Amodal Instance Segmentation with KINS Dataset

Lu Qi[1,2] Li Jiang[1,2] Shu Liu[2] Xiaoyong Shen[2] Jiaya Jia[1,2]
[1]The Chinese University of Hong Kong [2]YouTu Lab, Tencent
{luqi, lijiang}@cse.cuhk.edu.hk {shawnshuliu, dylanshen, jiayajia}@tencent.com

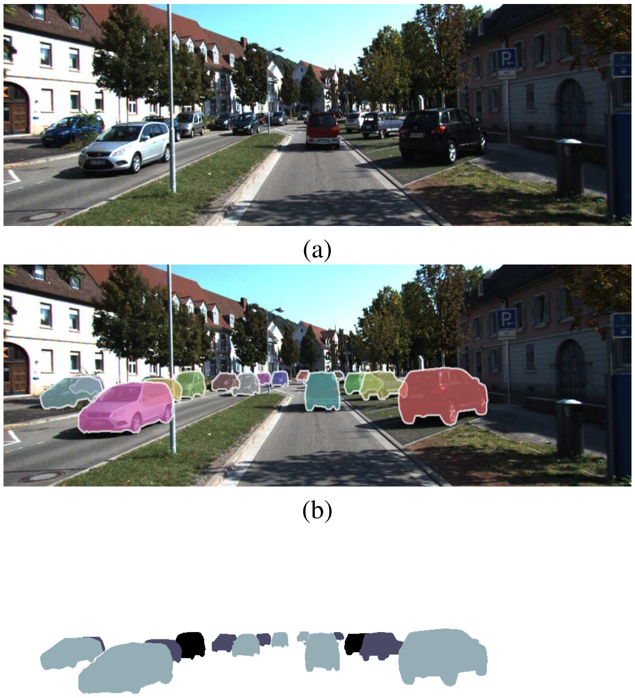
Abstract

Amodal instance segmentation, a new direction of instance segmentation, aims to segment each object instance involving its invisible, occluded parts to imitate human ability. This task requires to reason objects' complex structure. Despite important and futuristic, this task lacks data with large-scale and detailed annotation, due to the difficulty of correctly and consistently labeling invisible parts, which creates the huge barrier to explore the frontier of visual recognition. In this paper, we augment KITTI with more instance pixel-level annotation for 8 categories, which we call KITTI INStance dataset (KINS). We propose the network structure to reason invisible parts via a new multi-task framework with Multi-Level Coding (MLC), which combines information in various recognition levels. Extensive experiments show that our MLC effectively improves both amodal and inmodal segmentation. The KINS dataset and our proposed method are made publicly available.

1. Introduction

Human has the natural ability to perceive objects' complete physical structure even under partial occlusion [24, 21]. This ability, which is called *amodal* perception, allows us to gather integral information from not only visible clues but also imperceptible signals. Practically, *amodal perception* in computer vision offers great benefit in many scenarios. Typical examples include enabling autonomous cars to infer the whole shape of vehicles and pedestrians within the range of vision, even if part of them is invisible, largely reducing the risk of collision. It, thus, makes the moving decision in complex traffic or living environment easier. We note most current autonomous cars and robots still do not have this ability.

Challenges Although *amodal perception* is a common human ability, most recent visual recognition tasks, including object detection [16, 17, 36, 20, 8, 28], edge detection [1, 11, 38], semantic segmentation [32, 41, 39] and instance segmentation [19, 31], only focus on the visible parts of instances. There are only a limited number of amodal segmentation methods [26, 43, 14] due to the difficulty in both data preparation and network design.


(a)

(b)

(c)

Figure 1. Images in the KINS dataset are densely annotated with object segments and contain relative occlusion order. (a) A sample image, (b) the *amodal* pixel-level annotation of instances, and (c) relative occlusion order of instances. Darker color means farther instances in the cluster.

Despite the community's great achievement in image collection, existing large-scale datasets [10, 12, 6, 29, 23] for visual understanding are annotated without indicating occlusion regions, thus cannot be used for amodal perception. ImageNet [10] and OpenImages [23] are mainly used for classification and detection in image-or-box understanding. PASCAL VOC [12], COCO [29] and Cityscapes [6] pay more attention to pixel-level segmentation, which can be further classified into semantic segmentation and instance segmentation. These datasets have greatly promoted the development of visual recognition techniques. However, they only take into consideration the visible part of each instance. The key challenge for amodal instance segmentation data preparation is that annotation for occluded part has to follow ground truth, where the latter may not be available occasionally.

Our Contributions We put great effort to establish the new KITTI [15] INStance segmentation dataset (KINS). Using KITTI images, KINS has a lot of additional annotations, including complicated amodal instance segmentation masks and relative occlusion order following strict pixel-level instance tagging rules. Each image is labeled by three experienced annotators. The final annotation of each instance is determined by crowd-sourcing to deal with the ambiguity. The final labeled data for the unseen parts is guaranteed to be consistent among all annotators.

