Guided SAM: Label-Efficient Part Segmentation

Sabina B. van Rooij(✉) and Gertjan J. Burghouts

TNO, The Hague, The Netherlands
sabina.vanrooij@tno.nl

Abstract. Localizing object parts precisely is essential for tasks such as object recognition and robotic manipulation. Recent part segmentation methods require extensive training data and labor-intensive annotations. Segment-Anything Model (SAM) has demonstrated good performance on a wide range of segmentation problems, but requires (manual) positional prompts to guide it where to segment. Furthermore, since it has been trained on full objects instead of object parts, it is prone to over-segmentation of parts. To address this, we propose a novel approach that guides SAM towards the relevant object parts. Our method learns positional prompts from coarse patch annotations that are easier and cheaper to acquire. We train classifiers on image patches to identify part classes and aggregate patches into regions of interest (ROIs) with positional prompts. SAM is conditioned on these ROIs and prompts. This approach, termed 'Guided SAM', enhances efficiency and reduces manual effort, allowing effective part segmentation with minimal labeled data. We demonstrate the efficacy of Guided SAM on a dataset of car parts, improving the average IoU on state of the art models from 0.37 to 0.49 with annotations that are on average five times more efficient to acquire.

Keywords: Image segmentation · Object parts · Foundation models

1 Introduction

Precise localization of object parts is essential for many tasks, including scene perception [11], recognizing objects by the their parts [4,15], part-whole understanding [1,5] and robotic manipulation [8]. A specific part indicates what the object can do, e.g. the sharp blade of the knife can be used to cut, whereas the handle can be used to hold it. Segmentation is helpful to localize where the part is exactly, which is a requirement for a robot to grasp it at the right point, use it in the right way, or to understand the attributes of the part such as size and shape. However, segmentation of parts is not trivial. Boundaries between parts are not always clear (e.g. the hood of a car), parts can be very small compared

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-78110-0_19.

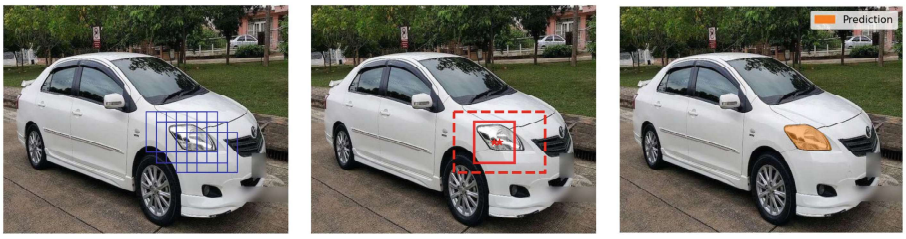
A. Antonacopoulos et al. (Eds.): ICPR 2024, LNCS 15329, pp. 291–306, 2025.
https://doi.org/10.1007/978-3-031-78110-0_19

to the full object size (e.g. the side mirror of a car), and they can have large inter-class variations (e.g. cars have very different lights).

Recently, advanced methods have become available for part segmentation. VLPart [17] trains a model on various granularities at the same time: parts, objects, and image annotations jointly provide multiscale learning signals. An object is parsed to find its parts, which provides the part segmentation with helpful contextual cues. OV-PARTS [18] builds on CLIP [13] and adapts it for part segmentation. The context of the part is provided by an object mask prompt and a compositional prompt shifts the model's attention to the parts [16]. Grounded SAM leverages Grounding DINO [7] as an open-vocabulary model to localize objects or parts, which are subsequently segmented by the Segment-Anything Model (SAM) [6]. The performance of these models is impressive. However, on common object parts they may still fail, see e.g. Fig. 1 (b) and (c).

Today's part segmentation models can be finetuned or retrained, but this typically requires large datasets. OV-PARTS was trained using ADE20K-Part-234 [18] and VLPart was trained using PACO [14] with 641K part masks. Moreover, part masks are labour intensive annotations, i.e. pixel-precise masks. Therefore, improving the models on specific parts of interest involves large datasets or labour intensive labelling. Our objective is a methodology that requires low amounts of labelled images, and moreover, annotations that are easy to acquire with a few clicks per image.

(a) Image and ground truth (b) Grounded SAM [16] (c) VLPart [17]



(d) Initial guidance (e) Refined guidance (f) Our prediction

Fig. 1. Guiding SAM (bottom row) for part segmentation, where SOTA methods fail (top row), our patch guidance (d) and refinement (e) is more effective.

