Dense Contrastive Loss for Instance Segmentation

Hang Chen[1]
chenhang20@mails.tsinghua.edu.cn
Chufeng Tang[1]
tcf18@mails.tsinghua.edu.cn
Xiaolin Hu[†1,2]
xlhu@mail.tsinghua.edu.cn

[1] Depart. of Comp. Sci. & Tech., State Key Lab of Intell. Tech. & Sys., THU-Bosch JCML Center, BNRist, Institute for AI, THBI, IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China

[2] Chinese Institute for Brain Research (CIBR), Beijing, China

Abstract

Instance segmentation, which requires instance-level mask prediction, is a fundamental task in computer vision. Many methods have been proposed in this field. However, the existing methods still do not perform well in complex scenarios such as occlusion. In this work, we analyzed the segmentation errors of some typical instance segmentation models. We found that false negatives (*i.e*. misclassification of foreground pixels as background) accounted for the majority of errors. It can be attributed to the inconsistent features of the same instance under complex scenarios. To address this problem, we proposed a dense contrastive loss to encourage the segmentation network to learn more consistent feature representations. Specifically, features on the same instance are pulled closer, while features on different instances and features between instances and the background are pushed farther apart. Without introducing any extra inference cost, the proposed method mitigated false-negative errors and achieved significant improvements on the Cityscapes and MS-COCO datasets. Code will be available at https://github.com/tinyalpha/DCL.

1 Introduction

Instance segmentation, a fundamental task in computer vision, has received extensive attention from the community [5, 8, 11, 12, 14, 20, 28, 32]. The purpose of instance segmentation is to predict an instance-level mask for each object in the image, separately. To achieve this, multi-stage methods (such as Mask R-CNN and its variants [5, 8, 14, 20]) follow a detect-then-segment manner. These algorithms first use a detector to localize the object, crop its corresponding features and apply a segmentation head to obtain the final mask. Recently, one-stage methods, which directly segment the objects, have gradually become a new trend [28, 32, 33, 35].

Despite some improvements in these methods, their performance (especially in complex scenarios) is still unsatisfactory. Instance segmentation is a complex task and a method can

[†]Corresponding Author.

make different types of errors. It is unclear which type of error is the dominant one. A systematic analysis is desired before proposing better methods. In this work, we classify the instance segmentation errors into detection and segmentation errors, providing a fine-grained analysis of the segmentation errors. As shown in Figure 1, segmentation errors are decoupled into three types and are analyzed separately.

Through error analysis of existing instance segmentation models, we found that false-negative error (*i.e*. misclassification of foreground as background) accounted for the majority of the errors. As shown in Figure 2, the amount of false-negative (FN) errors are much more than the foreground false-positive (fFP) and background false-positive (bFP) errors (corresponding to over-segmented pixels belonging to the foreground and background, respectively; see Section 3 for details). For example, on the Cityscapes dataset [13], FN errors caused a severe performance drop (18.8% AP) for Mask R-CNN [14], while fFP and bFP errors caused small drops (about 5% AP). This can be attributed to the fact that in complex scenes (occlusion, bad illumination, etc.), segmentation networks cannot predict consistent feature representations for different pixels on the same instance, which causes that some foreground pixels are mis-classified as background.

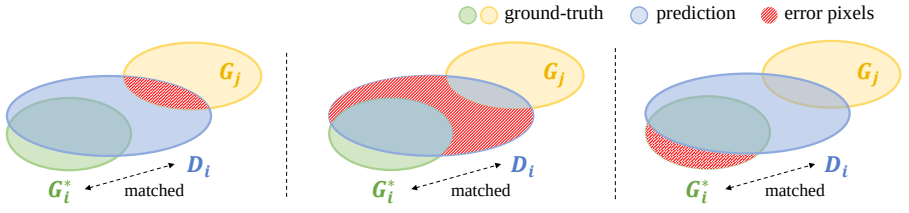
To alleviate this problem, we propose a dense contrastive loss to encourage the model to learn compact instance feature representations. Specifically, we apply the proposed loss function to guide the feature learning of the segmentation head. Features on the same instance are pulled closer, while features on different instances and features between instances and the background are pushed farther apart. As a result, the model is guided to learn a consistent feature representation for different pixels on the same instance. Without introducing any inference overhead, the proposed method mitigated false-negative errors and significantly improved the performance of various baselines.

Our contributions are summarized as follows: (1) We for the first time provide a detailed analysis of the segmentation errors for the task of instance segmentation and reveal that false negatives accounted for the majority of the errors. (2) To mitigate the false-negative errors, we proposed a dense contrastive loss function. (3) Without any inference overhead, our method effectively mitigated the false-negative errors and significantly improved the segmentation results. Extensive experiments on the Cityscapes [13] and MS-COCO [24] datasets validated the effectiveness of our approach.