So far, KINS is the largest amodal instance segmentation dataset. Amodal instance segmentation is closely related to other tasks such as scene flow estimation, which means that KINS also profits other vision tasks to provide extra information.

With this new large-scale dataset, we propose the effective Multi-Level Coding (MLC) to enhance the amodal perception ability of conjecturing the integral pixel-level segmentation masks for existing instance segmentation methods [31, 19]. MLC consists of two parts of *extraction* and *combination*. The extraction part is mainly used to obtain the abstract global representation of instances; and the combination part integrates the abstract semantic information and per-pixel specific features to produce the final amodal (or inmodal on visible parts) masks. A new branch for discriminating the occluded regions is introduced to make the network more sensitive to capture the amodal notion. Extensive experiments on the large-scale dataset verify that our MLC improves both amodal and inmodal instance segmentation by a large margin over different baselines.

2. Related Work

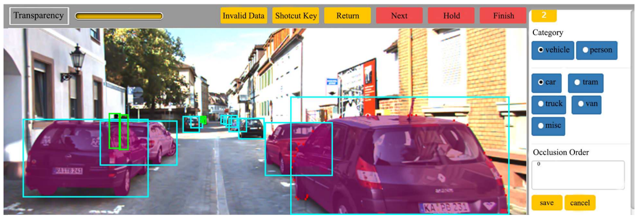
Object Recognition Datasets Most large-scale visual recognition datasets [10, 12, 6, 29, 23, 42] facilitate recognizing visible objects in images. ImageNet [10] and Open-Images [23] are used for classification and detection without considering objects' precise mask. Meanwhile, segmentation datasets are built to explore the semantic mask of each object in the pixel level. Pascal VOC [12], COCO [29] and ADE20K [42] collect a large number of images in common scenes. KITTI [15] and Cityscapes [6] are created for specific street scenarios. Although widely used in computer vision, these datasets do not contain labeling of invisible and occluded part of objects, thus cannot be used for amodal understanding.

Li and Malik [26] pioneered in building an amodal dataset. Since the training data directly uses instance segmentation annotation from Semantic Boundaries Dataset (SBD) [18], there are inevitable noise and outliers. In [43], Zhu *et al*. annotated part of the original COCO images [29] and provided COCO amodal dataset, which contains 5,000 images. Empirically, we found it is hard for a network to converge to an optimal point with this small-scale dataset due to a large variety of instances, which motivates us to establish the KITTI INStance Segmentation Dataset (KINS) with accurate annotation and image data in a much larger scale. We show in experiments that KINS is rather beneficial and general for a variety of advanced vision understanding tasks.

Amodal Instance Segmentation Traditional instance segmentation is only concerned with visible part of each instance. Popular frameworks are mainly proposal-based, which exploits state-of-the-art detection models (*e.g*., R-CNN [16], Fast R-CNN [17], Faster R-CNN [36], R-FCN [8], FPN [28], *etc*.) to either classify mask regions or refine the predicted boxes to obtain masks. MNC [7] is the first end-to-end instance segmentation network, cascading detection, segmentation and classification. FCIS [27] employs the position-sensitive inside/outside score maps to encode the foreground/background segmentation information. Mask R-CNN [19] adds a mask head to obtain refined mask results from box prediction generated by FPN and demonstrates outstanding performance. PANet [31] boosts information flow by bottom-up path augmentation, adaptive feature pooling and fully-connected fusion, which further improves Mask R-CNN. The other stream is mainly segmentation-based [2, 30, 22] with two-stage processing: segmentation and clustering. They learn specially designed transformation or instance boundaries. Instance masks are then decoded from predicted transformation.

Research on amodal instance segmentation begins to advance. Li and Malik [26] proposed the first method for amodal instance segmentation. They extended their instance segmentation approach [25] by iteratively enlarging the modal bounding box of an object and recomputing the mask. In order to evaluate on COCO amodal dataset, Zhu *et al*. [43] use AmodalMask as a baseline, which is the Sharp-Mask [34] trained on the amodal ground truth. Inspired by multi-task ROI-based networks [37], in [14], instances are segmented for both the amodal and inmodal setting. It adds an independent segmentation branch for amodal mask prediction on top of Mask R-CNN.

Several tasks encourage models to learn robust representation of input in a variety of applications, such as facial landmark detection [40], natural language processing [5], and steering prediction in autonomous driving [4]. Our design also extracts high-level semantic information to guide the segmentation branch to better infer the occluded parts.



Category
● vehicle ○ person
● car ○ tram
○ truck ○ van
○ misc
Occlusion Order
0

Figure 2. A screenshot of our annotation tool for amodal segmentation.

3. KINS: Amodal Instance Dataset

We annotate a total of $14,991$ images from KITTI to form a large-scale amodal instance dataset, namely KINS. The dataset is split into two parts where $7,474$ images are for training and the other $7,517$ are for testing. All images are densely annotated with instances by three skilled annotators. The annotation includes amodal instance masks, semantic labels and relative occlusion order, from which inmodal instance masks can be easily inferred. In this section, we describe our KINS dataset and analyze it with a variety of informative statistics.