Our starting point is the Segment-Anything Model (SAM) [6], because it has demonstrated a very strong performance on a wide range of image contents and across various granularities from objects to parts. But, SAM is not directly applicable to part segmentation, because it requires manual guidance where to segment. This guidance comes in the form of one or more locations in the image, which are referred to as positional prompts. We want to substitute the manual prompting by automated prompting, such that the part segmentation can be performed in a fully automated manner. This positional prompting is tailored to the part of interest. Our approach is to *learn* the positional prompting, from coarser annotations that are easy to acquire. Coarse annotations come from image patches of approximately 1/14th of the image width and height. The annotation says whether a patch contains the part. Such patch annotations are much coarser and simpler to acquire than pixel-precise masks, therefore this strategy is significantly more efficient. To advance the efficiency further, we leverage prototypical patches [9] that group the parts already reasonably well before annotation. For each part of interest, we learn a patch classifier to predict whether a test patch contains the part. For the representation of a patch, we use DINOv2 for its strong representational power for a wide variety of image contents [10]. For a sense of context, the predicted patches are locally grouped into regions of interest (ROIs). For the positional prompt within the ROI, a location is inferred using a maximum likelihood formulation. SAM is invoked on the ROI with the positional prompt. The advantage of a ROI is two-fold: it provides a contextual cue and avoids the necessity to process the full image. Processing only the ROI is advantageous for reducing computations and avoiding false alarms in irrelevant image regions. We coin our method 'Guided SAM' and it is illustrated in Fig. 1.

The efficacy of Guided SAM is measured on a dataset of car parts. This is an interesting testset, because the parts vary significantly in size, from very small (a tiny back light), small (side mirror, front light), medium (bumper, trunk) to large (door, hood). We compare our method with recent models that have shown impressive performance, namely vision-language models that take textual prompts: Grounded SAM [16] and VLPart [17]. Also, we compare various positional prompting strategies combined with SAM [6]. We will show the efficiency of acquiring patch annotations and their suitability for Guided SAM. It is possible to learn a good segmentation model for a part from only 16 to 64 images, which outperforms state of the art (SOTA) models, while requiring only 5 clicks per image on average.

2 Related Work

For part segmentation, vision-language models have been proposed recently, which can be prompted with a textual description of the part. VLPart [17] trains the model on the part-, object- and image-level to align language and image. An object is parsed by dense semantic correspondence. This approach benefits from various data sources and foundation models, as demonstrated in experiments where the model was applied to unseen object-part combinations

(open-vocabulary). OV-PARTS [18] modifies and tailors CLIP [13] for part segmentation. An object mask prompt is proposed to enable the model to take the context into account. To attend more to the parts than whole objects or scenes, a compositional prompt was proposed to reshift attention [16]. Since OV-PARTS and VLPart are both designed to perform open-vocabulary part segmentation, we only consider VLPart in our experiments.

Grounded SAM [16] combines two powerful models: Grounding DINO [7] and SAM [6]. Grounding DINO localizes boxes in the image based on a textual description. Each box contains a prediction where the target may be, in the form of a initial mask. This box is represented by an embedding pair of the top-left corner and the bottom-right corner that serve as positional prompts. These prompts are provided to SAM [6], which segments the target.

Grounded SAM [16] inspired us to look more deeply into the positional prompts themselves. Rather than using the box representations as input for SAM, we aim for a regional prediction that is centered around the part already, to acquire a small but tailored sense of context. Moreover, we want to predict the positional prompts more precisely. For that purpose, we take inspiration from OV-PARTS [18] and Grounded SAM [16], by following their strategy to incorporate some image context around the part. Instead of an implicit context via multiscale annotation (VLPart [17]), we follow OV-PARTS and Grounded SAM by providing an explicit context in the form of a mask or box. Our ROI approach differs because the ROI is more centered around the part, instead of the full object.

