DETECTING OUT OF DISTRIBUTION OBJECTS IN SEMANTIC SEGMENTATION OF STREET SCENES

Dominik Brüggemann and Stefan Bracke

Chair of Reliability Engineering and Risk Analytics
University of Wuppertal, Gaußstraße 20, Wuppertal, Germany.
E-mail: {*dbrueggemann, bracke*}*@uni-wuppertal.de*

Robin Chan, Matthias Rottmann and Hanno Gottschalk

School of Mathematics and Natural Sciences
University of Wuppertal, Gaußstraße 20, Wuppertal, Germany.
E-mail: {*robin.chan, rottmann, hanno.gottschalk*}*@uni-wuppertal.de*

Convolutional neural networks (CNNs) have seen spectacular advances over the past century, particularly improving the state-of-the-art in computer vision tasks. Semantic segmentation, an image classification at pixel-level, is an essential step in understanding a vehicle's surroundings via camera images for autonomous driving. While CNNs keep becoming more and more powerful predictive models, they still often fail if an input is outside of their learned concepts. The non-detection of objects in street scenes, including out-of-distribution (OOD) objects, poses serious hazard and may cause public harm. Therefore, methods determining when a model has failed are crucial in order to ensure a safe and responsible usage of CNNs in real-world applications. In this work we present a method for the detection of OOD objects. We extend work from image classification to more complex semantic segmentation. Our approach is based on pixel-wise entropy derived from the CNN's probabilistic Softmax output. This dispersion measure can be understood as prediction uncertainty indicating a failure per pixel. Paired with methods from image processing, we determine image regions in which an OOD object might be present but overlooked by the CNN. We show the course of our method's development performed on the semantic segmentation Lost and Found dataset that was inferred using the state-of-the-art CNN DeeplabV3+ with Xception65 network backbone. We provide an in-depth statistical evaluation and discuss strength, but also weakness, of our presented method. Additionally, we perform a brief analysis of the topic from the point of view of safety engineering, including a critical evaluation why common standards like ISO 26262 cannot be applied.

Keywords: computer vision, convolutional neural networks, semantic segmentation, out-of-distribution, autonomous driving, image processing, functional safety.

1. Introduction

Deep learning algorithms are to be deployed in a variety of real world tasks as they consistently improve the state-of-the-art in fields such as natural language processing, image processing or medical diagnosis. In particular in the field of computer vision the introduction of convolutional neural networks (CNNs, Krizhevsky et al. (2012)) led to a great impact on recent advances as most deep learning models are based on the convolutional architecture. Even for difficult problems like semantic segmentation, the task of assigning the object class for each pixel within an image, CNNs are the most commonly used network architecture (Long et al., 2014). Despite the high performance that such CNNs provide, the adoption of deep learning in everyday applications is yet to come. One reason is the lack of mandatory safety guarantees in systems that are becoming more and more complex. For instance, the capability of indicating when the underlying model is likely mistaken is still a missing but safety critical prerequisite for the usage of deep learning. Therefore, the behavior of CNNs when they encounter data unlike the training data is an important topic in this area. In this context, samples, that are possibly seen at test time but are different to the ones seen during training, are called out-of-distribution (OOD) samples. One property that one expects CNNs to satisfy is to output high prediction uncertainty for these OOD samples. However, the opposite behavior, i.e., producing high confidence predictions, have been reported in several works, see Yosinski and Clune (2015); Hendrycks and Gimpel (2016); Guo et al. (2017).

Tackling the OOD detection in CNNs is usually approached by either modifying the training process or by post processing given CNN out-

Proceedings of the 30th European Safety and Reliability Conference and
the 15th Probabilistic Safety Assessment and Management Conference
Edited by Piero Baraldi, Francesco Di Maio and Enrico Zio
Published by Research Publishing, Singapore.
ISBN: 978-981-14-8593-0; doi:10.3850/978-981-14-8593-0

puts. The training approach aims at improving the models' discrimination capabilities between out-of-distribution and in-distribution data (Lee et al., 2018) or at enforcing models to output uniform confidence scores on OOD inputs (Hendrycks et al., 2018). Mathematical guarantees for low confidence scores on OOD samples are additionally provided in the work by Meinke and Hein (2019). On the contrary, post processing methods do not require any changes to a pre-trained model and identify OOD samples by deriving patterns from the CNNs' output. As baseline approach Hendrycks and Gimpel (2016) introduced thresholding on the maximum softmax probability that is outperformed by techniques adjusting the estimated confidence (DeVries and Taylor, 2018; Liang et al., 2018). However, all these presented methods are limited to simple image classification problems only.

