

# Criteria for Uncertainty-based Corner Cases Detection in Instance Segmentation

Florian Heidecker<sup>1</sup>, Ahmad El-Khateeb<sup>1</sup>, Maarten Bieshaar<sup>2</sup>, Bernhard Sick<sup>1</sup>

<sup>1</sup> Intelligent Embedded Systems, University of Kassel  
Wilhelmshöher Allee 73, 34121 Kassel, Germany

{florian.heidecker, akhateeb, bsick}@uni-kassel.de

<sup>2</sup> Bosch Center for Artificial Intelligence  
31132 Hildesheim, Germany

maarten.bieshaar@de.bosch.com

## Abstract

*The operating environment of a highly automated vehicle is subject to change, e.g., weather, illumination, or the scenario containing different objects and other participants in which the highly automated vehicle has to navigate its passengers safely. These situations must be considered when developing and validating highly automated driving functions. This already poses a problem for training and evaluating deep learning models because without the costly labeling of thousands of recordings, not knowing whether the data contains relevant, interesting data for further model training, it is a guess under which conditions and situations the model performs poorly. For this purpose, we present corner case criteria based on the predictive uncertainty. With our corner case criteria, we are able to detect uncertainty-based corner cases of an object instance segmentation model without relying on ground truth (GT) data. We evaluated each corner case criterion using the COCO and the NuImages dataset to analyze the potential of our approach. We also provide a corner case decision function that allows us to distinguish each object into True Positive (TP), localization and/or classification corner case, or False Positive (FP). We also present our first results of an iterative training cycle that outperforms the baseline and where the data added to the training dataset is selected based on the corner case decision function.*

## 1. Introduction

Various driving assistance systems rely on object detection, such as pedestrian or traffic sign detection. Object detection is also essential for highly automated driving, enabling the vehicle to navigate safely among other road users. Object instance segmentation further provides the instance mask besides the object class and bounding box. The object in-

stance segmentation model must work properly even under constantly changing weather, illumination, scenarios, participants, and other conditions, including corner cases.

Corner Cases [3, 19, 39] are strongly related to anomalies [9, 14, 37], outliers [14, 17], and novelties [9, 14] but also cover samples where the model fails [12, 19, 22, 36] and data relevant for model improvement [6, 39]. In [19], a categorization of corner cases for perception in automated driving is introduced, which includes a bunch of potential corner case sources covering the sensor, data content, temporal aspect, and the ML method. Corner cases of the ML method [19] are caused due to a lack of knowledge as the model has never encountered a similar sample before, the applied ML method themselves or adversarial samples [13]. However, it is difficult to find the data we are looking for because there are no labels, and we do not know whether they represent corner cases for the model and thus provide a value for training to improve the model.

Corner case detection enables data selection to be guided to identify valuable data and label it more efficiently, offering tremendous cost-saving potential. Besides, there are also other use cases, e.g., active learning [25], novelty detection [14, 37], and dataset construction, i.e., creating a training and testing dataset covering all relevant and therefore crucial situations.

This article presents novel corner case criteria for detecting uncertainty-based corner cases in an object instance segmentation task. These criteria cover the uncertainty in classification, bounding box regression, and instance mask prediction. Our goal is to provide the reader with a clear understanding of how the corner case criteria are made up and function. Furthermore, we evaluate the corner case criteria and thus discuss the limitations and advantages of each corner case criterion. The evaluation is based on two real datasets, COCO [29] and NuImages [7], to show that our corner case criteria are not dataset-specific and can be ap-

plied to other datasets. A first approach shows that with our corner case criteria as input feature and a decision function, it is possible to detect corner cases based on the predictive uncertainty and classify predicted objects as True Positive (TP), localization and/or classification corner case, or False Positive (FP) reliably.

The remainder of this article is structured as follows: Section 2 provides a brief overview of related work. Section 3 describes the overall approach and provides a basic understanding of using the presented corner case criteria. In Section 4, we explain each corner case criteria and evaluate their feature importance. Section 5 explains the experimental setup. With the corner case criteria, we can classify whether an object is a corner case or not in Section 6 and also perform a cycle based model training (cf. Section 7). Finally, Section 8 concludes the article’s key message and provides possible future work.

## 2. Related Work

Before we continue, we should briefly discuss what uncertainty-based corner cases are and how they are defined. Regarding ML, corner cases are cases caused by an erroneous, malfunction, or incorrect behavior of the trained ML model [12, 19, 22, 36]. According to the examples given in [19], these can be anomalies, outliers, novelties, distribution shifts, and the ML model itself, which delivers erroneous predictions due to the implemented architecture, methodology, training parameters, etc. Investigating uncertainty is important for evaluating the prediction of the model. Moreover, considering model performance, it is possible to reveal information on individual samples or identify samples in which the model is maximally uncertain — which are corner cases.

Various approaches already address the challenge of detecting corner cases, such as Out-of-Distribution detection, e.g., using auxiliary Generative Adversarial Networks [33], uncertainty-based Out-of-Distribution [40] with Monte-Carlo (MC) Dropout [11] or techniques for entropy maximization [8]. Lis et al. [30], and Xia et al. [43] use image resynthesis to detect pixel anomalies in image data. Bolte et al. [4] also detect corner cases by fusing information from semantic segmentation and object predictions. In [34], Ouyang et al. present an approach for corner case detection for a classification task using modified distance-based surprise adequacy. Pfeil et al. [36] use ensembles to detect corner cases in trajectory data, and in [42], a variational autoencoder is used.

However, we take a different approach by modeling the uncertainty with MC-Dropout [11] comparable to [20, 21, 31, 38]. This allows us to derive the uncertainty with respect to each detected object from the model results. Subsequently, the criteria presented in this article are used to identify corner cases based on the predictive uncertainty and

additionally to classify them into different corner case categories (cf. Section 4).

## 3. Overall Approach

Our approach aims to identify specific corner cases [19] based on criteria derived from the uncertainty.

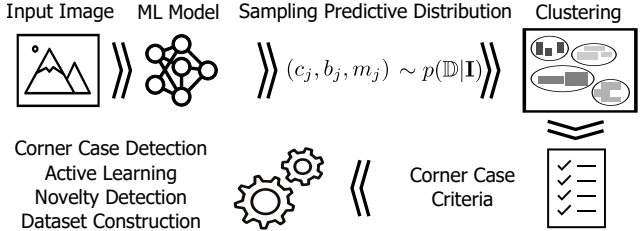


Figure 1. Overall approach to derive and apply uncertainty-based corner case criteria.

### 3.1. Uncertainty and Probabilistic Object Detection

We consider a probabilistic definition of object detectors [10] and instance segmentation networks [15]. Probabilistic object detection extends conventional object detection and instance segmentation by quantifying uncertainty. Let  $\mathbb{D} = \{d_1, \dots, d_M\}$ ,  $M \in \mathbb{N}^+$  denote object detections for a given input image  $\mathbf{I}$  with width  $W$  and height  $H$ . A detection  $d_j$  incorporating instance segmentation is a tuple  $(c_j, b_j, m_j)$ <sup>1</sup> consisting of information about the object class score  $c = (c_1 \dots c_k)$  of  $k \in \mathbb{N}^+$  classes, bounding box  $b = (x_1, y_1, x_2, y_2) \in \mathbb{R}^4$ , and binary instance mask  $m \in \{0, 1\}^{W \times H}$ . Our approach relies on the modeling of the predictive distribution  $p(\mathbb{D}|\mathbf{I})$  [1]. This predictive distribution includes both the aleatoric and epistemic model uncertainty [23]. However, modeling this distribution is challenging due to the high dimensionality of the input and the target space (e.g., there can be any number of predictions in the image). There are numerous methods for approximating this distribution, including Bayesian methods, variational inference, and sample-based representation of the distribution.

Our approach relies on the latter (i.e., sampling from  $p(\mathbb{D}|\mathbf{I})$ ), cf. [10, 20, 21, 31, 38]. There are numerous methods, including MC-Dropout [11], ensembles [26], which also model epistemic uncertainty, or probFRCNN [15], which only considers the aleatoric uncertainty. Our experiments use an instance segmentation model with MC-Dropout (modified Mask R-CNN model [21]) and cluster multiple forward pass predictions  $R$  (repetitions) of the same original input image. Hence, instead of  $M$  detections, we end up with  $L \in \mathbb{N}^+$  detections with  $L \gg M$ . In the following, we denote the set of detections (i.e., samples) as  $\tilde{\mathbb{D}} = \{d_1, \dots, d_L\}$ . Subsequently, we cluster the samples

<sup>1</sup>For object detection, it is a tuple with class score and bounding box.

