

RESEARCH

Open Access



Background-foreground segmentation for interior sensing in automotive industry

Claudia Drygala^{1*} , Matthias Rottmann¹, Hanno Gottschalk¹, Klaus Friedrichs² and Thomas Kurbiel²

*Correspondence:

drygala@uni-wuppertal.de

¹School of Mathematics and Science, IZMD & IMACM, University of Wuppertal, Wuppertal, Germany
Full list of author information is available at the end of the article

Abstract

To ensure safety in automated driving, the correct perception of the situation inside the car is as important as its environment. Thus, seat occupancy detection and classification of detected instances play an important role in interior sensing. By the knowledge of the seat occupancy status, it is possible to, e.g., automate the airbag deployment control. Furthermore, the presence of a driver, which is necessary for partially automated driving cars at the automation levels two to four can be verified. In this work, we compare different statistical methods from the field of image segmentation to approach the problem of background-foreground segmentation in camera based interior sensing. In the recent years, several methods based on different techniques have been developed and applied to images or videos from different applications. The peculiarity of the given scenarios of interior sensing is, that the foreground instances and the background both contain static as well as dynamic elements. In data considered in this work, even the camera position is not completely fixed. We review and benchmark three different methods ranging, i.e., Gaussian Mixture Models (GMM), Morphological Snakes and a deep neural network, namely a Mask R-CNN. In particular, the limitations of the classical methods, GMM and Morphological Snakes, for interior sensing are shown. Furthermore, it turns, that it is possible to overcome these limitations by deep learning, e.g. using a Mask R-CNN. Although only a small amount of ground truth data was available for training, we enabled the Mask R-CNN to produce high quality background-foreground masks via transfer learning. Moreover, we demonstrate that certain augmentation as well as pre- and post-processing methods further enhance the performance of the investigated methods.

Keywords: Background-foreground segmentation; Background subtraction; Interior sensing; Gaussian mixture model; Morphological snakes; Mask R-CNN; Augmentation; Color spaces; Contrast enhancement

1 Introduction

Interior Sensing is of high importance for automated driving. For instance, interior sensing aims at seat occupancy detection and classification [1, 2]. The classes may range from “Person”, “Child seat” and “Animal” to “Everyday object”. This knowledge about the seat occupancy can be used e.g. for smart airbag deployment control systems [3]. While the activation of the airbag could save a person’s life in case of an accident, it could lead to

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



Figure 1 Left: An RGB image of the test set. Right: The corresponding binary segmentation mask by which the foreground instances are separated of the background

serious injuries [4, 5] or even to death [6–8] for a child, which is sitting in a rear-facing child seat on the passenger seat. In the case of partially autonomous vehicles at the levels two to four (defined by [9]), a driver has to be present in the car. Thus it is necessary to verify, if a person is present on the driver seat [2]. Lastly, also the back seats of the car are of interest just as the front seats. Thinking of the so called “Forgotten Baby Syndrome” [7, 10] the system could give an alarm, if a forgotten child would be detected on a back seat.

In this work, we suggest seat occupancy detection by background-foreground segmentation methods. Only the extracted foreground instances, belonging to the classes “Person”, “Child Seat” or “Object”, should be considered for the classification. The motivation behind this approach is to realize the classification task independently of the car’s interior features and thus achieve better generalization.

Background-foreground segmentation [11–13], also known as background-foreground detection [14–16] or background subtraction [16–20], is an intensively studied field in computer vision. In recent years, several methods have been developed addressing various scenarios. These methods are based on completely different techniques. As it is described extensively in the survey [16], the approaches range from classical statistics based to modern methods, which incorporate deep convolutional neural networks [21].

The goal of this work is to introduce a dataset for the training of the background-foreground segmentation task and benchmark three methods in the given setting. The benchmark is performed on the quality of the generated background-foreground masks (see Fig. 1). Note that the minimization of the computational costs is not of interest for this work. The three methods we selected are based on different techniques:

1. Gaussian Mixture Model: A classical statistical method for background subtraction.
2. Morphological Snakes: A classical approach to object detection bases on active contours.
3. Mask R-CNN: A modern method solving the instance segmentation task by using deep neural networks.

In this work, we investigate the limitations of each approach. To ensure the comparability of the methods, all of them have been tested on a challenging test set of 100 real-world images. The test set is part of the dataset which is introduced by this work, named the ISSO dataset (Interior Sensing and Seat Occupancy). The dataset consists of 1300 annotated real-world images extracted of videos recorded by employees of the company APTIV in Wuppertal, Germany which are splitted into a training set of 1100 images, a validation and a test set of 100 images each. The images of the ISSO dataset describe scenarios of the interior of 13 different cars, as shown in Fig. 1. Scenarios of interior sensing are highly complex, since the foreground instances and the background can be both, dynamic and static. In this work, even the camera position varies slightly from car to car. Additionally, the impact of environmental effects has to be taken into account, like different weather conditions, shadows, traffic lights and vibrations.

Although only a rather small amount of 1100 real-world annotated images for the training is available, we demonstrate with the help transfer learning that it is possible to generate background-foreground masks of high quality with a Mask R-CNN. Furthermore, we investigate to what extent the performance of the methods can be leveraged by certain pre- and post-processing methods for the data as well as by applying data augmentation techniques during the training of the neural network. In particular we study the effect of

- the conversion to different color spaces (RGB, HSV, CIEL*a*b*),
- contrast enhancement methods (Histogram Equalization, CLAHE),
- morphological operators (Closing, Opening) and
- data augmentation before and during the training of neural networks.

The paper is organized as follows: The theoretical background to the methods considered is briefly explained in Sect. 2. This is followed by Sect. 3, where the description of the data set designed, registered and annotated for this work is given. In Sect. 4 the metrics by which the methods are evaluated are introduced. The results to the experiments are presented and discussed in Sect. 5. In particular, the three methods are compared and the limitations of each method is discussed. Finally, we provide our conclusion and an outlook in Sect. 6.

2 A choice of methods for background-foreground segmentation

2.1 Gaussian mixture model (GMM)

The GMM, introduced for foreground-background segmentation in [22], is based on the principle of background subtraction. As described in the Sect. A, the values of a pixel are defined by a certain color space. For example, the pixel values of a gray scale image are given by single scalars, whereas the pixel values of a color image are given by a vector with the number of channels as dimension. In the GMM framework, these values of each pixel are modeled by a mixture of adaptive Gaussian distributions. The underlying data for the computation of those mixture models is given by a so called pixel process $\{X_1, \dots, X_t\}$ which is a time series of pixel values. Thus, at any time t , the history for each pixel at position (w_0, h_0) is known:

$$\{X_1, \dots, X_t\} = \{I((w_0, h_0), i) | 1 \leq i \leq t\} \quad (1)$$

with I being the image sequence. Now, the probability of observing a certain pixel value at time t is defined as

$$p(X_t) = \sum_{m=1}^K \hat{w}_{m,t} \mathcal{N}(X_t; \hat{\mu}_{m,t}, \hat{\Sigma}_{m,t}) \quad (2)$$

with

- \mathcal{N} : The multivariate Gaussian distribution.
- K : Number of Gaussian distributions in a mixture.
- $\hat{\mu}_{m,t}$: The estimate of the mean value of the m -th Gaussian at time t .
- $\hat{\Sigma}_{m,t}$: The estimate of the covariance matrix of the m -th Gaussian at time t defined as

$$\hat{\Sigma}_{m,t} = \hat{\sigma}_{m,t}^2 \mathbf{I}$$

with $\hat{\sigma}_{m,t}^2$ the estimate of the variance of the m -th Gaussian at time t and \mathbf{I} the identity matrix of appropriate dimension.

- $\hat{w}_{m,t}$: The estimated weight of the m -th Gaussian at time t . Moreover, the weights fulfill the properties of non-negativity and normalization, i.e., $\hat{w}_{m,t} \geq 0$ and $\sum_{m=1}^M \hat{w}_{m,t} = 1$.

If a new frame of the image sequence is considered at current time t , a new pixel value enters the pixel process. By the update of the pixel process, the estimates of the Gaussian distributions also have to be updated. For the estimation of the parameters, the Maximum Likelihood estimator for the currently observed data is computed. A well-known approach for this computation is the Expectation Maximization (EM) [23]. However, here the application of the exact EM algorithm would be costly, since the values of each pixel are modeled by a mixture of Gaussians and thus, the parameters are updated pixel-wise. Hence, the parameter update is realized by the implementation of an online K -means approximation [22]. For each new pixel value it is checked whether the pixel is represented by one of the already existing K Gaussians. This check is performed until a match is found. For example, a match is given if the new pixel value is within 2.5 standard deviations of a distribution.

Estimation of the model parameters Whether there is a match or not, the weight parameters are iteratively updated via

$$\hat{w}_{m,t} = (1 - \alpha) \hat{w}_{m,t-1} + \alpha \mathbb{M}_{m,t}. \quad (3)$$

Here, $\alpha \in [0, 1]$ is the learning rate which determines the influence of data from past points in time and the speed at which the model parameters are updated and

$$\mathbb{M}_{m,t} = \begin{cases} 1 & \text{in case of a match,} \\ 0 & \text{else.} \end{cases} \quad (4)$$

The distribution parameters $\hat{\mu}$ and $\hat{\sigma}^2$ are only updated for the distribution that matches the new pixel value X_t , otherwise no update is performed:

$$\hat{\mu}_{m,t} = \begin{cases} (1 - \rho)\hat{\mu}_{m,t-1} + \rho X_t & \mathbb{M}_{m,t} = 1, \\ \hat{\mu}_{m,t-1} & \text{else.} \end{cases} \quad (5)$$

$$\hat{\sigma}_{m,t}^2 = \begin{cases} (1 - \rho)\hat{\sigma}_{m,t-1}^2 + \rho\hat{\delta}_{m,t}^T\hat{\delta}_{m,t} & \mathbb{M}_{m,t} = 1, \\ \hat{\sigma}_{m,t-1}^2 & \text{else.} \end{cases} \quad (6)$$

with $\rho = \alpha \mathcal{N}(X_t | \hat{\mu}_{m,t}, \hat{\sigma}_{m,t}^2)$ and $\hat{\delta}_{m,t} = X_t - \hat{\mu}_{m,t}$. If no match is given at all, the distribution that assigns the lowest probability to the data is replaced by a new distribution with the initial parameters $\hat{w}_{\text{new}} = \alpha$, $\hat{\mu}_{\text{new}} = X_t$, $\hat{\sigma}_{\text{new}} = \sigma_0$ and σ_0 an appropriate initial variance [20, 22].

Estimation of the background model Now, it should be determined by which of the computed Gaussian distributions, the background can be modeled. In particular, the Gaussians with the highest weights and the lowest variances are of interest. Generally, it can be assumed that pixel values describing the background of a scenario are repeated and thus, also their distributions. Hence, if a new pixel value enters to a pixel process which describes the background, a high probability for a match is given. By the update rule (3), it can be observed that the weights are increasing in the case of a match. Moreover, the background consists mostly of static elements that produce less variance than dynamic ones. Therefore, to determine the distributions of the mixture model that describe the background the best, the Gaussians are sorted firstly in descending order by the value $\hat{w}/\hat{\sigma}$. Hence, the distributions which are most likely representing the background are at the top of the list. Then, the first B distributions are chosen to model the background

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{m=1}^b \hat{w}_m > \tau \right) \quad (7)$$

with τ the percentage of the pixel process that should affect the background model. The pixel values that cannot be assigned to a distribution which belongs to the background model are grouped by a two-pass connected components algorithm [24].

Number of mixtures In the introduced GMM framework of above, the number of Gaussian distributions in a mixture is given by a constant value that is determined by the available memory and computational power. In this work, a modified version of the original GMM framework is used, where the number of Gaussians is also adaptive. In [20] the update rule of \hat{w} is reformulated such that the weights may take negative values. This aims omitting weights for Gaussians which are not relevant for the background estimation. Hence, the distributions that do not describe the background with high certainty are directly excluded. We refer to [20] for a detailed derivation of the modified update rule.