2 Related Work

Instance Segmentation. Existing instance segmentation algorithms can be divided into multi-stage and one-stage ones. The multi-stage methods follow a detect-then-segment manner. Mask R-CNN [14] utilizes RoIAlign to extract the features corresponding to the detection box and apply a mask head for instance-level segmentation. Cascade R-CNN [5], HTC [8], and RefineMask [38] further improve the performance by cascading and refining. Instead, one-stage methods directly predict instance-level masks. CondInst [28] and SOLOv2 [33] directly predict a set of instance-level convolutional kernels for segmentation. PolarMask [35] predicts the polar coordinate representation of the instance mask. Recently, transformer-based models [6, 10, 19] have also achieved competitive results. Our method applies to arbitrary frameworks. We mainly demonstrate the effect with multi-stage models as an example.

Contrastive Learning. Contrastive learning is a flexible technique in that different objec-



(a) fFP (b) bFP (c) FN

Figure 1: Illustration of the three types of errors defined (red shaded areas). D_i denotes a predicted mask, G_i^* is its matched ground-truth mask, and G_j is another adjacent instance of the same class. (a) Foreground false positive: over-segmented pixels which belong to the foreground. (b) Background false positive: over-segmented pixels which belong to the background. (c) False negative: under-segmented pixels. Best viewed in digit with color.

tives can be realized by different positive and negative sample definitions and loss function designs. Several works [7, 9, 15, 27, 29] applied contrastive learning to self-supervised pre-training and succeeded greatly. Recently, some work has attempted to apply this technique to dense prediction tasks. Some [18, 31, 37] improve semantic segmentation by mining semantic relations between pixels across different images via contrastive learning. Another line of work [2, 34, 39, 40] explores the use of contrastive learning to improve the performance of semi-supervised segmentation networks. In contrast, we focus on fully supervised instance segmentation and utilize contrastive learning to fix the false-negative errors.

Error Analysis Tools. Several works [1, 3, 4, 16, 17] have been proposed for the error analysis of object detection. COCO analysis toolkit [1] evaluates the impact of different error types on the precision-recall curve on the COCO dataset. UpperBound [4] analyzes the upper bound of the object detection. TIDE [3] isolates and compares different error types for object detection and instance segmentation. These tools are mainly designed for object detection. Although one can apply these tools on instance segmentation by replacing box IoU with mask IoU, they cannot analyze segmentation errors separately. As a complement, we propose an analysis tool for segmentation errors that can lead to more fine-grained conclusions. The Supplementary Material provides a detailed comparison.

3 Error Analysis

We decouple detection and segmentation errors in two steps: (1) for each predicted mask, we first try to match it with a ground-truth mask by IoU. If the predicted mask does not match any ground-truth mask, then we name it an unmatched prediction. The unmatched predictions and the missed ground-truth masks (not matched by any prediction) lead to detection errors, which have been analyzed in the previous error analysis tools. (2) For the matched masks pair, we compare their difference (defined as segmentation errors). Instead of analyzing detection errors, we focused on segmentation errors, which were not covered in previous works.

We classify the segmentation errors into the three types as shown in Figure 1. For a given image, we denote the M ground-truth masks and the N predicted masks as $\{G_i\}_{i=1}^{M}$ and $\{D_i\}_{i=1}^{N}$, respectively. Each element in the sets (*i.e.* G_i or D_i) is a set of pixels. For each prediction D_i, we first try to assign it with a ground-truth mask G_i^* with the largest mask

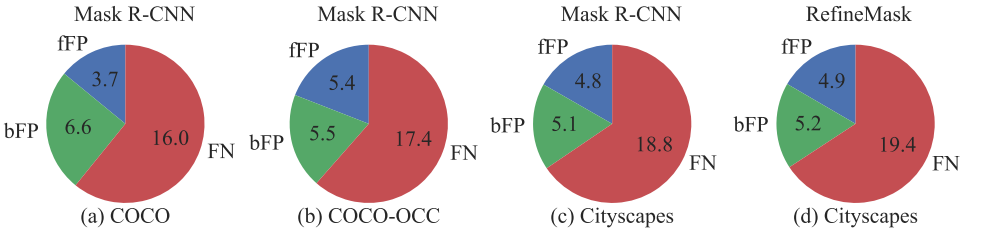


Figure 2: Segmentation error analysis of various models and datasets. The values indicate the expected improvements if we corrected each error.

