and we do not make any assumptions about object categories.

**Video-Object Mining.** Video-Object mining (VOM) refers to a task of collecting frequently-occurring patterns (*i.e.* object candidates) from video or streams of sensory recordings in general. Teichman *et al.* [45] propose a method for tracking-based semi-supervised learning by mining LiDAR streams, captured from a vehicle. Similarly, [28], [29] propose tracking-based semi-supervised learning based on video. Furthermore, in the context of vision, VOM has been used for improving object detectors by mining hard-negatives for specific object categories from web-videos [44], [20] and for learning new detectors for objects by localizing dominant video tubes in YouTube videos [34].

## III. METHOD

Towards the goal of object discovery using unlabeled video sequences, we first need to be able to obtain potential object candidates in these video streams. There are large amounts of unlabeled video data available [27], but finding new patterns in such data is challenging, as state-of-the-art object proposal methods such as Sharpmask [33] need to produce 100-1000 proposals per frame to achieve a high recall. This would result in a very large set of object candidates on the level of an entire video.

We propose to leverage temporal information and prior knowledge about common object categories. By forming object tracks from image-level object proposals (Fig. 2), we i) reduce the object candidate space considerably and ii) suppress noise and clutter in image-level object proposals, as these are typically unable to form stable object tracks. Recognition of common object types additionally helps reducing the proposal space and helps to suppress noise (see Tab. I). These object candidates form our track collection.

### A. Object Track Mining

For tracking, we build upon our recent work and utilize our category-agnostic multi-object tracker (CAMOT) [32]. In

<sup>1</sup>Project website: <https://vision.rwth-aachen.de/page/lsm>

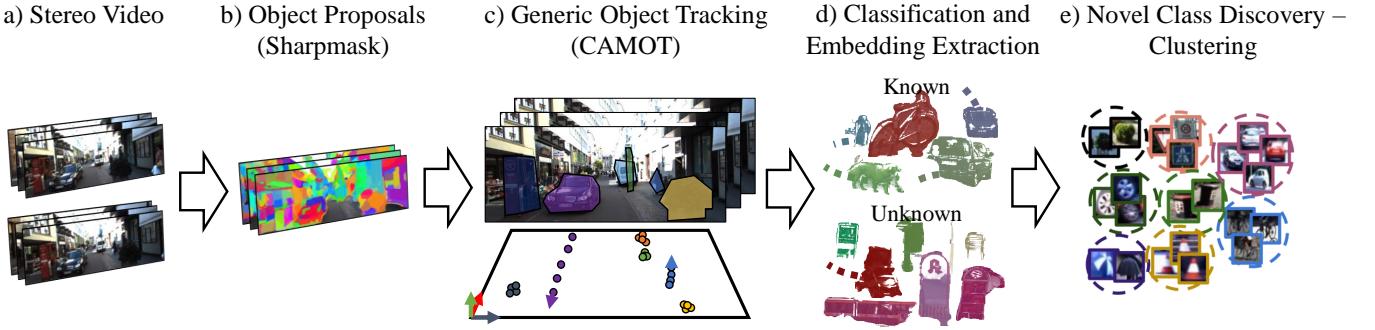


Fig. 2. Our proposed method processes large amounts of stereo video data (a) using a generic object tracker (b-c). We compute track embedding vectors (d) that allow us to perform clustering efficiently in Euclidean space using standard clustering approaches, such as (H)DBSCAN or KMeans (e). This way, we can discover novel object categories among previously unknown (non-recognized) object tracks.

a nutshell, using this tracker object candidates are obtained as follows (see Fig. 2). The tracker takes as input stereo images and frame-level mask proposals from Sharpmask [33]. CAMOT then uses these proposals in order to create a set of category-agnostic tracks. Afterwards, the classifier component of a Faster R-CNN [36] based detector (trained on the COCO dataset) is used to classify the tracks on an image crop level.