2.2 Morphological snakes

Originally, the object detection method based on active contours (also called “snakes”) was presented in [25]. The idea behind this approach is to detect foreground instances of

an image I by evolving an initial curve C_0 towards the instances boundaries. In particular, the evolution of this curve is achieved by minimizing the energy functional

$$E(C) = \alpha \int_0^1 \|C'(q)\|_2^2 dq + \beta \int_0^1 \|C''(q)\|_2^2 dq - \lambda \int_0^1 \|\nabla I(C(q))\|_2 dq \quad (8)$$

with $C(q) : [0, 1] \rightarrow \mathbb{R}^2$ a parameterized planar curve, which represents the contour, $I : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$ the considered image, $a, b \in \mathbb{R}^+$ and $\alpha, \beta, \lambda \in \mathbb{R}^+$ constant parameters. By the design of the functional, the smoothness of the curve is controlled by the first two terms, while the third term attracts the curve towards the boundary of the object. Therein, the gradient of the image ∇I acts as an edge detector. Hence, the (local) minimum should be obtained at the objects boundary.

To handle topological changes, such as splitting and merging, automatically, the original energy functional is modified in different ways. The “Geodesic Active Contours” (GAC) [26] and the “Active Contours Without Edges” (ACWE) [27] are based on the level-set-method [28] which are successfully applied to conduct curve evolution [26]. By this, the curve $C : [0, 1] \times \mathbb{R}^+ \rightarrow \mathbb{R}^2$, $(q, t) \mapsto C(q, t)$ parameterized over time $t \in \mathbb{R}^+$ is included into a level-set of an arbitrary smooth embedding function $u : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$, such that it holds $C(q, t) = \{(x, y) | u((x, y); t) = 0\}$. Hence, the curve C is represented implicitly by u [29].

To receive the level-set formulation, the evolution of C is defined by a partial differential equation (PDE) C_t obtained by minimizing the respective energy functional $E(C)$ with the steepest descent method. Then, this curve evolution C_t can be reformulated into the level-set equation $u_t = \frac{\partial u}{\partial t}$ for $t > 0$ with the initial value $u_0 = u((x, y); 0)$, as shown in [26, 30]. For the GAC- and ACWE method, this approach results in the following level-set equations for $t > 0$.

Geodesic active contours (GAC) For this approach, the level-set equation is given by

$$u_t = g(I)\tilde{\kappa}\|\nabla u\|_2 + g(I)\nu\|\nabla u\|_2 + \nabla g(I)\nabla u \quad (9)$$

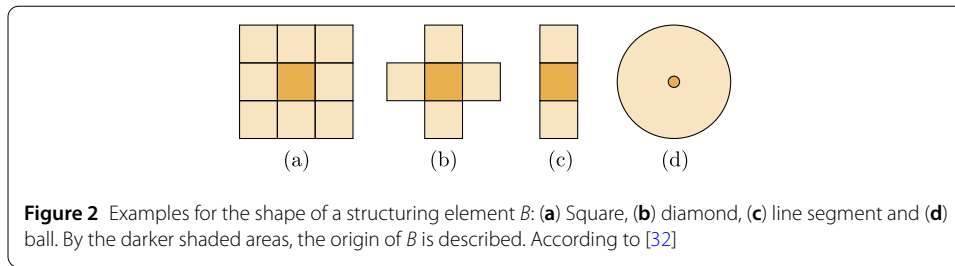
with

- $I : [0, a] \times [0, b] \rightarrow \mathbb{R}^+$, $a, b \in \mathbb{R}^+$ an image.
- $g(I) : [0, \infty) \rightarrow \mathbb{R}^+$ a strictly decreasing function. By the values of g , the image regions of interest can be selected, e.g. the object boundaries in the case of image segmentation. In this work, g is defined as

$$g(I) = \frac{1}{\sqrt{1 + \alpha\|G_\sigma * I\|}} \quad (10)$$

with $G_\sigma * I$ a Gaussian filter ($*$ being the convolution operator), σ the standard deviation and $\alpha > 0$ a non-linear scaling parameter. On object boundaries, $g(I)$ takes smaller values than on homogeneous image areas.

- $\tilde{\kappa} := \operatorname{div}(\frac{\nabla u}{\|\nabla u\|})$ the euclidean curvature of the embedding function u , proven in [30].
- $\nu \in \mathbb{R}$ the balloon force parameter.



Active contours without edges (ACWE) Herein, the level-set equation is given by

$$u_t = \|\nabla u\|_2 (\mu \tilde{\kappa} - \nu - \lambda_1 (I - c_1)^2 + \lambda_2 (I - c_2)^2) \quad (11)$$

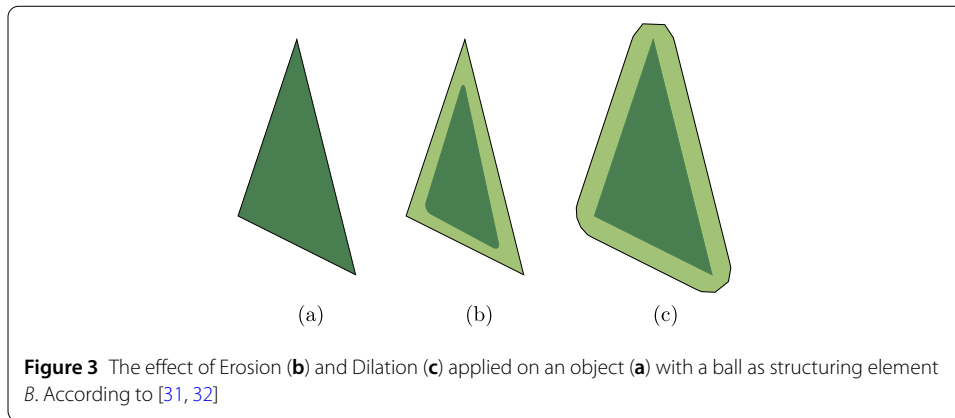
with

- I , $\tilde{\kappa}$ and ν analogous to above.
- c_1, c_2 constants that depend on the curve $C : [0, 1] \rightarrow \mathbb{R}^2$. In particular, c_1 represents the average of the pixel values of I inside C and c_2 the average of $I(x, y)$ outside C .
- $\lambda_1, \lambda_2 > 0$ and $\mu \geq 0$ fixed weight parameters.

Both level-set equations are composed of a smoothing term, a balloon force term and an attraction force or image attachment term. In particular, the parts of the curve with a high curvature will be smoothed by the smoothing term. The balloon force term should help to accelerate the curve evolution especially in areas, where the attraction force is too weak due to small values of the gradients in less informative areas. So, the evolution of C is inflated ($\nu > 0$) or deflated ($\nu < 0$) by determining the velocity $\nu \in \mathbb{R}$. If $\nu = 0$, the balloon force is switched off.

Now, the solutions of the level-set equations (9) and (11) are obtained by solving the time-dependent PDEs iteratively. However, the numerical methods for the computation of PDEs are costly, challenging in the implementation and suffer from stability constraints. In [29] it is shown, that it is possible to overcome these difficulties by approximating the PDEs of which (9) and (11) are composed of, by binary morphological operators. These operators are formulated as sup-inf operators as given in (18). The implementation of such a sup-inf operator is much easier and the computation is more stable and faster compared to the one of PDEs. Hence, the PDEs (9) and (11) are approximated by the composition of mathematical morphological operators, whereby the implicit representation of C is maintained.

Generally, a morphological operator T satisfies the properties of standard monotony, translation- and contrast invariance [31]. Furthermore, T is defined uniquely by a structuring element B , which is a set of arbitrary but small size and shape with a predefined origin as described in Fig. 2. Usually, B is significantly smaller than the considered image I . Mathematically, B is a matrix of dimension $c \times d$, $c, d \geq 1$ consisting of zeros and ones. The role of B is to probe a given image pixel-wise whereby the positioning of B at a pixel is given by its defined origin. According to the rule which is specified by a morphological operator, every pixel is evaluated by the comparison with the origin of B and its corresponding neighborhood which is represented by the values equal to one in the matrix [32]. Hence, the structuring element can be interpreted as a kernel in context of machine learning [31].



Morphological operators of interest for this work are the dilation, the erosion (see Fig. 3) and the curvature morphological operators, due to their properties regarding the infinitesimal behaviour.

Remark 2.1 (Notation) Let $\mathcal{F} \subseteq C_b^k(\mathbb{R}^n)$ be a set of bounded continuous differentiable functions up to order k over \mathbb{R}^n . The function operator is denoted by $T : \mathcal{F} \rightarrow \mathcal{F}$ which is assumed to be well-defined on $C_b^k(\mathbb{R}^n)$ [33].

Definition 2.2 (Dilation and Erosion [34]) Let $u \in \mathcal{F}$, B the structuring element and $h \geq 1$, $h \in \mathbb{R}$ a scaling parameter. The Dilation of u by hB , written as $D_h = D_{hB}$, is defined by

$$D_h u(\mathbf{x}) = \sup_{\mathbf{y} \in hB} u(\mathbf{x} - \mathbf{y}). \quad (12)$$

Whereas, the Erosion of u by hB , written as $E_h = E_{hB}$, is given by

$$E_h u(\mathbf{x}) = \inf_{\mathbf{y} \in -hB} u(\mathbf{x} - \mathbf{y}). \quad (13)$$

Infinitesimal behaviour of dilation and erosion Let be B convex bounded and the B -norm on \mathbb{R}^n is defined by $\|\mathbf{x}\|_B = \sup_{\mathbf{y} \in B} (\mathbf{x} \cdot \mathbf{y})$ with \cdot the Euclidean scalar product. Furthermore, the initial value of u at time $t = 0$ is given by $u_0(\mathbf{x}) = u(\mathbf{x}; 0)$ with $\mathbf{x} \in \mathbb{R}^n$. By defining $u : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}$ as the dilation of the initial value u_0 by tB , such that $u(\mathbf{x}; t) = D_t u_0(\mathbf{x})$, it holds (see [33, 34]):

$$\frac{\partial u}{\partial t} = \|\nabla u\|_B. \quad (14)$$

Analogously it holds for $u(\mathbf{x}; t) = E_t u_0(\mathbf{x})$ (see [33, 34]):

$$\frac{\partial u}{\partial t} = -\|\nabla u\|_B. \quad (15)$$

Here, it holds in particular that

$$\|\nabla u\|_B = \|\nabla u\|_2 \quad (16)$$

if the structuring element is defined as the unit ball

$$B_1(0) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 < 1\} \quad (17)$$

on \mathbb{R}^n [34, 35]. Hence, under certain conditions, the infinitesimal behaviour of the Dilation and the Erosion is equivalent to the PDE $\frac{\partial u}{\partial t} = \pm \|\nabla u\|_2$, which is a component of the level-set equations (9) and (11).

The sup-inf representation of morphological operators The authors of [34] show that every morphological operator has a sup-inf representation and that also the dual inf-sup form exists.

Let \mathcal{B} be a set of structuring elements and $T : \mathcal{F} \rightarrow \mathcal{F}$ an arbitrary morphological operator. Then T can be represented by the sup-inf operator

$$SI_h := T_h u(\mathbf{x}) = \sup_{B \in \mathcal{B}} \inf_{\mathbf{y} \in \mathbf{x} + hB} u(\mathbf{y}). \quad (18)$$

The dual operator of T is defined as $\tilde{T}(u) = -T(-u)$, which is also a morphological operator. Thus, the inf-sup representation of T is given by

$$IS_h := T_h u(\mathbf{x}) = \inf_{B \in \tilde{\mathcal{B}}} \sup_{\mathbf{y} \in \mathbf{x} + hB} u(\mathbf{y}) \quad (19)$$

with $\tilde{\mathcal{B}}$ the set of structuring elements of \tilde{T} .