3 Guided SAM

Our method segments parts of objects, such as the light of a car, see Fig. 1 (a). Two state of the art methods, Grounding SAM and VLPart, fail on this task, leading to respectively false positives in Fig. 1 (b) and false negatives in Fig. 1 (c). Our objective is to train a capable part segmentation model, while requiring a small amount and labour-efficient type of human annotations. For label-efficiency, we leverage a model M that has strong performance on segmentation already: SAM [6]. This model cannot be applied directly to an image in order to segment a specific part. It requires a spatial cue, provided as a positional prompt $P_{(x,\,y)}$ (a pixel location). Our rationale is to *learn* the spatial cue for a part, in order to guide SAM towards regions in the image where the part is located, P_{ROI}. Our guidance model $\mathcal{G}$ takes an image I and a part C and produces a set of tuples:

$$\mathcal{G}(I \,|\, C) \rightarrow \{(P^i_{ROI},\, P^i_{(x,\,y)})\}_{i \in 1:N} \tag{1}$$

Here, P^i_{ROI} serves as a region of interest (ROI) that conditions where SAM is applied. For each P^i_{ROI}, $P^i_{(x,\,y)}$ serves as the positional prompt for SAM to segment the part. The guidance model $\mathcal{G}$ involves a learner $\mathcal{L}$ that classifies whether an image patch p^j contains the part C: $\mathcal{L}(I \,|\, p^j) \rightarrow c$, where c is the confidence for the part class. Classifying patches is a much simpler learning task

than predicting pixel-precise segments. Moreover, the learning requires a simpler form of annotation, i.e., a binary label for the patch if it contains the part or not. Our hypothesis is that such a patch classifier can be learned with a small amount of labels that are simple to annotate. Figure 1 (d) shows the classified patches that are likely to contain the part. A ROI P^i_{ROI} is generated by grouping the classified patches. The positional prompt $P^i_{(x,\,y)}$ for each ROI P^i_{ROI} is inferred from its constituent patches and their respective confidences. Figure 1 (e) shows $(P^i_{ROI}, P^i_{(x,\,y)})$ that was inferred from the classified patches in Fig. 1 (d).

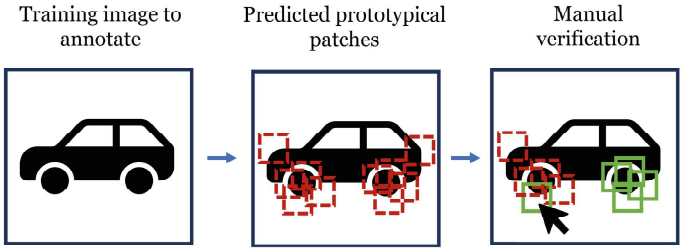
For the segmentation of an object part, both the ROI P^i_{ROI} and the positional prompt $P^i_{(x,\,y)}$ are used. SAM is conditioned on P^i_{ROI}, by passing only the respective image contents. This avoids false positives at irrelevant image regions. SAM is also conditioned on $P^i_{(x,\,y)}$, in order to give it a good starting point for segmentation. Figure 1 (f) shows the part segmentation. Our method enables to use SAM for part segmentation after providing a few labeled patches.

The flow diagram of GuidedSAM is illustrated in Fig. 2. The annotation of patches is shown in Fig. 2 (a) and will be further explained in Sect. 3.1. The inference steps before the segmentation are shown in Fig. 2 (b) and will be covered in Sect. 3.2. Finally, Fig. 2 (c) shows the guided segmentation with multiple model variants that will be explained in Sects. 3.3 and 3.4.

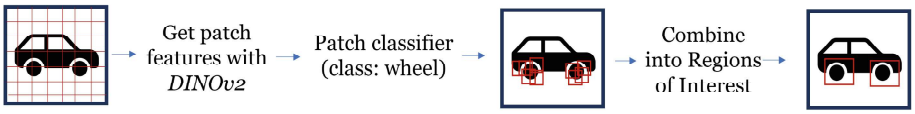
3.1 Prototypical Patches

Our learner $\mathcal{L}$ requires a set of binary labels for respective patches whether they contain the part of interest: $\mathcal{D} = \{(z_i, l_i)\}_{i=1}^{M}$ with M samples, each consisting of a patch z_i and a label $l_i \in [0, 1]$ indicating presence of the part. To arrive at $\mathcal{D}$, the problem is that patches containing parts have a low prevalence, considering that the parts are typically small. Drawing a random selection of patches for annotation, is not efficient. Instead, we select patches that have a larger probability of containing the part. We group similar patches by means of prototypical patches [9]. The prototypes do not have a name, neither are they necessarily related to the part of interest. To relate the prototypes to the part, we match each prototype to the part name, using the visual-textual similarity measure of CLIP [13]. Each prototype is assigned a score for the part of interest. Figure 3 shows examples for various car parts, illustrating that the prototypes group together patches that relate to the respective parts.

