Cityscapes-Panoptic-Parts and PASCAL-Panoptic-Parts datasets for Scene Understanding

Citation for published version (APA):

Document status and date:
Published: 16/04/2020

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 23. Apr. 2020
Cityscapes-Panoptic-Parts
and PASCAL-Panoptic-Parts
datasets for Scene Understanding

Panagiotis Meletis*,†, Xiaoxiao Wen*,‡,
Chenyang Lu†, Daan de Geus†, Gijs Dubbelman†

† Eindhoven University of Technology
‡ University of Amsterdam

Abstract

In this technical report, we present two novel datasets for image scene understanding. Both datasets have annotations compatible with panoptic segmentation and additionally they have part-level labels for selected semantic classes. This report describes the format of the two datasets, the annotation protocols, the merging strategies, and presents the datasets statistics. The datasets labels together with code for processing and visualization will be published at github.com/tue-mps/panoptic_parts.

1 Introduction

Our goal is to provide two datasets with consistent, multi-level annotations covering a wide spectrum of image understanding tasks, both holistic (panoptic segmentation), as well as detailed ones (semantic segmentation, part-level segmentation, object detection).

To create the datasets we use existing published datasets, and we either augment them with our manual annotations or merged them, to create a multi-level hierarchy of labels (see Figure 1). The whole hierarchy is given on the same set of images, and is free of any conflicts or inconsistencies that would arise if the constituent datasets were used separately. This enables researchers to use our datasets in a wide range of image understanding problems, and create and test their algorithms on an arbitrary level of label granularity.

Each pixel in the datasets is labeled on three levels: semantic level, instance level, and parts level. On the semantic level we maintain the intuitive separation that panoptic segmentation introduces between stuff and things classes [5].

*Panagiotis Meletis (p.c.meletis@tue.nl) and Xiaoxiao Wen (xiaoxiao.wen@student.uva.nl) can be contacted for any suggestions or questions. Authors contributed equally.
Currently, we provide part-level annotations only for the things classes, however we note that, on a conceptual level, stuff classes may also have parts and we leave this for future research.

The first dataset, Cityscapes-Panoptic-Parts extends the established Cityscapes dataset for urban street scenes understanding, with our manual annotations for 23 part-level classes. The second dataset, PASCAL-Panoptic-Parts extends the recognized PASCAL dataset for generic, everyday life scenes by merging two datasets, PASCAL-Parts, which has part-level annotations only for the classes of PASCAL and PASCAL-Context with semantic annotations only. A comparison with related datasets can be seen in Table 1.

2 Cityscapes-Panoptic-Parts Dataset

Cityscapes dataset [2] is a large-scale dataset of real-world urban scenes recorded in Germany and neighboring countries. Its densely annotated portion consists of three per-pixel manually labeled splits, i.e., training, validation, and test, which contain 2975, 500, and 1525 images, respectively. The complexity of urban scenes and the variety in number and pose of traffic participants, together with the fact that it is thoroughly studied, make it a perfect candidate for use in image and object understanding research.

In this work, we introduce Cityscapes-Panoptic-Parts, a new dataset that extends Cityscapes with part-level annotations, in order to facilitate part-level object understanding in complex urban environments. Both Cityscapes and Cityscapes-Panoptic-Parts have labels compatible with panoptic segmentation [5]. In Figure 2, we provide examples of images and labels from Cityscapes-Panoptic-Parts.

