Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation

Philipp Oberdiek[1], Matthias Rottmann[2] and Gernot A. Fink[1]

[1]Department of Computer Science, TU Dortmund University
[2]School of Mathematics and Natural Sciences, University of Wuppertal

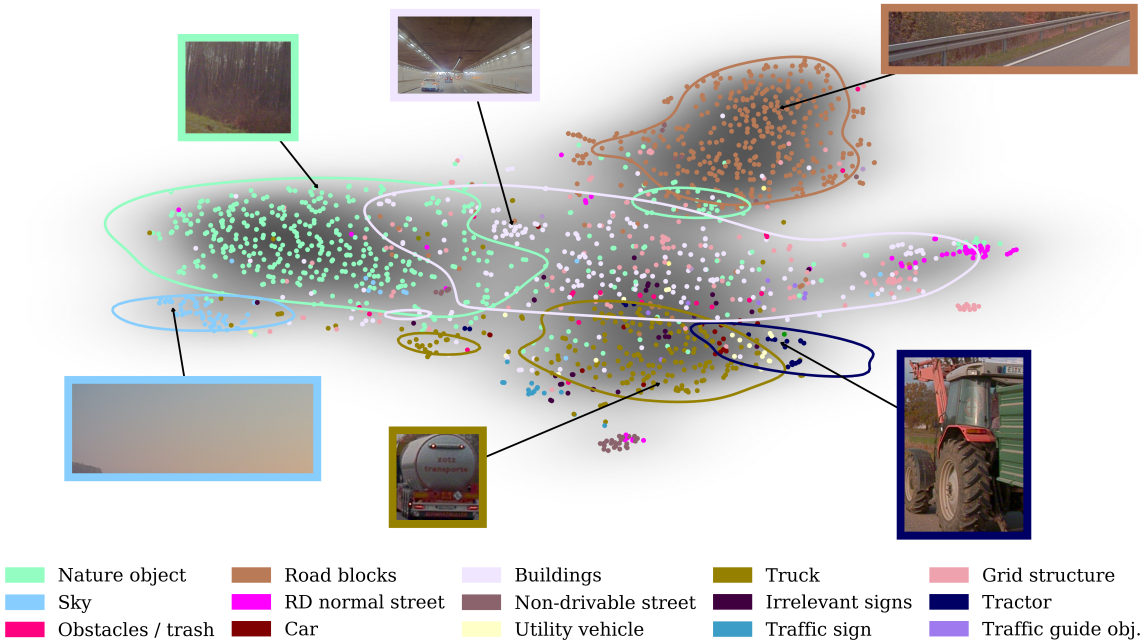

Figure 1: Embedding space of unknown and badly segmented objects based on ResNet152 features. Darker regions symbolize higher global density. Contour lines are *regions of highest density* [13] of gaussian kernel density estimates of the data conditioned to the classes depicted on the thumbnails. Bandwidth selection for the gaussian kernel density estimates is done using *Scott's rule* [31] and we select $\alpha = 0.2$ for the *highest density regions*. Dimensionality reduction has been performed using PCA down to 50 dimensions followed by *t-SNE* [34] with perplexity of 30, early exaggeration of 12 and learning rate of 200.

Abstract

When deploying deep learning technology in self-driving cars, deep neural networks are constantly exposed to domain shifts. These include, e.g.*, changes in weather conditions, time of day, and long-term temporal shift. In this work we utilize a deep neural network trained on the Cityscapes dataset containing urban street scenes and infer images from a different dataset, the A2D2 dataset, containing also countryside and highway images. We present a novel pipeline for semantic segmenation that detects out-of-distribution (OOD) segments by means of the deep neural network's prediction and performs image retrieval after feature extraction and dimensionality reduction on image patches. In our experiments we demonstrate that the deployed OOD approach is suitable for detecting out-of-distribution concepts. Furthermore, we evaluate the image patch retrieval qualitatively as well as quantitatively by means of the semi-compatible A2D2 ground truth and*

obtain mAP values of up to 52.2%.

1. Introduction

The advances of convolutional neural networks (CNNs) in the recent years enabled the use of machine learning for complex computer vision tasks that had been considered out of reach before. Among them is the semantic segmentation that facilitates complex scene understanding [20]. Applications like *e.g*. autonomous driving, medical imaging or surveillance are problem domains that induce a high risk and lead to fatal consequences when using CNNs in an autonomous and unsupervised fashion. Thus it is of utmost importance to monitor CNNs and ask for human intervention when questionable predictions are detected [11].

In general, there are many aspects of a machine learning pipeline for computer vision that require supervision, not necessarily by humans. Starting with data collection at the very beginning it is of high importance to collect a sample of the visual world that represents the sub environment for which a CNN's deployment is desired. As the visual world has basically an infinite variability, a sufficient representation (in particular in safety relevant scenarios) is not easy to accomplish. Additionally, the acquisition of annotation can be an expensive and time consuming task. A variety of publications presented different possible approaches. While some publications try to reduce the cost of label acquisition with techniques like *semi supervised learning* [24], *weakly supervised learning* [2, 19, 24] or *active learning* [28, 32], more recent works on *self-supervised learning* [7, 16] try to extract visual features without requiring labeled data. Furthermore, generating synthetic data for training neural networks is also considered. Publications in this direction include the rendering of highly realistic images [35] or using generative adversarial networks (GANs) for generation and augmentation of already collected data [30].