Definition 2.3 (Curvature morphological operator [29]) Given the morphological operators SI_h and IS_h with the set of structuring elements $\mathcal{B} = \{[-1, 1]_\theta \subset \mathbb{R}^2; \theta \in [0, \pi)\}$ and h sufficiently small. Then the composition

$$SI_{\sqrt{h}} \circ IS_{\sqrt{h}} \quad (20)$$

is defined as the curvature morphological operator.

Infinitesimal behaviour of $SI_{\sqrt{h}} \circ IS_{\sqrt{h}}$ It is shown in [34], that the mean operator

$$F_h u(\mathbf{x}) = \frac{SI_{2h}(\mathbf{x}) + IS_{2h}(\mathbf{x})}{2} \quad (21)$$

has an infinitesimal behaviour, which is equivalent to the mean curvature motion $\tilde{\kappa} \|\nabla u\|_2$. However, the problem about F_h or rather $F_{\sqrt{h}}$ is, that it is not a morphological operator since the property of contrast invariance is not satisfied [36]. Due to this reason, the curvature morphological operator is introduced by [29], which approximates the mean operator. Thus, (20) has the same infinitesimal behaviour as (21), namely $\tilde{\kappa} \|\nabla u\|_2$, which is also a component of the level-set equations (9) and (11).

Morphological GAC (MGAC) and ACWE (MACWE) Summarizingly, the introduced morphological operators approximate PDEs, which are defined by the level-set equations u_t of the GAC and ACWE method. By this knowledge, it is possible to derive the morphological versions of the both methods [29].

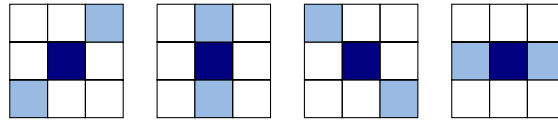


Figure 4 A discrete set of structuring elements \mathcal{B}_d with the origin at the center. Adapted from: [29]

To this end, the embedding function u needs to be redefined. Firstly, u should be discrete in practice and secondly, u has to be binary, since the morphological operators are also binary. Hence, $u : \mathbb{Z}^2 \rightarrow \{0, 1\}$ is defined as a binary piece-wise constant function with

$$u(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is inside the curve boundaries,} \\ 0 & \text{if } \mathbf{x} \text{ is outside the curve boundaries.} \end{cases} \quad (22)$$

Due to the discretization of u , the (sets of) structuring elements also have to be discretized. This realizes the discretization of the morphological operators. In Fig. 4, a possible discrete version \mathcal{B}^d of \mathcal{B} in definition 2.3 is described. Here, \mathcal{B}^d consists of four discrete line segments with a length of three pixels and the origin at the pixel coordinate $(0, 0)$. Analogously, the structuring element of the Dilation and the Erosion given by the unit ball $B_1(0)$ can be discretized.

Intuitively, the balloon force operator acts in a similar way as the Dilation or the Erosion by inflating or deflating a contour, respectively. So the PDE of the balloon force term can be approximated by the Dilation, if $\nu > 0$ and vice versa by the Erosion. The smoothing term represents the mean curvature motion, such that the PDE of this component is approximated by the curvature morphological operator. Finally, the remaining attraction force $\nabla g(I) \nabla u$ and image attachment term $\|\nabla u\|_2 (\lambda_2 (I - c_2)^2 - \lambda_1 (I - c_1)^2)$ can be discretized directly, as the remaining factors $g(I)$ of the balloon and the smoothing term. In [29] it is described how this discretization is realized.

In conclusion, the level-set equations (9) and (11) are solved by the successive computation of the composition of three discrete and morphological operators in the mGAC or the mACWE approach. The algorithms of both approaches can be found in [29].

2.3 Mask R-CNN

The Mask R-CNN [37] is a Region-based Convolutional Neural Network for instance segmentation. Hence, the goal is to detect and classify each object of an image, whereby it should be also distinguished between every individual instance within a class. By a Mask R-CNN, the detection and classification task and the generation of a mask for each instance are managed simultaneously.

In particular, the Mask R-CNN is an extension of the Faster R-CNN [38]. This framework for object detection consists of two components: A Region Proposal Network (RPN) and a region-based object detection network, here given by the Fast R-CNN [39] (the predecessor of Faster R-CNN). The RPN generates candidate object locations, called “proposals”, which the Fast R-CNN uses to determine the exact locations of the detected images. The innovation of the Faster R-CNN is to unify those both networks into one framework by developing training algorithms in which both networks share some of their layers. Now, the extension of the Faster R-CNN is realized by adding a mask branch. As described in

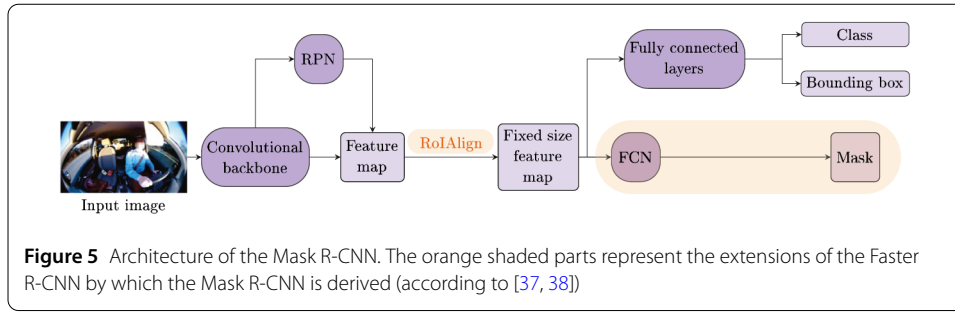


Fig. 5, the instance masks are generated by a Fully Convolutional Network (FCN) [40] which aims at classifying for each pixel, whether it belongs to a certain class or not.

For the achievement of the prediction of high quality masks, the authors of [37] show, that the introduction of the following two novelties play a key role. Firstly, the “RoIPooling”-layer (RoI: Region of Interest) of the Faster R-CNN is substituted by the “RoIAlign”-layer. Actually, the Faster R-CNN is not designed for a pixel-to-pixel relation between the in- and output. By RoIAlign the features which are extracted by the convolutional backbone, can be properly aligned according to the input image (see Fig. 5). Thus, the generation of pixel-accurate instance segmentation masks is possible. Secondly, the classification task and the prediction of the mask for each instance is decoupled. The loss function is defined in such a manner, that binary masks are predicted for all of the K classes independently, such that no competition exists among these classes during inference. Hence, the prediction of the class is not based on a predicted mask, but solely on the classification branch. Since all desired outputs are computed in parallel, the multi-task loss

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_B + \mathcal{L}_M \quad (23)$$

is defined on each RoI which is composed of the classification loss \mathcal{L}_C [39], the bounding box regression loss \mathcal{L}_B [39] and the loss of the mask branch \mathcal{L}_M [37, 41]. These loss functions are defined as described below.

Remark 2.4 (Notation) The set of ground truth labels is given by \mathcal{Y} , whereby $\#\mathcal{Y} = K + 1$, and consists of K predefined object classes and an additional background class. In particular, the background class is denoted by $y = 0$. Moreover, the ground truth class y of each instance is assigned to each RoI.

1. Classification loss \mathcal{L}_C The output of the classification branch is given by a discrete probability distribution $p = (p_0, p_1, \dots, p_y, \dots, p_K)$ over all $K + 1$ classes whereby

$$p_k = \frac{e^{z_k}}{\sum_{i=0}^K e^{z_i}} \quad \forall k = 0, \dots, K \quad (24)$$

with $z \in \mathbb{R}^{K+1}$ as the output of the last fully connected layer. Then the classification loss is defined as the log loss of the true class y :

$$\mathcal{L}_C = -\log(p_y) \quad (25)$$

2. Bounding box regression loss \mathcal{L}_B The output of the bounding box regression branch is given by a four tuple of pixel values $\hat{b}^k = (\hat{b}_c^k, \hat{b}_d^k, \hat{b}_w^k, \hat{b}_h^k) \forall k = 1, \dots, K$. Here, (\cdot_c, \cdot_d) describe the pixel coordinates of the center of the bounding box, while the width and height of the bounding box are given by the pixel values with the indices w and h . For detailed information on the derivation of the certain pixel values we refer to [39]. With the ground truth bounding box $b^y = (b_c^y, b_d^y, b_w^y, b_h^y)$, assigned to each RoI if $y \neq 0$, the loss function is defined by

$$\mathcal{L}_B = \mathbb{1}_{\{y>0\}} \sum_{j \in \{c,d,w,h\}} \mathcal{H}(\hat{b}_j^y - b_j^y) \quad (26)$$

with $\mathcal{H}(\phi)$ the Huber loss function [42]

$$\mathcal{H}(\phi) = \begin{cases} 0.5\phi^2 & \text{if } |\phi| < 1, \\ |\phi| - 0.5 & \text{else} \end{cases} \quad (27)$$

and the indicator function

$$\mathbb{1}_{\{y>0\}} = \begin{cases} 1 & \text{if } y > 0, \\ 0 & \text{if } y = 0. \end{cases} \quad (28)$$

3. Loss of the mask branch \mathcal{L}_M The output of the mask branch has the dimension Km^2 since it encodes a binary mask of a spatial dimension of $m \times m$ for all K object classes except the background class. The values of each pixel of the predicted mask $\hat{\tau}_{rs}^k, k = 1, \dots, K$ are derived by applying a sigmoid activation function to the outputs of the last feature map. With the pixel values τ_{rs}^y of the ground truth mask, the loss function is defined as the average binary cross-entropy:

$$\mathcal{L}_M = \mathbb{1}_{\{y>0\}} \frac{1}{m^2} \sum_{r=1}^m \sum_{s=1}^m \tau_{rs}^y \log(\hat{\tau}_{rs}^y) + (1 - \tau_{rs}^y) \log(1 - \hat{\tau}_{rs}^y) \quad (29)$$

with $\mathbb{1}_{\{y>0\}}$ defined as in (28).

Those three tasks of classification, bounding box regression and mask generation are solved in the head architecture of the Mask R-CNN which operates on each RoI, whereas the important features are extracted in the convolutional backbone architecture. In this work, the backbone architecture is given by a combination of the ResNet with 101 layers [43] and the Feature Pyramid Network (FPN) [44]. The head architecture is given only by the FPN.

Summarizingly, the Mask R-CNN creates instance segmentation masks that separate and individually distinguish the relevant instances from the background.

3 Datasets for interior sensing

Scenarios of interior sensing are highly complex. The instances separated from the background belong to the classes “Person”, “Child seat” and “Object” in this work. Thus, the foreground instances can be both, dynamic and static. Furthermore, only those detected instances which are positioned on the front and back seats of the car are of interest. Also

modelling the background is non-trivial since it contains dynamic elements due to the motion of objects visible through the car windows. Moreover, the camera position changes slightly from car to car in this work. Additionally, the impact of environmental effects has to be taken into account, like different weather conditions, shadows, traffic lights and vibrations.

To solve the background-foreground segmentation task in this high complex setting adequately, it is of importance, especially for the training of the Mask R-CNN, that an appropriate training set is available by which a wide range of the challenges is covered. To this end, the ISSO dataset has been created by APTIV and the authors of this work. It consists of 1300 real-world images extracted from videos of different interiors of stationary or driving cars. The images contain a high variety regarding the foreground instances, the background and the environmental conditions. Further details are provided in the upcoming section.