In this work, we extend OOD detection from image classification to the more demanding and application-relevant task of semantic segmentation. Specifically, we present a post processing method that extracts (pixel-wise) inference uncertainty via the entropy measure from the pixel-wise probabilistic softmax output. Connected components of pixels with high entropy values, that are additionally aggregated using methods from image processing, consequently are supposed to indicate the presence of an (possibly overlooked) OOD object. We perform numerical experiments with the state-of-the-art semantic segmentation network DeeplabV3+ with Xception65 backbone (Chen et al., 2018) trained on the Cityscapes street scenes dataset (Cordts et al., 2016) and test the performance of our presented method on the Lost and Found dataset (Pinggera et al., 2016) which includes OOD objects. We provide an in-depth statistical evaluation and discuss strength but also weakness of our presented method.

The remainder of this work is structured as follows: In the second chapter the methodology is presented by building up the necessary fundamental knowledge and explaining it step by step. Moreover the thought process is presented in a traceable way. In the third chapter the results obtained by applying our method to the test set of the Lost and Found data set are presented and analyzed. Moreover strengths and weaknesses of the method and the used evaluations are discussed. In the following, a brief consideration of the problem from a safety perspective is given. The work is concluded with a summary of the results and an outlook.

2. Methodology

The method of post processing the semantic segmentation is divided into three sub-processes. During the first sub-process the pixelwise entropy of the Softmax-outputs is determined. The entropy can be used as a measure of the uncertainty of a neural network. In the second sub-process the entropy is filtered in order to find dense regions of high uncertainty. These regions are assumed to contain an overlooked object. In the third sub-process the prediction, i.e. the semantic segmentation, is modified using the information obtained in the second sub-process.

2.1. *Determination of the entropy*

At the beginning of the first sub process the Softmax-output of a segmentation-CNN is given for every image of the data set. The Softmax function is given by the following formula.

$$S(z_i^l) = \frac{e^{(z_i^l)}}{\sum_j e^{(z_j^l)}} \tag{1}$$

Here, z_i^l is the activation of the i-th neuron, of the l-th layer. Applying the Softmax function to the activations of a layer converts these activations into easy-to-read probability-like values. The Softmax function is usually applied to the output layer of a neural network.
In this case, the CNN was trained to distinguish 19 different classes. Consequently, there are 19 softmax probabilities for each pixel of an image. To measure the uncertainty of the CNN for a single pixel, the entropy is calculated.
Entropy is a measure for the average information value of a message and was defined by Claude E. Shannon (Shannon, 1948). In general, the entropy H of a message is calculated as follows.

$$H_I = -\sum_{z \in Z} p_z log_2(p_z) \tag{2}$$

Here, H_I denotes the entropy of a message I, which is built from the letters z of the finite alphabet Z. p_z denotes the probability of occurrence of the letter z.
This formula is adapted to measure the uncertainty of the CNN. As stated before, the CNN was trained to distinguish 19 different classes, which leads to arrays of the size $2048 \times 1024 \times 19$ as the Softmax-output. The entropy is calculated for every individual pixel, using the following formula.

$$\begin{aligned} H_{pixelwise} &= \frac{-\sum_{i=1}^{19} p_i log_2(p_i)}{log_2(19)} \\ &= \sum_{i=1}^{19} \frac{p_i log_2(p_i)}{log_2(\frac{1}{19})} \end{aligned} \tag{3}$$

Here, H denotes the entropy again. More specifically, the normalized entropy, since the value is divided by $log_2(\frac{1}{19})$, leading to values in the range of $[0; 1]$. p_i denotes the Softmax-Output of the i-th class that the CNN was trained on.
An exemplary result of this first sub process can be seen in figure 1. It can be observed that high entropy values occur in the region of the left object near the center of the image.