Similar to the data acquisition phase, monitoring is also required when a trained model is deployed in the (open) real world – which is under constant change over time and space. As we cannot assume that available training samples represent the target real world environment well, it is highly relevant to track possible prediction failures and situations that are completely new to the model at hand. Important research areas that tackle these problems include *uncertainty* or *confidence* estimation as well as *out-of-distribution* (OOD) detection. Works in the field of *uncertainty* or *confidence* estimation include Bayesian methods [8, 15, 38], ensemble methods [17] as well as approaches that acquire information from intermediate layers of the network or from its predictions to train a second model that serves as confidence estimator [6, 11, 23, 27]. The task of OOD detection has been broadly studied for image recognition [6,11,18,23] and most of these methods are applicable to the problem of semantic image segmentation. Approaches specifically designed for semantic segmentation include e.g. [3, 22]. Both try to measure confidence on pixel level which is in contrast to the method used in this work. The authors of [3] utilize shared convolutional features to predict segmentation confidence with an additional output branch. Using a negative dataset as a proxy for OOD objects, they train their auxiliary model to predict model confidence. In [22] the authors propose to use ensembles of models to calibrate the prediction confidence. This is however computationally expensive, especially for state-of-the-art semantic segmentation models in the context of street scene segmentation.

During autonomous driving, even a simple change of location can result in a severe domain shift resulting in unseen objects. Additionally, the real world is subject to continuous transformation. Therefore it is indispensable to update deep learning models regularly. This results in a continuous feedback loop between the previously described steps of a machine learning pipeline, the data acquisition stage and the deployment phase. Our work makes important contributions to a more efficient workflow of this process. Using a meta classification and regression approach termed *MetaSeg* [27] to find unknown objects, we can group the detected entities into visually and semantically related groups in order to enhance data exploration in the presence of domain shift. Using predicted segmentation masks and image retrieval within newly collected data, our approach can be used to find classes that may be underrepresented or missing in the training dataset. This knowledge can be used for example to improve the existing model by partly labeling novel object classes and including them into the next training round. In summary the contributions of this work are as follows:

1. We show that MetaSeg predicts the intersection over union of out of domain samples reliably.

2. Using MetaSeg we demonstrate that we are able to detect unknown object classes.

3. By extracting visual features we are able to group the found entities into an embedding space with semantically related neighborhoods.

4. We perform an evaluation on the task of image retrieval with a variety of common deep learning architectures as feature extractors.

To the best of our knowledge, this is the first work that reliably detects OOD samples in semantic segmentation and reveals their semantic similarity.

The remainder of this work is structured as follows: In section 2 and section 3 we describe the theoretical foundations of our OOD detection and retrieval pipeline. Using the Cityscapes dataset [4] as source domain and the A2D2 [9]

dataset as target domain (out of domain sample) we demonstrate in section 4.1 that we are able to reliably detect unknown objects in the presence of domain shift. Complementing the meta segmentation analysis we conduct experiments on an image retrieval task and present in section 4.2 qualitative and quantitative results to demonstrate that this technique can be used to enhance data exploration for semantic image segmentation.

2. Out-of-Distribution Detection

Under the premise that objects of unknown classes mostly cause suspicious predictions, we can quantify this effect and use it for OOD detection. Therefore we deploy an approach that estimates segmentation quality for each predicted segment by means of statistical properties. This approach is termed *MetaSeg* [27] and was developed further in [21, 29]. Based on a structured dataset of metrics that aggregate dispersion measures of the softmax output as well as geometric properties of each predicted segment, a regression model is trained to predict the segmentation quality (in terms of segment-wise intersection over union (IoU) with the ground truth, also known as the Jaccard index [14]). To this end, we train a small fully-connected neural network on the set of metrics solely corresponding to in-distribution data. In what follows we describe this procedure in more detail, starting with the construction of metrics as proposed in [29]:

Given the output $f_z(y|x,w)$ of the semantic segmentation model for input x, weights w and pixel z over class labels $y \in \mathcal{C} = \{y_1, \ldots, y_K\}$, we compute the pixel-wise classification *entropy*

$$E_z(x,w) = -\frac{1}{\log(K)} \sum_{y \in \mathcal{C}} f_z(y|x,w) \log f_z(y|x,w)\,, \quad (1)$$

the *probability margin*

$$\begin{aligned} M_z(x,w) = 1 &- f_z(\hat{y}_z(x,w)|x,w) \\ &+ \max_{y \in \mathcal{C} \setminus \{\hat{y}_z(x,w)|x,w)\}} f_z(y|x,w)\,, \end{aligned} \quad (2)$$

and the *variation ratio*

$$V_z(x,w) = 1 - f_z(\hat{y}_z(x,w)|x,w)\,, \quad (3)$$

with $\hat{y}_z(x,w) = \arg\max_{y \in \mathcal{C}} f_z(y|x,w)$ being the predicted class of pixel z.