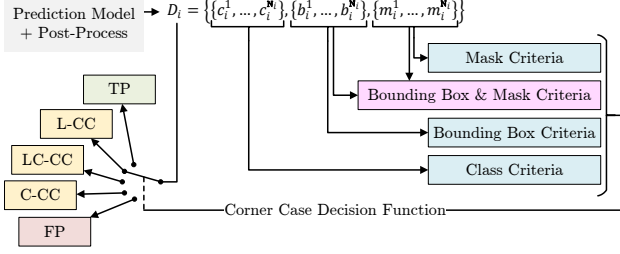


Figure 2. Our approach assumes that the object instance segmentation model plus a possible post-process (gray) provides several predictions, consisting of class score  $c$ , bounding box  $b$ , and instance mask  $m$ , per object detection  $D_i$ . The prediction variance is used to determine the uncertainty in the ML model. The uncertainty analysis is utilized in the class score, bounding box, and instance mask criteria (single knowledge criteria – blue) or combined in the bounding box & mask criteria (combined knowledge criteria – magenta). The values from the criteria are interpreted as features to assign the object detections  $D_i$  with a decision function to one of the defined corner case categories. The defined categories are True Positive (TP – green), False Positive (FP – red), and the corner cases (yellow) for location (L-CC), class (C-CC), and both (LC-CC).

from the predictive distribution based on the bounding boxes (cf. [21]). After this step we have  $i$ -th clusters  $D_i = \{\{c_i^{(1)}, \dots, c_i^{(N_i)}\}, \{b_i^{(1)}, \dots, b_i^{(N_i)}\}, \{m_i^{(1)}, \dots, m_i^{(N_i)}\}\}$  where  $c_i^{(1)}$ ,  $b_i^{(1)}$ , and  $m_i^{(1)}$  denote the first detection associated with the  $i$ -th cluster. Moreover, let  $N_i \in \mathbb{N}^+$  be the size of the  $i$ -th cluster. Clustering the detections from the predictive distribution is the starting point for our uncertainty-driven corner case criteria. An overall schematic representation of our approach to derive uncertainty-based corner case criteria and possible applications using them is depicted in Fig. 1.

The key idea behind the uncertainty-based criteria is to use them to detect corner cases. In other words, these criteria should permit us to draw conclusions (in an unsupervised fashion without any labels) about possible deficiencies in the model’s performance. In this context, we differentiate between corner case categories introduced later in Section 3.2. We derive the criteria based on the clustering and the associated classification scores, bounding boxes, and instance masks. Our corner case criteria can be distinguished into single and combined knowledge criteria; see Fig. 2. As single knowledge criteria in Section 4.1, 4.2, and 4.3, we refer to criteria that only use the uncertainty of one feature, e.g., class score criteria, while combined knowledge criteria in Section 4.4 use several sources, e.g., bounding box and mask. Hence, it depends on the application which criterion is useful for the uncertainty analysis. It is also possible to exclude specific criteria if the model provides, for example, no instance mask.

Two further steps are necessary to enable the detection of

corner cases with the given uncertainty-based criteria. First, the definition of corner case categories, as well as the conditions that must apply so that a detected object may represent a corner case, see Section 3.2, and secondly, a decision function that assigns the predicted detection  $D_i$  to one of the defined corner case based on the criteria, see Section 6.

### 3.2. Corner Cases Categories

To define corner case categories, we need an object-based metric to decide if the detections are corner cases with which the trained model has issues. Typically, we would think of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative as categories, but this separation is much too rough. Hence, we decided to use the  $\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{TP}{TP+FP+FN}$  combined with information if the predicted class is correct or wrong to differentiate the categories. With the metric in place, we defined that a match of ground truth (GT) and detection  $D_i$  with an  $\text{IoU} > 0.5$  and correct class is a good prediction (True Positive (TP)). The threshold of  $\text{IoU} = 0.5$  is freely selectable and separates good object prediction where no further model improvement is required from the rest. In the future, this threshold might be too low, as a higher IoU value would definitely be preferable. However, objects below an IoU value of 0.5 are not considered in any mean average precision (mAP) calculations, so we think these samples improve the model the most. We also defined a lower limit of  $\text{IoU} = 0.1$  because, beyond a certain point, the match between GT and prediction is too low. Altogether, we then get the following corner case categories for the class assignment of the object detections:

1. True Positive (TP): Class is correctly predicted, and there is a match between detection  $D_i$  and GT with an IoU above 0.5.
2. Localization Corner Case (L-CC): Class is correctly predicted, but the match between detection  $D_i$  and GT has an IoU between 0.1 and 0.5.
3. Classification Corner Case (C-CC): Class is wrongly predicted, but the match between detection  $D_i$  and GT is still above an IoU of 0.5.
4. Localization & Classification Corner Case (LC-CC): Class is wrongly predicted, and the match between detection  $D_i$  and GT has an IoU between 0.1 and 0.5.
5. False Positive (FP): All detections  $D_i$  with an IoU below 0.1, which also includes all detections without a GT match.
6. False Negative (FN): It would be very interesting to know, but there is no way to get this information from the model because there is no prediction to work with.

In [5, 24], we find an almost identical categorization with the same metric where they talk about ML model errors in image and video data instead of corner cases as we do. But these model errors, i.e., corner cases, are exactly the

problematic samples where the model needs further training samples to improve. On top, the presented categories from [5, 24] are used to evaluate the model results better but can also be used for labeling the detections as we do. To then classify the objects using the uncertainty-based corner case criteria as features (cf. Section 4). Also, the purpose of the categorization differs as [5, 24] uses them to compare the ML model evaluation results. Apart from the different naming of the categories, one difference in the categorization is that we do not consider duplicate detections as corner cases — we treat them as FP as another detection with a higher IoU exists. [24] also has another category that deals with a temporal error since video data is considered. This category currently has no meaning because we consider only single images and no sequences. However, this category must be considered for potential corner cases as soon as our methodology is applied to video data and temporal corner cases arising from uncertainty.

## 4. Corner Case Criteria

The corner case criteria are essential to the approach presented in Section 3. Each criterion below is calculated based on the knowledge of a single cluster  $D_i$ , which describes and quantifies the model’s uncertainty with a value. Besides, we treat the detected objects separately and do not aggregate them over one image to get one or multiple averaged values for each image.

### 4.1. Class Score Criteria

First, we examine the class uncertainty, which greatly influences object detection since it determines which objects are in the image. For the class score, we define four criteria that address the confidence of the class probability. First, we determine the detections’ mean class scores  $\bar{D}_{c^k}$  and standard deviation  $\sigma_{c^k}$  for all classes  $k$ , see Eq. 1. Then, the index of  $k$  is calculated for the class with the highest  $k_{\max} = \operatorname{argmax} \bar{D}_{c^k}$  and second-highest  $k_{2\text{nd}}$  class confidence. In summary, we obtain four class score criteria  $\bar{D}_{c^{k_{\max}}}$ ,  $\sigma_{c^{k_{\max}}}$ ,  $\bar{D}_{c^{k_{2\text{nd}}}}$ , and  $\sigma_{c^{k_{2\text{nd}}}}$ .

$$\bar{D}_{c^k} = \frac{\sum_{j=1}^N c_j^k}{N}, \quad \sigma_{c^k} = \sqrt{\frac{\sum_{j=1}^N (c_j^k - \bar{D}_{c^k})^2}{N-1}} \quad (1)$$

The class probability gives us the same information as any other classification model without uncertainty modeling. Still, in combination with the  $\sigma$ , we can measure how confident the ML model is in its detections.

### 4.2. Bounding Box Criteria

For the spatial uncertainty of the model’s detections, i.e., the bounding box predictions, we primarily consider the standard deviation of the four bounding box edges as a criterion

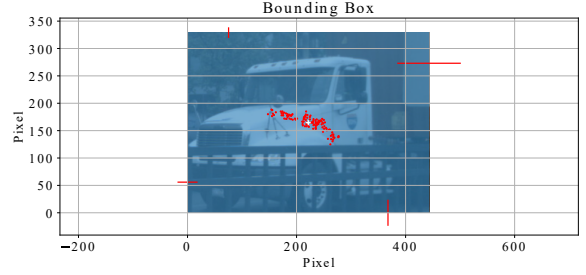


Figure 3. Bounding box uncertainty. The standard deviation  $\sigma_b$  of each bounding box edge is represented by red lines, and red dots represent the spread of the bounding box center point. The blue box with the truck in the middle represents the mean bounding box.

for the bounding box uncertainty. An example of the bounding box uncertainty is depicted in Fig. 3.