IoU (0 for different categories). The predictions with IoU less than 0.1 for any ground-truth masks are unmatched, which are not covered in our analysis. Formally, the segmentation error pixels could be represented as the symmetric difference of D_i and G_i^*:

$$\begin{aligned}\mathcal{P}_i &= D_i \ominus G_i^* \\ &= (D_i - G_i^*) \cup (G_i^* - D_i) \\ &= \underbrace{(D_i \cap (G - G_i^*))}_{\text{fFP}} \cup \underbrace{(D_i \cap \bar{G})}_{\text{bFP}} \cup \underbrace{(G_i^* - D_i)}_{\text{FN}},\end{aligned} \tag{1}$$

where $G = \bigcup_{i=1}^{M} G_i$ contains all foreground pixels, $\bar{G}$ is its complement set (*i.e.*, all background pixels). As shown in Equation (1), the segmentation error pixels can be divided into three orthogonal sets (see Supplementary Material for a detailed derivation). The meaning of each set is as follows:

- **Foreground false positive (fFP)**: over-segmented pixels which belong to the foreground (Figure 1 (a)), indicating that the model is confusing different foreground instances (*e.g.* predict only one mask for two adjacent persons).
- **Background false positive (bFP)**: over-segmented pixels which belong to the background (Figure 1 (b)), indicating that the model incorrectly treats background pixels as part of the foreground.
- **False negative (FN)**: under-segmented pixels (Figure 1 (c)), indicating that the model incorrectly treats foreground pixels as background.

Following [3], we quantitatively measure each type of error as the expected mAP improvement if we corrected this error. Specifically, for a specific error type (denoted as o), correcting its corresponding error pixels improves the segmentation results from AP to AP_o. The AP gap indicates the severity of the specified error type:

$$\Delta\text{AP}_o = \text{AP}_o - \text{AP}. \tag{2}$$

Figure 2 shows the segmentation errors of Mask R-CNN [14] and RefineMask [33]. We report the results on Cityscapes [13], COCO [24], and COCO-OCC [22] (subset of COCO validation dataset with more occlusion) datasets for Mask R-CNN and results on Cityscapes for RefineMask. More analysis can be found in the Supplementary Material. From Figure 2, we draw two conclusions:

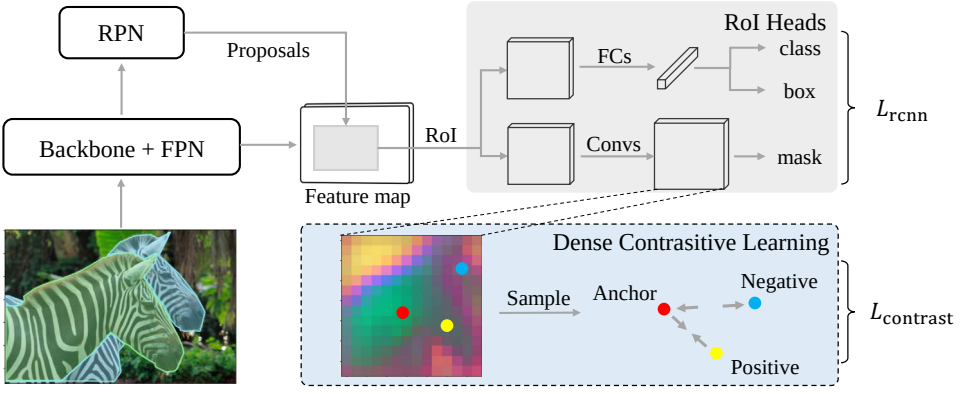


Figure 3: Overview of the proposed method. The proposed dense contrastive learning module can be integrated into Mask R-CNN in a plug-and-play manner. Anchor and positive samples are randomly selected from the target instance, while negative samples are selected from the rest of the RoI. Best viewed in color.

- *False-negative errors accounted for the majority.* Comparing different errors in Figure 2, the amount of FN errors is significantly more than fFP and bFP (*e.g.* 18.8% vs. 4.8% and 5.1% on Cityscapes for Mask R-CNN), implying that the model tends to predict incomplete masks.
- *Errors vary on different datasets.* The fFP and FN errors are higher on the COCO-OCC dataset than on the COCO dataset, which can be attributed to the occlusion between foreground objects and the occlusion of foreground objects by the background, respectively. FN error accounts for more on the Cityscapes dataset, indicating that the model yields more FN errors in crowded scenarios such as streets.

4 Dense Contrastive Loss

We first describe the motivation and overview of our method. Then, we present how to apply dense contrastive learning. Finally, we formalize the overall loss function for end-to-end training. We use Mask R-CNN [14] as an example, but our method is also applicable to other frameworks such as RefineMask [38] and SparseInst [12] (see Section 5.2 for the results).