After computing these dispersion measures for each pixel they are aggregated over each segment using different schemes like mean/variance over boundary/inner pixels and relative quantities between these. This results in a total of 75 metrics for each segment which are used as input for the meta segmentation network to predict the segment-wise IoU (for further details we refer to [27] and [29]). This approach has the advantage that predicted segments are rated as a whole whereas other methods that predict the confidence pixel wise, such as eqs. (1) to (3)) (hence providing uncertainty heat maps), typically have a concentration of low confidence at the boundary of objects.

Recalling the assumption that unknown concepts are coming with suspicious segmentations, we detect predicted segments with low estimated IoU values below a chosen threshold. Subsequently, each detected segment is framed by a bounding box, i.e., the rectangular box containing all pixels of the predicted segment with *minimal* width and height. The corresponding crops of the original image are then subject to further processing and retrieval analysis.

3. Retrieval

After detecting image crops corresponding to segments with low estimated quality (potential unknown OOD objects) from newly collected data, further exploration of these crops can reveal weaknesses of the CNN with respect to the given domain shift. One promising approach is content-based image retrieval [1] which can help finding similarities in the crops and ultimately rate the relevance of clusters with low effort.

Image retrieval is a well known problem with numerous applications in, *e.g.*, search engines, automatic 3D reconstruction or document analysis. Typically retrieval starts with a query image which acts as an anchor to sort the available data points. Ranking is almost always done by calculating a visual similarity and sorting all available samples according to the similarity value. Thus there are two decisions to make: First one needs to extract visual features and then apply a distance function.

For extracting visual features we evaluate a *VGG16* network [33], different sizes of a *ResNet* [10], *WideResNet* [37] and *DenseNet* [12] all pretrained on *ImageNet* [5] (implementation and pretrained weights are taken from the PyTorch [25] library). For all these networks we remove the final fully connected layers, i.e., we only use their backbones for feature generation. Unlike the other evaluated architectures, the original *VGG* network does not include global average pooling before the fully connected layers. To be able to extract features of a fixed dimensionality, independently of the size of the input image, we also perform global average pooling to the *VGG* backbone. As all evaluated network architectures have a limit on the minimum input size we only detect segments with a predefined minimum bounding box height and width. Another necessity for this choice is that the visual information in a small window of the image would be very low which is not beneficial for grouping objects based on visual similarity.

After computing feature vectors of image crops – which we term *embeddings* – we desire to explore their similarities. The most common and intuitive distance function is probably the *euclidean distance* (eq. (4)). Another fre-

quently used metric is the *cosine similarity* (eq. (5)) which can be beneficial for high dimensional data [26] like the feature vectors extracted from neural networks,

$$L^2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad x, y \in \mathbb{R}^n\,, \tag{4}$$

$$\cos(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad x, y \in \mathbb{R}^n \setminus \{0\}\,. \tag{5}$$

In order to reduce noise in the computed embeddings and to focus on features that are relevant for the detected objects, a dimensionality reduction can improve retrieval performance significantly. To accomplish this we will evaluate in section 4.2 different numbers of dimensions. For a dimension lower than four we will utilize the t-distributed stochastic neighbor embedding (t-SNE) [34] method. It is a method specifically designed for visualizing high dimensional data in low dimensional spaces while minimizing the Kullback-Leibler divergence between joint probabilities of the reduced and original space. For a more detailed explanation we refer to [34]. All other dimensionality reductions are performed using principal component analysis (PCA).

In summary, our complete OOD detection and retrieval pipeline looks as follows:

1. Gather semantic segmentation predictions of newly collected samples.
2. Rate all predicted segments according to their IoU estimated by MetaSeg (trained solely on the in-distribution domain).
3. Detect segments with estimated IoU < 0.5 that are, however, still predicted to belong to classes that are of high interest [1]. For each candidate segment, the corresponding bounding box is used to provide a crop of the original input image.
4. Feed each crop through an embedding network pretrained on ImageNet and compute vectors of visual features.
5. (Optional) Reduce the dimensionality of the embedding space.
6. Perform retrieval by nearest neighbor search in the resulting embedding space.

4. Experimental Evaluation

For our experimental evaluation we use a state-of-the-art DeepLabv3+ semantic segmentation model. The implementation and pretrained weights on the Cityscapes training dataset are taken from the GitHub repository of [36]. The architecture uses a WideResNet38 backbone and achieves a mIoU of 83.5% on the Cityscapes test dataset on the standard label set and an mIoU of 92.2% on the category labels. We use the Cityscapes training set as source domain and the A2D2 dataset [9] as target domain in which we try to find objects that are not well represented in the source domain. While Cityscapes only contains urban street scenes, A2D2 in addition provides scenes from highways and countryside. Therefore A2D2 is a suitable choice for exploring concepts not contained in Cityscapes. In fact, there are classes in A2D2, for example *tractor* or *obstacles / trash*, that are not present in Cityscapes. The class *road blocks* has a large portion in common with the Cityscapes classes *fence*, *wall* and *guard rail* but also contains "highway fences" from German highways that are not present in urban environments. This makes A2D2 highly suitable as a target domain to simulate newly collected data that has to be explored and analysed in terms of domain shift and possible new object classes that should be integrated into the next version of the model. As A2D2 provides quite a large number of images (30 000 in total), we randomly sample 2 000 images and additionally include all images that contain instances of the *tractor* class as this class is completely new to the semantic segmentation model. This leaves us with approximately 2 100 images.