3.1. Image Annotation

To obtain high-quality and consistent annotation, we strictly follow three instance tagging rules of (1) only objects in specific semantic categories are annotated; (2) the relative occlusion order among instances in an image is annotated; (3) each instance, including the occluded part, is annotated in the pixel level. These rules make the annotators to label instances in two steps. First, for each image, one expert annotator locates instances within specific categories in the box level and indicates their relative occlusion order. Afterwards, three annotators label the corresponding amodal masks for each image regarding these box-level instances. This process makes it easy for annotators to consider instance relationship and infer scene geometry. As shown in Figure 2, the annotation tool also well meets the tagging requirement. The detailed process is as follows.

(1) Semantic Labels Our instances are in specific categories. Semantic labels in our KINS dataset are organized into a 2-layer hierarchical structure, which defines an inclusion relationship between general- and sub-categories. Given that all images in KITTI are street scenes, a total of 8 representative categories are chosen for annotation in the second layer. General categories in KINS consist of 'people' and 'vehicle'. To keep consistency with KITTI detection dataset, the general category 'people' is further subdivided into 'pedestrian', 'cyclist', and 'person-siting', while the general category 'vehicle' is split into 5 sub-categories of 'car', 'tram', 'truck', 'van', and 'misc'. Here 'misc' refers to ambiguous vehicles that even experienced annotators cannot specify the category.

(2) Occlusion Ordering For each image, an expert annotator is asked to annotate instances with bounding boxes and order them in relative occlusion. For the order among objects, instances in an image are first partitioned into several disconnected clusters, each with a few connected instances for easy occlusion detection. Relative occlusion order is based on the distance of each instance to the camera. In addition, as shown in Figure 1(c), instances in one cluster are annotated in an order for near to distant objects where orders of non-overlapping instances are labeled as 0. As for the occluded instances in a cluster, order starts from 1 and increases by 1 when occluded once. For occasional complex occlusion situation (e.g. Figure 3), we impose another important criterion that instances with the same relative occlusion order should not occlude each other.

(3) Dense Annotation Three annotators then label each instance densely in its corresponding bounding box. A special focus in this step is to figure out occluded invisible parts by three annotators independently. Given slightly different predictions for the occluded pixels, our final annotation is decided by majority voting on the instance mask. For the parts that do not reach consensus, e.g., location of invisible car wheels as shown in Figure 3, more annotation iterations are involved until high confidence is reached for the wheel position. An inmodal mask is also drawn if the instance is occluded.

3.2. Dataset Statistics

In our KINS dataset, images are annotated following aforementioned strict criteria. On average, each image has 12.53 labeled instances, and each object polygon consists of 33.70 points. About 8.3% of image pixels are covered by at least one object polygon. Of all regions, 53.6% are partially occluded and the average occlusion ratio is 31.7%.

Annotating an entire image takes about 8 minutes where each single instance needs 0.38 minute on average. 30% of the time is spent on box-level localization and occlusion ordering; the rest is on pixel-level annotation. The time cost varies according to image and object structure complexity. We analyze the detailed properties in several major aspects.

Semantic Labels Table 1 shows the distribution of instance categories. 'vehicle' contains mostly cars, while 'tram' and 'truck' both contribute only 1% of the instances. The occurrence frequency of 'people' is relatively low, taking 14.43% of all instances. Among them, 10.56% are 'pedestrian' and 2.69% are 'cyclist'. Overall, the distribution follows Zipf's law, same as the Cityscapes dataset [6].

Shape Complexity Intuitively, independent of scene geometry and occlusion patterns, amodal segments should have relatively simpler shape than inmodel segments [43] that are possibly occluded in any way. We calculate shape