We first calculate the mean  $\bar{D}_b$  of the bounding box cluster and then the standard deviation  $\sigma_b$  in Eq. 2.

$$\bar{D}_b = \frac{\sum_{j=1}^N b_j}{N}, \quad \sigma_b = \sqrt{\frac{\sum_{j=1}^N (b_j - \bar{D}_b)^2}{N-1}} \quad (2)$$

To see if the standard deviation of the bounding boxes is huge or small compared to its size, we normalize the standard deviation  $\sigma_b$  based on the mean bounding box size. A low standard deviation  $\sigma_b$  indicates that the detector predicts the corresponding edge reliably. In contrast, a high standard deviation  $\sigma_b$  indicates difficulties separating the object from the neighboring object or background. Instead of the format  $b = \{x_1, y_1, x_2, y_2\}$  also,  $b^* = \{c_x, c_y, w, h\}$  can be used, whereby it is to be considered that both variants behave differently regarding the uncertainty. For example, width  $w$  and height  $h$  describe only the uncertainty of the bounding box size and no spatial uncertainty in the  $x$ - or  $y$ -direction, as all others do.

We also consider the IoU between the mean bounding box  $\bar{D}_b$  and all other predictions  $b_j$  as a further criterion. Therefore, we calculate the mean IoU  $\bar{iou}_b$  and the IoU standard deviation  $\sigma_{iou_b}$  in Eq. 3 and 4.

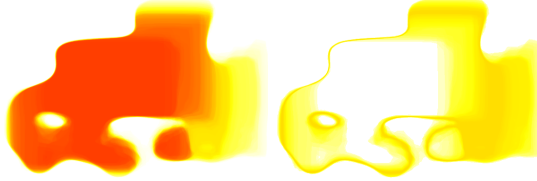
$$\bar{iou}_b = \frac{\sum_{j=1}^N \text{IoU}(b_j, \bar{D}_b)}{N} \quad (3)$$

$$\sigma_{iou_b} = \sqrt{\frac{\sum_{j=1}^N (\text{IoU}(b_j, \bar{D}_b) - \bar{iou}_b)^2}{N-1}} \quad (4)$$

The mean value  $\bar{iou}_b$  tells us how significant the overlap is between the individual bounding boxes within a cluster. The larger the value, the less uncertain the ML model is about its predictions. The standard deviation of the IoU  $\sigma_{iou_b}$ , on the other hand, provides information about the spread of the IoUs.



(a) The left shows the binary mask mean  $\bar{D}_m$ . In the right image, the bounding box of the mask mean  $\bar{D}_{m_{\text{box}}}$  is drawn in green, and the bounding box mean  $\bar{D}_b$  in red.



(b) The left side shows the mean value of the masks as a heat map before the binary mask was generated. The right shows the standard deviation of the mask also in a heat map representation.

Figure 4. Instance mask uncertainty example.

### 4.3. Instance Mask Criteria

Besides the bounding box, the used detection model also provides instance masks, which we use to analyze the spatial uncertainty. In Fig. 4, we see an example of the mask mean  $\bar{D}_m$  (top left). Besides the mean, we also calculate the mask’s standard deviation  $\sigma_m$ , which is depicted in the bottom right of Fig. 4. The heat map of the standard deviation  $\sigma_m$  reveals that the model well recognizes the left side of the driver’s cab since the standard deviation  $\sigma_m$  is small and represented by a thin line. There is greater uncertainty in the vicinity of the truck radiator as the area is also slightly colored. However, the model shows the largest uncertainty in the transition from truck to trailer, which the intense coloring indicates.

We need some criteria for the instance mask to use this knowledge. The first criterion reuses the bounding box uncertainty criterion described in Eq. 2. To calculate the uncertainty for the instance masks related to the individual edges, the bounding box surrounding the mask must be determined with  $m_j^{\text{box}} = \{c_x, c_y, w, h\} = \text{size}(m_j)$ . The mean instance mask  $\bar{D}_m$  of the object detection is calculated and converted to binary, by

$$\bar{D}_m = \begin{cases} 1 & \text{if } \frac{\sum_{j=1}^N m_j}{N} > 0.5 \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

We obtain the mean value of the mask’s bounding box with  $\bar{D}_{m_{\text{box}}} = \{c_x, c_y, w, h\} = \text{size}(\bar{D}_m)$  and the standard deviation  $\sigma_{m_{\text{box}}}$  with Eq. 6, which can be used to analyze the edges of the mask bounding box. As with the bounding box in Section 4.2, we normalize the standard deviation  $\sigma_{m_{\text{box}}}$

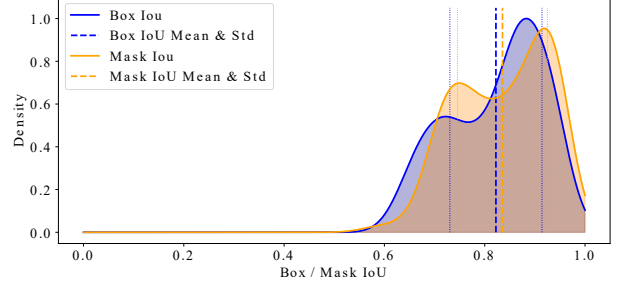


Figure 5. Distribution of the IoU between the mean bounding box  $\bar{D}_b$  (blue) or mean mask  $\bar{D}_m$  (orange) and all other bounding box  $b_j$  or instance mask  $m_j$ .

based on the mean mask bounding box size.

$$\sigma_{m_{\text{box}}} = \sqrt{\frac{\sum_{j=1}^N (m_j^{\text{box}} - \bar{D}_{m_{\text{box}}})^2}{N - 1}} \quad (6)$$

The next criterion we consider appears in [31] for result evaluation purposes and is comparable to our bounding box criterion Eq. 3 already listed above. Besides Eq. 3, we also adapt Eq. 4, where the  $\text{IoU}(m_j, \bar{D}_m)$  calculation is slightly changed and results from the mean instance mask  $\bar{D}_m$  and all other instance masks  $m_j$ . As with the bounding box criterion in Section 4.2, the mean  $\overline{\text{IoU}_m}$  and the standard deviation  $\sigma_{\text{IoU}_m}$  provide essential information about the similarity of the masks in the cluster and their spatial uncertainty with respect to the model prediction.

The first mask criterion interprets the instance mask like a bounding box, which reduces the information when considering the uncertainty. Therefore, we propose another criterion to capture the area of the binary instance masks better. To do this, we first calculate the area for each instance mask  $A_j = \sum_{H \times W} m_j$  and then the mean and standard deviation with Eq. 7. We then normalize  $\sigma_{A_m}$  by the mean  $\bar{A}_m$ . The standard deviation  $\sigma_{A_m}$  provides information about the variance in the size of the mask. A small value is good because all instance masks are similar in size. However,  $\sigma_{A_m}$  is not decisive whether they lie spatially together, but this knowledge provides the mask IoU criterion  $\overline{\text{IoU}_m}$  and  $\sigma_{\text{IoU}_m}$ .

$$\bar{A}_m = \frac{\sum_{j=1}^N A_j}{N}, \quad \sigma_{A_m} = \sqrt{\frac{\sum_{j=1}^N (A_j - \bar{A}_m)^2}{N - 1}} \quad (7)$$

### 4.4. Bounding Box & Mask Criteria

Finally, we present criteria that combine the bounding box and instance mask uncertainty. The assumption for these criteria is that a well-trained ML model provides predictions where the bounding box and instance mask match very well and increase if not. In other words, they match regarding location and size. The first criterion covers the mismatch

in terms of IoU  $iou_{\text{mis}}$  between the mean bounding box  $\bar{D}_b$  and the mean bounding box enclosing the instant mask  $\bar{D}_{m_{\text{box}}}$  by calculating the IoU between both. An example is depicted in Fig. 4 (top right). A high IoU value indicates that both matching well.

$$iou_{\text{mis}} = \text{IoU}(\bar{D}_b, \bar{D}_{m_{\text{box}}}) \quad (8)$$

The first criterion only considers the similarity of the mean values but not the distribution of the predicted bounding boxes or instance masks. However, to compare both distributions, we must first calculate the IoU between the mean and each predicted bounding box  $iou_{b_j}$  and instance mask  $iou_{m_j}$  within the cluster. If the distribution of the bounding boxes and instance masks is equal, then the two model heads are identical in their predictions, even if the variance of the distribution is large. If the distribution is unequal, the predictions are unequal as the two model heads produce contradictory predictions.

$$iou_{b_j} = \text{IoU}(b_j, \bar{D}_b), \quad iou_{m_j} = \text{IoU}(m_j, \bar{D}_m) \quad (9)$$