4.1 Overview

We first briefly review the classical Mask R-CNN baseline. Given an image, Mask R-CNN first uses a backbone network and a feature pyramid network (FPN) to extract the image-level features. Then, they go through the region proposal network (RPN) to obtain the box proposals. These proposals feed into RoIAlign for the corresponding instance-level RoI features. Based on the RoI features, the RoI heads predict the class, the bounding box, and the mask of the corresponding instance. Typically, the mask head is a fully convolutional network containing several convolutional and upsample layers supervised by binary cross-entropy loss.

Based on the analysis in Section 3, we argue that instance features supervised only by binary cross-entropy are sub-optimal. Consistency of features on the same instance is not guaranteed due to the individual supervision of pixels. To address this issue, we propose a loss function based on dense contrastive learning, which works in a plug-and-play manner (as shown in Figure 3). Given the ground-truth instance mask, we sample several pixels from the mask as anchors or positive samples, and several pixels outside the mask as negative samples. Then, we extract the corresponding features and apply infoNCE loss [29] to pull closer those on the same instance. The proposed training strategy encourages the model to learn a more consistent feature representation for different pixels on the same instance, thus improving the completeness of the masks.

4.2 Dense Contrastive Learning

Sampling Strategy. For each RoI, we randomly sample the pixels on its corresponding ground-truth instance mask as positive samples or anchors, and the pixels on the remaining regions as negative samples. Specifically, for the i-th RoI, we denote the instance mask as M_i and its complement in the RoI as $\bar{M}_i$[1]. Both positive samples P_i and anchors A_i are uniformly sampled from M_i. Meanwhile, the negative samples N_i are uniformly sampled from $\bar{M}_i$. For simplicity, we sample the same number of positive samples, negative samples, and anchors, *i.e.* $|P_i| = |A_i| = |N_i| = K$. If the instance mask within an RoI is too large or small to sample enough points (*i.e.* $|M_i| < K$ or $|\bar{M}_i| < K$), then we disregard this RoI when computing the dense contrastive loss.

Dense Contrastive Loss. After obtaining the sampled pixels P_i, A_i and N_i, we perform dense contrastive learning via the infoNCE loss [29]. For the i-th RoI, the loss function is defined as

$$L^i_{\text{contrast}} = -\sum_{u\in A_i}\sum_{v\in P_i} \log \frac{\exp(\hat{F}^i_u \cdot \hat{F}^i_v/\tau)}{\exp(\hat{F}^i_u \cdot \hat{F}^i_v/\tau) + \sum_{w\in N_i}\exp(\hat{F}^i_u \cdot \hat{F}^i_w/\tau)}, \tag{3}$$

where F^i is the RoI feature (input for the last layer of the mask head), $\hat{F}^i$ is its L_2-normalized version in the channel dimension. $\hat{F}^i_u, \hat{F}^i_v$, and $\hat{F}^i_w$ denote the feature vectors corresponding to the spatial locations of u, v, and w, respectively. Temperature τ is a hyperparameter. The contrastive loss is the summation of all valid RoIs:

$$L_{\text{contrast}} = \sum_i L^i_{\text{contrast}}, \quad \text{if } |M_i| \geq K \text{ and } |\bar{M}_i| \geq K. \tag{4}$$

Intuitively, under the supervision of L_{contrast}, the features corresponding to the anchor will be close to the positive samples and away from the negative samples, resulting in better consistency of features on the same instance.

Learnable Similarity. Empirically, we found that it was suboptimal to define the similarity of features as cosine similarity directly. Following [9, 37], we add a projection layer $\phi_\theta(\cdot)$ (implemented as two fully connected layers) to make the similarity learnable. The similarity between the feature vectors f_1 and f_2 is thus formalized as

$$s_\theta(f_1, f_2) = \frac{\phi_\theta(f_1) \cdot \phi_\theta(f_2)}{||\phi_\theta(f_1)||_2 ||\phi_\theta(f_2)||_2}. \tag{5}$$

[1] M_i, $\bar{M}_i$ P_i, A_i and N_i are all sets of pixel positions.

λ	AP_{box}	AP_{seg}	AP^{50}	AP^{75}
0	37.6	32.4	58.2	29.7
0.2	39.3	34.5	60.6	33.6
0.4	39.0	35.1	60.2	34.1
0.6	39.7	35.0	60.9	33.9
0.8	38.9	35.3	61.1	33.8
1.0	39.6	36.1	**61.9**	**35.9**
1.2	**39.8**	**36.5**	61.8	35.8
1.4	39.2	35.3	60.1	34.8

Table 1: Results of different loss weight. The AP of the segmentation continues to increase as the loss weight increases and reaches saturation at a weight of 1.2.

learnable	AP_{box}	AP_{seg}	AP^{50}	AP^{75}
	38.8	35.1	60.7	32.9
✓	**39.8**	**36.5**	**61.8**	**35.8**

Table 2: Effects of learnable similarity. Learnable similarity leads to better performance.