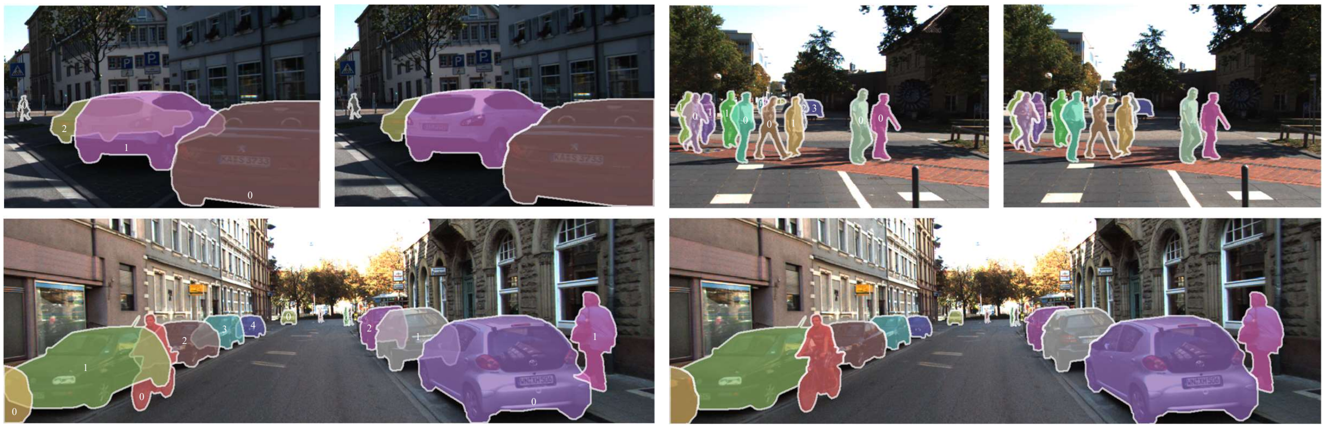


Figure 3. Examples of our amodal/inmodal masks. The digit in each amodal mask represents its relative occlusion order. Inmodal masks are obtained with the amodal mask and relative occlusion order.

category	people			vehicle				
subcategory	pedestrian	cyclist	person-siting	car	van	tram	truck	misc
number	20134	5120	2250	129164	11306	2074	1756	18822
ratio	10.56%	2.69%	1.18%	67.76%	5.93%	1.09%	0.92%	9.87%

Table 1. Class distribution of KINS.

	simplicity		convexity	
	inmodal	amodal	inmodal	amodal
BSDS-A [43]	.718	.834	.616	.643
COCO-A [43]	.746	.856	.658	.685
KINS	.709	.830	.610	.639

Table 2. Comparison of shape statistics among amodal and inmodal segments on BSDS, COCO and KINS.

convexity and simplicity following [43] as

$$\begin{aligned} \textit{convexity(S)} &= \frac{\textit{Area(S)}}{\textit{Area(ConvexHull(S))}} \\ \textit{simplicity(S)} &= \frac{\sqrt{4\pi * Area(S)}}{\textit{Perimeter(S)}}. \end{aligned} \tag{1}$$

Both metrics achieve the maximum value 1.0 when the shape is a circle. Thus, simple segments statistically should yield a large convexity-simplicity average value. Table 2 shows the comparison of shape simplicity and convexity among three amodal datasets of KINS, BSDS and COCO. The values of our KINS dataset are slightly smaller than those of BSDS and COCO because KINS contains more complex instances such as 'cyclist' and 'person-siting'. We also show the comparison between inmodal and amodal annotations of KINS. The amodal data yields stronger convexity and simplicity, verifying that the amodal masks are usually with more compact shapes.

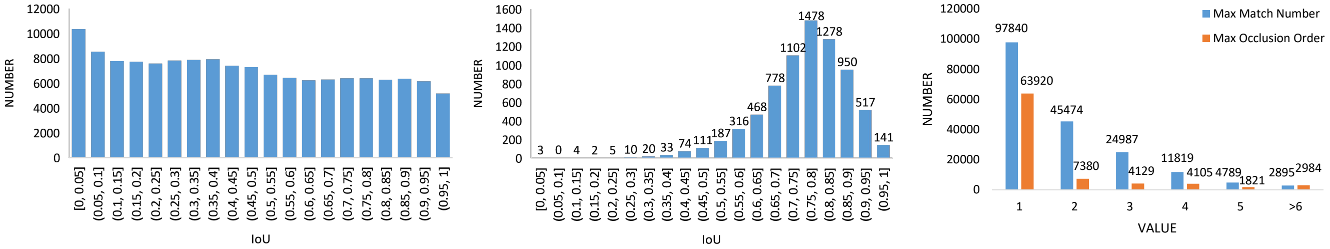
Amodal Occlusion Occlusion level is defined as the fraction of area that is occluded. Figure 4(a) illustrates that the occlusion level is nearly uniformly distributed in KINS. Compared with COCO Amodal dataset, heavy occlusion is more common in KINS. Occluded examples at different occlusion levels are displayed in Figure 3. It is challenging to reason out the exact shape of the car (Figure 3(a)) when the occlusion level is high. This is why amodal segmentation task is difficult.

Max Occlusion Order The *relative occlusion order* is valid only for instances in the same cluster. We accordingly define the *max occlusion order* of a cluster as the number of occluded instances in it. Besides, the *max match number* is the number of overlapping instances for each object. The distributions of the order and number are drawn in Figure 4(c). Most clusters only contain a small amount of instances. Clusters with the max occlusion order larger than 6 are only 1.54% in the whole dataset.

Segmentation Transformation With our amodal instance segments provided in KINS, the inmodal masks can be easily obtained given the amodal masks and occlusion orders. As shown in Figure 3, for two overlapping instances, the intersection regions should belong to the instance with the smaller occlusion order for inmodal annotation.