To obtain the IoU distribution, we use the kernel density estimation Eq. 10 with a Gaussian kernel and compute the probability distribution function with Eq. 11 [1]. An example of the IoU distribution for the bounding box and instance mask is shown in Fig. 5. The two distributions of the bounding box and mask are not identical. In addition, the distribution reveals the uncertainty arising from the instance mask Fig. 4 and bounding box Fig. 3. Afterward, we obtain the probability distribution function of the IoU kernel density estimation for the bounding box  $p_b = f(iou_{b_j})$  and the instance mask  $p_m = f(iou_{m_j})$ .

$$KDE(x, t) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h\sqrt{2\pi}} \exp^{-0.5(\frac{x_i-t}{h})^2} \quad (10)$$

$$f(t) = \int_0^1 KDE(x, t) dx \quad (11)$$

With both distributions at hand, we can calculate the Kullback-Leibler-Divergence (KL) [1] to compare the bounding box distribution with the mask distribution  $\text{KL}(p_b|p_m)$  and the opposite  $\text{KL}(p_m|p_b)$ . Besides, we also calculated the Jensen-Shannon Distance (JS) [32]  $\text{JS}(p_b|p_m)$  in Eq. 12 and the Wasserstein Distance [28]  $\text{EMD}(p_b|p_m)$  in Eq. 13, also called Earth Mover Distance, to compare both distributions. While the JS can be interpreted as merging and averaging the total KL divergence to the average distribution of  $\frac{p+q}{2}$ , the EMD is more complex.

$$\text{JS}(p, q) = \sqrt{\frac{\text{KL}(p|m) + \text{KL}(q|m)}{2}}, \quad m = \frac{p+q}{2} \quad (12)$$

While  $d_{ij}$  describes the distance between  $p_i$  and  $q_i$ ,  $f_{ij}$  represents the flow. The flow between  $p_i$  and  $q_i$  has to be minimized to reduce the overall costs with  $\sum_{i=1}^m \sum_{j=1}^n f_{ij} =$

$\min(\sum_{i=1}^m p_i \sum_{j=1}^n q_j)$ . If the optimal flow is found, the EMD [28] is defined as

$$\text{EMD}(p, q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}}. \quad (13)$$

## 4.5. Criteria Evaluation

Next, we examine the value of our proposed uncertainty-based corner case criteria. To this end, we performed forward and backward sequential feature selection<sup>2</sup> in combination with a decision tree to determine the top 10 corner case criteria for the classification task.

Top 10 features selected by forward sequential selection:

- class score criteria:  $\sigma_{c^{\text{kmax}}}$ ,  $\bar{D}_{c^{\text{k2nd}}}$ , and  $\sigma_{c^{\text{k2nd}}}$
- bounding box criteria:  $x_1, x_2, w$  of  $\sigma_b$ ,  $iou_b$ , and  $\sigma_{iou_b}$
- mask criteria:  $h$  of  $\sigma_m$
- bounding box & mask criteria:  $\text{KL}(p_b|p_m)$

Top 10 features selected by backward sequential selection:

- class score criteria:  $\sigma_{c^{\text{kmax}}}$ ,  $\bar{D}_{c^{\text{k2nd}}}$ , and  $\sigma_{c^{\text{k2nd}}}$
- bounding box criteria:  $c_x$  of  $\sigma_b$ , and  $iou_b$
- mask criteria:  $c_x$  of  $\sigma_m$ , and  $\sigma_{A_m}$
- bounding box & mask criteria:  $iou_{\text{mis}}$ ,  $\text{KL}(p_m|p_b)$ , JS

Furthermore, we evaluated the Pearson and Spearman correlation of bounding box criteria and box IoU, instance mask criteria and mask IoU, and bounding box & mask criteria concerning box IoU and mask IoU (cf. Supplementary Material, Section 10). The correlations were very low except for some corner case criteria, which have better results for the COCO and the NuImages dataset. These include  $iou_b$  and  $iou_{\text{mis}}$  in correlation with the box IoU, as well as  $iou_m$ ,  $\sigma_{iou_m}$ ,  $iou_{\text{mis}}$  and EMD in correlation with the mask IoU. Overall, the mask criteria and bounding box & mask criteria showed better correlation than the bounding box criteria. However, in Section 6, we used all features for classification because the performance was worse when using any subset of corner case criteria. This can also be assumed due to the poor correlation, which requires all available information to be considered when classifying the objects.

## 5. Experimental Setup

As mentioned in Section 3, we used a modified Mask R-CNN model [20, 21, 31, 38]. The major difference in comparison to the original Mask R-CNN model from He et al. [18] are the added MC-Dropout [11] layers to sample from the model, which allows us to model the predictive uncertainty [23]. We used a dropout rate of 0.2 and did  $R = 10$  and  $R = 100$  repetitions for each input image to see the difference in the results, as it takes 10 times longer to do 100 compared to 10 repetitions. In the best case of 100 repetitions, we get 400 predictions if the input

<sup>2</sup>sklearn.feature\_selection.SequentialFeatureSelector

Mask R-CNN with	no NMS	$N_i = 10$	$N_i = 100$	no NMS	$N_i = 10$	$N_i = 100$
Dataset	COCO			NuImages		
Size of Test Subset (Images)	4952			14884		
Labeled Objects (Test Subset)	36335			136074		
Model Predictions	34827	67569	73344	106133	180557	197743
TP Correct Class & IoU > 0.5	16040 (46.1%)	23778 (35.2%)	24554 (33.5%)	59921 (56.5%)	100309 (55.6%)	106441 (53.8%)
L-CC Correct Class & 0.5 > IoU > 0.1	3378 (9.7%)	3213 (4.8%)	2901 (4.0%)	11667 (11.0%)	9800 (5.4%)	7546 (3.8%)
C-CC Wrong Class & IoU > 0.5	1275 (3.7%)	1714 (2.5%)	1749 (2.4%)	3206 (3.0%)	4373 (2.4%)	4429 (2.2%)
LC-CC Wrong Class & 0.5 > IoU > 0.1	1077 (3.1%)	1189 (1.8%)	1205 (1.6%)	2931 (2.8%)	3050 (1.7%)	2893 (1.5%)
FP No Matches	13057 (37.5%)	37675 (55.8%)	42935 (58.5%)	28408 (26.8%)	63025 (34.9%)	76434 (38.7%)
FN Missed Objects	14565	6441	5926	58349	18542	14765
Bounding Box $mAP_{IoU>0.5}$	0.374	0.488	0.504	0.266	0.414	0.439
Mask $mAP_{IoU>0.5}$	0.372	0.470	0.484	0.245	0.357	0.376

Table 1. Test dataset details, number of matching predictions, corner cases, and model performance.

images contain, for example, four objects. To cluster these predictions, we used Bayesian Gaussian Mixture [2] in the post-process, and as features, the bounding box parameters  $b = (x_1, y_1, x_2, y_2)$  as described in [21]. We get four object detections  $D_i$  containing  $N_i = 100$  predictions each under ideal conditions. For comparison reasons, we also used the original Mask R-CNN model and disabled Non-maximum Suppression (NMS) as proposed in [10, 15, 16] to get, in the best case, multiple predictions of each object to cluster and determine the corner case criteria.

For the evaluation, we selected the COCO [29] and NuImages [7] datasets to analyze our corner case criteria because both datasets are very different. For example, the NuImages test subset is around three times bigger than COCO, as listed in Table 1. In both cases, we are using the official validation subset as a test subset to evaluate the corner case criteria and the overall approach. Furthermore, we randomly picked 10% from the training subset to get a new validation split because both datasets offer only two labeled subsets. With the rest of the training subset, we trained the model. In the case of the COCO dataset, we trained the model with all 81 provided classes. However, in the NuImages dataset, we trained the model with the following 17 classes (*animal, child, pedestrian, other pedestrians (e.g., police), e-scooter, stroller, barrier, traffic cone, debris, bicycle, motorcycle, car, bus, truck, trailer, off-highway, other vehicles (e.g., police)*). For the assignment of detection to the corner case categories defined in Section 3.2, we use the Hungarian Matching Algorithm [41] combined with the class to identify the matches between GT and the predicted object detections  $D_i$ .

As mentioned before, both datasets are quite different.

Also, the mAP of the model performance differs; on the COCO [29] dataset, the model is better than on the NuImages [7] dataset, which is surprising because COCO has 81 classes and NuImage 17. Since the model without NMS misses many objects, the number of predicted objects is also quite low compared to the MC-Dropout model. Another expected behavior also occurs, namely that more objects are predicted as repetitions increase. The model ideally manages to predict every object in the image at least once during the repetition, which in turn also leads to objects being predicted where there are none (No Matches in Table 1). It is also interesting to note that the predictions classified as corner cases have a similar quantity of objects. Our current assumption for the MC-Dropout models is that the contained objects overlap to a high degree, i.e., objects representing a corner case at a repetition of 10 also represent one at 100 repetitions.

