


Article

Image Segmentation from Sparse Decomposition with a Pretrained Object-Detection Network

Yulin Wu [†] , Chuandong Lv [†], Baoqing Ding, Lei Chen, Bin Zhou ^{*} and Hongchao Zhou ^{*}

School of Information Science and Engineering, Shandong University, Qingdao 266237, China; yulinw@mail.sdu.edu.cn (Y.W.); sdulcd@mail.sdu.edu.cn (C.L.); 201912470@mail.sdu.edu.cn (B.D.); lei.chen@sdu.edu.cn (L.C.)

^{*} Correspondence: binzhou@sdu.edu.cn (B.Z.); hongchao@sdu.edu.cn (H.Z.)

[†] These authors contributed equally to this work.

Abstract: Annotations for image segmentation are expensive and time-consuming. In contrast to image segmentation, the task of object detection is in general easier in terms of the acquisition of labeled training data and the design of training models. In this paper, we combine the idea of unsupervised learning and a pretrained object-detection network to perform image segmentation, without using expensive segmentation labels. Specially, we designed a pretext task based on the sparse decomposition of object instances in videos to obtain the segmentation mask of the objects, which benefits from the sparsity of image instances and the inter-frame structure of videos. To improve the accuracy of identifying the ‘right’ object, we used a pretrained object-detection network to provide the location information of the object instances, and propose an Object Location Segmentation (OLSeg) model of three branches with bounding box prior. The model is trained from videos and is able to capture the foreground, background and segmentation mask in a single image. The performance gain benefits from the sparsity of object instances (the foreground and background in our experiments) and the provided location information (bounding box prior), which work together to produce a comprehensive and robust visual representation for the input. The experimental results demonstrate that the proposed model boosts the performance effectively on various image segmentation benchmarks.

Keywords: image segmentation; object detection; pretext task; sparse decomposition



Citation: Wu, Y.; Lv, C.; Ding, B.; Chen, L.; Zhou, B.; Zhou, H. Image Segmentation from Sparse Decomposition with a Pretrained Object-Detection Network. *Electronics* **2022**, *11*, 639. <https://doi.org/10.3390/electronics11040639>

Academic Editor: Gemma Piella

Received: 11 January 2022

Accepted: 16 February 2022

Published: 18 February 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Supervised learning has been successfully applied to many complex computer vision tasks [1,2]. However, training deep neural networks requires a tremendous amount of manual labeling and prior knowledge from experts. Especially in image segmentation tasks, the cost of pixel-level annotations acquisition is significantly larger than that of region-level and image-level labels, respectively [3]. This motivates the development of weakly supervised image segmentation methods [4]. Different forms of supervision are explored to promote the performance of image segmentation [5,6].

Previous works [7,8] study the forms of supervision to solve the image segmentation problems. Co-segmentation regards a set of images with the same category as the supervision information and segments the common objects. The initial works are mainly based on effective computational models [9–11] to process an image pair. More recent co-segmentation methods are formulated as modeling optimization [12], object saliency [13], or data clustering [14–16] problems in more than two images. However, the existing methods still suffer from certain limitations including hand-crafted features and unscalable prior [17]. Moreover, these methods rely on the collections of images to segment the objects and fail to segment in a single image. Recent works [18,19] have demonstrated that the pretrained network could provide powerful semantic representation for a given image. Therefore, it is necessary to explore the potential of exploiting the pretrained networks:

networks trained for one task, but that could be reused in scenarios that are different from their original purposes [20].

Self-supervised learning aims at designing various pretext tasks to assist the main task in exploring the properties of unlabeled data [21–25]. The methods [26,27] use the generative models to reconstruct inputs. However, most methods [21–23,26,27] are mainly based on static images to train the models. Video sequences give rise to characteristic patterns of visual change including geometric shape, motion tendencies and many other properties. Natural videos could serve as a more powerful signal for both static and dynamic visual tasks. Self-supervised methods based on videos [24,25] utilize optical flow estimation to find correspondences between pixels across video frames. All moving objects can result in optical flow; therefore, these methods cannot perceive the objects of interest in the scene.

To address these limitations, we aimed to design a pretext task based on the sparse decomposition of object instances in videos, and use the encoder–decoder structures to reconstruct the input to promote the image segmentation tasks. Our motivation is based on two observations. (1) The sparsity of image instances: An image typically consists of multiple objects instances (such as the foreground and background). The objects can be sparsely represented by a deep neural network to learn the features. (2) The inter-frame structure of videos. Compared with static images, the objects instances between continuous frames have high correlation and can be sparsely represented. Video data is more common and conducive to visual learning. We explore the two ideas and utilize the sparsity of image instances and the inter-frame structure of videos.