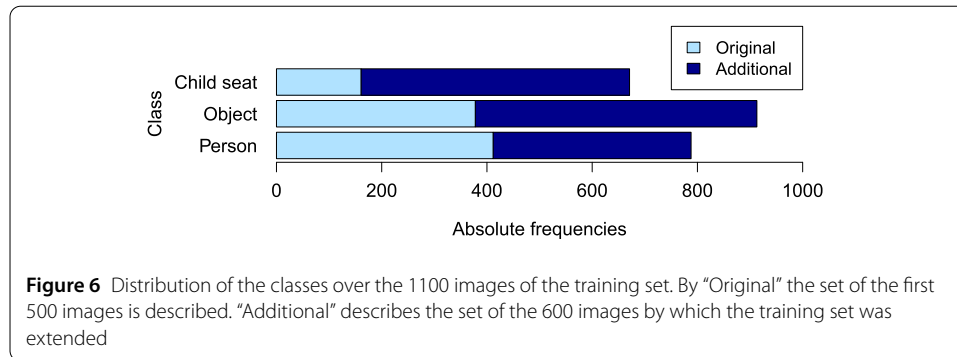
Since the annotation of images is time-consuming and costly, only a small amount of 1100 real-world images is available for training. To overcome this problem, we apply transfer learning. Here, the training of the Mask R-CNN is initialized by a model pretrained on the COCO dataset. This pretrained model is able to detect persons and certain everyday objects outside the scope of car interiors. Therefore, we also consider the impact of the COCO dataset maintained during the training. Moreover, images of the synthetic dataset SVIRO are used for the training of the Mask R-CNN to overcome the problem of the small amount of real-world annotated data. The SVIRO dataset consists of rendered images describing scenarios in the passenger compartment of different cars.

Hence, in total, we consider images and videos of three different datasets—the COCO dataset, the SVIRO dataset and our ISSO dataset. In the next section, we describe these datasets in more detail.

3.1 The ISSO dataset

The Interior Sensing and Seat Occupancy (ISSO) dataset has been created by APTIV and the authors of this work. It consists of images which are extracted from videos that have been recorded in driving or stationary cars by APTIV in Wuppertal, Germany. The purpose of this dataset is to enable the feasibility study provided by the present article and it is not meant to be representative for a local or global population. While selecting the images for the dataset, it was taken into account, that a high variation within the instances, the backgrounds and the light conditions is given over all images. The camera is mounted on the upper or lower area of the windshield in each car. Hence, the position changes slightly per car. Since the camera is not integrated, its position might even change slightly in one and the same car. In total, 1300 images were labeled from which we define test, training and validation sets.

1. Training set The training set is used to train the Mask R-CNN. In total, it consists of 1100 labeled images recorded in five different cars. 500 images were selected and labeled at the beginning of this work. Since the class “Child seat” suffered from a lack of variation, it was paid attention to recording as many different child seats as possible in later recording session. Of these new videos, 600 additional images were chosen and labeled, such that the number of instances for the class “Child seat” increased in particular. Nevertheless, as one can see from Fig. 6, the instances of this class are least represented in the training set. For



the class “Child seat” it is to remark that instances are not clearly visible in two situations. Here, a situation refers to a specific camera position in a specific car. Firstly, due to the camera position, only a small part of a child seat is visible if it is mounted on the back seat of the car interior. Second, a child seat is hardly visible if it is occupied by a child. Hence, especially for the training it is of interest in how many cases the child seats are mounted on the front passenger seat and thus clearly visible. In particular, the child seats are mounted on the passenger front seat of the car in about 60% of the images that contain a child seat. Of these front mounted child seats over 80% are not occupied. Detailed statistics for the training set are given in the Sect. B.

2. Validation set The validation set consists of 100 images distributed over three different cars. It contains five persons, among them one child, one female and four male. The class “Object” is represented by four instances of five main categories (laptop, PC-keyboard, backpack and beverage crate). Moreover, two child seats are available in the validation dataset. In 29 images, a child seat is contained whereby only seven of these images show a child seat mounted on the passenger front seat. None of the front mounted child seats is occupied by a child. All cars, child seats, objects and persons are different from those shown by the training set.

3. Test set The test set is used to evaluate and to compare the performance of the implemented foreground-background detection methods. It consists of 100 labeled images extracted from 70 videos that are recorded in five different cars. 50 images are extracted from videos inside a driving car and the other 50 images are extracted from videos inside stationary cars.

The test set contains 13 persons, among them three children and one baby, three female and ten male. In the class “Object”, everyday items are collected, like a bag pack or a wallet. As described in Table 14, 42 instances of 16 main categories are included in the test set. Furthermore, four different child seats are available in the test set. The child seat is mounted on the passenger front seat of the car in about 30% of the images that contain a child seat. Additionally, about 53% of these front mounted child seats are occupied. All cars, child seats, objects and persons are different from those shown by the training and validation sets.

Creation of the ground truth The annotations of the images are created by the tool “Labelme” from MIT [45] extended by the function of an eraser. With “Labelme”, it is possible



Figure 7 An image of the training set (left) and its corresponding gray scale ground truth instance segmentation mask (right)



Figure 8 An image of the test set (left) and its corresponding binary ground truth segmentation mask (right)

to annotate the instances of an image by closed polygon courses. By this information, the ground truth segmentation masks with different gray scale values for each instance can be created (see Fig. 7).

In this feasibility study we only focus on foreground-background segmentation. Hence, for the images of the test set, we generated binary ground truth segmentation masks. As described by Fig. 8, the foreground instances are represented by white pixels, while the background is given by black pixels.

3.2 The SVIRO dataset

As the name suggests, the Synthetic Vehicle Interior Rear Seat Occupancy (SVIRO) dataset [46] consists of images produced artificially by a rendering software which is Blender, version 2.79. These images depict randomly generated scenarios in the passenger compartment of ten different vehicles. For each of these cars 2500 images were generated and split into a training and a test set. Here, each training set contains 2000 labeled images and each test set 500 images. The labeled instances of the classes “Person”, “Child seat” and “Everyday object” differ between the training and the test set.

Herein, the training datasets of the five car models Ford Escape, the Lexus GSE, the Tesla Model 3, the VW Tiguan and the RenaultZoe are used. For the training set of each car, the following statistics apply: Each training set contains 23 persons, whereby six persons are children and three persons are babies. Moreover, three different child seats and four different everyday objects are used in one training dataset. Furthermore, different light conditions are taken into account. For the task of instance segmentation, the ground truth corresponding to the RGB image is given by an instance segmentation mask as described in Fig. 9.



Figure 9 An RGB image (left) and the corresponding ground truth instance segmentation mask (right) of the SVIRO dataset. Source: [46]

3.3 The COCO dataset

The “Common Objects in Context” (COCO, [47]) dataset consists of images that depict everyday objects in typical environments. Mainly, the images are non-iconic. This means for example, that an object is not shown in front of a calm background but in a complex scene. For this work the training dataset of the year 2017 is used. This dataset consists of over 118,000 labeled images covering over 60 object categories of 10 main categories including the category “background” as described in Table 15. The annotations are created by labeling each instance in an image by a closed polygon course and their corresponding bounding box.

4 Evaluation metrics

The foreground-background masks generated by the implemented frameworks are evaluated pixel-wise. The metrics by which this evaluation is realized are composed of the following four terms:

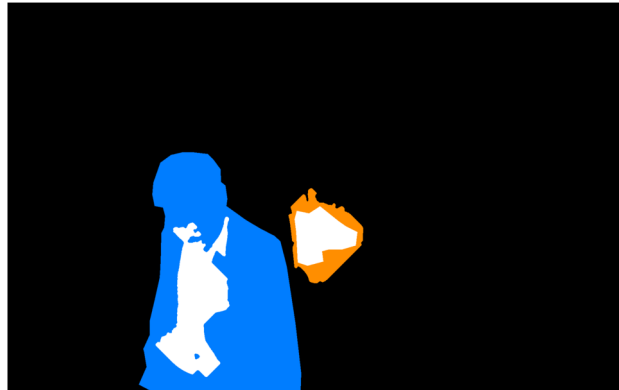
- True positives (TP): Pixels that belong to the foreground and which are correctly classified.
- False positives (FP): Pixels that belong to the background, but which are misclassified.
- True negatives (TN): Pixels that belong to the background and which are correctly classified.
- False negatives (FN): Pixels that belong to the foreground, but which are misclassified.

Thereof, commonly used metrics for the evaluation of background-foreground segmentation tasks [48] can be defined as given in Table 1. While the precision describes the amount correctly predicted foreground pixels relative to the total number of predicted foreground pixels, recall considers the predicted foreground pixels relative to the total number of actual (true) foreground pixels, corresponding to the ground truth. The specificity describes the proportion of true background pixels that are correctly classified. The accuracy provides the proportion of correct classifications overall. The similarity is also known as the Jaccard index or the Intersection over Union (IoU) [48–50]. This value measures to what extent the ground truth mask and the predicted mask resemble one another. Finally, the F_1 -score describes the harmonic mean of precision and recall [51].

Remark 4.1 (Evaluation on images of test set) The performance of all methods is evaluated on the test set presented in Sect. 3.1. In section Sect. 5, the averaged values over the 100 images are reported for all evaluation metrics. Furthermore, in our comparisons we mainly rely on the metric “similarity” which we compute for each test image separately and

Table 1 Definition of the evaluation metrics

Name	Formula
Precision	$Pr = \frac{TP}{TP+FP}$
Recall	$Re = \frac{TP}{TP+FN}$
Specificity	$Sp = \frac{TN}{TN+FP}$
Accuracy	$Acc = \frac{TP+TN}{TP+TN+FP+FN}$
Similarity	$Sim = \frac{TP}{TP+FP+FN}$
F_1 -score	$F_1 = \frac{Pr \cdot Re}{Pr + Re} = \frac{2TP}{2TP+FP+FN}$

**Figure 10** A predicted foreground-background mask. Here, the colored pixels are coded as follows: TP = white pixels, FP = orange pixels, TN = black pixels, FN = blue pixels

then compute the average over the whole test set. The similarity is the most sensitive and intuitive metric to measure the differences between the ground truth and the predicted mask. As can be seen in Fig. 10, the background of the given scenarios is very dominant in relation to the foreground instances. In this case, the accuracy would be still on a high level, if the foreground instances would not be segmented properly or detected at all. Due to this reason, the accuracy can lead to misinterpretations of the results. Contrary, the similarity is not affected by the distribution of background and foreground pixels over a given image. So, this metric produces more robust results. For example, for the predicted mask shown in Fig. 10 the similarity is only 26% but the accuracy is about 90%. In section Sect. 5 the similarity of the prediction and ground truth masks is denoted by the term “similarity score”.

5 Results of experiments

For each of the methods introduced in Sect. 2, implementation details and the results of our experiments are summarized in the following. All codes have been written in Python.

5.1 Gaussian mixture model (GMM)

Implementation details For the implementation of the GMM, the OpenCV-function BackgroundSubtractorMOG2 [52, 53] implemented in Python was used. By this function, a background model can be created. Contrary to the other two considered methods, the input of the GMM has to be a video since it is a motion based approach. Thus, the GMM was applied on the entire videos, from which the predicted masks of the test images were extracted. In particular, the video sequences are considered in the RGB, HSV

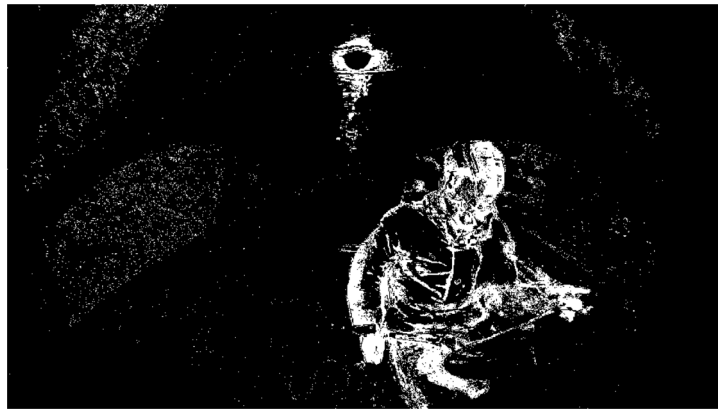


Figure 11 A foreground-background mask predicted by the GMM

and Lab color spaces introduced in Sect. A. Hence, the individual pixels are modeled by three-dimensional Gaussian distributions in the experiments performed here.