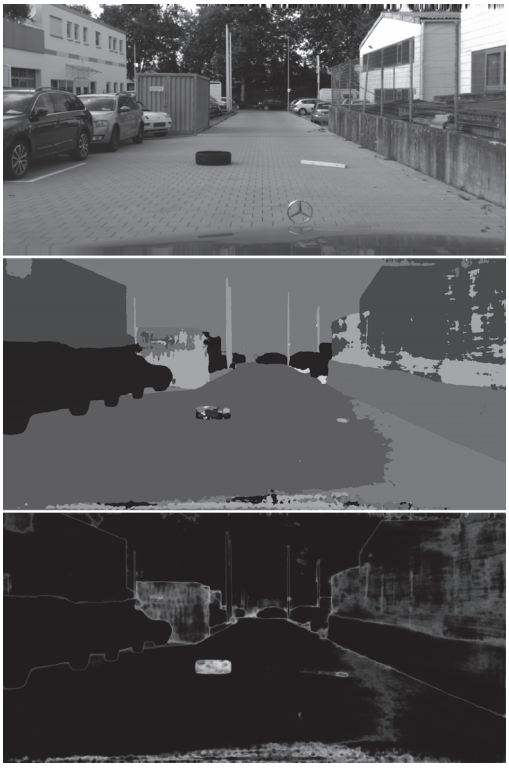

Fig. 1. The Softmax probabilities that were obtained by inferring the real image (top) visualized in the semantic segmentation (middle) provide the basis for the calculation of the entropy. The values of the entropy are visualized (bottom) from white for $H = 1$ to black for $H = 0$.

2.2. *Filtering of the entropy*

During the next sub-process, the entropy is filtered in order to find dense regions of high uncertainty. To do this, the relevant area of the image is narrowed down first with the goal to obtain the driveable area seen in the image. We propose two different methods to do so.

2.2.1. *Defining the relevant area of the image*

The first method uses a fixed mask to define the relevant area of the image (hereinafter referred to as "**fixed** restriction"). The fixed mask, that is used to define the relevant area, was determined by "adding up" all objects that can be seen in a pixel over all images in the training portion of the data set. The resulting image is then blurred using `gaussian_filter(sigma = 25)` from the scipy.ndimage.filters library. Afterwards all values below the threshold of $e - 1$ are discarded. The pixels whose value exceeds the limit form the mask, that is used to define the relevant area. The resulting mask, that was used for the evaluations is shown in figure 2.

Fig. 2. The mask that is used to define the relevant area if the fixed restriction is used.

The second method to define the relevant area of the image uses an adaptive approach to define the relevant area of the image (hereinafter referred to as **adaptive** restriction). As the name suggests, the adaptive restriction determines a different relevant area for every image. The approach uses the Softmax output by defining the relevant area as all pixels, that are in the convex hull of pixels, that are labeled as either street or sidewalk. This area is extended by the area of all pixels that have been labelled as terrain. A pixel is considered to be labeled as category X if the corresponding Softmax output is maximal for the pixel. An exemplary result of this method can be seen in figure 3.

Note that both methods have not been optimized yet. The parameters used were determined by randomly testing some of the values and making a decision based on sound judgment.

2.2.2. *Procedure of the method*

After the relevant area has been narrowed down, the entropy in that relevant area is filtered. Here, the following observations on the obtained entropy images are considered:

(1) In multiple cases, the neural network seems to be unsure, whether the road area is to be considered as "road" or "sidewalk" resulting in low amounts of entropy in the area. The cause for this uncertainty is most likely, that in several images the road surface is made out of paving stones, which is typically used

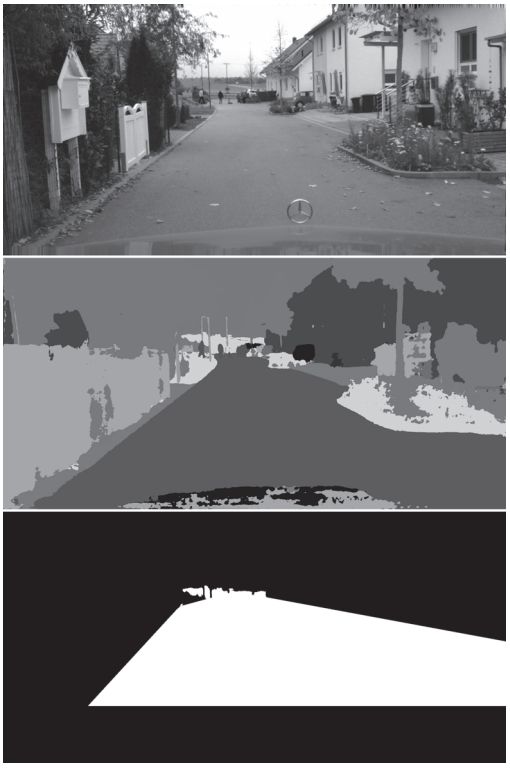

Fig. 3. Real image (top), semantic segmentation (middle) and adaptive restriction of the relevant area of this image (bottom).