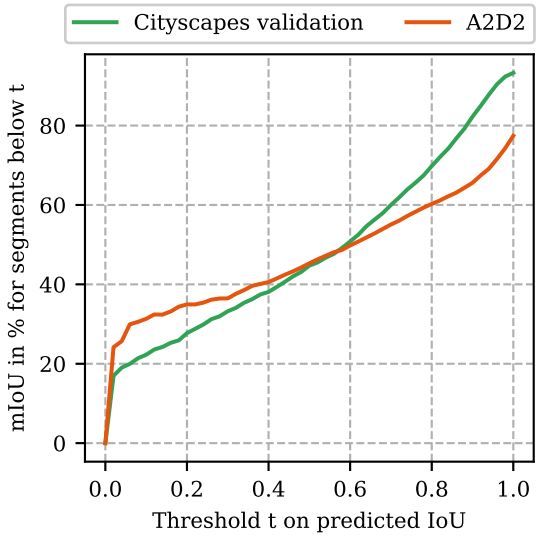

Figure 2: Intersection over union for segments that have a predicted IoU below a given threshold.

All models and experiments were implemented using the PyTorch framework [25] and the source files can be found in our GitHub repository[2].

[1] In our experiments we focus on the classes wall, fence, traffic light, traffic sign, person, rider, car, truck, bus, train, motorcycle and bicycle.

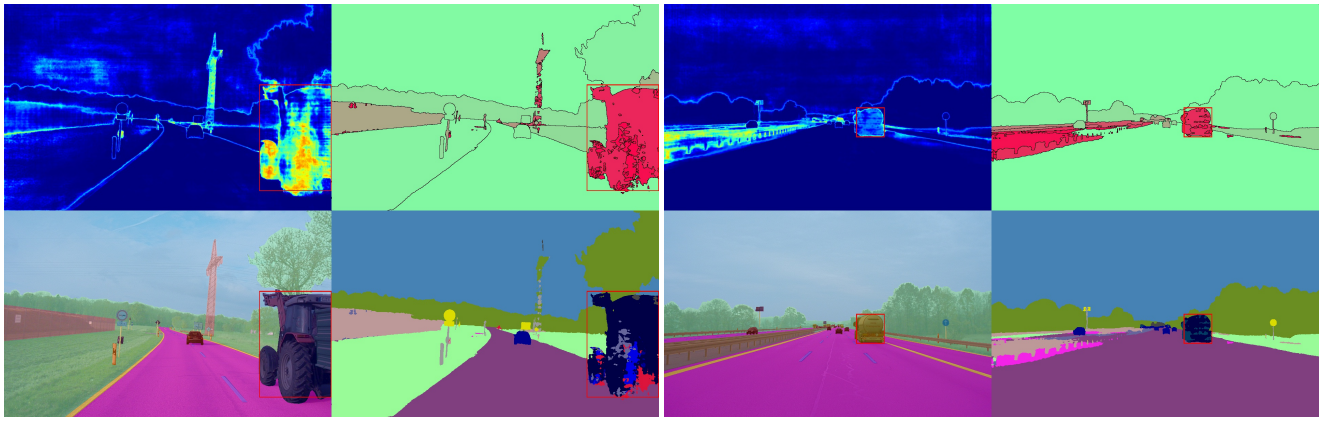

Figure 3: Two sample images from the A2D2 dataset. Each of them consists of four panels. Top left: Per pixel entropy heatmap, top right: prediction of MetaSeg (green color represents high predicted IoU values, red represents low ones), bottom left: annotation over input image, bottom right: predicted semantic segmentation.

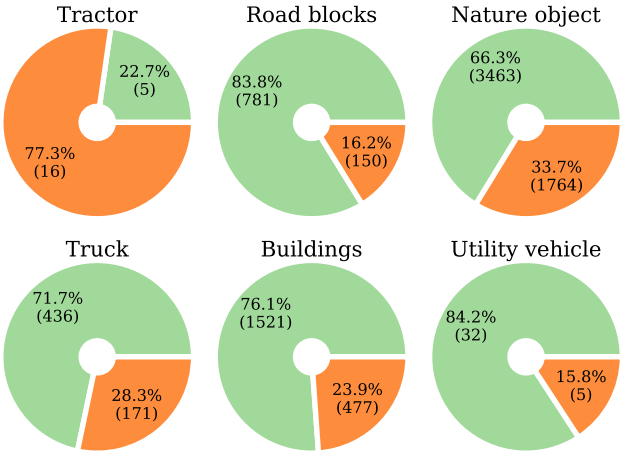

Figure 4: Number of (not) detected instances of selected classes from the A2D2 dataset. The minimum size was set to 128×128 pixels, green: not detected, red: detected.