τ	AP_{box}	AP_{seg}	AP^{50}	AP^{75}
0.01	38.7	35.0	60.5	33.8
0.04	38.7	35.0	60.4	34.1
0.07	39.8	**36.5**	**61.8**	35.8
0.10	**39.9**	36.4	61.2	**36.8**
0.13	38.5	35.7	61.3	35.2

Table 3: Results of different temperature. The best results were obtained when $\tau = 0.07$.

K	AP_{box}	AP_{seg}	AP^{50}	AP^{75}
0	37.6	32.4	58.2	29.7
8	38.5	35.1	61.0	34.0
16	39.2	35.3	59.9	34.5
32	**39.8**	**36.5**	61.8	**35.8**
64	38.5	35.9	**62.1**	35.5
128	38.2	35.1	60.3	32.9

Table 4: Results of different sample numbers. Too large or too small sample numbers lead to performance degradation.

4.3 Overall Loss

The overall training loss function contains the original Mask R-CNN loss and our dense contrastive loss. We denote the loss of Mask R-CNN as L_{rcnn} (including both RPN and RoI heads). Then, we introduce the weight λ to balance the newly added loss, giving the following overall loss form:

$$L = L_{\text{rcnn}} + \lambda L_{\text{contrast}}. \tag{6}$$

The model parameters, including the projection layer ϕ_θ, are trained in an end-to-end manner.

5 Experiments

Experimental Settings: We mainly conducted experiments on the Cityscapes dataset (with `fine` annotations only) [13]. Cityscapes is a real-world dataset on urban street scenes, containing 2975, 500, and 1525 images as the training, validation and test set, respectively. Our implementation was based on the popular `detectron2` framework. Unless otherwise specified, Mask R-CNN with ResNet-50 (pre-trained on ImageNet [26]) and FPN as backbone was used as the baseline. The training lasted 24000 iterations with a batch size of 8 (on 4× 2080Ti GPUs). All other settings were the same as the default settings of `detectron2`.

5.1 Ablation Study

In this section, we performed ablation experiments on the different settings of the proposed method. We used the COCO API for evaluation, which yields slightly lower results than the standard Cityscapes API, but with more detailed metrics. We reported on box AP (AP_{box})

Method	COCO	AP_{val}	AP_{test}	person	rider	car	truck	bus	train	mcycle	bicycle
PointRend [23]		35.8	-	-	-	-	-	-	-	-	-
Mask R-CNN [14]	✓	36.4	32.0	34.8	27.0	49.1	30.1	40.9	30.9	24.1	18.7
BShapeNet+ [21]	✓	-	32.9	36.6	24.8	50.4	**33.7**	41.0	33.7	**25.4**	17.8
UPSNet [36]	✓	37.8	33.0	35.9	27.4	51.9	31.8	**43.1**	31.4	23.8	19.1
CondInst [28]	✓	37.5	33.2	35.1	27.7	54.5	29.5	42.3	**33.8**	23.9	18.9
Mask R-CNN*		33.1	28.4	32.9	25.6	49.6	23.7	36.0	22.4	20.1	17.2
w/ Ours		37.1	31.1	37.1	29.1	53.7	25.2	37.7	23.1	22.6	20.4
Mask R-CNN*	✓	37.3	32.0	36.1	29.1	51.8	29.2	38.3	28.0	23.5	19.6
w/ Ours	✓	38.6	33.5	38.3	**30.6**	54.2	29.2	38.6	30.4	25.3	21.1
RefineMask [38]		37.6	32.0	37.4	29.3	55.6	26.6	36.5	26.6	23.4	20.8
w/ Ours		**39.0**	**33.6**	**39.3**	30.4	**56.9**	27.5	40.2	28.3	24.5	**21.5**

Table 5: Comparison with previous methods on Cityscapes dataset. "✓" indicates pre-training on COCO dataset. "*" denotes our implementation. "-" means that data is not available. We report Cityscapes-style results here to be consistent with previous methods.

and segmentation AP (AP_{seg}), and segmentation AP at different IoU thresholds (AP^{50}, AP^{75}). Without specification, we used $\lambda = 1.2, \tau = 0.07, K = 32$, and applied learnable similarity as the default setting.

Loss Weight. Since a new loss term was added, we first investigated how to balance it with the existing loss. We started from 0 and gradually increased its weight λ to obtain the results in Table 1. The segmentation AP gradually increased as the loss weight increased and saturated at $\lambda = 1.2$. Excessive weight led to performance degradation. With appropriate loss weight, our model improved the segmentation AP by 4.1% compared to the baseline. The greater improvement in AP^{75} indicates the advantages of our method for predicting more precise masks. The box AP also has a significant improvement (+2.2%), which can be attributed to the benefits of better instance features for localization.