To initialize the background model, mainly one parameter has to be set, namely $T = \tau N$ (recall Sect. 2.1, N denotes the number of previous frames to be considered). This parameter indicates, the number of the last frames which are affecting the background model. After parameter tuning, we fixed the value $T = 250$ for all presented experiments below. Furthermore, the algorithm choses a learning rate $lr \in (0, 1)$ automatically where $lr = 0$ would mean that the background model is never updated and $lr = 1$ would mean that the background model is newly initialized for every frame. For further details, see [54].

Results The masks predicted by the GMM achieve similarity of 15.5% (averaged over the whole test set). We investigated if the quality of the masks can be enhanced by the application of a post-processing and pre-processing method. In particular, one can observe, that the predicted foreground-background masks contain a lot of noise as described by Fig. 11. Furthermore, the boundaries of the detected instances are clearly visible, however big parts of the areas inside the instance boundaries are not predicted as foreground. Therefore, we investigated the effect of the morphological operators “Opening” and “Closing”. Both operators are defined by a composition of the morphological operators Dilation and Erosion presented in Sect. 2.2. While noise can be removed from binary segmentation masks by the operator Opening, the areas of the instance boundaries can be filled by white pixel values with the help of the operator Closing.

Remark 5.1 (Notation) In the subsequent tables, the columns “MO”, “C” and “CE” contain the following information:

- MO describes which morphological operator was applied: C = Closing and O = Opening.
- C shows if the color channel V or L was considered.
- CE represents the applied contrast enhancement method: HE (Histogram equalization) or CHE (CLAHE) (see Sect. A).
- The baseline is given by the model with no application of a pre- or post-processing marked by the entry “-” of the columns MO, C and CE.

Table 2 Evaluation results for the predicted masks of the GMM after applying post-processing methods

MO	Pr	Re	Sp	Acc	Sim	F_1
–	0.333	0.325	0.811	0.744	0.155	0.256
C	0.337	0.368	0.794	0.736	0.172	0.279
O	0.357	0.269	0.837	0.756	0.131	0.220

Table 3 Evaluation results for the predicted masks of the GMM generated of preprocessed frames

C	CE	Pr	Re	Sp	Acc	Sim	F_1
–	–	0.333	0.325	0.811	0.744	0.155	0.256
V	HE	0.289	0.380	0.776	0.720	0.163	0.268
V	CHE	0.309	0.434	0.770	0.725	0.192	0.307
L	HE	0.298	0.351	0.808	0.744	0.161	0.267
L	CHE	0.315	0.418	0.785	0.736	0.192	0.308

Table 4 Evaluation results for the predicted masks of the GMM after applying post-processing methods on the masks. The input images are pre-processed by applying CLAHE on the respective color channel

MO	C	Pr	Re	Sp	Acc	Sim	F_1
–	–	0.333	0.325	0.811	0.744	0.155	0.256
C	V	0.311	0.503	0.740	0.710	0.212	0.332
O	V	0.345	0.358	0.815	0.750	0.175	0.282
C	L	0.319	0.492	0.754	0.722	0.214	0.336
O	L	0.350	0.335	0.831	0.762	0.171	0.278

As shown by Table 2, the application of Closing on the predicted masks leads to a slight improvement of about 2 percent points (pp.) for the similarity, whereas the quality of the predicted masks decreases when removing noise via Opening. Moreover, the performance of the GMM suffers from changes in the light conditions. To address this problem, a pre-processing step P_{CC} was introduced. Here, an image of the RGB color space is converted into the color spaces HSV or Lab. Then, the respective color channels V or L are extracted since it is possible to control the brightness of an image by them as described in Sect. A. Additionally, Histogram equalization or CLAHE can be applied to the extracted channels to approximate a uniform distribution regarding the brightness of an image and to enhance the contrast at the same time. Thus, each frame of a video sequence was pre-processed by P_{CC} before the GMM algorithm was applied on it. The corresponding results are given in Table 3. We observe that the similarity score increases slightly by about 4 pp. by applying CLAHE on the color channels V or L. If this pre-processing method was applied together with the morphological operator Closing as post-processing, the similarity even reached a value of about 21% as presented in Table 4.

Limitations The prediction of foreground-background masks by the GMM approach poses problems for the given scenario. The main reason is that the GMM is a motion based approach. The detection of static foreground instances that primarily belong to the classes “Object” or “Child seat” depends highly on external effects as the movement by a person or by the vibrations during the drive. But also foreground instances, which are dynamic in general, might not be detected if they do not move over a longer time period, such that they can be incorporated to the background. Additionally, a high number of false positives is generated when the car is driving. In our experiments, we observed that

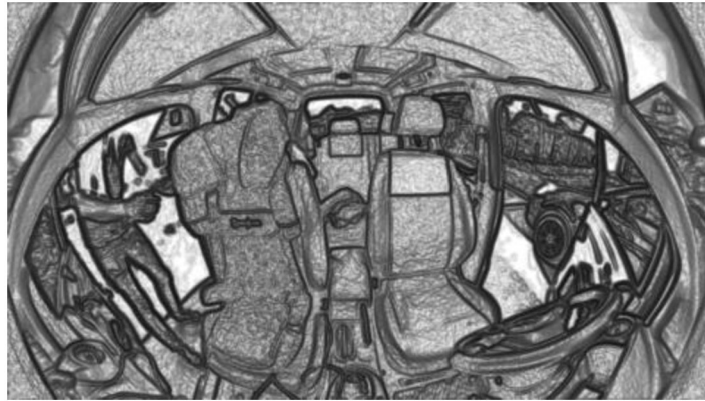


Figure 12 A gray scale real-world image after the application of the pre-processing P_{MGAC}

this can be attributed to motion visible in the car windows but also changing illumination. Moreover, the GMM takes a while to learn the background model. Hence, predicted masks for frames at the beginning are of bad quality.

5.2 Morphological snakes

Implementation details To implement the morphological snakes, the Python package *morphsnakes* [55] was used. The input of the algorithm is a gray scale image. For the MACWE as well as for the MGAC the initial contour is given by a circle for which the center (x, y) and the radius r have to be determined by the user. Here, a contour was initialized on each car seat which is occupied by at least one foreground instance. Furthermore, the number of iterations i for the curve evolution and the number of smoothing steps $s \in \{1, 2, 3, 4\}$ have to be defined for both methods, here $s = 2$. Besides that, the following parameters have been set for each method individually.

1. MACWE For the MACWE, the weight parameters for the region outside λ_1 and inside λ_2 the evolving curve have to be defined. In the case $\lambda_1 > \lambda_2$, it is assumed that the region outside the evolving curve contains more variation in their pixel values, compared to the region inside the curve and vice versa.

2. MGAC To perform the MGAC algorithm properly on an image, the contours of the foreground instances need to be clearly visible. Due to this reason, a pre-processing P_{MGAC} was performed on the images to highlight these contours. Here, P_{MGAC} is given by an “inverse gaussian gradient magnitude”-filter defined in (10). Applying this filter to an image leads to all pixel values being inside the interval $[0, 1]$. By this, the pixel values are close to zero, particularly in the areas which are close to the contours of the foreground instances as shown in Fig. 12. To conduct this pre-processing, two parameters have to be defined. The standard deviation σ of the Gaussian filter and the non-linear scaling parameter α acting as a steepness parameter. The larger α is, the steeper the transition between the areas of the instance contours and the flat regions inside and outside their contours is. Here, $\sigma = 3$ and $\alpha = 1000$.

Additionally, two parameters have to be set for the actual MGAC algorithm. The balloon force term $\nu \in \mathbb{R}$ determines if a Dilation ($\nu > 0$) or Erosion ($\nu < 0$) should be performed.

Table 5 Evaluation results for MACWE with $\lambda_1 = 1$ fixed

i	λ_2	Pr	Re	Sp	Acc	Sim	F_1
100	1	0.810	0.343	0.989	0.892	0.313	0.465
100	2	0.762	0.400	0.984	0.896	0.345	0.501
100	3	0.740	0.437	0.980	0.897	0.366	0.525
200	1	0.671	0.384	0.970	0.882	0.315	0.463
200	2	0.616	0.455	0.956	0.880	0.338	0.491
200	3	0.588	0.514	0.943	0.877	0.358	0.513
300	1	0.593	0.406	0.951	0.870	0.309	0.454
300	2	0.533	0.484	0.927	0.860	0.323	0.471
300	3	0.506	0.558	0.905	0.852	0.340	0.491

Table 6 Best evaluation results for each i for MGAC

i	τ	Pr	Re	Sp	Acc	Sim	F_1
100	0.3	0.788	0.572	0.981	0.914	0.469	0.633
200	0.4	0.778	0.565	0.975	0.911	0.466	0.627
300	0.4	0.712	0.603	0.958	0.902	0.456	0.616

By $\nu = 0$, no balloon force is applied. As a result of parameter tuning, we set the value to $\nu = 1.2$. Secondly, the stopping threshold τ has to be defined. Regions of the image with smaller values than τ are considered as the contours of the foreground instances. Thus, the evolution of the curve stops in those regions. In the experiments, we consider different values of τ .

Results

1. MACWE We study the influence of the number of iterations i and of the weights parameters λ_1, λ_2 . In particular, three different values are considered for i, λ_1 and λ_2 respectively, namely $i \in \{100, 200, 300\}$ and $\lambda_1, \lambda_2 \in \{1, 2, 3\}$. The baseline is represented by the assumption, that the pixel values of the regions inside and outside the evolving curve contain the same amount of variation, so $\lambda_1 = \lambda_2 = 1$.

As shown by Table 5, the similarity score increases with the increase of λ_2 for the case $\lambda_2 \geq \lambda_1 = 1$. During the experiments, we observed that the number of false positives increases by the increase of iterations i . Therefore, the values for the precision, accuracy, similarity and F_1 -score decrease with the increase of i . We made the same observation for the case $\lambda_1 \geq \lambda_2 = 1$. Here, the results of the baseline could not be exceeded for any i . Therefore, the hypothesis that the variation in the pixel values of the region outside the evolving the curve is higher compared to the pixel values inside the curve can be rejected for the given test set.

2. MGAC Analogously to the MACWE, the influence of the number of iterations and the stopping threshold on the performance of MGAC were investigated. In particular, the values $\tau \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ are considered for each number of iterations $i \in \{100, 200, 300\}$ in the experiments. Table 6 does not show a clear trend regarding the influence of the number of iterations for MGAC. For each $i \in \{100, 200, 300\}$ we observe that the values of Sim, Acc and F_1 increase with increasing values of τ until the respective best values are obtained. For $\tau > 0.4$ we observe in our experiments that the similarity, accuracy and F_1 -score decrease with increasing τ .

Limitations Although the similarity score of the MGAC is about 10 pp. higher than the score of MACWE, both approaches have the same limitations. The generated masks lose in quality if the pixel values of the foreground instances and the background have similar values in RGB color space. Furthermore, even slight changes in the light conditions and shadows lead to poorly generated foreground-background masks. One can observe that the curve evolution quickly gets stuck in areas with strong sunlight or at boundaries of shadows. The performance of the approaches depends highly on exterior factors since the essential parameters have to be determined by the user at the beginning. Hence, the configuration of the algorithms depends on the experience of the user.

Improvement of performance Analogously to the GMM, the performance of the morphological snakes suffers from changes in the light conditions. For this reason we also investigated if the performance could be positively affected by the application of the pre-processing P_{CC} . Thus, instead of a general gray scale image, the contrast enhanced image of the color channels V or L is presented to the respective morphological algorithm.