for sidewalks. To ignore the entropy values resulting from the uncertainty between two classes, the maximum value of entropy, that can occur in this case, is calculated.

$$\begin{aligned}\max(H)|[2C] &= \sum_{i=1}^{19} \frac{p_i log_2(p_i)}{log_2(19)} \\ &\approx \sum_{i=1}^{2} \frac{p_i log_2(p_i)}{log_2(19)} \\ &= \frac{2 \cdot 0.5 \cdot log_2(0.5)}{log_2(\frac{1}{19})} \\ &= 0,2354\end{aligned} \tag{4}$$

Here, $\max(H)|[2C]$ denotes the maximum entropy value under the assumption, that all Softmax outputs except two are approximately zero. In order not to take into account entropy values that occur due to the uncertainty between two classes, a threshold value greater than 0,2354 is applied in the method.

(2) Between two semantic segments there is almost always a fine line, at which the output of the neural network transitions from one segments class to the other segments class. This results in a fine line of high entropy values. In order to discard those fine lines of high entropy and instead focus on dense regions of high entropy, a series of image filters is applied in the method.

The method starts with importing the Softmax outputs and calculating the pixelwise entropy. In the following first filtering step all entropy values that are below the threshold of 0.5 or outside of the defined relevant area of the image are discarded. The values are than scaled up to brightness values by multiplying them by 255. Afterwards a gaussian blur is applied using `cv2.GaussianBlur` with a kernel-size of 11×11. After that, all pixels with a value greater or equal 90 are set to 255 by applying `cv2.threshold(im, 90, 255, cv2.THRESH_BINARY)`. Thereafter `cv2.erode` is applied with two iterations and `cv2.dilate` is applied with four iterations. The idea for this method of filtering entropy values is taken from Rosebrock (2016). The image filters used are taken from the OpenCV library. As before, all used parameters were determined by randomly testing some values and making a decision based on sound judgment. An optimization of the parameters should be done in the future.

2.2.3. *Modifying the segmentation*

The remaining entropy segments have their origin in dense fields of high entropy. Therefore they can be used to modify the original segmentation in the third and final sub-process of the method. For this purpose a new 20^{th} class is introduced. All pixels that are filtered out by the method are then labeled with that 20^{th} class that was named "USO" (for "unidentified street object"). An exemplary result of the method can be seen in figure 4.

3. Results

The evaluations are performed using the `compute_metrics_per_image` function from the MetaSeg-framework by Matthias Rottmann (Rottmann et al., 2018). The function compares two images by scanning all segments of the first image and comparing them to the segments of the second image. The most important measure, that is taken is the Intersection-over-Union (abbr: IoU), which indicates the ratio of intersection to union. An exemplary application of that function is to scan a Ground Truth image and comparing its segments with the segments of a prediction for the same image. By doing so, the quality of the prediction can be evaluated.