3.3. Dataset Consistency

Annotation consistency is a key property of any human-labeled datasets since it determines whether the annotation task is well defined or not. It is worth mentioning that inferring the occluded part is subjective and open-ended. However, due to our strict tagging criteria and human prior knowledge of instances, the amodal annotations in KINS are rather consistent. We evaluate it based on bounding box

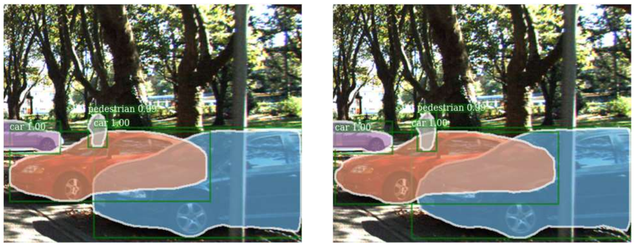


(a) Occlusion Level

(b) Box Consistency

(c) Cluster Statistics

Figure 4. Three metrics to further evaluate our dataset.



(a) 20000 Iteration

(b) 24000 Iteration

Figure 5. Visualization of Mask R-CNN prediction for different iterations. With more training iterations, masks of the orange car and person shrink.

consistency and mask consistency. Considering that Intersection over Union (IoU) can measure the matching degree of instance masks and bounding boxes from different annotators, we calculate average IoU for all annotations.

First we measure the bounding box consistency by comparing the bounding boxes in KINS with those in original KITTI detection dataset. Difference is found: bounding boxes in KITTI detection dataset are annotated without considering occluded pixels. Hence in KINS, the boxes are generally larger. To fairly evaluate the consistency, we generate our own inmodal boxes by tightening the corresponding inmodal masks. For each image, there are 12.74 objects in KINS on average compared with 6.93 of them in KITTI Detection dataset. The histogram in Figure 4(b) shows that most annotations are consistent with the original detection bounding boxes. Over 78.34% of the images have average IoU larger than 0.65.

Second, for measuring mask consistency, we randomly select 1,000 images from KINS (about 6.7% of the entire dataset), and ask the three annotators to process them again. There is a 4-month gap between the two annotation stages. We denote the annotator i $(i = 1, 2, 3, mv)$ in stage j $(j = 1, 2)$ as a_{ij}. The consistency scores between every two annotators are shown in Table 3. Here, a_{mvj} represents the annotation result after majority voting of the three annotators in stage j $(j = 1, 2)$.

Although the two annotation periods have a few months in between, annotators still tend to make similar prediction about unseen parts. Thus the average IoUs of all images in the diagonal of Table 3 are relatively high. We get the highest score when matching the final comprehensive results a_{mv1} and a_{mv2}, which manifests that integrating annotation from the three annotators into a final output by majority voting further improves data consistency.

	ann_{11}	ann_{21}	ann_{31}	ann_{mv1}
ann_{12}	0.836	0.802	0.805	0.834
ann_{22}	0.809	0.840	0.818	0.836
ann_{32}	0.804	0.816	0.835	0.833
ann_{mv2}	0.838	0.836	0.837	0.843

Table 3. Consistency scores for three annotators in two stages.

4. Amodal Segmentation Network

Since amodal segmentation is a general and advanced version of instance segmentation, we first evaluate state-of-the-art Mask R-CNN and PANet on amodal segmentation. Though designed for inmodal segmentation, these frameworks are still applicable here by simply using amodal masks and boxes. They can produce reasonable results. But the problem is that increasing training iterations makes the network suffer from severe overfitting, as shown in Figure 5. With more iterations, occlusion regions shrink or disappear while the prediction of visible parts becomes stable.

Analysis of Amodel Properties To propose a suitable framework for amodal segmentation, we first analyze the above overfitting issue by discussing important properties of CNNs. (1) Convolution operations, which are widely used in mask prediction, help capture accurate local features while losing a level of global information for the whole instance region. (2) Fully Connected (FC) operations enable the network to have comprehensive understanding instances by integrating information in space and channels. In existing instance segmentation frameworks, the mask head usually consists of four convolution layers and one deconvolution layer, making good use of local information. However, without global guidance or prior knowledge of the instance, it is difficult for the mask head to predict *invisible part* caused by occlusion with only local information.

Importance of global information for inmodal mask prediction was also mentioned in [31] especially for disconnected instances. Empirically, we also observe that *strong perception ability by global information is key for the net-*

work to recognize the occluded area.

We utilize more global information to infer occluded parts. We first explain the global features in Mask R-CNN. Besides the region proposal network (RPN), there are three branches in instance segmentation frameworks, including box classification, box regression and mask segmentation. The first two branches, sharing the same weight except two independent FC layers, are respectively used to predict what and where the instance is. They give attention to overall perception where the features can be taken to help integral instance inference.

Occlusion Classification Branch We note only the global box features are not enough for amodal segmentation because several instances may exist in one region of interest (RoI). Features of other instances may cause ambiguity in mask prediction. We accordingly introduce the *occlusion classification branch* to judge the possible existence of occlusion regions. The high classification accuracy in Table 6 shows that the occlusion features in this branch provide essential invisible information and make mask prediction balance the influence of several instances.

Amodal Segmentation Network (ASN) Based on above consideration, Amodal Segmentation Network (ASN) is proposed to predict complete shape of instances by combining box and occlusion features. As shown in Figure 6, our framework is also a multi-task network with box, occlusion, and mask branches.