We repeated the experiments with the best results for both methods, MACWE and MGAC, now incorporating the P_{CC} pre-processing. The MACWE approach was repeated with the parameters $\lambda_1 = 1$ and $\lambda_2 = 3$ for all $i \in \{100, 200, 300\}$. The MGAC approach was repeated for the combination of the parameters, which are recorded in Table 6 whereby the pre-processing P_{CC} was performed before P_{MGAC} . Especially for the MACWE, the similarity of the generated and the ground truth masks increased by up to 6 pp. by applying HE to V. The similarity of the masks generated by the MGAC approach and the ground truth masks increased only slightly by 2.05 pp.

Overall, the highest accuracy (91.86%), F_1 -score (64.76%) and similarity (48.63%) was achieved by applying MGAC with the parameters $i = 100$ and $\tau = 3$ on the pre-processed image of the color channel V whose contrast was enhanced by using histogram equalization.

5.3 Mask R-CNN

Implementation details For the experiments with the Mask R-CNN, we used the implementation from [56]. Furthermore, we applied transfer learning, i.e., we trained the Mask R-CNN by using the weights of a pre-trained model as initial weights. This pre-trained model was trained on the entire COCO dataset presented in Sect. 3.3. The training of the Mask R-CNN was performed on one Titan XP GPU with 12 gigabytes working memory over 100 epochs with a batch size of 1. Per epoch, 1000 gradient descent steps were performed. Here, the momentum of the adam optimizer [57] was fixed to 0.9. During the inference, the detection of an instance was accepted if the predicted probability was ≥ 0.9 . The results of the experiments below are given for a confidence level of 90%. Within our experiments, we study the influence of three factors on the performance of the model:

1. *The influence of the learning rate lr and the weight decay λ* Here, the values $lr \in \{0.0005, 0.001, 0.002, 0.01\}$ and $\lambda \in \{0.0001, 0.001, 0.01\}$ were considered.

2. *The influence of data augmentation* The goal of data augmentation is to increase variability of a dataset. This of particular interest when the dataset is small. Since the ISSO

training dataset consists only of 1100 annotated real-world images, we consider data of-line augmentation (before training) and online augmentation (during training). We utilize both online and offline augmentation since different kinds of augmentation are readily available in python.

To this end, each image and its corresponding ground truth mask were horizontally flipped, randomly cropped and as a combination of both stored to the offline augmented data set. Additionally, we utilize further augmentation methods in an online augmentation pipeline. To this end, we use the Python package *albumentations* [58]. In detail, the pipeline performs four steps, in each step randomly choosing one of the following augmentation methods:

1. Gaussian blur, glass blur;
2. Gaussian noise, ISO noise;
3. Random brightness contrast, CLAHE (see Sect. A), random sunflair;
4. Grid distortion, elastic transformation, optical distortion.

In particular, the methods of (4) affect also the appearance of the ground truth masks.

3. The influence of the data which is used during training The Mask R-CNN was trained on the three different datasets presented in Sect. 3. Below, the datasets used in the training are denoted as follows:

- A^j : The ISSO training dataset which contains j real-world images, $j \in \{500, 1100\}$.
- A_{aug}^j : The ISSO training dataset which contains additionally the images of the offline augmentation. Thus, in total, this dataset consists of $4j$ images, $j \in \{500, 1100\}$.
- S^j : A subset of the SVIRO training set. This subset consists of j images of five different car interiors of the SVIRO dataset, $j \in \{2000, 4400\}$. The number of images of each car is uniformly distributed in the dataset S^j .
- C^j , $j \in \{4000, T\}$: A subset of the COCO dataset. The subset C^T consists of all images, which do not belong to the main category “vehicle”, “outdoor” or “animal” (see Table 15). That is, C^T contains images illustrating persons and everyday objects. The instances of all main categories are summarized to the class “Object”, except the instances of the main category “Person”. C^{4000} consists of 4000 randomly sampled images of C^T .

Lastly, the ISSO validation set was used for monitoring the training progress and tuning parameters.

Results Firstly, we discuss the results of experiments that were obtained from the Mask R-CNN trained on the datasets A_{aug}^{500} as well as S^{2000} (4000 images in total) and consider the effect of online augmentation. Offline augmentation is performed by default and consists of different augmentations than the online augmentation. The learning rate and weight decay were adopted from [37]. As documented by Table 8 the similarity is about 4 pp. higher if additional online augmentation is performed. Likewise, the precision also increases. Due to this reason, we applied online augmentation in all subsequent experiments.

Next, the influence of the learning rate and the weight decay are considered. To test the influence of the parameter lr the weight decay was set to $\lambda = 0.0001$. Table 9 suggest that $lr = 0.001$ is a decent choice. For a higher learning rate the final similarity score after training decreases. In all subsequent experiments, we fixed $lr = 0.001$. In order to test the influence of the weight decay we increased λ and considered $lr = 0.001$ and $lr = 0.01$. This

Table 7 Evaluation results for models trained on different datasets by applying online augmentation with the parameters $lr = 0.001$ and $\lambda = 0.0001$

Data	Pr	Re	Sp	Acc	Sim	F_1
$A_{aug}^{500} + S^{2000}$	0.904	0.790	0.989	0.965	0.735	0.840
A_{aug}^{500}	0.904	0.752	0.990	0.961	0.705	0.822
A_{aug}^{500}	0.939	0.736	0.991	0.960	0.700	0.805
S^{2000}	0.630	0.659	0.953	0.925	0.520	0.737
$A_{aug}^{500} + C^T + S^{2000}$	0.739	0.717	0.967	0.940	0.594	0.742
$A_{aug}^{500} + C^{4000} + S^{2000}$	0.863	0.782	0.985	0.962	0.708	0.813
$A_{aug}^{500} + C^{4000}$	0.837	0.778	0.979	0.958	0.680	0.788
A_{aug}^{1100}	0.919	0.797	0.991	0.969	0.755	0.866
A_{aug}^{1100}	0.888	0.759	0.988	0.962	0.705	0.813
$A_{aug}^{1100} + S^{4400}$	0.804	0.801	0.967	0.947	0.669	0.778

Table 8 Evaluation results for models trained on A_{aug}^{500} and S^{2000} with $lr = 0.002$ and $\lambda = 0.0001$

Online aug.	Pr	Re	Sp	Acc	Sim	F_1
No	0.830	0.762	0.982	0.960	0.677	0.801
Yes	0.920	0.767	0.989	0.961	0.716	0.817

Table 9 Evaluation results for models trained on A_{aug}^{500} and S^{2000} with $\lambda = 0.0001$

lr	Pr	Re	Sp	Acc	Sim	F_1
0.002	0.920	0.767	0.989	0.961	0.716	0.817
0.001	0.904	0.790	0.989	0.965	0.735	0.840
0.0005	0.894	0.752	0.987	0.961	0.694	0.800
0.01	0.000	0.000	1.000	0.855	0.000	NA

results in similarity scores of 69.61% and 70.58%, respectively, therefore neither showing a clear tendency nor a performance increase. Hence, we fix $\lambda = 0.0001$.

In addition to the experiments on data augmentation, we consider different compositions of training sets and study their influence on the similarity score after training. The results are summarized in Table 7. For this discussion, the previously best model trained on A_{aug}^{500} and S^{2000} with a similarity of 73.5% sets the baseline. As described in Table 7, the Mask R-CNN trained only on the real-world images of the ISSO dataset A^{500} or A_{aug}^{500} reaches a similarity of about 70%. A model solely trained on the synthetic data S^{2000} performs significantly worse, achieving a similarity score of 52.0%. This signals the presence of a strong domain shift when going from synthetic to real data which is typical for machine learning in computer vision [59]. Since the initial weights were pretrained on the whole COCO dataset, we study if the model's performance can be improved by adding the subset C^T or C^{4000} to the training. This approach aims at making the model memorize at least some of the features from the COCO dataset. Contrary to the ISSO and the SVIRO dataset, the images of the COCO dataset show the foreground instances not in the setting of car interiors but in an arbitrary environment. In Table 7 it can be observed that the performance of the model decreases significantly, compared to the baseline, if the number of images of the COCO dataset is much bigger than the number of images describing the foreground instances in the setting of car interiors during the training. Only the model trained on a balanced dataset with 4000 images of the COCO dataset (C^{4000}) and 4000 images of the ISSO and SVIRO dataset ($A_{aug}^{500} + S^{2000}$) reaches a similarity of about 71%. Nonetheless, the model does not outperform the baseline.

Table 10 Summary of the best evaluation result for each implemented method

Method	Pr	Re	Sp	Acc	Sim	F_1
Morph snakes	0.868	0.538	0.990	0.919	0.486	0.648
Mask R-CNN	0.919	0.797	0.991	0.969	0.755	0.866
GMM	0.342	0.465	0.777	0.737	0.216	0.337

When studying the predicted segmentations per image, the problem observed for the baseline model is that child seats are not detected reliably. This instability might occur due to the small variation of four child seats in the datasets A^{500} and A_{aug}^{500} . To test this hypothesis the Mask R-CNN was trained another time (after an extensions of the data collection process) by the extended datasets A^{1100} and A_{aug}^{1100} which contain 16 different child seats in total. To balance the ratio between synthetic and real-world data, the amount of images of the SVIRO dataset was also enlarged for the training. Table 7 shows that the similarity increases by training the Mask R-CNN just on the 1100 real-world images of the extended dataset. By looking through the background-foreground masks predicted by this model we observed that all unoccupied child seats are clearly segmented. However, occupied ones still remain challenging.

In conclusion, the best model with a similarity score of 75.5% was obtained by training the model on the dataset A^{1100} with the use of online augmentation and the determination of the parameters $lr = 0.001$ and $\lambda = 0.0001$.

Limitations The detection of occupied child seats and instances on the back seats of the car represent still difficult cases. As described in Sect. 3.1, only a small number of occupied child seats is given in the ISSO training dataset. Hence, this problem might be solved analogously to the problem with the unoccupied child seats by extending the dataset by more images of occupied child seats. The problem of the detection on the back seats could be solved by additional cameras, such that the instances on the back seats become sufficiently visible for the detection task.

5.4 Comparison

In summary, the Mask R-CNN clearly outperforms the two classical methods GMM and morphological snakes. As described in Table 10 the best model of the Mask R-CNN achieves a similarity score of 75.5% which significantly surpasses the best results of the morphological snakes by 26.9 pp. and of the GMM even by 53.9 pp. Moreover, we can observe by Fig. 13 that the Mask R-CNN is the only method which provides a clear segmentation of the foreground instances. By our experiments we also found out that the problems of the classical methods are addressed by the Mask R-CNN. Contrary to the morphological snakes and the GMM, the Mask R-CNN is much more robust against changes in the light conditions, shadows and other exterior factors, like traffic lights.