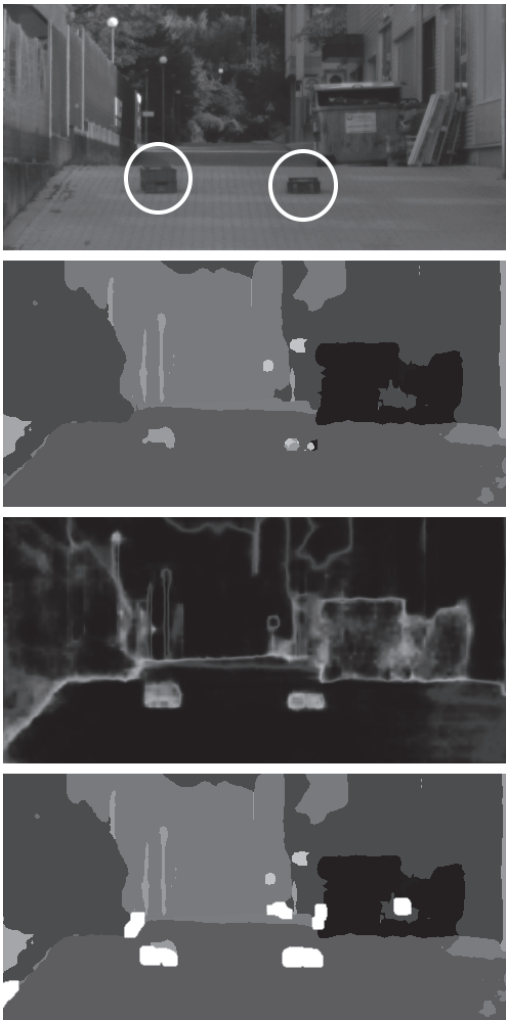

Fig. 4. From top to bottom: 1) Real image (with the two objects circled in white for enhanced visibility), 2) semantic segmentation / Inference, 3) visualization of the entropy, 4) modified segmentation (new segments in white for enhanced visibility).
The segmentation was modified and improved in the region of the two objects that were placed on the street. Further segments that were labeled as "USO" can be found in the region of the dumpster on the right side, slightly above the two objects and in the left foreground of the picture. (total of five false-positives).

3.1. *Evaluation of the restriction of the relevant area*

To evaluate the two different restriction methods, the number of Ground Truth Objects that are contained in the relevant area to a certain percentage is considered. The results of this evaluation are shown in Table 1.

Table 1. Number of objects that are contained in the relevant area for the specified proportion.

	adaptive	(%)	fixed	(%)
$>$ 0%	1864	100	1719	92.22
$>$ 20%	1864	100	1688	90.56
$>$ 40%	1857	99.62	1667	89.43
$>$ 60%	1842	98.82	1647	88.36
$>$ 80%	1823	97.80	1623	87.07
100%	1751	93.94	1566	84.01

Note: The columns headed with "(%)" indicate the proportion of the total quantity represented by the respective quantity to the left.

As can be seen in the table the adaptive restriction seems to outperform the fixed restriction, since more objects are contained for every specified proportion. By using the adaptive restriction, all 1864 objects of the dataset are contained in the relevant areas of the images by at least 20% of their area. If the fixed restriction is used, only 90.5% of the objects are contained in the relevant area by at least 20%. The adaptive restriction outperforms the fixed restriction by about 10% with only slight variations for containments of 40% to 100%. When using the adaptive restriction, 93.94% of the objects are completely contained in the relevant area. By using the fixed restriction only 84.01% of the objects are entirely contained. These evaluations have little meaning, if the size of the respective relevant area isn't taken into account. [Imagine not restricting the area at all. All objects would intersect by 100%. However, a lot of objects that are in "unimportant" areas of the image would be found by applying the method.] This is done indirectly by measuring the numbers of false positives, i.e. segments of the prediction that don't intersect with objects of the ground truth. By doing so, we can evaluate how effectively the relevant area was restricted by the respective methods.

3.2. *Analysis of false-positives*

For this evaluation the segments of the modified segmentations which are denoted USO_a (using the adaptive approach) and USO_f (using the fixed approach) were scanned for intersections with segments of the ground truth. Although the amount of false-positives might seem high at first glance, it is to be considered, that the ground-truth of the Lost and Found data set is very sparsely labeled, i.e. many objects aren't labeled at all. Thus, many of the false-positives shown in this evaluation aren't *true* false-positives, since they intersect with objects which are simply not labeled as such.

A better evaluation of the method regarding the false-positives should be done in the future by applying the method to denser labeled data sets or by subsequently labeling previously unlabeled objects in the data set.

Table 2. Number of false-positives.

	USO_a	USO_f
# obj	10170	5210
IoU $= 0.00$	9310	4428
IoU < 0.25	9905	4971
IoU < 0.50	10061	5110
IoU < 0.75	10148	5188

Note: The first line shows the total number of segments in the respective predictions.