Box branches, including classification and regression, share the same weight except the independent head. The occlusion classification branch is used to determine whether occlusion exists in a RoI or not. The mask branch aims to segment each instance. The input to all branches is the RoI features; each branch consists of 4 cascaded convolutions and ReLu operations. To predict the occlusion part, Multi-Level Coding (MLC) is proposed to let the mask branch segment complete instances by visible clues and inherent perception of the integral region at the same time.

Moreover, to prove that our MLC is not restricted to amodal segmentation, mask prediction of our network consists of independent amodal and inmodal branches. For each mask branch, the corresponding ground-truth is used respectively. In the following, we explain the two most important and effective components in our framework, i.e., the occlusion classification branch and Multi-Level Coding.

4.1. Occlusion Classification Branch

In general, 512 proposals are sampled from the result of RPN with 128 foreground RoIs. According to our statistics, at most 40 RoIs, in general, have overlapping parts even considering background samples. What makes things more challenging is that several occlusion regions only contain 1 to 10 pixels. The extreme imbalance between occlusion and non-occlusion samples imposes extra difficulty for previous networks to work here [35].

Based on common knowledge that features of extremely small regions are weakened or even missed after RoI feature extraction [28, 9], we only regard regions with overlapping area larger than 5% of the total mask as occlusion samples. To relieve the imbalance between occlusion and non-occlusion samples, we set the weight loss of positive RoI to 8. Besides, this branch leads to the backbone of our network to extract robust image features under occlusion.

4.2. Multi-Level Coding

Our network now contains occlusion information. To further enhance the ability of predicting amodal or inmodal masks given the currently long distance between the backbone and mask head, Multi-Level Coding (MLC) is proposed to amplify the global information in mask prediction.

Albeit the same structure as box and occlusion branches, the mask branch has its unique characteristics. First, this branch only aims to segment positive RoIs. Therefore, in the box/occlusion classification branch, only features of positive samples are extracted and fed to MLC as global guidance. Besides, sizes of feature maps for the box/occlusion classification branch and mask branch are respectively 7×7 and 14×14. To utilize these features and extract more information, our MLC has two modules of extraction and combination. The extraction part incorporates category and occlusion information into a global feature. Then the combination part fuses the global feature and local mask feature to help segment complete instances. More details are given below. By default, the kernel size of the convolution layer is $C \times C \times 3 \times 3$, with stride and padding size 1. C denotes the number of channels.

Extraction In this module, the box and occlusion classification features are first concatenated and then up-sampled by a deconvolution layer with a $2C \times C \times 3 \times 3$ kernel. Next, to integrate information in two features, the upsampled features are fed into two sequential convolution layers followed by the ReLU operation.

Combination To combine the global and specific local clues in the mask branch, the features from extraction part are first concatenated with mask feature. They are then fed into three cascaded convolution layers followed by a ReLU operation. The last convolution layer reduces the feature channels by half, making the output dimension the same as that of features in mask branches. At last, the output feature is sent to the mask branch for final semantic segmentation.

4.3. Multi-Task Learning

Our network treats all branches of RPN, box recognition, occlusion classification and mask prediction similarly important with weight for each loss set to 1. It works decently

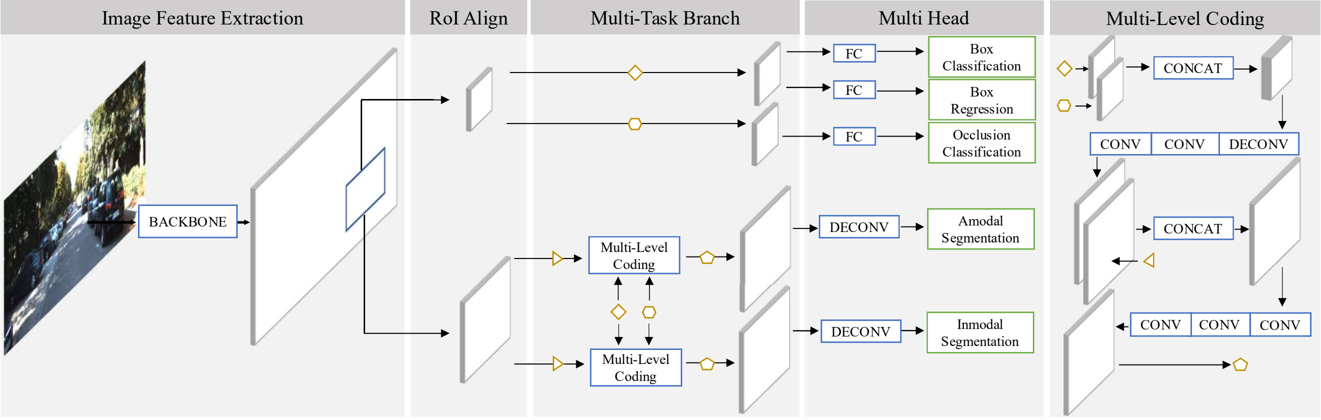


Figure 6. Apart from similar structure of Mask R-CNN, amodal segmentation network consists of an occlusion classification branch and Multi-Level Coding. Multi-Level Coding is used for combining multi-branch features to guide mask prediction by two modules including extraction and combination. Yellow symbols represent features in corresponding branches.

for amodel segmentation in our experiments. The final loss is expressed as

$$L = L_{cls} + L_{box} + L_{occlusion} + L_{mask}, \quad \text{where} \\ L_{mask} = L_{mask_a} + L_{mask_i}. \tag{2}$$

For inference, there is a small modification. We calculate the regressed boxes according to the output of the box branch and proposal location. Then the updated boxes are fed into the box branch again to extract class and occlusion features. Afterwards, we only select remaining boxes after NMS [13] for final mask prediction.