Tracks are thus automatically labeled by the recognized category type (*i.e.*, as one of the COCO [25] categories) or as *unknown* object track. Finally, for each frame, a mutually consistent subset of tracks is picked by performing MAP inference using a conditional random field (CRF) model (for details, see [32]). This way, we obtain a reduced set of object tracks. Tracks that are labeled as *unknown* are considered object candidates and are used for object discovery.

**Track Postprocessing.** After applying the tracker, we obtain a large collection of selected tracklets of both *known* and *unknown* categories. Since model selection is performed on a per-frame basis, one object may be split into several short tracklets. As a final postprocessing step extending our tracker [32] we progressively merge short selected tracklets into final tracks. In each frame, either a) existing tracklet  $h_i$  is re-selected and trivially continues an existing track  $H_k$ , or b) tracklet  $h_i$  is not re-selected and its track is continued by another selected tracklet  $h_j$  if they have a sufficient overlap. If  $h_i$  and  $h_j$  do not have sufficient overlap,  $H_k$  is terminated and  $h_j$  starts a new track. As an overlap criterion, we use the fraction of matching masks to the length of the shorter tracklet:

$$\lambda(h_i, h_j) = \frac{|\{t | IoU(h_i^t, h_j^t) > \gamma\}|}{\min(|h_i|, |h_j|)}. \quad (1)$$

Here, two masks are considered to be a match when mask IoU is higher than a threshold  $\gamma$  in frame  $t$ .

**Video Mining.** We applied the tracker on two publicly available datasets, KITTI Raw [9] and Oxford RobotCar [27]. For both we use stereo for estimating depth [11]. Compared to the original CAMOT [32], we replace the dense scene flow [49] by a sparse scene flow [24] for initialization. Sparse scene flow is less accurate, but has a lower processing time by several orders of magnitude. For egomotion estimation we use the visual odometry method by [12].

We perform track mining using a computer cluster by processing chunks of 500 frames. This way, object mining of large datasets can be processed efficiently in parallel in a matter of hours. In particular, a dataset containing 9 h of video and 521,500 frames can be processed in 5-24 hours using 1043 computing nodes. The total runtime depends on the tracking parameters and the number of proposals per frame used for tracking. In our experiments, we input the top-100 proposals to the tracker in each video frame.

### B. Object Discovery via Clustering

After running the tracker, we obtain a reduced set of object tracks, each of which is either classified as one of the COCO [25] categories or marked as *unknown*. We aim to find patterns using the *unknown* set of tracks via clustering. This is a challenging problem: i) we are dealing with large amounts of data, ii) the mined tracks will always contain outliers and occasionally imprecise localization of objects and iii) novel objects appear rarely (*i.e.* they appear only in the long tail of the category distribution, see Sec. IV-A).

We consider several possibilities of how to tackle this problem. Many clustering methods work in two steps. First a suitable feature representation is generated, and then the clustering is performed using these features by one of the standard clustering algorithms, such as KMeans or DBSCAN. Recently, clustering has also been tackled in an end-to-end fashion using deep learning [18], [16]. In the following, we will describe the methods which we utilized for either extracting features or directly performing clustering.

**Extracting Features from a Pre-trained Network.** A simple method is to utilize a pre-trained network to extract features from its internal activations and optionally reduce their dimensionality. Since our aim is to cluster small crops of objects, a pre-trained object detector is well suited.

**Learning an Embedding on a Labeled Dataset.** Another possibility to obtain features is to make use of the recent advances in the area of feature embedding learning [50], [39], [41]. The idea here is to use a labeled dataset to learn a feature embedding in which images of the same class have a small distance and images of different classes are far away.