Our main contributions with respect to the original Cityscapes dataset are:
Table 1: Dataset statistics for related parts segmentation datasets and our proposed dataset. The difference of the average number of instances with parts between Cityscapes and Cityscapes-Panoptic-Parts is due to small instances and parts that are not distinguishable. PASCAL-Context has 459 semantic classes but only 59 of them are included in the official split.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Inst. aware</th>
<th>Panop. aware</th>
<th>Stuff classes</th>
<th>Things classes</th>
<th>Parts classes</th>
<th># Images train / val</th>
<th>Average image size</th>
<th>Average # insts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL-Context [8]</td>
<td></td>
<td></td>
<td>59</td>
<td>-</td>
<td>-</td>
<td>4998/5105</td>
<td>387 x 470</td>
<td>-</td>
</tr>
<tr>
<td>LIP [7]</td>
<td></td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>30.5k/10k</td>
<td>325 x 240</td>
<td>1</td>
</tr>
<tr>
<td>CIHP [4]</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>1</td>
<td>20</td>
<td>28.3k/5k</td>
<td>484 x 578</td>
<td>3.4</td>
</tr>
<tr>
<td>MHP v2.0 [9]</td>
<td>✓</td>
<td></td>
<td>-</td>
<td>1</td>
<td>59</td>
<td>15.4k/5k</td>
<td>644 x 718</td>
<td>3</td>
</tr>
<tr>
<td>PASCAL-Person-Parts [1]</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>6</td>
<td>1716/1817</td>
<td>387 x 470</td>
<td>2.2</td>
</tr>
<tr>
<td>PASCAL-Parts [1]</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>20</td>
<td>193</td>
<td>4998/5105</td>
<td>387 x 470</td>
<td>4</td>
</tr>
<tr>
<td>Cityscapes [2]</td>
<td></td>
<td></td>
<td>-</td>
<td></td>
<td>69</td>
<td>2975/500</td>
<td>1024 x 2048</td>
<td>15.6</td>
</tr>
<tr>
<td>This work</td>
<td>✓</td>
<td>✓</td>
<td>100</td>
<td>20</td>
<td>193</td>
<td>4998/5105</td>
<td>387 x 470</td>
<td>4</td>
</tr>
<tr>
<td>Cityscapes-Panoptic-Parts</td>
<td>✓</td>
<td>✓</td>
<td>23</td>
<td>8</td>
<td>23</td>
<td>2975/500</td>
<td>1024 x 2048</td>
<td>15.6</td>
</tr>
</tbody>
</table>

1) a rich and detailed set of manual part-level annotations 2) a procedure for combining existing Cityscapes labels with our new part-level annotations, which does not create conflicts and minimizes the extra annotation time per image to an average of 7 minutes, described in Section 2.2, and 3) a hierarchical encoding of the part-level panoptic labels explained in Section 2.1.

### 2.1 Dataset format

We have manually annotated 5 semantic classes of Cityscapes with 9 parts classes in total. Specifically, the two human classes (person, rider) with 4 part-level classes, namely torso, head, arms, legs, and three vehicle classes (car, truck, bus) with 5 part-level classes, namely windows, wheels, lights, license plate, and chassis are annotated. Cityscapes-Panoptic-Parts is the first public dataset, to the best of our knowledge, with part-level panoptic labels.

We designed the labeling protocol and format with the following goals: 1) maximizing the compatibility with existing datasets, 2) providing a compact part-level panoptic label representation, which allows for future extensions, and 3) incorporating the hierarchical nature of part-level panoptic segmentation into the label format. These requirements led to a labeling scheme that is compatible with the Cityscapes panoptic format and extends the single-file ground truth format in an intuitive hierarchical manner.
2.2 Annotation procedure

We employed a team of 11 annotators and three experts, and we split the load such that every Cityscapes city is labeled by multiple annotators to ensure that human bias for parts class definitions and error are minimized per-city. The procedure for creating the part-level panoptic ground truth has the following steps:

1. An image is masked according to human and vehicle Cityscapes class labels, and these two masked images are provided to the annotators.

2. The annotators create polygons to label part regions. The polygons at the boundaries of objects and outside of the masked pixels do not have to be precise since those areas can be discarded in post-processing.
2.2 Dataset statistics

We have annotated 2975 images from the training split and 500 images from the validation split. Since the test split is not public, we cannot provide annotations. In addition to the statistics provided in the main paper, we also investigate scene complexity. We compute the frequency of images with a certain number of traffic participants, of the human or vehicle categories in the histogram of Figure 3. All vehicle instances of Cityscapes-Panoptic-Parts are annotated with at least one parts class, while for human instances, only a few are not labeled at the part-level.

Furthermore, for the five things classes that we annotated on the parts level,
we provide the absolute number of pixels per semantic class and per parts class in Figure 4. As can be seen, some part classes (bus lights, truck license plate) have very few pixels compared to others (vehicle chassis), and we anticipate interesting algorithms to be proposed to tackle this extreme imbalance.

3 PASCAL-Panoptic-Parts Dataset

The annual PASCAL Visual Object Classes (VOC) challenges have been a major benchmark for visual object recognition and detection tasks, and its corresponding datasets serve as an essential resource for relevant researches in the field. The PASCAL VOC 2010 challenge [3] has the main goal of recognizing several classes of visual objects in realistic scenes. It has 21738 images in total, which are split into three splits, i.e., training, validation, and test, containing 4998, 5105 and 11635 images, respectively.