5. Experiments

All experiments are performed on our new dataset with 7 object categories. 'person-siting' is excluded due to the large number of annotation of crowds. Since the ground-truth annotation of the testing set is available, 7,474 images are used for training; evaluation are conducted on the 7,518 test images. We integrate our occlusion classification branch and Multi-Level Coding on two baseline networks.

We train these network modules using the Pytorch library on 8 NVIDIA P40 GPUs with batch size 8 for $24,000$ iterations. Stochastic gradient descent with 0.02 learning rate and 0.9 momentum is used as the optimizer. We decay learning rate with 0.1 at $20,000$ and $22,000$ iterations respectively. Results are reported in terms of mAP, which is commonly used for detection and instance segmentation. We use amodal bounding boxes as our ground truth in the box branch in case of missing occlusion parts. We conducted the same experiments five times and report the average results. The variance is 0.3.

5.1. Instance Segmentation

Table 4 shows that our amodal segmentation network produces decent mAPs for both amodal and inmodal instance segmentation, since the mask branch in our framework can determine if the feature of invisible parts should be enhanced or weakened. For amodal mask prediction, MLC prefers to enlarge the mask area of invisible part by global perception and prior knowledge about category and occlusion prediction. Besides, connection of box, occlusion and mask branches makes the feature in each branch robust when serving different tasks, compared with independently working in previous networks. PANet with 44,056,576 parameters still performs worse than Mask R-CNN + ASN with 13,402,240 parameters, indicating that the performance gain is not only related to the number of parameters. Note that the structure of ORCNN is similar to Mask R-CNN with two independent mask heads, except for a unique branch for predicting invisible parts.

Model	Det	Amodal Seg	Inmodal Seg
MNC [7]	20.9	18.5	16.1
FCIS [27]	25.6	23.5	20.8
ORCNN [14]	30.9	29.0	26.4
Mask R-CNN [19]	31.3	29.3	26.6
Mask R-CNN + ASN	32.7	31.1	28.7
PANet [31]	32.3	30.4	27.6
PANet + ASN	**33.4**	**32.2**	**29.7**

Table 4. Comparison of our approach and other alternatives. All super-parameters in both methods are the same.

5.2. Ablation Study

Ablation study on specific modules and their features fusion locations is performed, as shown in Table 5. Inmodal and amodal mask prediction along with Mask R-CNN performs slightly better than each single mask prediction because more features in different aspects are learned.

Performance further improves by adding the occlusion classification branch, which indicates that exploiting im-

Model	Det	Amodal Seg	Inmodal Seg
Mask R-CNN [19]	31.0	×	26.4
Mask R-CNN [19]	31.1	29.2	×
Mask R-CNN [19]	31.3	29.3	26.6
Mask R-CNN + OC	31.9	30.0	27.9
Mask R-CNN + OC + MLC(0,0)	32.5	31.0	28.6
Mask R-CNN + OC + MLC(1,1)	**32.7**	**31.1**	**28.7**
Mask R-CNN + OC + MLC(2,2)	32.3	30.6	28.2
Mask R-CNN + OC + MLC(3,3)	31.7	29.8	28.0
Mask R-CNN + OC + MLC(0,3)	31.9	29.8	27.9
Mask R-CNN + OC + MLC(3,0)	31.8	29.7	27.8

Table 5. The first three rows list Mask R-CNN performance for inmodal segmentation, amodal segmentation and tackling both simultaneously. The fourth row is for the model that adds occlusion classification branch into Mask R-CNN. The remaining rows show results with different fusion locations of modules' features. MLC(a, b) means the combination between features after the a^{th} convolution layer of box/occlusion classification branch and that after the b^{th} convolution layer of the mask branch.

Model	Det	Amodal Seg	Inmodal Seg	Occlusion Accuracy
MR [19]	31.3	29.3	26.6	0.866
MR + OC(0%)	31.7	29.7	27.5	0.871
MR + OC(5%)	**31.9**	**30.0**	**27.9**	**0.872**
MR + OC(15%)	31.4	29.6	27.3	0.869
MR + OC(20%)	31.2	29.4	26.7	0.866

Table 6. The ablation study for overlapping threshold in the occlusion classification branch. MR refers to Mask R-CNN.