A weakness of both approaches is that the source domain, on which the network or embedding is trained might differ from the target domain. Alternatively, one could pre-train an

	KTC	OTC
Frames	42,407	521,500
Duration (h)	1.18	9.06
Proposals (total)	4,240,700	52,150,000
Tracks (total)	8,005	359,503
Tracks (labeled)	8,005	12,308
Tracks (unknown)	1,190	4,198
Tracking Errors	745	787

TABLE I

STATISTICS OF TRACK MINING FROM UNLABELED VIDEOS. WE ACHIEVE A SIGNIFICANT REDUCTION OF THE PROPOSALS USING TRACKING.

embedding on a different domain and iterate between clustering and re-learning the embedding on the target domain using the obtained clusterings [53]. However, such an approach is slow and does not scale well to large amounts of data.

**Clustering Algorithm.** To perform clustering using a feature embedding, we propose to use the recent, hierarchical density-based clustering algorithm HDBSCAN [2] due to its scalability to large datasets and its inherent ability to deal with outliers in the data. We show in Sec. IV-B that this approach outperforms simpler alternatives. As a distance measure between tracked objects, we use the Euclidean distance in the learned embedding space.

**Track Similarity Measure.** One of the central questions in clustering is how to define a distance measure between data points, in our case, object tracks. Object tracks are defined by a collection of image crops, representing the appearance of the tracked object over time. When applying the embedding network on tracks, we first extract a representative embedding vector for each track. We take the embedding vector of the crop that is closest to the mean of the embedding vectors of the track’s image crops. This proved to be more robust than simply taking the mean. After clustering, the resulting cluster label is transferred to the whole track.

**End-to-end Clustering.** Recently, Hsu *et al.* [18] proposed ClusterNet, a scalable end-to-end clustering solution based on deep learning. They propose to train a Similarity Prediction Network (SPN), that is used to produce binary labels (same / different category) for image pairs. These labels are then used to train the actual clustering network (ClusterNet) on the unlabeled target data using a softmax output layer with a fixed number of classes corresponding to cluster labels. We perform an evaluation of ClusterNet trained on our data.

#### IV. EXPERIMENTAL EVALUATION

The KITTI Raw [9] dataset was recorded in street scenes from a moving vehicle in Karlsruhe, Germany. For our experiments we only use the stereo cameras and a subset of 1.18 h (42,407 frames) of video data. The Oxford RobotCar dataset [27] has a similar setup as KITTI and it has been collected from a mobile vehicle in street scenes, mainly in the inner city of Oxford, UK. In our experiments, we only used the stereo setup. In total 1,000 km have been recorded over 1 year, from which we use a representative subset of 9 h of video (521,500 frames).

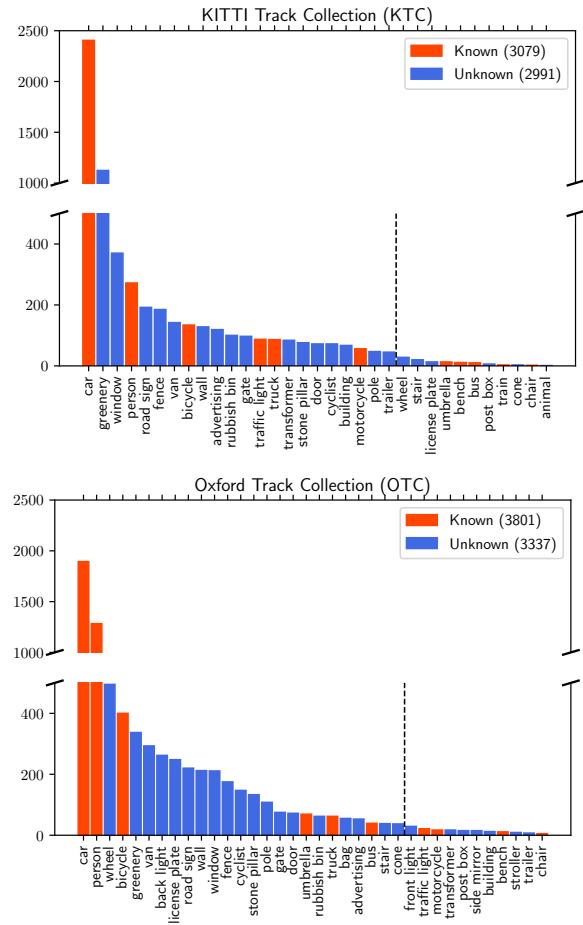


Fig. 3. Object category distributions in KITTI Track Collection (*top*) and Oxford Track Collection (*bottom*). As can be seen, the majority of the object categories appear in the long tail of the category distribution, rendering discovery of novel objects a challenging problem. The dashed line marks a cutoff, object categories beyond that are extremely rare (less than 30 instances) and are therefore excluded from the object discovery evaluation.