As stated before, the evaluation of the false-positives shown in table 2 is only there to provide an indication of the size of the relevant areas. As the values show, the segmentation that was made using the adaptive restriction contains 10170 segments and therefore almost twice as much segments as the segmentation that was made using the fixed restriction, which contains 5210 segments. Note, that there is no difference between the two methods other than the restriction of the relevant area. From the values it can be deduced that the fixed restriction provides a relevant area, that is on average significantly smaller than the relevant area provided by the adaptive restriction.
Based on this insight, the adaptive restriction should no longer be considered to be outperforming the fixed restriction. The relevant area of the adaptive restriction contains a greater number of objects, which is at the expense of an averagely larger relevant area and thereby a larger number of false-positives.

3.3. *Evaluation of the different Predictions*

To evaluate the method, the following three segmentations were evaluated:

- **USO:** This segmentation contains only the unidentified street objects (abbr.: USOs) found using the method described above, i.e. everything that remained after the method was applied to the entropy of the Softmax Outputs (white segments in last image of figure 4). This segmentation is used to test whether objects are actually found by filtering the entropy.
- **NN&USO:** This segmentation contains the USOs as well as everything that has been labeled as non-driveable space in the segmentation, i.e. everything that was not labeled as street, sidewalk or terrain by the neural network. This segmentation is used to evaluate the final result, i.e. the modified segmentation, and contains all objects that were recognized by the neural network and the proposed method.
- **NN:** This segmentation contains everything, that was labeled as non-driveable space in the inference of the neural network. This segmentation serves as a baseline for the aforementioned modified segmentation and contains all objects that were recognized (but not necessarily identified, see 3.4) by the neural network.

Each of these segmentations can be made with either method to restrict the relevant area, which were explained earlier (fixed relevant area will be denoted with by $_f$; adaptive relevant area will be denoted by $_a$), leading to a total of six different segmentations that are evaluated in this section.
For the evaluation, the ground truth objects were scanned for intersections with the segmentations. The results of the evaluation are shown in table 3.

Table 3. Number of Ground Truth objects that fulfill the condition in the column on the left

	NU_a	NU_f	N_a	N_f	U_a	U_f
IoU > 0.0	1238	1144	1235	1142	541	494
IoU > 0.1	909	724	855	690	410	376
IoU > 0.2	795	642	738	595	333	304
IoU > 0.3	698	584	617	518	267	242
IoU > 0.4	609	525	523	442	197	176
IoU > 0.5	526	464	431	375	131	119
IoU > 0.6	419	370	325	283	69	65
IoU > 0.7	285	256	226	195	35	34
IoU > 0.8	156	139	113	97	13	13
IoU > 0.9	16	16	8	8	1	1

Note: The designations for the different predictions had to be further abbreviated in order to fit the table.
NN&USO $\rightarrow$ NU; NN $\rightarrow$ N; USO $\rightarrow$ U.

As can be seen in the table, the original objective of the work was only partially achieved. The number of ground-truth objects that intersect with the segmentation was increased from 1235 (N_a, baseline) to 1238 (NU_a, modified segmentation) for the adaptive restriction, and from 1142 (N_f) to 1144 (NU_f) for the fixed restriction. This means that only three respectively two objects that were previously undetected were detected by the application of our method. However, the quality of the prediction was significantly improved. Depending on the condition, the improvement on the number of objects ranges from 6,3% (from 855 to 909 objects at 'IoU > 0.1') to 22.0% (from 431 to 526 objects at 'IoU > 0.5') to 34.5% (from 113 to 156

objects at 'IoU > 0.8') up to 100% (from 8 to 16 objects at 'IoU > 0.9') when using the adaptive restriction. For the fixed restriction the number of objects was increased, too. The improvement on the number of objects ranges from 4.9% (from 690 to 724 objects at 'IoU > 0.1') to 23.7% (from 375 to 464 objects at 'IoU > 0.5') up to 100% (from 8 to 16 objects at 'IoU > 0.9'). The relatively low numbers of intersections with the segmentations U_a and U_f indicate, that many of the 1864 objects were recognized and identified by the neural network in good quality. This is why the entropy in the region of those objects was low and thus, no USO-segment was added.

3.4. *Further observations*

A phenomenon that emerged was, that objects were recognized but not identified by the neural network. This resulted in a potpourri of different classes that pixels of the object were labeled with. In almost all cases of that happening, the entropy was very high in the region of the object. This resulted in a clean detection of the object by our method, which then resulted in a single segment showing the object in the modified prediction. This kind of improvement of the prediction doesn't show in the evaluations, since even though there's a potpourri of classes in the region of the object, the object is recognized and therefore treated as a single segment during the evaluations. An example for the described phenomenon is shown in figure 5.