Learnable Similarity. We verified the necessity for learnable similarity with the experiments in Table 2. The model trained with learnable similarity achieved 1.4% higher segmentation AP than trained with original cosine similarity.

Temperature. Temperature τ can affect the strength of penalties of the contrastive loss on the hard negative samples. Smaller temperatures lead to more emphasis on to hard negative samples, and vice versa [30]. We studied the effects of different temperature values in Table 3. The best results were obtained when $\tau = 0.07$.

Number of Samples. We studied the effect of the different sample numbers and presented the results in Table 4. Inadequate samples led to insufficient supervision. When the sample number K was too large, the number of invalid RoIs (with ground-truth mask too large or too small to be sampled without repetition) increased. For example, when $K = 128$, about 10% of the RoI was invalid during training. The absence of RoI led to performance degradation. We found that $K = 32$ achieved the best balance. The Supplementary Material provides the results when $|P_i|$, $|A_i|$ and $|N_i|$ are not equal.

Dataset	DCL	AP_{dev}	AP_{val}	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
COCO [24]		35.4	35.2	56.2	37.5	**17.1**	37.5	50.4
	✓	**36.0**	**35.7**	**56.4**	**38.3**	16.9	**38.1**	**51.0**
COCO-OCC [22]		-	31.5	52.7	33.4	13.6	27.8	42.0
	✓	-	**32.3**	**53.5**	**33.9**	**14.3**	**28.4**	**43.7**

Table 6: Segmentation results on COCO and COCO-OCC dataset. "✓" indicates that trained with dense contrastive loss (DCL).

	Backbone	DCL	AP_{dev}	AP_{val}	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}
Mask R-CNN	Swin-T		39.9	39.3	62.2	42.2	**20.5**	41.8	57.8
		✓	**40.2**	**39.9**	**62.6**	**42.7**	20.1	**43.0**	**58.3**
SparseInst	ResNet-50		32.0	31.6	**51.4**	32.9	**12.7**	33.3	48.5
		✓	**32.4**	**32.1**	51.3	**33.9**	12.3	**34.8**	**48.9**

Table 7: Application on other models on COCO dataset. SparseInst was trained for 73 epochs (half of the original) for faster convergence.

5.2 Overall Results

Comparison with Previous Methods. We compared the proposed method with the previous methods on Cityscapes dataset in Table 5. To verify the universality of our method on different models, we also report the results on RefineMask [33]. RefineMask is a multi-stage refinement model. We applied the proposed loss function to its first stage (instance head). Our method achieved consistent improvement on different baseline models, improving the test set AP by 1.6% on the powerful RefineMask baseline. Compared with other methods such as UPSNet [36] and CondInst [28], our method also yielded superior results. Note that our method does not introduce any test-time overhead and therefore does not affect the efficiency.

Results on Other Datasets. We extended our method (with Mask R-CNN) to the COCO [24] and COCO-OCC [22] (subset of COCO validation set with more occlusion) datasets. The results are shown in Table 6. Incorporating the dense contrastive loss into the mask head improved the AP significantly. The AP_{val} on COCO and COCO-OCC were improved by 0.5% and 0.8%, respectively. The improvement on COCO-OCC is larger than COCO, indicating that our method is more advantageous in complex scenes (*e.g.*, occlusion). Compared to the Cityscapes dataset, the improvement on these two datasets is relatively less. We attribute this to the differences in the distribution of the datasets. Figure 2 illustrates that the false-negative errors on these datasets are fewer than on Cityscapes, which limits the gain of our method.

Application on Other Models. We report in Table 7 the results on SparseInst [12] (a recent state-of-the-art real-time method) and Mask R-CNN with larger backbone (*e.g.*, Swin Transformer [25]). The experiments were conducted on the COCO dataset. For SparseInst, we applied DCL to the output features of the mask branch. Since proposals are absent from SparseInst, we use the ground-truth bounding box as the RoI. The RoI features were extracted using RoIAlign and supervised by DCL in the same way as in Figure 3. Our method achieves consistent improvement on these recent and stronger baselines.

Analysis on the Improvement. We analyzed the sources of improvement based on the error analysis tool presented in Section 3. As shown in Table 8, the model trained with our dense contrastive loss yielded significantly fewer false-negative errors. Meanwhile, fFP and bFP

	AP_{val}	$\Delta AP_{fFP} \downarrow$	$\Delta AP_{bFP} \downarrow$	$\Delta AP_{FN} \downarrow$
Baseline	32.3	**5.0**	**4.7**	21.2
+Ours	**36.5**	5.4	5.3	**18.8**

Table 8: Analysis on the improvement. "↓" means the lower the better. Our method significantly alleviates false-negative errors.