## 6. Corner Case Decision Function

With the criteria shown in Section 4, we have a proper baseline to quantify how certain or uncertain the model is with respect to its prediction about each object. It should also be noted that we do not rely on GT data to calculate the corner case criteria, apart from the training of the object detection model in Section 5. The question is, can we classify the corner case categories from the overview in Section 4? – Yes, to some extent.

A "corner case" decision function is needed to achieve this goal. For this purpose, we used five different classification methods from sklearn [35]: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Gaussian Process Classifier (GPC), and a Multi-layer Perceptron

(MLP) classifier with only one hidden layer (100 neurons). Besides that, all default parameter settings were kept.

Mask R-CNN	no NMS	$N_i = 10$	$N_i = 100$	no NMS	$N_i = 10$	$N_i = 100$
Dataset	COCO			NuImages		
DT	0.56	<b>0.5</b>	0.43	0.56	<b>0.5</b>	0.43
RF	0.5	0.38	0.39	0.5	0.38	0.39
SVM	0.43	0.31	0.28	0.42	0.31	0.28
GPC	<b>0.58</b>	<b>0.5</b>	<b>0.45</b>	<b>0.58</b>	<b>0.5</b>	<b>0.45</b>
MLP	0.56	0.44	0.4	0.56	0.44	0.4

Table 2. Class-weighted F1-score results using Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Gaussian Process Classifier (GPC) and Multi-layer Perception (MLP). Input features are the corner cases criteria (cf. Section 4), derived from the mentioned Mask R-CNN model and the class labels provided by the corner case categories (cf. Section 3.2).

As input feature, we used all presented corner case criteria, and as output, the five classes: TP, L-CC, C-CC, LC-CC, and FP (cf. Table 1). We trained each classifier model with the validation dataset, but we had to use Random Undersampling [27] because of the class imbalance. Table 2 reveals that the Gaussian Process Classifier outperforms all other methods. Interestingly, the classification results for the corner case criteria of the Mask R-CNN model with disabled NMS are much better than for those models where the corner case criteria were calculated on the MC-Dropout model results. Thus, there seems to be a connection that many object predictions (cf. Table 1), which mainly concern non-existent objects, have a negative effect on the classification, e.g., COCO with a repetition of 10 and 100.



Figure 6. The Confusion matrix of our decision function, using a Gaussian Process Classifier ( $N_i = 10$ ) and presented corner cases criteria as input, to distinguish objects in TP, corner case, or FP.

The class-weighted F1-score of the Gaussian Process

Classifier ( $N_i = 10$ ) decision function is 0.5, and the related confusion matrix in Fig. 6 shows further details. A breakdown of the corner cases into individual classes is possible to a certain extent. The L-CC and LC-CC corner cases are challenging to separate from FP, which is not surprising since FP contains object detections whose IoU is only minimally smaller than 0.1 and would otherwise perhaps represent a corner case. Furthermore, FP contains duplicates of objects from other categories. This happens if there are several matches for a GT label, but only the best match is selected as a corner case or TP.

## 7. Corner Case Detection for Data Reduction

In this section, we apply our approach, as shown in Fig. 2, to a real scenario where each data and annotation costs money. Therefore, we evaluate using our corner case criteria to reduce the amount of training data so that we need fewer annotations but still achieve good results compared to the baseline model trained with the entire training dataset. We perform multiple training cycles to improve the model iteratively by adding model-selected corner case data to the training dataset to achieve this goal. We split the NuImages training dataset into four equal-sized subsets. In the 1st cycle, we train the model on the 1st training data subset and execute the inference and corner case decision on the 2nd training data subset. Every image that contains at least one corner case (L-CC, C-CC, or LC-CC) is added to the training of the 2nd cycle, and we repeat the process.

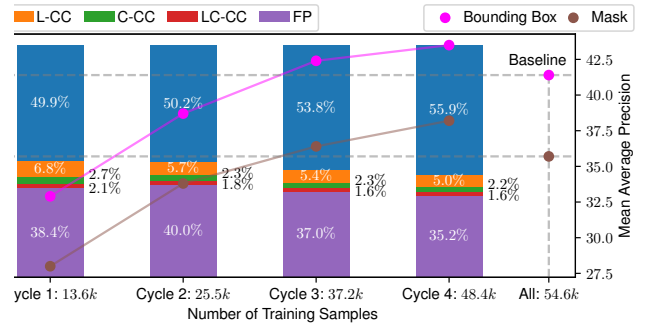


Figure 7. Four training cycles to iteratively improve the model by adding model-selected corner case data to the training dataset. Performance comparison between the cyclically trained model and a baseline model trained with all data.

As shown in Fig. 7, the proportion of TP prediction on the testing dataset increases after each training cycle, and the FN decreases simultaneously. Adding only the model detected corner cases to the training dataset improves the model performance at each cycle iteration. We also see that the number of corner cases (for the model) decreases due to the improvement. Since we find a corner case in almost every image, the reduction in training data used after four cycles is moderate, with 6k images (~10%) compared to

when all data is used. However, our model performs better than the baseline trained on all data. In future works, it would be interesting to investigate the model performance if only images with two or more corner cases were added to the training dataset, and we have maybe a reduction of 30% or more and an even higher cost saving.

## 8. Conclusion and Future Work

The approach presented in this article aims to detect corner cases based on uncertainties and use them for better model training (reduction of annotation costs). Therefore, we introduce corner case criteria derived from the predictive uncertainty of probabilistic object detectors. The corner case criteria determine the uncertainty of the predicted class score, bounding box, and instance mask, which classifies the detected object. It should be noted that the criteria are computed purely from the modeled uncertainty, assuming a sampling predictive distribution with subsequent clustering, where no ground truth (GT) is required. The presented corner case decision function utilizes the output of the corner case criteria as input features and assigns the detected objects to one of the following corner case categories: True Positive (TP), Localization Corner Case (L-CC), Classification Corner Case (C-CC), Localization & Classification Corner Case (LC-CC), and False Positive (FP). The classification results show we can identify many corner cases based on the introduced criteria. On the other hand, it also shows that the distinction between FP and corner cases is challenging. Finally, we presented our first results on an iterative training cycle that outperforms the baseline model by using the corner cases as a selection strategy for model training to reduce costs regarding annotations.

In the future, we also want to investigate further how the model performance behaves if we only add images to the training dataset containing multiple corner cases and if our results are also statistically consistent. With the ability to detect and classify uncertainty-based corner cases, we are also interested in developing relevance metrics to determine the value of the recognized corner cases [6, 39]. To explicitly state how important or valuable a corner case is, e.g., concerning vehicle safety or model training.

## Acknowledgements

This work results from the project KI Data Tooling (19A20001O) funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

## References

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, New York, NY, USA, 2006. 2, 6
- [2] David M. Blei and Michael I. Jordan. Variational Inference for Dirichlet Process Mixtures. *Bayesian Analysis*, 1(1):121–144, 2006. 7
- [3] Daniel Bogdoll, Jasmin Breitenstein, Florian Heidecker, Maarten Bieshaar, Bernhard Sick, Tim Fingscheidt, and J. Marius Zöllner. Description of Corner Cases in Automated Driving: Goals and Challenges. In *Proc. of ICCV, ERCVAD Workshop*, pages 1023–1028, Virtual, 2021. 1
- [4] Jan-Aike Bolte, Andreas Bär, Daniel Lipinski, and Tim Fingscheidt. Towards Corner Case Detection for Autonomous Driving. In *Proc. of IV*, pages 438–445, Paris, France, 2019. 2
- [5] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. TIDE: A General Toolbox for Identifying Object Detection Errors. In *Proc. of ECCV*, pages 558–573, Virtual, 2020. 3, 4
- [6] Jasmin Breitenstein, Florian Heidecker, Maria Lyssenko, Daniel Bogdoll, Maarten Bieshaar, J. Marius Zöllner, Bernhard Sick, and Tim Fingscheidt. What Does Really Count? Estimating Relevance of Corner Cases for Semantic Segmentation in Automated Driving. In *Proc. of ICCV, Workshops*, pages 3991–4000, Paris, France, 2023. 1, 9
- [7] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proc. of CVPR*, pages 11618–11628, Seattle, WA, USA, 2020. 1, 7
- [8] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation. In *Proc. of ICCV*, pages 5108–5117, Montreal, QC, Canada, 2021. 2
- [9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3):1–58, 2009. 1
- [10] Di Feng, Ali Harakeh, Steven Waslander, and Klaus Dietmayer. A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving. *IEEE Trans. on ITS*, pages 1–20, 2021. 2, 7
- [11] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. of ICML*, pages 1050–1059, New York, NY, USA, 2016. 2, 6
- [12] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A Survey of Uncertainty in Deep Neural Networks. *arXiv preprint arXiv:2107.03342*, 2022. 1, 2
- [13] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In *Proc. of ICLR*, pages 1–10, San Diego, CA, USA, 2015. 1
- [14] Christian Gruhl, Bernhard Sick, and Sven Tomforde. Novelty Detection in Continuously Changing Environments. *Future Generation Computer Systems*, 114:138–154, 2021. 1
- [15] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic Object Detec-