#### A. Video-Object Mining

In this subsection, we describe and analyze the tracks we mined from the Oxford and KITTI Raw datasets. We input 100 mask proposals to the object tracker per frame, of which  $\sim 85$  pass the geometric consistency checks in a typical inner-city sequence. The tracker internally maintains on average  $\sim 97$  tracklet proposals per frame, of which  $\sim 13$  are selected as most prominent object candidates. Tab. I displays a short summary of the track mining for both datasets and Fig. 7 and Fig. 8 show qualitative tracking results, obtained on KITTI Raw and Oxford RobotCar datasets, respectively. Even state-of-the-art object proposal approaches require an extremely large number of object candidates to achieve high recall for such sequences, rendering direct object discovery from proposals infeasible. Using tracking, we are able to reduce the number of object hypotheses to a manageable level and achieve a significant compression factor per image (*i.e.*, from 100 mask proposals per image to  $\sim 13$  object tracks), and an even greater compression factor on the sequence level.

For the purpose of a detailed analysis of tracks and clustering evaluation, we manually annotate all 8,005 tracks mined on the KITTI Raw dataset and a subset of 12,308

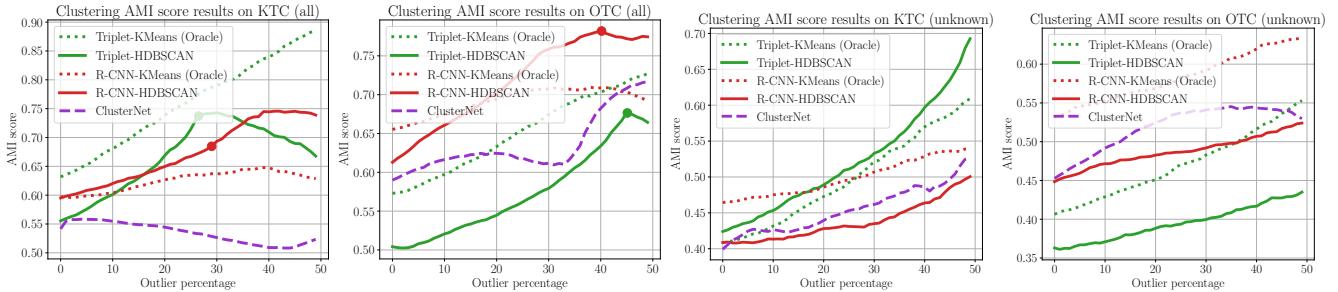


Fig. 4. Clustering results measured by AMI (all objects). Circle markers represent the automatically selected fraction of outliers by HDBSCAN. “Oracle” means that the ground truth number of clusters is used as  $k$ .

tracks, mined from Oxford RobotCar dataset. Thus, we obtain the KITTI Track Collection (KTC) and Oxford Track Collection (OTC), which we make publicly available in order to facilitate further research in the area of object discovery in automotive scenarios. These annotations have only been used for evaluation of the clusterings. We label each track as one of 36 categories (33 in case of KTC) which we manually identified in the tracks. Tracks that diverge from the tracked object are marked as a tracking error. When the tracked object was recognized as a valid object but does not fit into any of the 36 classes, it was labeled as a valid *unknown* object. Both the erroneous tracks and *unknown* tracks are excluded for the object discovery evaluation.