	Dataset	Device	Epochs	DCL	Time	GPU Mem.
SparseInst	COCO	V100	73		33.5h	12.4G
				✓	40.0h	12.4G
Mask R-CNN	COCO	2080Ti	12		17.5h	7.7G
				✓	21.5h	8.4G
RefineMask	Cityscapes	3080	64		8.6h	6.9G
				✓	9.3h	7.0G

Table 9: Computation and GPU memory cost (counted by Pytorch's API) during training.

errors slightly increased, suggesting a trade-off between the different errors.

Training-time Efficiency. We report the training time and GPU memory usage in Table 9. Compared to baselines, our method introduced only 10-20% extra training time and negligible GPU memory increase.

Qualitative Results. We present the visualization results in the Supplementary Material. Compared with the baseline, our method yielded higher quality and more complete masks, consistent with the quantitative results in Table 8. The t-SNE visualization (Figure 4) also verified that our model learns a more compact feature representation.

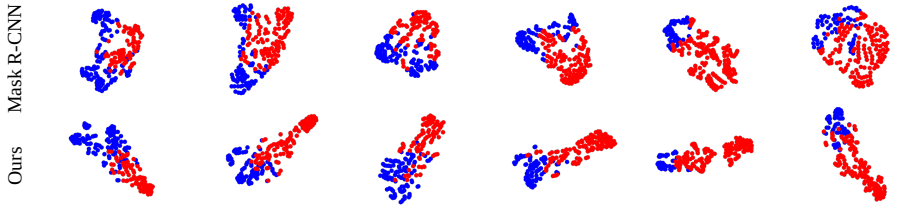


Figure 4: t-SNE visualization of the instance features. Features corresponding to the foreground and background pixels are shown in red and blue, respectively.

6 Conclusions

We analyzed segmentation errors in instance segmentation and proposed a dense contrastive loss for alleviating the false-negatives errors. Without introducing any inference overhead, our method achieved consistent improvement across various baselines and various datasets. Analysis of the improvement showed that false-negative errors were significantly mitigated. We hope our work will improve the understanding of errors in instance segmentation and advance the study of contrastive learning for downstream tasks such as instance segmentation.

Acknowledgements: This work was supported by the National Natural Science Foundation of China (Nos. 62061136001, U19B2034, 61836014) and THU-Bosch JCML center.

References

[1] Coco analysis toolkit. http://cocodataset.org/#detection-eval. Accessed: 2022-06-28.

[2] Iñigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C. Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 8199–8208. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00811. URL https://doi.org/10.1109/ICCV48922.2021.00811.

[3] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. TIDE: A general toolbox for identifying object detection errors. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part III*, volume 12348 of *Lecture Notes in Computer Science*, pages 558–573. Springer, 2020. doi: 10.1007/978-3-030-58580-8_33. URL https://doi.org/10.1007/978-3-030-58580-8_33.

[4] Ali Borji and Seyed Mehdi Iranmanesh. Empirical upper bound in object detection and more. *CoRR*, abs/1911.12451, 2019. URL http://arxiv.org/abs/1911.12451.

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1483–1498, 2021. doi: 10.1109/TPAMI.2019.2956516. URL https://doi.org/10.1109/TPAMI.2019.2956516.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020. doi: 10.1007/978-3-030-58452-8_13. URL https://doi.org/10.1007/978-3-030-58452-8_13.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951.

[8] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and

Dahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4974–4983. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00511. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Hybrid_Task_Cascade_for_Instance_Segmentation_CVPR_2019_paper.html.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. URL http://proceedings.mlr.press/v119/chen20j.html.

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *CoRR*, abs/2112.01527, 2021. URL https://arxiv.org/abs/2112.01527.

[11] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask R-CNN. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV*, volume 12359 of *Lecture Notes in Computer Science*, pages 660–676. Springer, 2020. doi: 10.1007/978-3-030-58568-6_39. URL https://doi.org/10.1007/978-3-030-58568-6_39.

[12] Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, and Wenyu Liu. Sparse instance activation for real-time instance segmentation. *CoRR*, abs/2203.12827, 2022. doi: 10.48550/arXiv.2203.12827. URL https://doi.org/10.48550/arXiv.2203.12827.

[13] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.350. URL https://doi.org/10.1109/CVPR.2016.350.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2980–2988. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.322. URL https://doi.org/10.1109/ICCV.2017.322.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019. URL http://arxiv.org/abs/1911.05722.