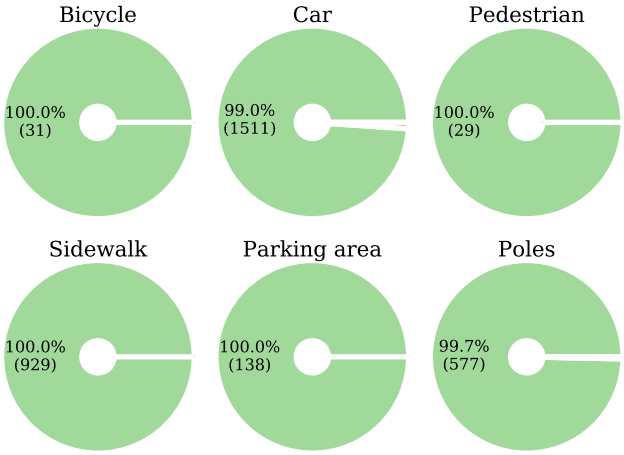

Figure 5: Number of not detected instances of selected classes from the A2D2 dataset. The minimum size was set to 128×128 pixels, green: not detected, red: detected.

4.1. Out-of-Distribution Detection

So far the performance of MetaSeg [27] has not been evaluated on datasets that are different from the domain MetaSeg has been trained on. To demonstrate the suitability of MetaSeg to find badly segmented objects on out of domain samples, we evaluate the DeepLabv3+ model [36] on the A2D2 dataset. First, we calculate the mIoU of the semantic segmentation model on the target domain to have a reference in terms of segmentation quality. However, the label sets of A2D2 and Cityscapes are not compatible, as discussed earlier. Hence, we perform a label mapping from the full A2D2 label set to the Cityscapes training set. To minimize mapping errors we further map the Cityscapes classes onto their coarse category ids. Excluding the *void* class this leaves us with the seven categories *flat*, *construction*, *object*, *nature*, *sky*, *human* and *vehicle*. Note that we made a small tweak where we mapped the *rider* class of Cityscapes to the *vehicle* instead of the default *human* category. This is motivated by the different annotation styles, i.e., a person riding a bicycle is annotated as bicycle in the A2D2 dataset which is in contrast to the annotation in Cityscapes where the person is annotated as a *rider*. DeepLabv3+ achieves a remarkable test accuracy of 99.2% on the Cityscapes test set with respect to the coarse categories. Evaluating the DeepLabv3+ on the A2D2 dataset with this label mapping still results in a mIoU of 77.4%.

[2] https://github.com/RonMcKay/OODRetrieval

In order to demonstrate the effectiveness of MetaSeg we compute the mIoU under different thresholds, removing segments with an estimated IoU above the specified threshold from the evaluation. This leads to a minimal performance of around 20% as shown in fig. 2. Although the thresholding does not work as nicely as for the Cityscapes validation set, the experiment shows that we are able to identify badly segmented regions and detect them confidently by means of the predicted IoU.

Two example predictions can be seen in fig. 3. On the left the unknown *tractor* object is segmented very poorly which is detected by MetaSeg. On the right the truck as well as the highway fence are badly segmented which is detected as well.

Next, we performed an experiment to test whether unknown objects are consistently segmented with low quality, i.e., low IoU, and also if MetaSeg predicts a low IoU. This would be an indispensable property to be able to find these objects reliably. Under the assumption that large objects with low predicted IoU are most critical, we collect all segments that have a predicted IoU of less than 0.5 and a minimum size of 128×128 pixels and count how many instances of each class are covered by at least one of these segments. We count an instance as *covered* by a segment when the total number of pixels of that instance inside the segment is at least 50% of the total segment size. For the evaluation we also only consider ground truth instances that have a minimum size of 128×128 pixels, for a given class their number represents the amount of instances. Note that, in this scenario we do not use the label mapping as we want to explore the behavior of our approach with respect to all A2D2 classes. Figure 4 shows that MetaSeg detects 77.3% of the instances belonging to the *tractor* class (which is a class not present in the source domain). Reviewing the remaining 22.7% of *tractor* instances that were not detected they are mostly in scenarios where the tractor was obscured to a large degree by other vehicles or in situations where the tractor was not on the road but on a nearby farming field. When analyzing the 16.2% of *road blocks* instances, we observe that they are almost exclusively segments of "highway fences". The classes *nature object* as well as *truck* have also a relatively high share of detected segments. This is due to the fact that large trucks and forests / grasslands that cover a big portion of the image are rather rare in urban environments. The *building* cluster consists to a large degree of tunnels and bridges that span the street. Figure 5 shows classes that are common in the Cityscapes dataset. The fact that almost no segments belonging to these classes are detected (except for a few cars, sidewalks and poles) further demonstrates the performance of MetaSeg on this out-of-distribution detection task. Performing this same evaluation on the Cityscapes test set leads to an average of 0.06 detected segments per image whereas we detect 0.82 segments per image in the A2D2 dataset. This shows that we are consistently detecting out of distribution objects.