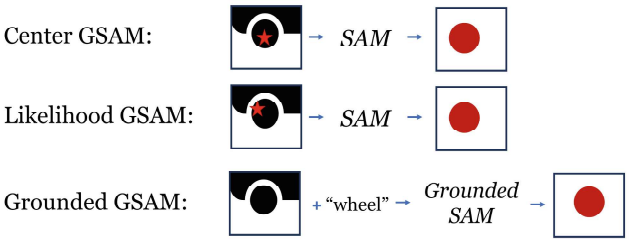
For a specific part, the prototypes are ranked by descending CLIP score. Each prototype is verified by a human annotator. For illustration purposes, we indicate this for an example image for the part 'wheel' in Fig. 2 (a). This involves one affirmative click if all patches of the prototype contain the part. Similarly, one negative click is required when none of the patches contain the part. More clicks are needed when most patches contain the part, by negating the fewer patches that do not contain it, or vice versa. This procedure yields (z_i, l_i) that constitute $\mathcal{D}$.



(a) Annotation with prototypical patches as guidance.



(b) Inference with GuidedSAM before segmentation.



(c) Segmentation using different model variants.

Fig. 2. Various elements of the pipeline for GuidedSAM, showing the efficient annotation process in (a), the inference until the segmentation in (b) and the segmentation of the regions of interest with different model variants in (c).

3.2 Guidance Classifier

Given $\mathcal{D} = \{(z_i, l_i)\}_{i=1}^{M}$, the classifier $\mathcal{L}$ is learned, which predicts for a test patch z_j the probability that it contains the part. The patch z_j is represented as a feature vector by a model $\phi(\cdot)$: $z_j^{\phi} = \phi(z_j)$. For $\phi(\cdot)$ we consider DINOv2 [10] which has proven to be a robust feature extractor. $\mathcal{L}_p$ is an SVM [2] with a radial basis function as the kernel. The parameters are learned from train samples $\{(z_i^{\phi}, l_i)\}$. During inference, the trained classifiers are used to predict a rough location of the parts of interest in the test image. This process is illustrated on the left side of Fig. 2 (b).

3.3 Guided Segmentation

A ROI P_{ROI}^{i} is generated by grouping the predicted patches $\{p_j\}$ that are likely to contain the part: $\{\mathcal{L}(I \mid p_j) > c_t\}$, where c_t is a threshold on the confidence

c. An example is provided in Fig. 1 (d). The grouping is based on the patches from $\{p_j\}$ that overlap: $\{p_k\}_{IoU>0}$, where $\{p_k\} \subset \{p_j\}$. The ROI P^i_{ROI} is the combination of the minimum and maximum coordinates of the patches in $\{p_k\}$. This is also shown in Fig. 2 (b), where the individual patches are combined into larger ROIs that take into account more context. The positional prompt $P^i_{(x,\,y)}$ for each ROI P^i_{ROI} is inferred from its constituent patches p_k and their respective confidences c_k by taking the center coordinates of the patch with the highest confidence. Figure 1 (e) shows $(P^i_{ROI}, P^i_{(x,\,y)})$. Guided SAM is the conditioning of SAM on $(P^i_{ROI}, P^i_{(x,\,y)})$. An example result is shown in Fig. 1 (f).

3.4 Model Variants

Besides the version of Guided SAM described in Sect. 3.3, we also consider other variants of our conditioning. We make a distinction between applying the segmentation on the ROI P^i_{ROI} (i.e. the combination of the classified patches) or taking the individual patches p_k as ROIs. This is also depicted in Fig. 1 (e), where the bounding box with the dashed line stands for a ROI of combined patches and the smaller box represents an individual patch. For these ROI types there are various options to segment or prompt. Firstly we can replace the segmentation model SAM by Grounded SAM [16] and apply it to the ROI: we coin this model Grounded Guided SAM (GGSAM). This version takes a textual prompt instead of the positional prompt $P^i_{(x,\,y)}$. Secondly, we can infer the positional prompt $P^i_{(x,\,y)}$ from the center coordinates of the ROI, which we coin Center Guided SAM (CGSAM). The version that was described before, where the positional prompt is inferred from the center of the patch with the maximum confidence in P^i_{ROI}, is coined Likelihood Guided SAM (LGSAM). This method can only be applied to P^i_{ROI}, since the other ROI is just a single patch. These model variants are illustrated in Fig. 2 (c) on the combined ROIs.