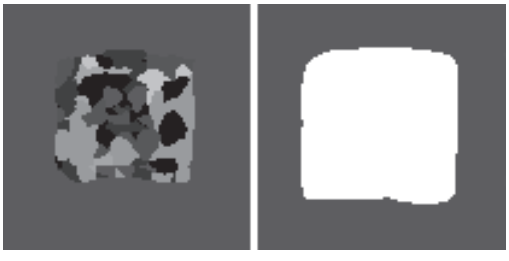

Fig. 5. A recognized but unidentified object before (left) and after (right) the method was applied.

4. Consideration from a safety perspective

From a safety perspective, the overlooking of objects during semantic segmentation represents a decisive threat to the functional safety of autonomous vehicle control. Our method represents a procedure to improve the quality of a semantic segmentation and does therefore contribute to the functional safety. However, there is currently no standard that defines the functional safety for autonomous driving. Standards to ensure functional safety in the automotive sector are usually given by ISO 26262 (International Organization for Standardization, 2018). However, these standards cannot be applied to autonomously driving vehicles, since, for example, lots of the measures used to rate a hazard in the ISO 26262 evaluate the controllability of a situation by the driver.
A standard which defines requirements and methods for the validation of artificial intelligence for autonomous driving is therefore urgently needed to ensure the safe and responsible usage of artificial intelligence for autonomous driving. Furthermore, a relevance rating of false-positives is needed, which extends the outdated and primitive counting of false-positives to a more informative metric.

5. Conclusion and outlook

We presented a method that can improve the quality of a semantic segmentation by calculating and filtering the entropy of the Softmax-outputs of a segmentation-CNN. As part of the method, two different methods to restrict the relevant area of the processed images were presented. Both methods have both advantages and disadvantages compared to the other method.
By applying the presented method on the Lost and Found data set, the original objectives could only be partially achieved. Only few previously undetected objects were detected by our method. However, the quality of the prediction could be significantly improved.
For the future, an optimization of the method by adjusting the used parameters is planned. Moreover, more precise metrics shall be sought in order to make the method more comparable to similar methods. Beyond that, the application of the method to other data sets will be performed. Here, we will strive to find a more completely labeled data set in order to be able to examine and evaluate our method in a more profound way.

References

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *The European Conference on Computer Vision (ECCV)*.

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

DeVries, T. and G. W. Taylor (2018, feb). Learning Confidence for Out-of-Distribution Detection in Neural Networks.

Guo, C., G. Pleiss, Y. Sun, and K. Q. Weinberger (2017). On calibration of modern neural networks. *34th International Conference on Machine Learning, ICML 2017 3*, 2130–2143.

Hendrycks, D. and K. Gimpel (2016). A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *CoRR abs/1610.02136*.

Hendrycks, D., M. Mazeika, and T. Dietterich (2018, dec). Deep Anomaly Detection with Outlier Exposure.

International Organization for Standardization (01.12.2018). Iso 26262:2018: Road vehicles - functional safety.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc.

Lee, K., H. Lee, K. Lee, and J. Shin (2018). Training confidence-calibrated classifiers for detecting out-of-distribution samples. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 1–16.

Liang, S., Y. Li, and R. Srikant (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (2017).

Long, J., E. Shelhamer, and T. Darrell (2014). Fully convolutional networks for semantic segmentation. *CoRR abs/1411.4038*.

Meinke, A. and M. Hein (2019). Towards neural networks that provably know when they don't know. pp. 1–14.

Pinggera, P., S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester (2016). Lost and found: Detecting small road hazards for self-driving vehicles. *IEEE International Conference on Intelligent Robots and Systems 2016-November*, 1099–1106.

Rosebrock, A. (2016). Detecting multiple bright spots in an image with python and opencv.

Rottmann, M., P. Colling, T.-P. Hack, R. Chan, F. Hüger, P. Schlicht, and H. Gottschalk (2018). Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities. *CoRR abs/1811.0*.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal 27*(3), 379–423.

Yosinski, J. and J. Clune (2015). Deep Neural Networks are Easily Fooled : High Confidence Predictions for Unrecognizable Images.