We introduce PASCAL-Panoptic-Parts, a summarized compound dataset that extends PASCAL VOC 2010 with panoptic-level annotations and part-level annotations, based on the extensions PASCAL-Context [8] and PASCAL-Parts [1]. In PASCAL-Panoptic-Parts we include the training and validation splits (10103 images in total) since they are the only common subset between the three datasets. By combining the PASCAL-Context and PASCAL-Parts annotations, the resulting PASCAL-Panoptic-Parts dataset has both stuff and things annotations, which is compatible with panoptic segmentation [5]. In Figure 5, we provide examples of images and labels from PASCAL-Panoptic-Parts.

PASCAL-Panoptic-Parts is labeled on semantic level with 80 stuff classes (including one unlabeled/unknown class) and 20 things classes. The 20 things classes are also labeled on instance level, and parts level with 193 parts classes.
Figure 5: Examples of PASCAL-Panoptic-Parts images and labels from the training split. The benefits of our “best-effort” merging strategy are clear in the last two images, where the semantic-level labels (from PASCAL-Context), boat and car, provide information for the unlabeled pixels of PASCAL-Parts.
The total 100 semantic classes are a subset of the original PASCAL-Context [8] 459 semantic classes, which are the top 100 classes selected based on the product of the average occupied area per image and the number of images it appears. The measure of the average occupied area of a class indicates the visual exposure of it in terms of pixels. It encodes the semantic importance of the class as it takes up a significant portion in the image it occurs. Selection based on the average occupied area ensures that the class distribution includes classes that are both generally frequent and obvious in the images and classes that are rare but are significantly obvious in the images. In this case, the class distribution is long-tailed, and the relatively rare classes can be included for diversity. PASCAL-Context official set contains only 59 out of 459 classes, and we included all of them in our selection.

The 20 things classes are common between PASCAL-Context and PASCAL-Parts. As it is difficult to assess the completeness and correctness of those annotations without a complete ground truth annotation set, we decided to keep labels from both datasets and count on a vigorous merging strategy in order to maximize the maintained label information. The benefits are visible in the last two images of Figure 5. The extent of some instance-level labeled regions is smaller than the corresponding object, leading the rest of the object to be labeled with semantic-level labels. If we had opted for ignoring the semantic-level labels for the things classes of PASCAL-Context we would have lost valuable information for a large number of pixels. According to our hierarchical labeling scheme we provide as much information as possible for the three levels of the hierarchy and we leave to the user of the dataset to select which level of information is needed or which some pixels should be ignored.

3.1 Merging process

PASCAL-Panoptic-Parts is created by merging the PASCAL-Context dataset with PASCAL-Parts. PASCAL-Context only contains semantic-level annotation for the semantic classes and PASCAL-Parts only contains object-level annotations and/or part-level annotations for the 20 things class (boat, chair, table and sofa classes only have object-level annotations).

During merging, we have three types of conflicts: firstly, because the two datasets have semantic classes in common, there are overlapping labels, which we define as inter-dataset conflicts. Secondly, given that PASCAL-Parts is annotated object-wise and not image-wise, a pixel may have multiple instance ids, which we define as object-level intra-dataset conflicts. The third type of overlap arises from partonomy in the part-level annotations from PASCAL-Parts where some parts are part of others, i.e. nose is a part of head. In the part-level annotations from PASCAL-Parts, since individual masks are annotated for each part class, having multiple hierarchical labels per pixels is allowed. However, in our definition, also seen in Section 4, only one hierarchical label is allowed per pixel. These conflicts are defined as part-level intra-dataset conflicts.

We outline our merging strategy using an “imaginary canvas” p for the pixel in question on which, in each step, we fuse information from the two datasets for
each level of our label hierarchy. Furthermore, we denote the label from PASCAL-Context as PC and that from PASCAL-Parts as PP. We propose the following merging strategy to make the PASCAL-Panoptic-Parts annotation compatible with the hierarchical panoptic format and to resolve the aforementioned conflicts in each step:

1. At the semantic level, we first select 100 semantic classes (informative labels) out of 459 from PASCAL-Context based on the selection criterion and treat the other classes as unlabeled/unknown. Then for each pixel in an image, we have the following cases:

   (i) Both PC and PP are informative (not unknown/unlabeled), and the labels agree, then we fill $p$ with this label; if the labels do not agree, we use PP with higher priority. This is because PASCAL-Parts is essentially encoding more in-depth information (object-level and/or part-level) compared to PASCAL-Context, which is what we aim to prefer to include. By using the semantic label of PP, we ensure that in the following merging steps we can include object-level annotations and potentially part-level annotations as much as possible.