3.5 Computational Load

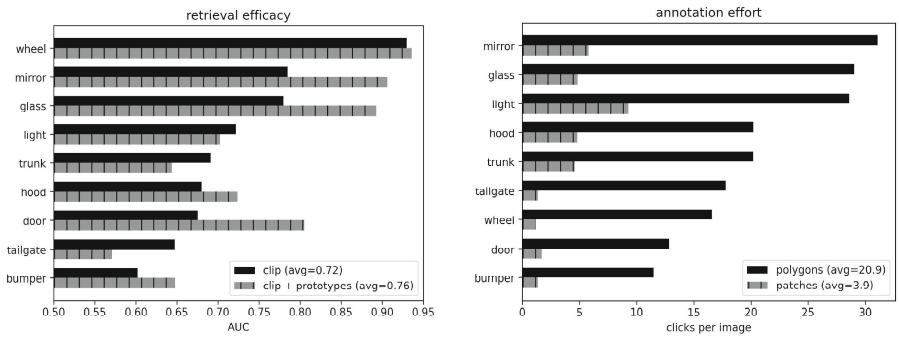
The computational steps for model inference are shown in Fig. 2 (b). These computations are required on top of the original SAM. We apply an efficient DINOv2 [10] variant to compute the patch features, i.e. ViT-B, which has only 86M parameters. For each part class, the same DINOv2 features are re-used, with a class-specific part classifier. This classifier is an SVM, which involves negligible computations compared to SAM. SAM has 94.7M parameters, comparable to DINOv2 ViT-B, so the computation time of our Guided SAM will be approximately doubled by the classifier guidance. If computational efficiency is essential, faster alternatives are available, e.g. [19], which has a faster backbone. Currently our method applies SAM to every region of interest that is proposed by the part classifiers. This can be implemented more efficiently by re-using its feature maps and only re-running SAM's efficient head on the various regions of interest.

Fig. 3. Prototypical patches group together similar object parts, which facilitates the annotation.

4 Experiments

4.1 Setup

For evaluation we consider the Car Parts Segmentation dataset [12], because of its large inter-class and intra-class variations. The part classes have very different sizes relative to the object. The same part can have different appearances, e.g. forms, sizes and colors. The dataset contains 400 images with annotated segmentation masks of 18 part classes. We merged the different sides (front vs. back, left vs. right) to one part class. There are a total of 9 part classes: bumper, glass, door, light, hood, mirror, tailgate, trunk, and wheel. For language-guided methods (i.e. VLPart and Grounded SAM) we made slight variations to these class names that better reflect the nature of the classes (e.g. replacing glass for window). As a metric, we consider the IoU for each part. The training efficiency is established by increasing the number of training images from 1, 2, 4, 8, ..., 64.



(a) Retrieval efficacy. (b) Annotation efficiency.

Fig. 4. Prototypical patches are helpful to find the parts (a). Annotating patches is on average >5x more efficient (b).

4.2 Patch Selection

For finding the patches that contain the part, we evaluate the merit of the prototypical patches. To that end, we compare the retrieval efficacy of CLIP with and without the prototypes. Figure 4(a) shows that for most part classes, there is an advantage to consider the prototypical patches. This means that it is helpful to consider the average CLIP score for each prototype before ranking.

To evaluate the annotation efficiency, we compare the amount of manual clicks that are necessary for conventional annotation of polygons to create pixel masks, and for our patch-based annotations using the prototypes. Figure 4(b) shows the annotation efforts for both strategies for the various part classes. On average, annotating patches with our prototype strategy is more than 5

times more efficient. The speedup is most prominent for tailgate and wheel. For conventional annotation, bumper involves the least clicks, because it has a simple shape. Even compared to bumper, all parts are annotated with fewer clicks per image when using the patch-based strategy.

4.3 Guidance Classifier

As shown in Table 1, the learned patch classifier $\mathcal{L}_p$ performs classification of object parts very accurately. For 'door' the performance is the highest: AUC = 0.994 when using 64 training images (in following subsections we will experiment with fewer training images). 'Wheel' also has a very high performance: AUC = 0.990, possibly because of its distinct visual features. For 'trunk' the performance is the lowest, but still very high: AUC = 0.977. Trunk does not have a distinctive boundary and the part is often a flat surface without much texture or distinctive visual features. Light is also somewhat harder to classify (AUC = 0.979). It is a very small part and shows a lot of intra-class variation, such as different shape, size, color (depending on whether it is on or off).