Model	Modification	Det	Amodal Seg	Inmodal Seg
0111	ADD	32.1	30.5	27.9
1011	Order Adjustment	32.4	30.6	28.0
1011	1 CONV	31.9	30.3	27.6
1011	3 CONV	**32.7**	31.0	28.6
1101	ADD	32.3	30.6	28.1
1110	Order Adjustment	32.6	31.0	28.5
1110	1 CONV	32.0	30.1	27.5
1110	3 CONV	32.6	**31.1**	28.6
1111	×	**32.7**	**31.1**	**28.7**

Table 7. Ablation study on differnet operations for each part of Multi-Level Coding. In column 'Modification', 'ADD' means adding features of two branches, 'Order Adjustment' means reversing the 'special' convolution, such as deconvolution with stride 2, and other two cascaded convolutions. '{x} CONV' means using x cascaded convolutions. '1011' in column 'model' means that we use default operations in column 'Modification' except for the second part.

age features with occlusion information to guide our mask prediction are effective. The performance of feature fusion location in Multi-Level Coding manifests that the box and occlusion features in former layers are helpful to determine whether the occlusion parts should be reasoned or not. The best fusion location for different types of features is after the first convolution layer of each branch. These features maintain not only the global information but also unique properties for each specific-task branch.

Table 6 shows that overlapping threshold in occlusion classification branch is important to help get robust global image features for the backbone network. Threshold 5% is used for the best effect. An overly small threshold may cause ambiguity to discriminate among RoIs with small occlusion part, which are usually on the border. Contrarily, it is also challenging for the network to capture sufficient amodal cases when using a too large threshold.

Further exploration for Multi-Level Coding is shown in Table 7. MLC consists of four parts altogether. Each module consists of concatenation or cascaded convolution, as shown in Figure 6. The design of MLC yields very good performance in both detection and segmentation. It achieves effective feature fusing only using these a few simple operations. Due to the page limit, we have to put visualization of our segmentation results into our supplementary material.

5.3. Further Applications

Model	D1	D2	F1	SF
OSF [33]	4.74	6.99	8.55	9.94
ISF [3]	3.61	4.84	6.50	7.46
ISF with KINS	**3.56**	**4.75**	**6.39**	**7.35**

Table 8. Disparity (D1,D2), flow (Fl) and scene flow (SF) errors for background and foreground are averaged over KITTI 2015 validation set.

Table 8 lists refined flow prediction assisted by instance segmentation with the KINS dataset. For simplicity's sake, we train our flow model following [3]. The improved performance on the validation set of KITTI 2015 scene flow dataset indicates that KINS can be broadly beneficial in other vision tasks to provide extra information.

6. Conclusion

We have built a large dataset and presented a new multi-task framework for amodal instance segmentation. KITTI INStance dataset (KINS) are densely annotated with the amodal mask and relative occlusion order for each specific instance. Belonging to the augmented KITTI family, KINS has great potential to benefit other tasks in autonomous driving. Besides, a generic network design was proposed to improve reasoning ability for invisible part with independent occlusion classification branch and Multi-Level Coding. More solutions for feature enhancement and models, such as GAN, will be researched in future work.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011. 1

[2] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017. 2

[3] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *ICCV*, 2017. 8

[4] Rich Caruana. Multitask learning. *Machine learning*, 1997. 2

[5] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2, 3

[7] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 2, 7

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, 2016. 1, 2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. 2017. 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[11] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. *PAMI*, 2015. 1

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 2

[13] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 7

[14] Patrick Follmann, Rebecca König, Philipp Härtinger, and Michael Klostermann. Learning to see the invisible: End-to-end trainable amodal instance segmentation. *WACV*, 2019. 1, 2, 7

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[17] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015. 1, 2

[18] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 2

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 7, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 1

[21] Philip J Kellman and Christine M Massey. Perceptual learning, cognition, and expertise. In *Psychology of learning and motivation*. 2013. 1

[22] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *CVPR*, 2017. 2

[23] Ivan Krasin, Tom Duerig, Neil Alldrin, Andreas Veit, Sami Abu-El-Haija, Serge Belongie, David Cai, Zheyun Feng, Vittorio Ferrari, Victor Gomes, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016. 1, 2

[24] Steven Lehar. Gestalt isomorphism and the quantification of spatial perception. *Gestalt theory*, 1999. 1

[25] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016. 2

[26] Ke Li and Jitendra Malik. Amodal instance segmentation. In *ECCV*, 2016. 1, 2

[27] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 2, 7

[28] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 2, 6

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2

[30] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In *ICCV*, 2017. 2

[31] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2, 5, 7

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[33] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 8

[34] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollr. Learning to refine object segments. In *ECCV*, 2016. 2

[35] Lu Qi, Shu Liu, Jianping Shi, and Jiaya Jia. Sequential context encoding for duplicate removal. In *NeuralPS*, 2018. 6

[36] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2

[37] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv:1706.05098*, 2017. 2

[38] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, 2015. 1

[39] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. *CVPR*, 2018. 1

[40] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 2

[41] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2

[43] Yan Zhu, Yuandong Tian, Dimitris N Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *CVPR*, 2017. 1, 2, 3, 4