[16] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi

Sato, and Cordelia Schmid, editors, *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, volume 7574 of *Lecture Notes in Computer Science*, pages 340–353. Springer, 2012. doi: 10.1007/978-3-642-33712-3_25. URL https://doi.org/10.1007/978-3-642-33712-3_25.

[17] Jan Hendrik Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? In Michel François Valstar, Andrew P. French, and Tony P. Pridmore, editors, *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press, 2014. URL http://www.bmva.org/bmvc/2014/papers/paper082/index.html.

[18] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 16271–16281. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01598. URL https://doi.org/10.1109/ICCV48922.2021.01598.

[19] Jie Hu, Liujuan Cao, Yao Lu, Shengchuan Zhang, Yan Wang, Ke Li, Feiyue Huang, Ling Shao, and Rongrong Ji. ISTR: end-to-end instance segmentation with transformers. *CoRR*, abs/2105.00637, 2021. URL https://arxiv.org/abs/2105.00637.

[20] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6409–6418. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00657. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Huang_Mask_Scoring_R-CNN_CVPR_2019_paper.html.

[21] Ba Rom Kang, Hyunku Lee, Keunju Park, Hyunsurk Ryu, and Ha Young Kim. Bshapenet: Object detection and instance segmentation with bounding shape masks. *Pattern Recognit. Lett.*, 131:449–455, 2020. doi: 10.1016/j.patrec.2020.01.024. URL https://doi.org/10.1016/j.patrec.2020.01.024.

[22] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4019–4028. Computer Vision Foundation / IEEE, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Ke_Deep_Occlusion-Aware_Instance_Segmentation_With_Overlapping_BiLayers_CVPR_2021_paper.html.

[23] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross B. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9796–9805. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00982. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kirillov_PointRend_Image_Segmentation_As_Rendering_CVPR_2020_paper.html.

[24] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9992–10002. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00986. URL https://doi.org/10.1109/ICCV48922.2021.00986.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

[27] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, volume 12356 of *Lecture Notes in Computer Science*, pages 776–794. Springer, 2020. doi: 10.1007/978-3-030-58621-8_45. URL https://doi.org/10.1007/978-3-030-58621-8_45.

[28] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2020. doi: 10.1007/978-3-030-58452-8_17. URL https://doi.org/10.1007/978-3-030-58452-8_17.

[29] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

[30] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2495–2504. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00252. URL https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html.

[31] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7283–7293. IEEE, 2021. doi: 10.1109/

ICCV48922.2021.00721. URL https://doi.org/10.1109/ICCV48922.2021.00721.

[32] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: segmenting objects by locations. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVIII*, volume 12363 of *Lecture Notes in Computer Science*, pages 649–665. Springer, 2020. doi: 10.1007/978-3-030-58523-5_38. URL https://doi.org/10.1007/978-3-030-58523-5_38.

[33] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/cd3afef9b8b89558cd56638c3631868a-Abstract.html.

[34] Xuehui Wang, Kai Zhao, Ruixin Zhang, Shouhong Ding, Yan Wang, and Wei Shen. Contrastmask: Contrastive learning to segment every thing. *CoRR*, abs/2203.09775, 2022. doi: 10.48550/arXiv.2203.09775. URL https://doi.org/10.48550/arXiv.2203.09775.

[35] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12190–12199. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01221. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Xie_PolarMask_Single_Shot_Instance_Segmentation_With_Polar_Representation_CVPR_2020_paper.html.

[36] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8818–8826. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00902. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Xiong_UPSNet_A_Unified_Panoptic_Segmentation_Network_CVPR_2019_paper.html.

[37] Feihu Zhang, Philip H. S. Torr, René Ranftl, and Stephan R. Richter. Looking beyond single images for contrastive semantic segmentation learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3285–3297, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1a68e5f4ade56ed1d4bf273e55510750-Abstract.html.

[38] Gang Zhang, Xin Lu, Jingru Tan, Jianmin Li, Zhaoxiang Zhang, Quanquan Li, and Xiaolin Hu. Refinemask: Towards high-quality instance segmentation with fine-grained features. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 6861–6869. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00679. URL https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_RefineMask_Towards_High-Quality_Instance_Segmentation_With_Fine-Grained_Features_CVPR_2021_paper.html.

[39] Yuanyi Zhong, Bodi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7253–7262. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00718. URL https://doi.org/10.1109/ICCV48922.2021.00718.

[40] Xiaoyu Zhu, Jeffrey Chen, Xiangrui Zeng, Junwei Liang, Chengqi Li, Sinuo Liu, Sima Behpour, and Min Xu. Weakly supervised 3d semantic segmentation using cross-image consensus and inter-voxel affinity relations. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2814–2824. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00283. URL https://doi.org/10.1109/ICCV48922.2021.00283.