Table 1. The patch classifier performs very accurate classification of object parts (average AUC≈0.985).

Class	door	wheel	mirror	hood	glass	tailgate	bumper	light	trunk
AUC	0.994	0.990	0.989	0.988	0.988	0.987	0.986	0.979	0.977

4.4 Comparison to SOTA

We compare against two methods: VLPart[1] [17] and Grounded SAM [16]. Our method is trained on 64 images. In following subsections we evaluate the impact of having fewer training images. Table 2 reveals that Guided SAM outperforms VLPart and Grounded SAM for most object parts. Overall it is the best performer, on average IoU = 0.493 compared to 0.370 (VLPart) and 0.124 (Grounded SAM). VLPart does perform best at larger or common parts, such as wheel, door and mirror. It is surprising that VLPart does not perform better at other parts, given that it was trained on datasets that include all parts from Table 2, i.e. LVIS [3] and PACO [14]. Grounded SAM performs much worse across the board, probably because it is optimized for objects and not for object parts, although on some parts it performs somewhat better: e.g. wheel and bumper. These are larger parts or parts with clear boundaries. Some parts have a very low performance for both VLPart and Grounded SAM: light, tailgate, and trunk, AUC≈0.04. For these parts, Guided SAM performs much better: AUC≈0.35.

[1] For VLPart we use a confidence threshold of 0.5. For the results of VLPart with varying confidence thresholds, see Supplementary Material.

Table 2. Performance of VLPart [17], Grounded SAM [16], and Guided SAM on the Car Parts dataset in terms of IoU. Bold numbers indicate the best performance per part for the three methods. Guided SAM outperforms VLPart and Grounded SAM for most object parts.

	VLPart	Grounded SAM	Guided SAM (ours)
wheel	**0.800**	0.305	0.683
glass	0.621	0.089	**0.638**
door	**0.736**	0.202	0.635
bumper	0.027	0.299	**0.605**
hood	0.440	0.089	**0.553**
light	0.000	0.041	**0.377**
tailgate	0.000	0.035	**0.370**
trunk	0.006	0.048	**0.314**
mirror	**0.696**	0.009	0.259
average	0.370	0.124	**0.493**

Predictions of the tested models are illustrated for three examples, see Fig. 5. The top row indicates the ground truth, where the other rows show the predicted part segments. VLPart (b) is sometimes very impressive (left), while at other times it misses the part completely (middle), or over-segments it (right). Grounded SAM (c) typically segments the full objects rather than the part. Guided SAM provides a balance, often segmenting the part well, while sometimes over-segmenting or segmenting the background rather than the part.

4.5 Evaluating Model Variants

We evaluate the model variants from Sect. 3.4. As a short recap, we have two main divisions: taking the P^i_{ROI} as the ROI, or its consituent patches $\{p_k\}$ as individual ROIs. This is the top row in Table 3. For each ROI type, there are various options to segment or prompt: Grounded Guided SAM (GGSAM) which uses Grounded SAM as the segmenter, Center Guided SAM (CGSAM) which uses the ROI center as the positional prompt, and Likelihood Guided SAM (LGSAM) which uses the most likely location (i.e. the center of the patch with the highest confidence) as the positional prompt. To establish the effect of the segmentation methods, we also compare with assigning the full patch as the segment, i.e. no segmentation. We refer to this variant as Naive.

Table 3 presents the IoU scores per part for the model variants. ROI guidance is more effective than patch guidance, in most cases. The exceptions are tailgate and trunk, but for these parts ROI guidance performs similarly. Using a positional prompt based on the likelihood (LGSAM) is best on average. There is no single model variant that performs best for all parts. CGSAM performs best on light, hood and mirror. GGSAM appears to perform well on larger car parts that have a distinct boundary, such as bumper, door and wheel. Interestingly,

Fig. 5. VLPart either segments the object part very well or misses it completely, whereas Grounded SAM typically over-segments severely and often segments the full object. Guided SAM provides a balance, often segmenting the part well, while sometimes over-segmenting.

the performance of GGSAM (i.e. Grounded SAM as segmenter) is much better than applying Grounded SAM on the full image, i.e. without our guidance (Table 2). We conclude that our guidance is also helpful for an existing model.