- tion: Definition and Evaluation. In *Proc. of WACV*, pages 1031–1040, Snowmass Village, CO, USA, 2020. 2, 7
- [16] Ali Harakeh, Michael Smart, and Steven L. Waslander. BayesOD: A Bayesian Approach for Uncertainty Estimation in Deep Object Detectors. *arXiv preprint arXiv:1903.03838*, 2019. 7
- [17] D. M. Hawkins. *Identification of Outliers*. Springer, Dordrecht, 1980. 1
- [18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. of ICCV*, pages 2980–2988, Venice, Italy, 2017. 6
- [19] Florian Heidecker, Jasmin Breitenstein, Kevin Rösch, Jonas Löhdefink, Maarten Bieshaar, Christoph Stiller, Tim Fingscheidt, and Bernhard Sick. An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving. In *Proc. of the IV*, pages 644–651, Nagoya, Japan, 2021. 1, 2
- [20] Florian Heidecker, Abdul Hannan, Maarten Bieshaar, and Bernhard Sick. Towards Corner Case Detection by Modeling the Uncertainty of Instance Segmentation Networks. In *Proc. of ICPR, IADS Workshop*, pages 361–374, Milan, Italy, 2021. 2, 6
- [21] Florian Heidecker, Ahmad El-Khateeb, and Bernhard Sick. Sampling-based Uncertainty Estimation for an Instance Segmentation Network. *arXiv preprint arXiv:2305.14977*, 2023. 2, 3, 6, 7
- [22] Sebastian Houben, Stephanie Abrecht, Maram Akila, Andreas Bär, Felix Brockherde, Patrick Feifel, Tim Fingscheidt, Sujan Sai Gannamaneni, Seyed Eghbal Ghobadi, Ahmed Hammam, Anselm Haselhoff, Felix Hauser, Christian Heinzemann, Marco Hoffmann, Nikhil Kapoor, Falk Kappel, Marvin Klingner, Jan Kronenberger, Fabian Küppers, Jonas Löhdefink, Michael Mlynarski, Michael Mock, Firas Mualla, Svetlana Pavlitskaya, Maximilian Poretschkin, Alexander Pohl, Varun Ravi-Kumar, Julia Rosenzweig, Matthias Rottmann, Stefan Rüping, Timo Sämann, Jan David Schneider, Elena Schulz, Gesina Schwalbe, Joachim Sicking, Toshika Srivastava, Serin Varghese, Michael Weber, Sebastian Wirkert, Tim Wirtz, and Matthias Woehrle. Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety. *arXiv preprint arXiv:2104.14235*, 2021. 1, 2
- [23] Eyke Hüllermeier and Willem Waegeman. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 110(3): 457–506, 2021. 2, 6
- [24] Wenhe Jia, Lu Yang, Zilong Jia, Wenyi Zhao, Yilin Zhou, and Qing Song. TIVE: A Toolbox for Identifying Video Instance Segmentation Errors. *arXiv preprint arXiv:2210.08856*, 2022. 3, 4
- [25] Daniel Kottke, Marek Herde, Christoph Sandrock, Denis Huseljic, Georg Kreml, and Bernhard Sick. Toward Optimal Probabilistic Active Learning Using a Bayesian Approach. *Machine Learning*, 110(6):1199–1231, 2021. 1
- [26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Proc. of NIPS*, pages 6402–6413, Long Beach, CA, USA, 2017. 2
- [27] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. 8
- [28] E. Levina and P. Bickel. The Earth Mover’s distance is the Mallows distance: some insights from statistics. In *Proc. of ICCV*, pages 251–256, Vancouver, BC, Canada, 2001. 6
- [29] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv preprint arXiv:1405.0312*, 2015. 1, 7
- [30] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the Unexpected via Image Resynthesis. In *Proc. of ICCV*, pages 2152–2161, Seoul, Korea (South), 2019. 2
- [31] Doug Morrison, Anton Milan, and Epameinondas Antonakos. Uncertainty-aware Instance Segmentation using Dropout Sampling, 2019. Accessed 15-Feb.-2019. 2, 5, 6
- [32] Frank Nielsen. On a Generalization of the Jensen–Shannon Divergence and the Jensen–Shannon Centroid. *Entropy*, 22(2):1–24, 2020. 6
- [33] Julia Nitsch, Masha Itkina, Ransalu Senanayake, Juan Nieto, Max Schmidt, Roland Siegwart, Mykel J. Kochenderfer, and Cesar Cadena. Out-of-Distribution Detection for Automotive Perception. In *Proc. of ITSC*, pages 2938–2943, Indianapolis, IN, USA, 2021. 2
- [34] Tinghui Ouyang, Vicent Sanz Marco, Yoshinao Isobe, Hideki Asoh, Yutaka Oiwa, and Yoshiki Seo. Corner Case Data Description and Detection. In *Proc. of ICSE Workshop*, pages 19–26, Madrid, Spain, 2021. 2
- [35] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *JMLR*, 12(85): 2825–2830, 2011. 7
- [36] Jerg Pfeil, Jochen Wieland, Thomas Michalke, and Andreas Theissler. On Why the System Makes the Corner Case: AI-based Holistic Anomaly Detection for Autonomous Driving. In *Proc. of IV*, pages 337–344, Aachen, Germany, 2022. 1, 2
- [37] Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. A Review of Novelty Detection. *Signal Processing*, 99:215–249, 2014. 1
- [38] Tobias Riedlinger, Marius Schubert, Karsten Kahl, and Matthias Rottmann. *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, chapter Uncertainty Quantification for Object Detection: Output- and Gradient-Based Approaches, pages 251–275. Springer International Publishing, Cham, Switzerland, 2022. 2, 6
- [39] Kevin Rösch, Florian Heidecker, Julian Truetsch, Kamil Kowol, Clemens Schick Tanz, Maarten Bieshaar, Bernhard Sick, and Christoph Stiller. Space, Time, and Interaction: A Taxonomy of Corner Cases in Trajectory Datasets for Automated Driving. In *Proc. of SSCI, IEEE CIVTS*, pages 1–8, Singapore, 2022. 1, 9

- [40] Andreas Sedlmeier, Thomas Gabor, Thomy Phan, Lenz Belzner, and Claudia Linnhoff-Popien. Uncertainty-Based Out-of-Distribution Detection in Deep Reinforcement Learning. *arXiv preprint arXiv:1901.02219*, 2019. 2
- [41] Russell Stewart, Mykhaylo Andriluka, and Andrew Y. Ng. End-to-End People Detection in Crowded Scenes. In *Proc. of CVPR*, pages 2325–2333, Las Vegas, NV, USA, 2016. 7
- [42] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. Misbehaviour Prediction for Autonomous Driving Systems. In *Proc. of ICSE*, pages 359–371, Seoul, Korea (South), 2020. 2
- [43] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L. Yuille. Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation. *arXiv preprint arXiv:2003.08440*, 2020. 2

# Criteria for Uncertainty-based Corner Cases Detection in Instance Segmentation

## Supplementary Material

### 9. Overview

This supplementary material gives further information about our main work, which has not been mentioned there mainly due to space limitations.

### 10. Evaluation

We evaluate the corner case criteria listed in Section 4 by testing them with our experimental setup from Section 5.

#### 10.1. Class Score Uncertainty

We depict the results for the NuImages datasets in Fig. 8 to evaluate the class score criteria defined in Section 4.1. The left y-axis of each plot shows the IoU between mean bounding box  $\bar{D}_b$  and GT, while the right y-axis represents the class score. The plot on the top contains the cases if the class is correct:  $L\text{-}CC$  and  $TP$ . The middle presents the cases  $LC\text{-}CC$  and  $C\text{-}CC$ , where the class is wrong, and the last plot finally shows the  $FP$ . The x-axis represents the predicted objects. Detections are sorted by IoU and smoothed to reveal the trend in class scoring criteria.