4.2. Retrieval Task

In this section we present a qualitative and quantitative evaluation of the retrieval task described in section 3. Figure 1 depicts an embedding space of features computed by a *ResNet152*. Dimensionality reduction has been performed using principal component analysis down to 50 dimensions followed by *t-SNE* [34] (the original feature space had a dimensionality of 2048). Therefore, each predicted segment that has been detected by MetaSeg is mapped to a data point / sample in $\mathbb{R}^2$. Qualitatively the space seems to be well separated into different clusters of objects. Note that this is achieved without using any ground truth of A2D2. In order to visualize that the embedded features of detected segments are clustered according to their semantics, we assign the A2D2 color code to the embedded samples. Herein, the class assigned to the predicted segment is the class of the ground truth segment that has maximal overlap with the predicted segment.

The classes that contribute the majority of data points in the embedding space are *nature object*, *road blocks*, *buildings* and *truck*. All of them are well separated into individual clusters. The *tractor* class, although having a relative share of only 1.3% does also form a cluster close to trucks, which are semantically related. Embeddings with the ground truth class *sky* seem to emerge from situations with borderline weather conditions like rain or direct sunlight. In both cases the neural network tends to make false predictions within sky regions. In addition, one can notice that in highway scenes the sky covers a much larger area than in urban scenes. However, due to the training data consisting of urban scenes, the segmentation network is potentially biased towards predicting buildings in case the sky looks unusual.

For measuring retrieval performance quantitatively we use the *mean average precision* (mAP) which is defined as

$$\text{mAP} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AP}_q \,, \quad \text{where} \tag{6}$$

$$\text{AP}_q = \frac{\sum_{i=1}^{n} p(i) \cdot r(i)}{t} \,, \tag{7}$$

with $p(i)$ being the precision when cutting off the retrieval list at position i, $r(i)$ an indicator function equaling 1 if element i is relevant with respect to the query and 0 otherwise, n is the total number of data points and Q the total number of queries. It holds that $\text{mAP} \in (0, 1]$, being 1 if all relevant objects are at the top of the retrieval list for each query and minimal if all relevant objects are at the end of each retrieval list.

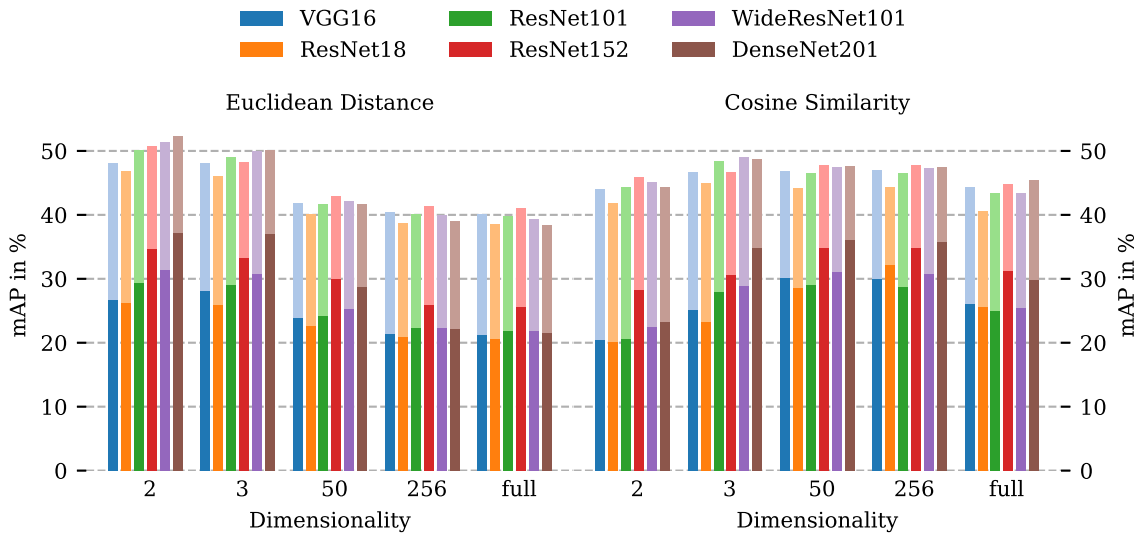

Figure 6: Retrieval results for different feature extractors, distance metrics and feature dimensionalities. For dimension two and three PCA down to 50 dimensions followed by t-SNE [34] has been used for dimensionality reduction. For the dimensions 50 and 256 PCA has been performed. A dimension of *full* means no dimensionality reduction. Transparent bars correspond to mean over all queries not taking their class into account thus being weighted by frequency. Non transparent bars correspond to the mAP when averaging over all queries first class-wise and then over all classes.