As can be seen in Fig. 3, the largest annotated categories in KTC are *car*, *greenery*, *window*, and *person* with 2,405, 1,124, 370, and 272 instances, respectively. A tracking error only occurred in 745 tracks (9.3%) which demonstrates the robustness of the tracker. In OTC, the largest annotated categories are *car*, *person*, *wheel*, and *bicycle* with 1,894, 1,283, 495, and 400 instances, respectively. Tracking errors occur in only 787 (2.2%) tracks. Some of the categories which are not annotated in COCO are *van*, *trailer*, and *rubbish bin*, for which we obtained 142, 45, and 100 instances in KTC, respectively. This demonstrates that the tracker can deliver tracks for interesting previously unseen categories, but the amount of data from the smaller KITTI Raw (1.18 h) might not yet be sufficient for discovering rare categories via clustering.

## B. Object Discovery

We evaluate the quality of the object discovery via clustering using the adjusted mutual information (AMI) criterion, which is a standard measure for assessing clustering performance. It measures how well the obtained clustering fits the ground truth classes. Since the tracks contain noise, we allow the clustering algorithm to mark tracks as outliers. We then measure the performance as a function of the allowed fraction of outliers which are excluded from the evaluation.

We compare one end-to-end trained method (ClusterNet) and a “standard” clustering pipeline, that utilizes trained embeddings in combination with KMeans and HDBSCAN. When running KMeans, we set the number of clusters to the ground truth number of classes to provide an upper bound on the achievable performance with KMeans. The outliers are selected based on the distance to the cluster centers. In the

following, we describe the details of each considered setup for clustering.

1) *Learned Triplet Embedding*: We train a feature embedding network on the COCO dataset [25]. We apply a triplet loss [50] to learn an embedding space with a dimensionality of 128, in which crops of different classes are separated and crops of the same class are grouped together. To this end, we adopt the batch-hard triplet mining and the soft-plus margin formulation of [15]. We trained the network to discriminate between the 80 object classes in the COCO dataset.

2) *Last Layer of Faster R-CNN Detector*: We use the activations of the last layer before the classification layer of the Faster R-CNN based detector with an Inception-ResNet-v2 [43] backbone which is also used in the tracker. For efficiency, we reduced their dimensionality from 1,536 to 50 using PCA, and found that the results are not very sensitive to the exact choice of dimensionality.

3) *ClusterNet*: In order to assess the performance of ClusterNet [18] on our data, we trained a Similarity Prediction Network (SPN) [54] as a Siamese network with a two-class softmax. We trained the SPN on COCO to predict whether two input crops belong to the same class. We then used the SPN output to train a ClusterNet with 50 cluster labels in the output layer on the tracks to directly predict cluster labels.

For our implementation of the SPN, ClusterNet, and for the triplet embedding network, we used a wide ResNet variant with 38 hidden layers [51] pre-trained on ImageNet [5] as base architecture. The crops which either come from the COCO ground truth or from tracks, were resized bilinearly to  $128 \times 128$  pixels before they were given into the networks.

On KTC, all 8,005 tracks which we use for clustering are labeled by us. On OTC, the clustering is performed on 359,503 tracks, and the evaluation is done on the 12,308 tracks which we labeled. For evaluation, the tracks labeled as *unknown*, tracking error, or with less than 30 labeled instances are excluded. Fig. 4 shows the quantitative results of the clustering evaluation on KTC and OTC. We provide a separate evaluation for i) considering all annotated ground truth categories (Fig. 4 left), and ii) only for the categories which are not in COCO (Fig. 4 right). Table II shows the results of each of our methods evaluated at zero outlier percentage for both all objects and unknown objects not in the COCO training data. Note that for KMeans we always set  $K$  to correspond to the ground truth number of categories while HDBSCAN estimates the number of clusters automatically. For the task of object discovery, it is important to evaluate



Fig. 5. Visualization of the clustering results on OTC using R-CNN features and HDBSCAN. The numbers on the right hand side indicate the number of tracks in each cluster. The cluster labels were assigned by hand. Newly discovered categories are marked in red.

clustering not just on all tracks but also on the unknown tracks only, because the track collection is dominated by known categories.