6 Conclusion and outlook

In this work we have introduced a benchmark for the task of foreground-background segmentation in interior sensing. We compared the segmentation performance of different variants of classical methods, i.e., Gaussian Mixture Models and Morphological Snakes, with the segmentation performance of a recent deep learning model, the Mask R-CNN. Similarly to other real-world computer vision applications, we observe that the

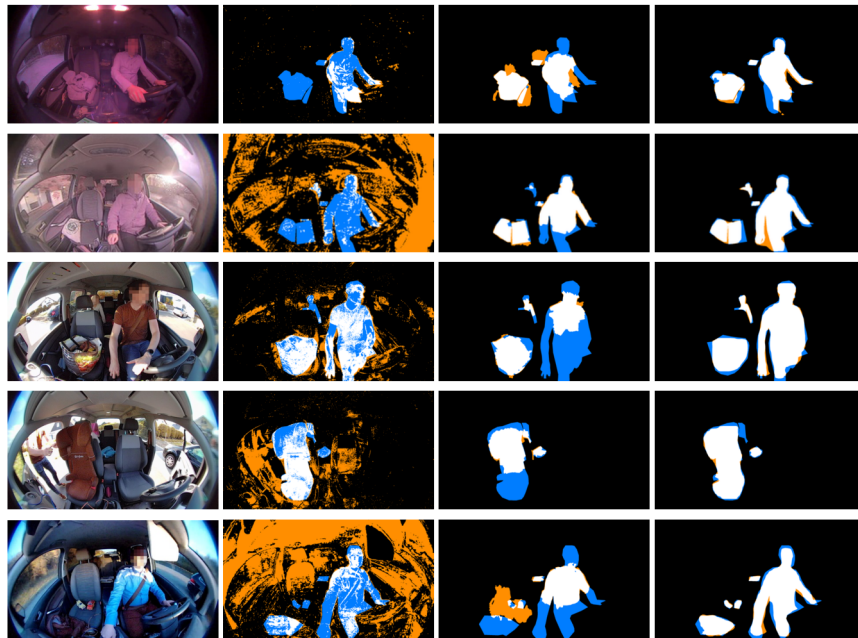


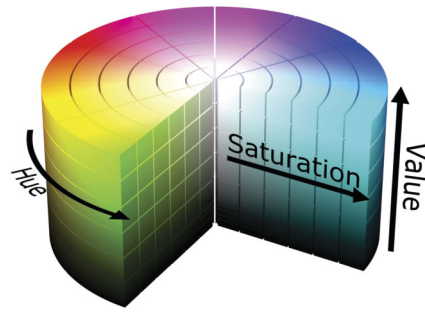
Figure 13 Comparison of the masks predicted by the investigated methods. 1st column: The RGB real-world images. 2nd column: The masks predicted by the GMM. 3rd column: The masks predicted by the Morphsnakes. 4th column: The masks predicted by the Mask R-CNN

Mask R-CNN is much more capable of handling the rather large variety in the recorded scenes. Static and moving objects / persons inside the car as well as static and moving backgrounds outside the car as well as varying illumination and shadows contribute to a complex scenery that classical methods cannot handle anymore. We also found out that the hunger for data of the Mask R-CNN to be reduced to some extent by state of the art data augmentation techniques. However, the only way to sate this hunger seems to be the recording and labeling of new data.

Interesting directions for the future are comparisons with other deep learning models, e.g. for semantic segmentation [60]. Besides that, recording additional data that covers difficult cases such as occupied child seats in the back of the car seems of importance. Also deep learning models that consider multiple frames as well as hybrid models that use the output of, e.g., the GMM as an input could be of interest. The latter technique could also be used to reduce the model complexity, making deep neural networks more suitable for embedded systems. Yet, the Mask R-CNN inference requires 0.34 seconds on a Titan XP GPU with 12 GB memory while the GMM model only requires 0.015 seconds per inference on an Intel Xeon E-2186M CPU with 2.90 GHz, a hardware component with way less compute resources. Furthermore, in the long run, the data collection and labeling process could be supported by methods that make proposals towards labeling those scenes that leverage the model's performance the most. This can be approached e.g. via active learning for image segmentation [61, 62].

Appendix A: Excursion: color spaces

Each pixel is modeled by a color space [63]. We investigate the influence of the choice of the color space for the performance of the implemented methods. To this end, the considered

Figure 14 The HSV color space. Source: [67]

color spaces RGB, HSV and Lab are briefly introduced below. For the mathematics behind the conversion between those colors spaces, we refer to [64] and [65].

RGB The RGB color space [63, 64, 66] can be described as an additive color system since the colors arise by a linear combination of the three primary colors Red, Green and Blue (RGB). Mathematically, the RGB color space can be comprehended as a cube which is located in a three dimensional Cartesian coordinate system. Thereby, each axis is represented by one of the color channels R, G or B. Thus, RGB colors are given by a three dimensional vector (r, g, b) where r, g and b describe the intensities of the corresponding color channels red, green and blue, respectively. The values of the color intensities lie in the interval $[0, M]$. The special case $r = g = b$ represents the colors white (M, M, M) , black $(0, 0, 0)$ and all gray-scale values inbetween.

HSV The problem about the RGB color space is that the colors are not created according to the color perception of a human. Intuitively, a human creates a color by selecting a color of a certain spectrum and then develop a desired saturation and brightness level. The HSV color space [63, 64, 66] is based on this idea. Thus, the goal of the HSV color space is to adapt the definition of colors to the color perception of humans. Hence, the HSV color space can be described as a perceptive color model. The colors in a HSV space derive by the determination of a Hue, Saturation and Value. Mathematically, the HSV color space can be described by a cylinder. The hue is represented by a pure color. All possible pure colors are organized on the borders of the circular base area of the cylinder as described by Fig. 14. Hence, the Hue is defined by an angle degree on the base area of the cylinder. The saturation describes how vibrant a color is and is determined by defining a radius inside the cylinder. Finally, the color channel V describes the brightness of a color and is represented by a height inside the cylinder. Usually, the value for this channel lies in the range of $[0, M]$ with $0 < M \leq 1$.

CIE $L^*a^*b^*$ (Lab) The CIE $L^*a^*b^*$, also denoted by Lab, was designed by the Commission Internationale d'Éclairage (CIE) in 1976. Lab [65, 68] is a uniform color space which should correlate with the color perception of humans analogous to the HSV color space. By the color channel L^* the luminosity of a color is described. The channels a^* and b^* represent the color pairs green-red and blue-yellow, respectively. By the value of those color pairs the saturation C_{ab}^* and the hue h_{ab}^* of a color are defined as

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}} \quad \text{and} \quad (30)$$

Table 11 Number of instances per main category of the object class in the training set

Images	Main group	Number of instances		
		Orig.	Add.	Total
Original	Laptop	3	0	3
	Bottle	4	5	9
	Smartphone	5	0	5
	Blanket	2	0	2
	Jacket	9	4	13
	Wallet	1	0	1
	PC-keyboard	1	0	1
	Sheet of paper	3	2	5
	Bag(pack)	10	10	20
	Baker's bag	1	1	2
	Socks	1	0	1
	Cardboard box	5	8	13
Additional	Suitcase	0	1	1
	Toys	0	3	3
	Face mask	0	1	1
	Hat	0	2	2
	Basket	0	1	1
	Food	0	2	2
	Cable drum	0	1	1
Total		45	41	86

Table 12 Number of instances per class in the training set

Class	Number of instances		
	Original	Additional	Total
Person	21	8	29
Object	45	41	86
Child seat	4	12	16

$$h_{ab}^* = \arctan\left(\frac{b^*}{a^*}\right), \quad (31)$$

respectively. In this work, especially the color channels V and L are of interest since it is possible to control the brightness of an image by both of these color channels independently of the hue and the saturation. Due to this reason, image enhancement methods regarding the contrast are generally applied on those two color channels. Common contrast enhancement methods, which are also investigated, are given by the Histogram Equalization (HE) [69] and the Contrast Limited Adaptive Histogram Equalization (CLAHE) [70].

Appendix B: Detailed statistics for the ISSO dataset

While the statistics over the images of the ISSO test set are given in Table 14, the statistics of the ISSO training set are described in Tables 11 to 13. Herein, the statistics over the first 500 images of the training set are described by column “Original”/“Orig.”. In column “Additional”/“Add.” the statistics over the 600 images by which the training set was extended are captured. The summary of all quantities is given by the column “Total”. In Table 11 it is also described by the row “Original” which objects the first 500 images of the training set contain. In row “Additional” the objects are recorded which have been added by the extension of the training set.

Table 13 Detailed description for the class “Person” of the training set regarding the characteristics gender and age

		Number of instances		
		Original	Additional	Total
Gender	Female	8	2	10
	Male	13	6	19
Age	Baby	1	1	2
	Child	0	1	1
	Adult	20	6	26

Table 14 Number of instances per main category of the class “Object” in the test set

Main category	Number of instances
Laptop	4
Jacket	10
Toys	6
Trolley	1
Glasses	1
Bag(pack)	5
Stack of books	1
Helmet	1
CD cover	1
Umbrella	2
Smartphone	3
Tissues	1
Wallet	2
Stack of clothes	1
Shoe	1
Beverage crate	1

Table 15 A list of the main categories and their corresponding object categories of the COCO training dataset from 2017

Main category	Object classes
person	person
vehicle	bicycle, car, motorcycle, airplane, bus, train, truck, boat
outdoor	traffic light, fire hydrant, stop sign, parking meter, bench
animal	bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe
accessory	backpack, umbrella, handbag, tie, suitcase
sports	frisbee, skis, snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket
kitchen	bottle, wine glass, cup, fork, knife, spoon, bowl
food	banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake
furniture	chair, couch, potted plant, bed

Appendix C: List of object classes of the COCO dataset

In Table 15 the main categories (exclusive the category “background”) and the corresponding object classes of the COCO training set, year 2017, are described.

Acknowledgements

The authors of this work would like to thank Fabian Kunst from the University of Wuppertal for providing the extended implementation of “Labelme”.

Funding

The authors acknowledge support from the Open Access Publication Fund of the University of Wuppertal. H.G. and M.R. acknowledge financial support through the research consortium bergisch.smart.mobility funded by the ministry for economy, innovation, digitalization and energy (MWIDE) of the state North Rhine Westphalia under the grant-no.

DMR-1-2. Open Access funding enabled and organized by Projekt DEAL.

bergisch.smart_mobility



Abbreviations

GMM, Gaussian Mixture Model; EM, Expectation Maximization; PDE, Partial differential equation; GAC, Geodesic active contours; ACWE, Active contours without edges; MGAC, Morphological geodesic active contours; MACWE, Morphological active contours without edges; R-CNN, Region-based Convolutional Neural Network; RPN, Region proposal network; FCN, Fully Convolutional Network; RoI, Region of Interest; FPN, Feature Pyramid Network; TP, True positives; FP, False positives; TN, True negatives; FN, False negatives; IoU, Intersection over Union; Pr, Precision; Re, Recall; Sp, Specificity; Acc, Accuracy; Sim, Similarity; F_1 , F_1 -score; HE, Histogram equalization; CHE (CLAHE), Contrast Limited Adaptive Histogram Equalization.

Availability of data and materials

The COCO and SVIRO dataset used during the current study are available under [47] and [46]. The ISSO dataset consists of images recorded by the company APTIV Services Deutschland GmbH and is not publicly available. The purpose of this dataset is to enable the feasibility study provided by the present article and it is not meant to be representative for a local or global population.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author contribution

CD prepared, conducted and documented the experiments for this work and drafted the manuscript. MR and HG supported this work by additional expert opinion and participated in the writing process. KF and TK supported this work by additional expert opinion and participated in organising and recording of the ISSO dataset. All authors read and approved the final manuscript.

Author details

¹School of Mathematics and Science, IZMD & IMACM, University of Wuppertal, Wuppertal, Germany. ²Aptiv Services Deutschland GmbH, Wuppertal, Germany.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 27 October 2021 Accepted: 20 December 2022 Published online: 30 December 2022

References

1. Koch C, Yoon JJ, Lii N. Evaluation of vision based in-vehicle applications. 2006.
2. Feld H, Mirbach B, Katrolia JS, Selim M, Wasenmüller O, Stricker D. Dfki cabin simulator: a test platform for visual in-cabin monitoring functions. In: Commercial vehicle technology 2020—proceedings of the 6th commercial vehicle technology symposium—CVT 2020. Commercial vehicle technology symposium (CVT), 6th international commercial vehicle technology symposium Kaiserslautern, Kaiserlautern, Germany. University of Kaiserslautern. Berlin: Springer; 2020.
3. Yoon JJ, Koch C, Ellis TJ. Vision based occupant detection system by monocular 3d surface reconstruction. In: Proceedings. The 7th international IEEE conference on intelligent transportation systems (IEEE cat. no. 04TH8749). 2004. p. 435–40. <https://doi.org/10.1109/ITSC.2004.1398939>.
4. Arbogast KB, DeNardo MB, Xavier AM, Durbin DR, Winston FK, Kallan MJ. Upper extremity fractures in restrained children exposed to passenger airbags. SAE Transact. 2003;112:540–7.
5. Mittal MK, Kallan MJ, Durbin DR. Breathing difficulty and tinnitus among children exposed to airbag deployment. Accid Anal Prev. 2007;39(3):624–8. <https://doi.org/10.1016/j.aap.2006.10.00>.
6. Nichols JL, Glassbrenner D, Compton RP. The impact of a nationwide effort to reduce airbag-related deaths among children: an examination of fatality trends among younger and older age groups. J Saf Res. 2005;36(4):309–20. <https://doi.org/10.1016/j.jsr.2005.05.00>.
7. Tatarinov D, Mica C, Di Mario Cola P, Watgen C, Landwehr J, Larsen P, Goniva T, Diewald AR, Gomez O. In: Proff H, editor. Radar basiertes Sensorsystem zur Kindererkennung in verlassenen Fahrzeugen. Wiesbaden: Springer; 2019. p. 265–72.
8. Centers for Disease Control and Prevention. Child Passenger Safety. Accessed 05 January 2021. 2020. <https://www.cdc.gov/injury/features/child-passenger-safety/index.html>
9. SAE International. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. 2016.