   (ii) PC is informative but PP is not, then we fill $p$ with PC;

   (iii) PP is informative but PC is not, then we fill $p$ with PP.

   which resolves the inter-dataset conflicts.

2. At the instance level, because PASCAL-Parts only provides object-level annotations, we first instantiate the semantic classes by grouping the object-level annotations by their semantic classes and yield the instance indices. Subsequently, we extend the semantic class labels to instance-level by appending the instance indices. With regarding this extended instance-level annotation as the flattened PASCAL-Parts annotation $PP_{flat}$, for each pixel in the image, we iterate through the object-level annotations from PASCAL-Parts by the provided object order, and if $p$ is annotated with PP in the previous step, we fill it with $PP_{flat}$. During this step, as we iterate through the order of objects provided in the PASCAL-Parts, if a pixel originally belongs to multiple objects, it will be assigned with the label of the last object following this order. This process then resolves the object-level intra-dataset conflicts.

3. At the part level, for each pixel in the image, if $p$ is annotated with $PP_{flat}$ from the previous step and its corresponding semantic class belongs to one of the things classes that contains part-level information, we first extend the annotation by appending the part-level unlabeled/unknown class labels. Then we annotate $p$ with specifying the priority of the part-level classes according to partonomy, i.e. we impose the part-level class nose over head, in order to resolve the part-level intra-dataset conflicts.
Figure 6: Selection criterion for choosing the 100 semantic classes to include in our dataset. The classes in bold denote things classes and have instances as well as part-level labels.

Figure 7: Number of pixels for PASCAL-Panoptic-Parts 100 semantic class labels for training and validation splits (10103 images). The classes in bold denote things classes and have instances as well as part-level labels.

### 3.2 Dataset Statistics

In Figure 6, the selection criterion is plotted against all the 99 semantic classes. The top 98 semantic classes (+1 unlabeled/unknown – not shown in Figure) are selected together with the mouse class that scores much lower (#166), but was included, so all classes from the official set of 59 classes are included. In Figure 7, the absolute number of pixels, is shown for all 100 semantic classes for the 10103 images of the training and validation splits. Figures 6 and 7 differ in the sorting order of classes due to the merging strategy explained in Section 3.1. Finally, in Figure 8 the 20 things classes that are labeled instance-wise and 16 of them part-wise are shown, together with a summarization of their 193 part-level classes to 73 for easier usage visualization. The classes of the reduced set can be found at github.com/tue-mps/panoptic_parts.
Figure 8: The 20 things classes (out of total 100 semantic classes) of PASCAL-Panoptic-Parts in the inner circle and their respective parts classes in the outer circle order by the number of pixels in the dataset. For some classes e.g., person, the parts classes are grouped for cleaner visualization. Colors do not to the colormap used for visualizations.

3.3 Future improvements

In addition, further exploration in the measure for selecting feasible semantic classes from PASCAL-Context can be done to include more semantically important classes, i.e., taking both the number of occurrences and the average occupied area into consideration. A refined version of the PASCAL-Panoptic-Parts dataset will also be investigated in future work.

4 Hierarchical label format

We decided to extend the Cityscapes dataset [2] label format due to its compactness and directness and include part-level labels in a hierarchical manner. Cityscapes dataset is labeled pixel-wise with an integer (base 10) id, which has up to 5 digits. Every pixel in an image has a semantic id (0-99), encoding either things or stuff semantic classes, e.g., car, person, building, traffic light. If a pixel belongs to a countable object (thing), it may also have an instance id (0-999), encoding different instances of the same semantic class in an image. The semantic classes are a fixed, predefined set for the whole dataset. The instance id is a counter per things semantic class and per image.

We decided to extend this format with a two-digit part id (0-99) denoting the semantic part-level class labels. This format enables to define up to 100 parts classes for every things semantic class. Moreover, the parts are bounded to a specific instance, which makes our format compatible with the recently introduced instance-wise object parsing task [6, 9, 4].
To summarize, each pixel in our hierarchical label format has:

- An up to 2-digit semantic id, encoding a things or stuff semantic class. If the pixel belongs to a things semantic class it can optionally have:
- An up to 3-digit instance id, a counter of instances per image.