## V. DISCUSSION

When evaluating different object discovery pipelines, we found that the last layer activations of a Fast R-CNN detector are surprisingly effective as a feature representation for clustering, outperforming the learned embedding dataset and ClusterNet. This is not only the case when clustering *known* object categories. These features achieve good performance also when clustering only *unknown* tracks, which suggests that they generalize well and are very well suited for clustering tasks. ClusterNet performs poorly when only a small amount of data is available but significantly improves when increasing the number of tracks, showing great potential for clustering of large amounts of unlabeled data. KMeans

	All		Unknown	
	KTC	OTC	KTC	OTC
Triplet-KMeans (Oracle)	<b>0.63</b>	0.58	0.40	0.41
Triplet-HDBSCAN	0.55	0.51	0.43	0.36
R-CNN-KMeans (Oracle)	0.60	<b>0.65</b>	<b>0.47</b>	<b>0.53</b>
R-CNN-HDBSCAN	0.60	0.62	0.41	0.45
ClusterNet	0.54	0.59	0.40	0.45

TABLE II

RESULTS OF EACH OF OUR METHODS EVALUATED AT ZERO OUTLIER PERCENTAGE FOR BOTH ALL OBJECTS AND UNKNOWN OBJECTS NOT IN THE COCO TRAINING DATA.



Fig. 6. Visualization of the clustering results on KTC using R-CNN features and HDBSCAN. The numbers on the right hand side indicate the number of tracks in each cluster. The cluster labels were assigned by hand. Newly discovered categories are marked in red.

(Oracle) is often the best-performing method, but here we use the ground truth number of categories as  $k$ , which is unrealistic in practice. HDBSCAN performs very well, often on par with KMeans (Oracle).

We show qualitative results of all clusters with a size of at least 80 tracks of the obtained clustering on Oxford using R-CNN features and HDBSCAN in Fig. 5. As can be seen, we obtain clusters for several object types, that are not present in the COCO dataset (highlighted in red in the figure): *wheel*, *window*, *greenery*, *number plate*, *rubbish bin*, *car window*, *road sign*, *car back light*, *cone*, *sign bollard*, *direction sign*, *bollard*, and *heads*. On KTC (Fig. 6) we identify the following novel object categories: *greenery*, *rubbish bin*, *road sign*, *traffic light pole*, *transformer*, and *striped sign*.

## VI. CONCLUSION

This work is an initial study about object discovery from unlabeled video by automatically extracting generic object tracks. We showed that it is indeed possible to automatically discover previously unseen categories through clustering. We collected over 350,000 object tracks and manually labeled



Fig. 7. Qualitative tracking results on the KITTI Raw [10] dataset. Beside tracked objects, recognized by the classifier, we also find new objects such as various traffic signs, car trailers, advertisements, poles, caterpillar machines, post boxes, *etc.*



Fig. 8. Qualitative tracking results on the Oxford RobotCar [27] dataset. Beside tracked objects, recognized by the classifier, we also find new objects such as various traffic signs, traffic cones, advertisements, poles, street cleaners, *etc.*

over 18,000 of them in order to facilitate further research in the area of object discovery. We believe that this work is a starting point and there is still a large potential for further exploiting such unlabeled data. For example, the automatically clustered tracks could be used to fully-automatically train object detectors for the newly discovered categories.

**Acknowledgements:** We would like to thank Bin Huang and Michael Krause for annotation work. This project was funded, in parts, by ERC Consolidator Grant DeeVise (ERC-2017-COG-773161). The experiments were performed with computing resources granted by RWTH Aachen University under project rwth0275.