10. Diewald AR, Landwehr J, Tatarinov D, Di Mario Cola P, Watgen C, Mica C, Lu-Dac M, Larsen P, Gomez O, Goniva T. RF-based child occupation detection in the vehicle interior. In: 2016 17th international radar symposium (IRS). 2016. p. 1–4. <https://doi.org/10.1109/IRS.2016.7497352>.
11. Harville M, Gordon G, Woodfill J. Foreground segmentation using adaptive mixture models in color and depth. In: Proceedings IEEE workshop on detection and recognition of events in video. 2001. p. 3–11. <https://doi.org/10.1109/Event.2001.938860>.
12. Camplani M, Salgado L. Background foreground segmentation with rgb-d kinect data: an efficient combination of classifiers. *J Vis Commun Image Represent*. 2014;25(1):122–36. <https://doi.org/10.1016/j.jvcir.2013.03.00>.
13. Kim K, Chalidabhongse TH, Harwood D, Davis L. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*. 2005;11(3):172–85. <https://doi.org/10.1016/j.rti.2004.12.00>. Special Issue on Video Object Processing.
14. Guo J, Liu Y, Hsia C, Shih M, Hsu C. Hierarchical method for foreground detection using codebook model. *IEEE Trans Circuits Syst Video Technol*. 2011;21(6):804–15. <https://doi.org/10.1109/TCSVT.2011.2133270>.
15. Guo X, Wang X, Yang L, Cao X, Ma Y. Robust foreground detection using smoothness and arbitrariness constraints. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. *Computer vision—ECCV 2014*. Cham: Springer; 2014. p. 535–50. ISBN 978-3-319-10584-0.
16. Bouwmans T. Traditional and recent approaches in background modeling for foreground detection: an overview. *Comput Sci Rev*. 2014;11. <https://doi.org/10.1016/j.cosrev.2014.04.001>.
17. McIvor AM. Background subtraction techniques. *Proc Image Vis Comput*. 2000;4:3099–104.
18. Barnich O, Van Droogenbroeck M. Vibe: a universal background subtraction algorithm for video sequences. *IEEE Trans Image Process*. 2011;20(6):1709–24. <https://doi.org/10.1109/TIP.2010.2101613>.
19. Sen-Ching C, Kamath C. Robust background subtraction with foreground validation for urban traffic video. *EURASIP J Adv Signal Process*. 2005;14. <https://doi.org/10.1155/ASP.2005.2330>.
20. Zivkovic Z. Improved adaptive gaussian mixture model for background subtraction. vol. 2. 2004. p. 28–312. ISBN 0-7695-2128-2.
21. Goodfellow I, Bengio Y, Courville A. *Deep learning*. Cambridge: MIT Press; 2016. <http://www.deeplearningbook.org>.
22. Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. In: *Proceedings of IEEE conf. computer vision patt. recog*. vol. 2. 2007.
23. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. *J R Stat Soc, Ser B, Methodol*. 1977;39(1):1–38.
24. Horn BKP. *Robot vision*. MIT electrical engineering and computer science series. Cambridge: MIT Press; 1986. ISBN 978-0-262-08159-7.
25. Kass M, Witkin A, Terzopoulos D. Snakes: active contour models. *Int J Comput Vis*. 1988;1:321–31. <https://doi.org/10.1007/BF00133570>.
26. Caselles V, Kimmel R, Sapiro G. Geodesic active contours. *Int J Comput Vis*. 1997;22:61–79. <https://doi.org/10.1109/ICCV.1995.466871>.
27. Chan TF, Vese LA. Active contours without edges. *IEEE Trans Image Process*. 2001;10(2):266–77. <https://doi.org/10.1109/83.902291>.
28. Osher S, Sethian JA. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys*. 1988;79:12–49.
29. Márquez-Neila P, Baumela L, Alvarez L. A morphological approach to curvature-based evolution of curves and surfaces. *IEEE Trans Pattern Anal Mach Intell*. 2014;36(1):2–17. <https://doi.org/10.1109/TPAMI.2013.106>.
30. Kimmel R. The Osher–Sethian level set method. *Numerical geometry of images: theory, algorithms, and applications*. New York: Springer; 2003.
31. Cao F. *Geometric curve evolution and image processing*. 1805th ed. Lecture notes in mathematics. vol. 1. Berlin: Springer; 2003.
32. Soille P. *Morphological image analysis. Principles and applications*. 2nd ed. Berlin: Springer; 2002.
33. Alvarez L, Guichard F, Lions P-L, Morel J-M. Axioms and fundamental equations of image processing. *Arch Ration Mech Anal*. 1993;123:199–257.
34. Guichard F, Morel J-M, Ryan R. *Contrast invariant image analysis and PDE's*. 2004.
35. Appell J, Vähä M. *Elemente der Funktionalanalysis*. Wiesbaden: Vieweg+Teubner Verlag; 2005.
36. Catté F, Dibos F, Koepfler G. A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets. vol. 32. 1994. p. 26–30. <https://doi.org/10.1109/ICIP.1994.413268>.
37. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. Facebook AI Research (FAIR). 2018. [arXiv:1703.06870v3](https://arxiv.org/abs/1703.06870v3).
38. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2016. [arXiv:1506.01497v3](https://arxiv.org/abs/1506.01497v3).
39. Girshick R. Fast R-CNN Microsoft Research. 2015. [arXiv:1504.08083v2](https://arxiv.org/abs/1504.08083v2).
40. Long J, Shelhamer E, Darrell T. Fully Convolutional Networks for Semantic Segmentation. UC Berkeley. 2015. [arXiv:1411.4038v2](https://arxiv.org/abs/1411.4038v2).
41. Ho Y, Wooley S. The real-world-weight cross-entropy loss function: modeling the costs of mislabeling. *IEEE Access*. 2019;8:4806–13. <https://doi.org/10.1109/ACCESS.2019.2962617>.
42. Huber P-J. Robust estimation of a location parameter. *Ann Math Stat*. 1964;35(1):73–101. <https://doi.org/10.1214/aoms/1177703732>.
43. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. Microsoft Research. 2015. [arXiv:1512.03385v1](https://arxiv.org/abs/1512.03385v1).
44. Lin T-Y, et al. Feature Pyramid Networks for Object Detection. 2017. Facebook AI Research (FAIR), Cornell University and Cornell Tech. [arXiv:1612.03144v2](https://arxiv.org/abs/1612.03144v2).
45. Kentaro W. labelme: Image Polygonal Annotation with Python. 2016. <https://github.com/wkentaro/labelme>.
46. Dias Da Cruz S, et al. SVIRO: Synthetic Vehicle Interior Rear Seat Occupancy Dataset and Benchmark. 2020. <https://sviro.kl.dfki.de/data/>. [arXiv:2001.03483v1](https://arxiv.org/abs/2001.03483v1).
47. Lin T-Y, et al. Microsoft COCO: Common Objects in Context. 2015. <https://cocodataset.org/#download>. [arXiv:1405.0312v3](https://arxiv.org/abs/1405.0312v3).

48. Maddalena L, Petrosino A. Background subtraction for moving object detection in rgb-d data: a survey. *J Imaging*. 2018;4:71. <https://doi.org/10.3390/jimaging4050071>.
49. Vorontsov I, Kulakovskiy I, Makeev V. Jaccard index based similarity measure to compare transcription factor binding site models. *Algorithms for molecular biology. AMB*. 2013;8:23. <https://doi.org/10.1186/1748-7188-8-23>.
50. Wang Y. Optimizing intersection-over-union in deep neural networks for image segmentation. vol. 10072. 2016. p. 234–244. ISBN 978-3-319-50834-4.
51. Sasaki Y. The truth of the f-measure. *Teach Tutor Mater*. 2007.
52. Mordvintsev A. OpenCV-Python Tutorials: Background Subtraction. Accessed 27 September 2020 (2013). https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html.
53. OpenCV: cv::BackgroundSubtractorMOG2 Class Reference. Documentation to the OpenCV functions. Accessed 27 September 2020. https://docs.opencv.org/3.4/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html.
54. OpenCV: cv::BackgroundSubtractorMOG2 Class Reference. Documentation to the OpenCV function apply. Accessed 27 September 2020. https://docs.opencv.org/3.4/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html#a682adde901148d85450435e6cc0de4a1.
55. Márquez-Neila P. Morphological Snakes. Github. 2018. <https://github.com/pmneila/morphsnakes>.
56. Abdulla W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Github. 2017. https://github.com/matterport/Mask_RCNN.
57. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2017. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
58. Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA. Albumentations: fast and flexible image augmentations. *Information*. 2020;11(2). <https://www.mdpi.com/2078-2489/11/2/125>.
59. Sun B, Feng J, Saenko K. Return of frustratingly easy domain adaptation. 2015. CoRR. [arXiv:1511.05547](https://arxiv.org/abs/1511.05547).
60. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. 2018. CoRR. [arXiv:1802.02611](https://arxiv.org/abs/1802.02611).
61. Colling P, Roesse-Koerner L, Gottschalk H, Rottmann M. MetaBox+: a new region based active learning method for semantic segmentation using priority maps. In: *Proceedings of the 10th international conference on pattern recognition applications and methods—volume 1: ICPRAM*. SciTePress; 2021. p. 51–62. <https://doi.org/10.5220/0010227500510062>.
62. Kasarla T, Nagendar G, Hegde GM, Balasubramanian V, Jawahar CV. Region-based active learning for efficient labeling in semantic segmentation. In: *2019 IEEE winter conference on applications of computer vision (WACV)*. 2019. p. 1109–17. <https://doi.org/10.1109/WACV.2019.00123>.
63. Zhang D. *Fundamentals of image data mining: analysis, features, classification and retrieval*. Texts in computer science. Cham: Springer; 2019. ISBN 978-3-030-17988-5.
64. Burger W, Burge MJ. *Principles of digital image processing. Fundamental techniques*. Undergraduate topics in computer science. London: Springer; 2009.
65. Burger W, Burge MJ. *Principles of digital image processing. Core algorithms*. Undergraduate topics in computer science. London: Springer; 2009.
66. Stratmann L. Color Systems. Accessed 27 May 2020. <https://web.cs.uni-paderborn.de/cgvb/colormaster/web/color-systems.html>.
67. Horvath M. Mike-Wikipedia-Illustrations. accessed: 17 September 2020. <https://github.com/mjhorvath/Mike-Wikipedia-Illustrations>.
68. Cheung V. Uniform color spaces. In: Chen J, Cranton W, Fihn M, editors. *Handbook of visual display technology*. Berlin Springer; 2012. p. 161–9.
69. Gonzalez RC, Woods RE. *Digital Image Processing*. 3rd ed. Pearson International Edition prepared by Pearson Education.
70. Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, Romeny BTH, Zimmerman JB. Adaptive histogram equalization and its variations. *Comput Vis Graph Image Process*. 1987;39(3):355–68. [https://doi.org/10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X).

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)