Finally, if the pixel belongs to a things semantic class and is labeled instance-wise it can optionally have:

- An up to 2-digit part id, encoding the parts semantic class per-instance and per-image.

We compactly encode the aforementioned ids into an up to 7-digit id, for which the first two digits (starting from the left) encode the semantic class, the next 3 encode the instance (after zero pre-padding), and the final two encode the parts class (after zero pre-padding). We use the following formula, which produces ids that can be stored in a single image-like file:

\[
\text{id} = \begin{cases} 
(\text{semantic id}) & \text{semantic level} \\
(\text{semantic id}) \cdot 10^3 + (\text{instance id}) & \text{semantic, instance levels} \\
(\text{semantic id}) \cdot 10^5 + (\text{instance id}) \cdot 10^2 + (\text{part id}) & \text{semantic, instance, parts levels}
\end{cases}
\]

For example, for Cityscapes-Panoptic-Parts a sky (stuff) pixel will have \( id = 23 \), a car (things) pixel that is labeled only on the semantic level will have \( id = 26 \), if it’s labeled also on instance level it can have \( id = 26002 \), and a person (things) pixel that is labeled on all three levels (of Figure 1) can have \( id = 2401002 \).

We handle the unlabeled/void/“do not care pixels” in the three levels as follows:

- Semantic level: For Cityscapes-Panoptic-Parts we use the original Cityscapes void class. For PASCAL-Panoptic-Parts we use the class with \( id = 0 \) (first class).
- Instance level: For instances the void class is not needed. If a pixel does not belong to an object or cannot be labeled on instance level then it has only an up to 2-digit semantic id.
- Parts level: For both datasets we use the convention that, for each semantic class, the part-level class with \( id = 0 \) (first class) represents the void pixels e.g. for a person pixel, \( id = 2401000 \) represents the void parts pixels of instance 10. The need for a void class arises during the manual annotation process but in principle it is not needed at the parts level. Thus, we try to minimize void parts level pixels and assign them instead only the semantic and/or instance level labels.

5 Acknowledgments

We kindly thank Chenyang Lu, Xiaoxiao Wen, and Gijs Kok, Guus Engels, for providing a large number of initial annotations. Moreover, we thank the following students for annotating the rest of the dataset: Iko Vloothuis, Quinten
Aalders, Gijs Cunnen, Noud van de Gevel, Jelin Leslin, Daan Hommersom, and 4 more who wished to maintain anonymity.

6 More examples

In Figures 9 and 10, more examples of images and corresponding labels from the two datasets are shown.

References


Figure 9: More examples from Cityscapes-Panoptic-Parts.
Figure 10: More examples from PASCAL-Panoptic-Parts.
A Semantic classes and part-level classes names

In this Section we list all the things semantic classes that have part-level class labels in the two datasets. The format is the following:

Dataset
  ┌ semantic level class
  │ parts level class

A.1 Cityscapes-Panoptic-Parts

Cityscapes-Panoptic-Parts things classes with part-level classes
  ┌ person
  │   ┌ torso
  │   │ head
  │   │ arm
  │   │ leg
  │ rider
  │   ┌ torso
  │   │ head
  │   │ arm
  │   │ leg
  │ car
  │   ┌ window
  │   │ wheel
  │   │ light
  │   │ license plate
  │   │ chassis
  │ truck
  │   ┌ window
  │   │ wheel
  │   │ light
  │   │ license plate
  │   │ chassis
  │ bus
  │   ┌ window
  │   │ wheel
  │   │ light
  │   │ license plate
  │   │ chassis

A.2 PASCAL-Panoptic-Parts

PASCAL-Panoptic-Parts things classes
  ┌ aeroplane
  │   ┌ body
  │   │ stern
___living
___rwing
___tail
___engine
___wheel
bicycle
___fwheel
___bwheel
___saddle
___handlebar
___chainwheel
___headlight
bird
___head
___leye
___reye
___beak
___torso
___neck
___living
___rwing
___lleg
___lfoot
___rleg
___rfoot
___tail
boat
bottle
___cap
___body
bus
___frontside
___leftside
___rightside
___backside
___roofside
___leftmirror
___rightmirror
___fliplate
___bliplate
___door
___wheel
___headlight
___window
car
___frontside