## REFERENCES

- [1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11):2189–2202, 2012.
- [2] R. J. G. B. Campello, D. Moulavi, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *TKDD*, 10(1):5:1–5:51, 2015.
- [3] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals for accurate object class detection. In *NIPS*, 2015.
- [4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [6] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(4):743–761, 2012.
- [7] F. Endres, C. Plagemann, C. Stachniss, and W. Burgard. Unsupervised discovery of object classes from range data using latent dirichlet allocation. In *RSS*, 2009.
- [8] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.
- [9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013.
- [10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.
- [11] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010.
- [12] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D reconstruction in real-time. In *Intel. Vehicles Symp.*, 2011.
- [13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.
- [14] E. Herbst, X. Ren, and D. Fox. RGB-D object discovery via multi-scene analysis. In *IROS*, 2011.
- [15] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [16] S. Hickson, A. Angelova, I. Essa, and R. Sukthankar. Object category learning and retrieval with weak supervision. In *NIPS Workshops*, 2017.
- [17] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.
- [18] Y. Hsu, Z. Lv, and Z. Kira. Deep image category discovery using a transferred similarity function. *arXiv preprint arXiv:1612.01253*, 2016.
- [19] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.
- [20] S. Jin, A. RoyChowdhury, H. Jiang, A. Singh, A. Prasad, D. Chakraborty, and E. Learned-Miller. Unsupervised hard example mining from videos for improved object detection. In *European Conference on Computer Vision (ECCV)*, 2018.
- [21] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015.
- [22] Y. J. Lee and K. Grauman. Object-graphs for context-aware visual category discovery. In *CVPR*, 2010.
- [23] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, 2011.
- [24] P. Lenz, J. Ziegler, A. Geiger, and M. Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *Intel. Vehicles Symp.*, 2011.
- [25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.
- [26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg. SSD: Single Shot Multibox Detector. In *ECCV*, 2016.
- [27] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 year, 1000km: The Oxford RobotCar dataset. *IJRR*, 36(1):3–15, 2017.
- [28] I. Misra, A. Shrivastava, and M. Hebert. Watch and learn: Semi-supervised learning of object detectors from videos. In *CVPR*, 2015.
- [29] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and Learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.
- [30] F. Moosmann and M. Sauerland. Unsupervised discovery of object classes in 3d outdoor scenarios. In *ICCV Workshops*, 2011.
- [31] A. Osep, A. Hermans, F. Engelmann, D. Klostermann, M. Mathias, and B. Leibe. Multi-scale object candidates for generic object tracking in street scenes. In *ICRA*, 2016.
- [32] A. Osep, W. Mehner, P. Voigtlaender, and B. Leibe. Track, then decide: Category-agnostic vision-based multi-object tracking. *ICRA*, 2018.
- [33] P. Pinheiro, T. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.
- [34] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
- [36] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [37] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.
- [38] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.
- [39] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.
- [40] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.
- [41] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Deep metric learning via facility location. In *CVPR*, 2017.
- [42] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.
- [43] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.
- [44] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, 2012.
- [45] A. Teichman and S. Thrun. Tracking-based semi-supervised learning. *IJRR*, 31(7):804–818, 2012.
- [46] R. Triebel, J. Shin, and R. Siegwart. Segmentation and unsupervised part-based discovery of repetitive objects. In *RSS*, 2010.
- [47] Y.-H. Tsai, G. Zhong, and M.-H. Yang. Semantic co-segmentation in videos. In *ECCV*, 2016.
- [48] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV*, 88:284–302, 2010.
- [49] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In *ICCV*, 2013.
- [50] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10:207–244, 2009.
- [51] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.
- [52] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. *ICML*, 48:478–487, 2016.
- [53] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016.
- [54] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.
- [55] Q. Zhang, X. Song, X. Shao, H. Zhao, and R. Shibasaki. Unsupervised 3d category discovery and point labeling from a large urban environment. In *ICRA*, 2013.
- [56] C. L. Zitnick and P. Dollár. Edge Boxes: Locating object proposals from edges. In *ECCV*, 2014.