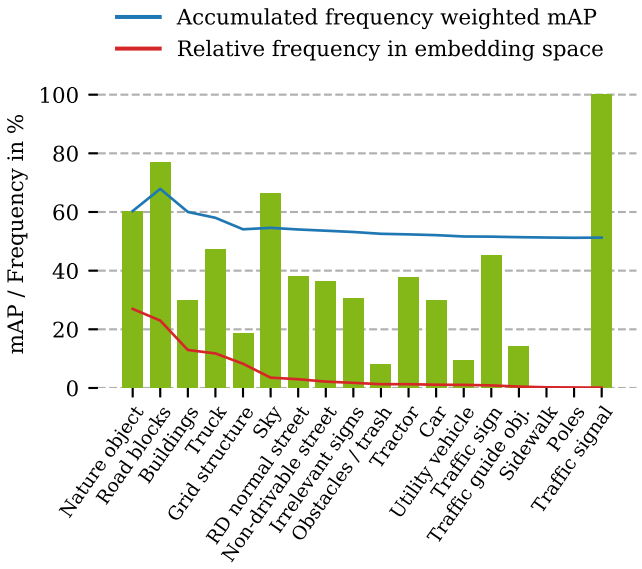

Figure 7: Class-wise retrieval results for a *DenseNet201* feature extractor using PCA to reduce to 50 dimensions followed by t-SNE [34] to reduce the embedding dimensionality to two dimensions. For measuring similarity the Euclidean distance has been used. The classes have been sorted from left to right in descending order according to their frequency in the embedding space. Results are measured in mAP percentage.

In fig. 6 the quantitative retrieval results with respect to different feature extractors, embedding space dimensionalities and distance metrics are summarized in terms of mAP. The Euclidean distance benefits from the t-SNE embedding into a lower dimensional space. This is in contrast to the cosine similarity that performs worse in low dimensional spaces. However, the results for cosine similarity show a small performance increase when going from two to three dimensions. Over the range of dimensions, the cosine similarity results appear to be more stable. Regarding the different feature extractors, deeper networks with more filters extract more meaningful features which are better in retrieving visually similar objects. The overall best performing network is the *DenseNet201* with 52.2% mAP followed by the *WideResNet101* with 51.3% mAP. The large gaps between global and class wise mAP is due to a few under represented classes like, *e.g.*, *poles* or *sidewalk*. In our experiments they have only a few samples and a high visual variability due to many distracting background objects. This is why these classes get a mAP in the range of $0.16 - 0.38\%$ and reduce the average over classes. Figure 7 depicts the mAP values for the best performing setup (*DenseNet201* with PCA/t-SNE reduced to two dimensions) split up into the different classes that are present in the embedding space. The results are sorted from left to right in descending order according to their frequency in the embedding space. In practice, the frequency of an unknown object is very likely to be an important indicator. Therefore, it should be estimated before making a decision whether to acquire new data for training.

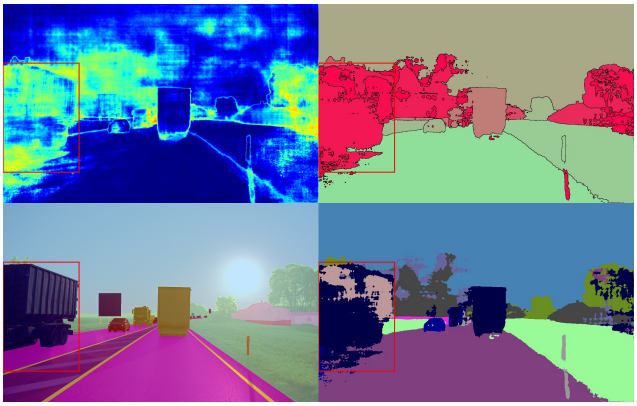

Figure 8: Sample prediction from the A2D2 dataset. Panels are the same as in fig. 3. The trailer on the very left hand side is labeled as *tractor*, presumably based on previous frames. This can hardly be established without utilizing temporal information or expert knowledge. The overall badly predicted segmentation is likely caused by recording against direct sunlight.

Our retrieval results in terms of mAP show that retrieval is useful for such an estimation and for data selection in general.

Intuitively the qualitative results in fig. 1 seem to be better than the achieved mAP results from fig. 6. The reasoning behind this is as follows. When looking at the label set of A2D2, which is where we extract the ground truth information for the retrieval task, the visual variability of some classes is too high to perform retrieval on them based on visual similarity. The class *utility vehicle*, *e.g.*, contains not only trams but also excavators and other construction machines which have a rather low visual similarity. Another example is the *tractor* class. Figure 8 shows a trailer on the very left hand side. Without any further context, classifying this trailer to belonging to a tractor is challenging. Previous frames however reveal that the trailer is mounted to a tractor, therefore it is presumably labeled as tractor. In order to compute visual features that are correlated with the other *tractor* class instances, sequential models could be considered for exploiting correlations in consecutive frames. The described issues only represent a few of the challenges that we face when extracting ground truth information from pixel label annotation for evaluation of the retrieval task. In general the retrieval evaluation suffers from the coarse semantic classes and inconsistencies among the datasets. Datasets that provide a label set with more fine-grained object classes might increase the quantitative retrieval results significantly. Nonetheless, the qualitative and quantitative results show that the embedding space is suitable for exploring newly collected data and that the proposed pipeline can be used to accelerate feedback from the deployment phase to an update of the training dataset. Note that the proposed pipeline is of generic nature, in a sense that the user has the freedom to choose any OOD detection method as well as any semantic segmentation model.