We note that the average  $\sigma_{c^{k\max}}$  is increasing while the averaged  $\bar{D}_{c^{k\max}}$  and IoU is decreasing, see Fig. 8 (top). The middle plot behaves unusually because the  $\bar{D}_{c^{k\max}}$  is slowly decreasing and then stagnates while the  $\sigma_{c^{k\max}}$  is decreasing along with the IoU, which is the opposite of the behavior in the top plot of Fig. 8. In our opinion, this is related to the increase of  $\bar{D}_{c^{k2\text{nd}}}$ , which is shown in green in Fig. 8.  $\bar{D}_{c^{k\max}}$  in the bottom plot forms almost a perfectly straight line with the value corresponding to approx. 0.67 for NuImages. The value of  $\sigma_{c^{k\max}}$  is almost constant. If we take this value and look at the top plot, we see that  $\bar{D}_{c^{k\max}}$  is above this value if the class is correct and the standard deviation is the same or smaller. In the middle of Fig. 8, where the class is wrong,  $\bar{D}_{c^{k\max}}$  is for most parts below this value, and the standard deviation is the same or larger. This circumstance indicates a threshold to separate correctly and incorrectly classified objects.

#### 10.2. Spatial Uncertainty

For the spatial uncertainty, we present three different groups of criteria. The first group covers the bounding box uncertainty (Section 4.2), the second the instance mask uncertainty (Section 4.3), and the last group includes criteria resulting from the combination of both uncertainties (Section 4.4). We consider the correlation between the IoU and the criteria to evaluate how well the model criteria describe the model performance. The IoU is calculated between GT, the mean bounding box (Box IoU), and the mean instance

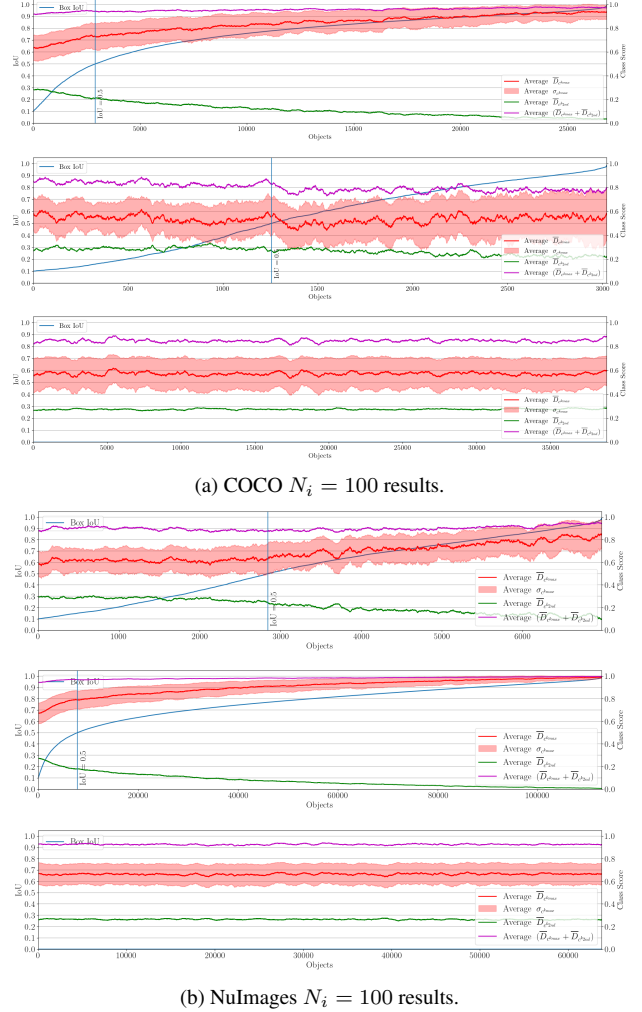


Figure 8. NuImages results for class score uncertainty. Object evaluation of the max class score mean  $\bar{D}_{c^{k\max}}$  (red), 2nd-max class score mean  $\bar{D}_{c^{k2\text{nd}}}$  (green), standard deviation  $\sigma_{c^{k\max}}$  (red area) and mean sum (magenta).

mask (Mask IoU). Subsequently, the linear correlation is calculated with *Pearson* (*Pear.*) and the monotonic correlation with *Spearman* (*Spear.*). The monotonic correlation is interesting as Box IoU and Mask IoU monotonically increase. Table 3 lists the calculated correlation values. The linear correlation is low, except for a few values. The *Spearman* correlation always delivers better results, but they are moderate. The criteria that stand out the most are  $iou_b$  for the bounding boxes,  $iou_m$  for the instance masks, and  $iou_{\text{mis}}$  for the combination of the bounding box and mask.

Bounding Box Criteria													
Criterion		COCO no NMS		COCO $N_i = 10$		COCO $N_i = 100$		NuImages no NMS		NuImages $N_i = 10$		NuImages $N_i = 100$	
		Pear.	Spear.	Pear.	Spear.	Pear.	Spear.	Pear.	Spear.	Pear.	Spear.	Pear.	Spear.
$\sigma_b$	$x_1$	-0.17	-0.05	-0.09	-0.05	-0.10	-0.17	-0.27	-0.35	-0.15	-0.26	-0.14	-0.38
	$y_1$	-0.16	-0.09	-0.10	-0.11	-0.13	-0.25	-0.21	-0.29	-0.12	-0.14	-0.16	-0.29
	$x_2$	-0.17	-0.05	-0.09	-0.06	-0.10	-0.19	-0.27	-0.35	-0.15	-0.26	-0.14	-0.39
	$y_2$	-0.16	-0.08	-0.10	-0.10	-0.13	-0.23	-0.20	-0.30	-0.13	-0.18	-0.17	-0.32
	$c_x$	-0.17	-0.08	-0.09	-0.08	-0.10	-0.21	-0.27	-0.37	-0.15	-0.29	-0.14	-0.41
	$c_y$	-0.16	-0.12	-0.10	-0.13	-0.13	-0.28	-0.21	-0.34	-0.14	-0.22	-0.16	-0.37
	$w$	-0.13	-0.06	-0.08	-0.07	-0.16	-0.21	-0.21	-0.31	-0.16	-0.23	-0.25	-0.40
	$h$	-0.15	-0.11	-0.10	-0.11	-0.18	-0.26	-0.21	-0.30	-0.13	-0.15	-0.22	-0.31
$\overline{iou}_b$		<b>0.23</b>	0.17	<b>0.16</b>	0.18	<b>0.24</b>	<b>0.33</b>	<b>0.35</b>	<b>0.43</b>	0.23	0.35	<b>0.32</b>	<b>0.50</b>
$\sigma_{iou_b}$		0.02	0.06	-0.02	-0.03	-0.14	-0.19	-0.01	-0.10	-0.09	-0.17	-0.21	-0.33
Mask Criteria													
$\sigma_{m_{box}}$	$c_x$	<b>-0.20</b>	-0.11	-0.08	-0.11	-0.09	-0.24	-0.32	-0.37	-0.14	-0.30	-0.14	-0.42
	$c_y$	-0.16	-0.13	-0.09	-0.15	-0.13	<b>-0.29</b>	-0.21	-0.36	-0.16	-0.28	-0.17	-0.43
	$w$	-0.19	-0.11	-0.12	-0.11	-0.20	-0.25	<b>-0.34</b>	-0.33	<b>-0.24</b>	-0.30	<b>-0.32</b>	-0.45
	$h$	<b>-0.20</b>	-0.13	-0.13	-0.14	<b>-0.21</b>	<b>-0.29</b>	-0.32	-0.34	-0.19	-0.25	-0.25	-0.41
$\overline{iou}_m$		<b>-0.22</b>	-0.17	<b>-0.15</b>	<b>-0.19</b>	<b>0.29</b>	<b>0.35</b>	<b>-0.34</b>	<b>-0.38</b>	<b>-0.24</b>	<b>-0.37</b>	<b>0.40</b>	<b>0.56</b>
$\sigma_{iou_m}$		<b>0.31</b>	<b>0.22</b>	<b>0.21</b>	<b>0.22</b>	-0.20	-0.25	<b>0.47</b>	<b>0.47</b>	<b>0.31</b>	<b>0.43</b>	-0.30	-0.44
$\sigma_{A_m}$		-0.06	0.01	-0.11	-0.15	<b>-0.21</b>	<b>-0.31</b>	-0.09	-0.07	-0.17	-0.32	-0.29	<b>-0.50</b>
Bounding Box & Mask Criteria													
Box IoU	$iou_{mis}$	<b>0.29</b>	<b>0.34</b>	<b>0.25</b>	<b>0.29</b>	<b>0.27</b>	<b>0.30</b>	<b>0.45</b>	<b>0.56</b>	<b>0.38</b>	<b>0.49</b>	<b>0.39</b>	<b>0.49</b>
	$KL(p_b p_m)$	0.07	0.19	0.10	0.16	0.06	0.07	0.08	0.10	0.15	0.28	0.12	0.19
	$KL(p_m p_b)$	-0.09	0.18	-0.06	0.13	-0.07	0.07	-0.12	-0.04	-0.07	0.22	-0.08	0.17
	JS	0.14	0.14	0.12	0.13	0.12	0.13	0.15	0.25	0.19	0.25	0.20	0.26
	EMD	-0.01	<b>0.30</b>	0.06	<b>0.25</b>	0.13	0.24	0.04	0.06	0.15	<b>0.44</b>	0.31	0.44
Mask IoU	$iou_{mis}$	<b>0.34</b>	<b>0.33</b>	<b>0.26</b>	<b>0.28</b>	<b>0.28</b>	<b>0.30</b>	<b>0.58</b>	<b>0.59</b>	<b>0.40</b>	<b>0.47</b>	<b>0.40</b>	<b>0.47</b>
	$KL(p_b p_m)$	0.08	<b>0.22</b>	0.11	0.17	0.07	0.08	0.27	0.35	0.16	0.30	0.15	0.21
	$KL(p_m p_b)$	-0.06	<b>0.26</b>	-0.07	0.14	-0.07	0.09	0.20	0.32	-0.07	0.26	-0.07	0.21
	JS	0.19	0.21	<b>0.15</b>	0.15	0.16	0.16	0.15	0.27	<b>0.24</b>	0.30	0.25	0.31
	EMD	0.01	<b>0.31</b>	0.06	<b>0.25</b>	0.14	0.25	<b>0.46</b>	<b>0.50</b>	0.17	<b>0.44</b>	<b>0.34</b>	<b>0.45</b>