Table 3. Performance for Region-of-interest (ROI) and Patch guidance for Grounded Guided SAM (GGSAM), Center Guided SAM (CGSAM), and Likelihood Guided SAM (LGSAM) in terms of IoU. Bold numbers indicate the best performance per part for all model variants. ROI guidance is more effective than patch guidance, where segmentation based on likelihood (LGSAM) is best on average.

	Region-of-interest				Patches	
	GGSAM	CGSAM	LGSAM	Naive	GGSAM	CGSAM
bumper	**0.605**	0.319	0.423	0.487	0.560	0.550
glass	0.317	0.626	**0.638**	0.354	0.458	0.480
door	**0.635**	0.447	0.399	0.440	0.508	0.582
light	0.173	**0.377**	0.371	0.170	0.211	0.217
hood	0.395	**0.553**	0.505	0.362	0.406	0.478
mirror	0.063	**0.259**	0.206	0.102	0.113	0.148
tailgate	0.325	0.165	0.370	0.318	0.348	**0.389**
trunk	0.281	0.173	0.314	0.264	0.293	**0.338**
wheel	**0.683**	0.369	0.513	0.246	0.440	0.329
average	0.386	0.365	**0.415**	0.305	0.371	0.390

4.6 Model Selection

There is no single model variant that performs best for all parts (see Table 3). Therefore, we explore model fusion. When selecting the best scores per part out of the three ROI-based methods we get an IoU of 0.493, which is a great improvement over the best model variant (LGGAM with 0.415). We want to understand how many images are needed to decide properly about this model selection. The upper bound is the best-case scenario, established from having seen the full set. The lower bound is worst-case model selection for each part. Now, we are interested in the performance of model fusion when selecting a model variant for each part, after seeing 1, 2, 4, ..., 64 random images. For each amount of images, the experiment is repeated 10 times, because it involves random draws of the images. The increasing performance is shown in Fig. 6. We observe that the average IoU starts way above the lower bound, indicating that just one image is already an indication of which model is most suitable for respective parts. After having seen a few images, e.g. 4 or 8, it is already possible to determine an effective selection of models to acquire better performance by fusion. With 32–64 images, the performance is close to the upper bound.

4.7 Label Efficiency

We hypothesize that the performance of Guided SAM largely depends on the accuracy of the guidance classifier. This classifier is trained with 1, 2, 4, ..., 64 images. We explore how many training images are needed for effective guidance. At various amounts of training images, we evaluate the model variants (Sect. 4.5)

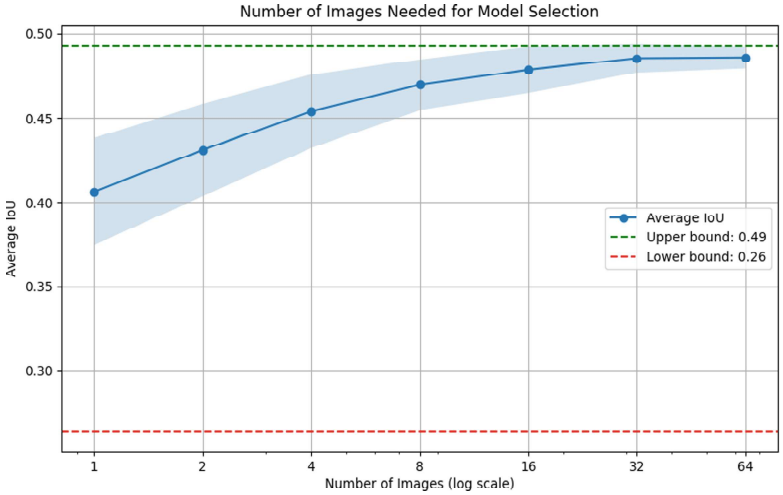


Fig. 6. After having seen a few images, e.g. 4 or 8, it is already possible to determine an effective selection of models to acquire better performance by fusion.