5. Conclusion and future work

In this work we have demonstrated and validated how to use prediction quality estimation methods, such as MetaSeg, and image retrieval to explore newly collected data that might be affected by domain shift. We are able to detect object classes that are unknown to the semantic segmentation network due to missing samples in the training set. Data exploration can be guided by image retrieval on visual features that are gathered by common deep learning architectures which are trained on the task of image classification. However, dataset selection for evaluating this kind of methods leaves room for further improvement. Also benchmarking in the fields of OOD detection and uncertainty on OOD samples in semantic segmentation remains tedious due to the lack of appropriate datasets. We believe that these subjects deserve to be further addressed in the future.

In terms of future work, we plan to explore possibilities to utilize the detected segments. Methods like *active learning* or *semi supervised learning* can be used to reduce annotation cost for new object classes and still incorporate them into a new training set. The knowledge gained from a human in the loop in an active learning setting could also be used to automatically retrain the embedding network. This way the visual features would be more meaningful in the context of the current environment and thus increase retrieval performance. Another possible research direction is the utilization of temporal correlation between adjacent frames or segmentation networks that are trained to be uncertain on unlabeled classes like, *e.g.*, the void classes of Cityscapes.

Acknowledgment

This work is in part funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) through the grant 19A19013Q, project AI-DeltaLearning. Furthermore we thank Hanno Gottschalk for useful advice and discussion.

References

[1] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky. Neural codes for image retrieval. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 584–599. Springer, 2014.

[2] Amy L. Bearman, Olga Russakovsky, Vittorio Ferrari, and Fei-Fei Li. What's the point: Semantic segmentation with point supervision. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, pages 549–565. Springer, 2016.

[3] Petra Bevandic, Ivan Kreso, Marin Orsic, and Sinisa Segvic. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang, editors, *Pattern Recognition - 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10-13, 2019, Proceedings*, volume 11824 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2019.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.

[6] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *CoRR*, abs/1802.04865, 2018.

[7] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1422–1430. IEEE Computer Society, 2015.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016.

[9] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: AEV Autonomous Driving Dataset. `http://www.a2d2.audi`, 2019.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[12] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[13] Rob J. Hyndman. Computing and graphing highest density regions. *The American Statistician*, 50(2):120–126, 1996.

[14] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, Feb. 1912.

[15] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017.

[16] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1920–1929. Computer Vision Foundation / IEEE, 2019.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6402–6413, 2017.

[18] Shiyu Liang, Yixuan Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017.

[19] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3159–3167. IEEE Computer Society, 2016.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3431–3440. IEEE Computer Society, 2015.

[21] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. Time-dynamic estimates of the reliability of deep semantic segmentation networks. *CoRR*, abs/1911.05075, 2019.

[22] Alireza Mehrtash, William M. Wells III, Clare M. Tempany, Purang Abolmaesumi, and Tina Kapur. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *CoRR*, abs/1911.13273, 2019.

[23] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk. Classification uncertainty of deep neural networks based on gradient information. In Luca Pancioni, Friedhelm Schwenker, and Edmondo Trentin, editors, *Artificial Neural Networks in Pattern Recognition - 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September*

19-21, 2018, Proceedings, volume 11081 of *Lecture Notes in Computer Science*, pages 113–125. Springer, 2018.

[24] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1742–1750. IEEE Computer Society, 2015.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019.

[26] Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-scale image retrieval with compressed fisher vectors. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3384–3391. IEEE Computer Society, 2010.

[27] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities. *CoRR*, abs/1811.00648, 2018.

[28] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. Deep bayesian active semi-supervised learning. In M. Arif Wani, Mehmed M. Kantardzic, Moamar Sayed Mouchaweh, João Gama, and Edwin Lughofer, editors, *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*, pages 158–164. IEEE, 2018.

[29] Matthias Rottmann and Marius Schubert. Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, page 0. Computer Vision Foundation / IEEE, 2019.

[30] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser-Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3752–3761. IEEE Computer Society, 2018.

[31] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley, 1992.

[32] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[34] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[35] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *CoRR*, abs/1810.08705, 2018.

[36] Fitsum A. Reda Kevin J. Shih Shawn Newsam Andrew Tao Bryan Catanzaro Yi Zhu*, Karan Sapra*. Improving semantic segmentation via video propagation and label relaxation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[38] Mengjie Zhao, Bin Song, Yue Zhang, and Hao Qin. Face verification based on deep bayesian convolutional neural network in unconstrained environment. *Signal, Image and Video Processing*, 12(5):819–826, 2018.