Table 3. Correlations of *Bounding Box Criteria* and Box IoU, *Instance Mask Criteria* and Mask IoU, and *Bounding Box & Mask Criteria* with Box IoU and Mask IoU.

## 11. Examples

This section depicts a few examples of object detections (Fig. 9) and the calculated criteria values (Table 4).

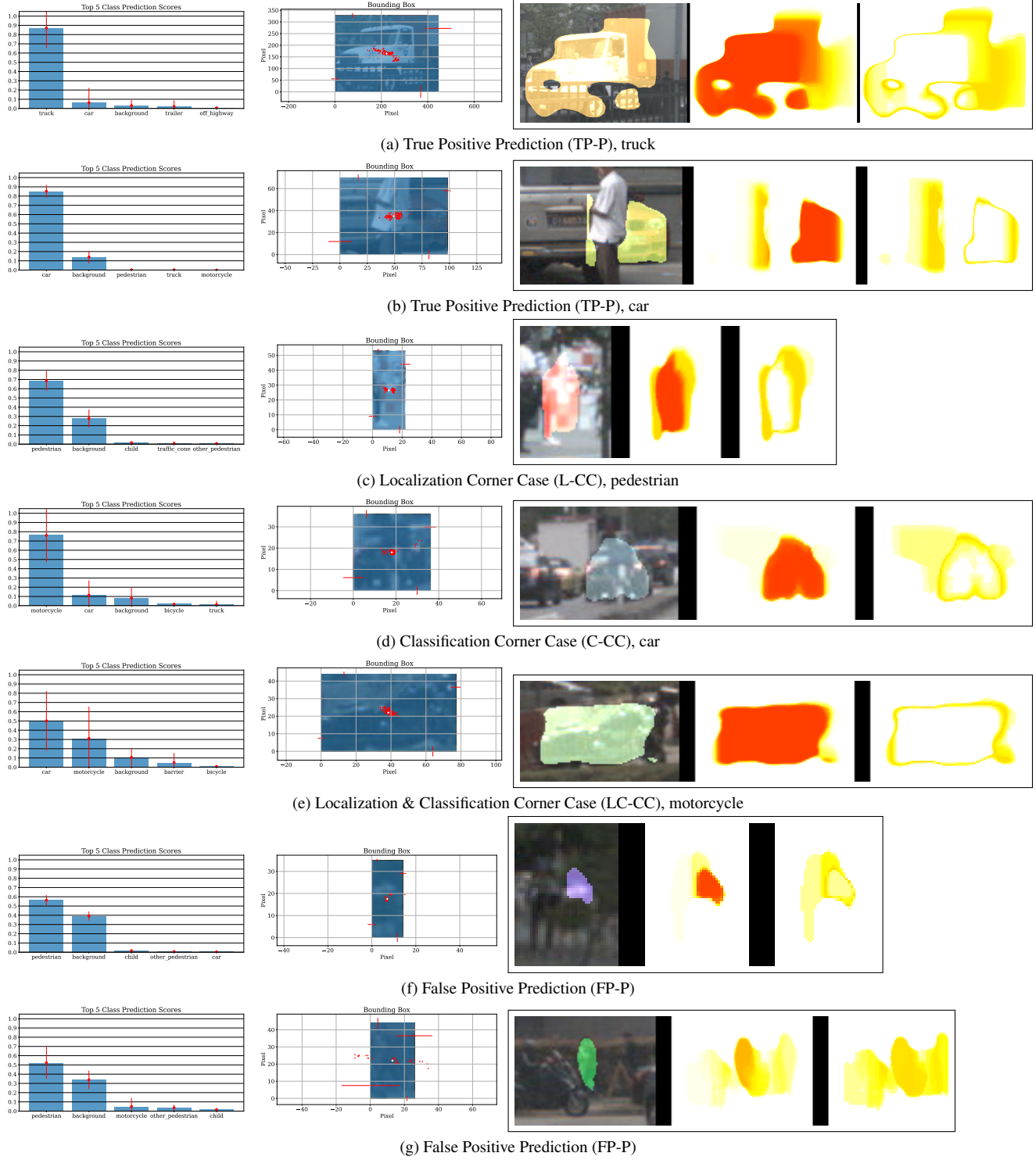


Figure 9. Uncertainty results of some NuImages examples.

		NuImages examples						
		9a	9c	9b	9d	9e	9f	9g
Corner Case Categories		TP-P	TP-P	L-CC	C-CC	LC-CC	FP-P	FP-P
GT	Box IoU	0.65	0.795	0.376	0.752	0.344	0.0	0.0
	Mask IoU	0.571	0.653	0.222	0.494	0.136	0.0	0.0
Class Score Criteria								
$\overline{D}_{c^{kmax}}$		0.871	0.852	0.686	0.761	0.501	0.563	0.523
$\sigma_{c^{kmax}}$		0.214	0.069	0.108	0.29	0.322	0.054	0.174
$\overline{D}_{c^{k2nd}}$		0.063	0.14	0.279	0.114	0.311	0.392	0.339
$\sigma_{c^{k2nd}}$		0.16	0.068	0.096	0.158	0.343	0.048	0.1
Bounding Box Criteria								
$\sigma_b$	$x_1$	0.032	0.105	0.098	0.129	0.02	0.123	0.627
	$y_1$	0.072	0.055	0.042	0.051	0.06	0.051	0.028
	$x_2$	0.125	0.031	0.145	0.072	0.038	0.098	0.379
	$y_2$	0.027	0.036	0.024	0.047	0.027	0.013	0.062
	$c_x$	0.068	0.052	0.118	0.077	0.024	0.109	0.493
	$c_y$	0.042	0.025	0.016	0.03	0.027	0.029	0.038
	$width$	0.12	0.116	0.076	0.14	0.038	0.046	0.316
	$height$	0.068	0.079	0.059	0.078	0.075	0.048	0.058
$\overline{iou}_b$		0.822	0.824	0.765	0.849	0.896	0.852	0.456
$\sigma_{iou_b}$		0.092	0.053	0.06	0.108	0.054	0.113	0.327
Instance Mask Criteria								
$\sigma_{m_{box}}$	$c_x$	0.069	0.064	0.119	0.077	0.024	0.163	0.678
	$c_y$	0.041	0.04	0.045	0.033	0.015	0.127	0.073
	$width$	0.123	0.129	0.101	0.126	0.04	0.088	0.308
	$height$	0.069	0.144	0.056	0.063	0.035	0.232	0.078
$\overline{iou}_m$		0.835	0.774	0.751	0.896	0.94	0.813	0.343
$\sigma_{iou_m}$		0.09	0.072	0.084	0.134	0.029	0.196	0.288
$\sigma_{A_m}$		0.159	0.222	0.109	0.078	0.044	0.251	0.311
Bounding Box & Mask Criteria								
$iou_{mis}$		0.952	0.654	0.808	0.918	0.814	0.301	0.362
$KL(p_b p_m)$		0.049	0.294	0.254	0.214	3.057	0.235	0.162
$KL(p_m p_b)$		0.044	0.697	0.383	0.401	0.44	0.959	0.135
$JS(p_b p_m)$		0.107	0.278	0.258	0.236	0.349	0.269	0.188
$EMD(p_b p_m)$		0.266	0.424	0.684	0.169	1.966	0.42	0.241

Table 4. Corner Case Criteria Results of the NuImages examples form Fig.9.