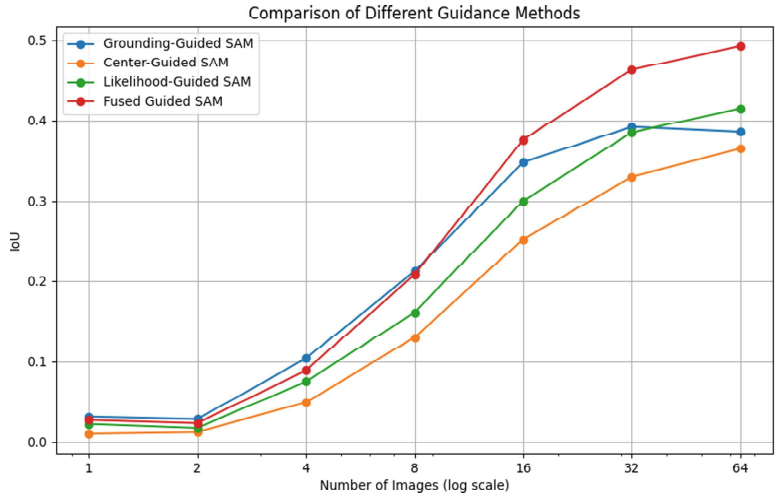


Fig. 7. Learning efficiency of our model variants and the fused model. With 64 images, a performance of IoU≈0.49 is achieved. With 32 or only 16 images, performance drops with respectively only 0.04 or 0.12.

and the fused model (Sect. 4.6), which combines the best performing model variants per part. Figure 7 shows the learning efficiency. The fused model is the best performer on average. Our guidance with Grounded SAM, i.e. Grounded Guided SAM (GGSAM), is the best performer at very low number of images. Probably this is because the confidences of that model are a useful source to filter out

wrong segmentations. With more than 8 training images, the fused model has a better performance, especially with 16, 32 or 64 images. With more training images, the guidance becomes better, hence all model variants become better. As a consequence, merit can be taken that the best variant is different across parts (Table 3). With 64 images, a performance of IoU≈0.49 is achieved. With 32 or only 16 images, object parts can be segmented reasonably well: performance drops with respectively only 0.04 or 0.12.

5 Conclusion

In this paper, we proposed a novel method for guiding segmentation models to accurately identify object parts. Our approach leverages regions of interest (ROIs) composed of patches predicted by a learnt classifier to identify specific parts of the object and indicate a positional prompt as starting point for part segmentation. It can be used as a guidance for advanced segmentation models such as (Grounded) SAM. We evaluated our method using the Car Parts dataset and demonstrated that it achieves good performance, even with a limited number of labeled patches. This approach significantly reduces the manual effort required for annotation, as it relies on labeling patches rather than creating full segmentation masks. The patch annotations must be centered around the object parts to ensure that the SAM positional prompts are correctly placed. Misalignment could lead the model to segment the background instead of the intended object parts. For future work, we plan to explore techniques to automatically refine patch placement to enhance segmentation accuracy further. Additionally, we aim to extend our method to other datasets and object categories to validate its generalizability and robustness across various domains.

References

1. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychol. Rev. **94**(2), 115 (1987)
2. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)
3. Gupta, A., Dollar, P., Girshick, R.: LVIS: a dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5356–5364 (2019)
4. Jain, A.K., Hoffman, R.: Evidence-based recognition of 3-D objects. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 783–802 (1988)
5. Jia, M., et al.: Fashionpedia: ontology, segmentation, and an attribute localization dataset. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 316–332. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_19
6. Kirillov, A., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015–4026 (2023)
7. Liu, S., et al.: Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)

8. Myers, A., Teo, C.L., Fermüller, C., Aloimonos, Y.: Affordance detection of tool parts from geometric features. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1374–1381. IEEE (2015)
9. Nauta, M., Schlötterer, J., van Keulen, M., Seifert, C.: PIP-Net: patch-based intuitive prototypes for interpretable image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2744–2753 (2023)
10. Oquab, M., et al.: DINOv2: learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
11. Palmer, S.E.: Vision Science: Photons to Phenomenology. MIT press (1999)
12. Pasupa, K., Kittiworapanya, P., Hongngern, N., Woraratpanya, K.: Evaluation of deep learning algorithms for semantic segmentation of car parts. Complex Intell. Syst. 1–13 (2021). https://doi.org/10.1007/s40747-021-00397-8
13. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
14. Ramanathan, V., et al.: PACO: parts and attributes of common objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7141–7151 (2023)
15. Reddy, N.D., Vo, M., Narasimhan, S.G.: CarFusion: combining point tracking and part detection for dynamic 3D reconstruction of vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1906–1915 (2018)
16. Ren, T., et al.: Grounded SAM: assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)
17. Sun, P., et al.: Going denser with open-vocabulary part segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15453–15465 (2023)
18. Wei, M., Yue, X., Zhang, W., Kong, S., Liu, X., Pang, J.: OV-PARTS: towards open-vocabulary part segmentation. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
19. Zhao, X., et al.: Fast segment anything (2023)