In this paper, we propose an Object Location Segmentation (OLSeg) model of three branches with bounding box prior. The model is trained from videos and is able to capture the foreground, background and segmentation mask in a single image. We consider a relatively simple scenario where each image consists of a foreground and a background, and construct the model based on the following three aspects. (1) The first of these is the foreground and background branches. On the one hand, we use an autoencoder [28] in the foreground and background branches to construct the foreground and background respectively. The encoder in the foreground branch outputs more channels than the encoder in the background branch to express more complex motion information of the foreground. On the other hand, we apply a gradient loss to smooth the background, which avoids the appearance of the foreground object. (2) The second is the mask branch. We use a U-Net [29] to generate the mask, and adopt an object loss to focus on the information in the bounding box of the foreground object. Our motivation is that the segmentation mask can be calculated if the object location is given. The location information is obtained by a pretrained object-detection network [30]. In addition, we consider a closed loss to ensure that the mask shows smooth contours without holes, and a binary loss to generate a binary mask. (3) The final aspect is image reconstruction. The original image is reconstructed by combining the foreground and background with the binary mask.

We conclude our contributions as follows:

- We designed a pretext task based on the sparse decomposition of object instances in videos for image segmentation. The task benefits from the sparsity of image instances and the inter-frame structure of videos;
- We propose an OLSeg model of three branches with bounding box prior. The location information of the object is obtained by a pretrained object-detection network. The model trained from videos is able to capture the foreground, background and segmentation mask in a single image;
- The proposed UnsupRL model is demonstrated to boost the performance effectively on various image segmentation benchmarks. The ablation study shows the gains of different components in OLSeg.

2. Related Work

2.1. Image Segmentation from Unlabeled Data

Image segmentation from unlabeled data is challenging due to the fact that it does not have the pixel-level annotations rather than a given unlabeled image. Recent methods [31,32] explore different forms of supervision to promote the segmentation performance. Stretcu et al. [33] matched multiple video frames for image segmentation. Papazoglou et al. [34] relied on the optical flow to identify moving objects. Koh et al. [35] iteratively generated the object proposals. Wang et al. [36] focused on the relations between pixels across different images. Zhou et al. [37] used the optical flow and attention mechanism to segment the video objects; image co-segmentation requires a set of images containing objects from the same category as a weak form of supervision. Rother et al. [38] minimised an energy function to segment image pairs. Kim et al. [12] explored an optimization problem of saliency to find the common object in multiple images. Joulin et al. [14] formulated the co-segmentation problem in terms of a discriminative clustering task. Joulin et al. [15] used spectral and discriminative clustering for fully unsupervised segmentation. Rubinstein et al. [13] combined a visual saliency and dense correspondences to capture the sparsity and visual variability of the common objects in a group of images. Quan et al. [16] used a pretrained network to obtain the semantic features, and proposed a manifold ranking method to discover the common objects. Zhao et al. [17] constrained the proportion of foreground object in the image. These methods segmented images based on hand-crafted features and unscalable prior. Our model does not depend on any assumptions about the existence of the common objects, and is able to segment in a single image.

2.2. Pretrained Networks

Current deep learning methods have achieved great success on a variety of visual tasks [39,40]. However, these deep frameworks still heavily rely on a large amount of training data and a time-consuming training process. Therefore, some researchers have recently turned to a new direction: network reuse [20]. Obviously it would be appealing if a pretrained network can be reused in a new domain that is different from its initial training purpose, and without needing to fine-tune it. Some recent works attempt to train deep models on a large image dataset collecting from web images [41], which can further leverage the generalization of pretrained networks. Fortunately, benefiting from the large-scale dataset COCO [3], which consists of over 330,000 images in over 80 categories, the pretrained networks have revealed powerful object detection ability. Inspired by [42], our proposed OLSeg extracts the location of the foreground object from a pretrained object-detection network without any training or fine-tuning process. This operation can be easily implemented and is conducive to object segmentation.

2.3. Self-Supervised Learning

Self-supervised learning methods usually construct a pretext task to learn features from raw data, and improve the performance of the main task. Some methods use generation-based models [43–45] to obtain the latent feature representations of the input. Various pretext tasks have been explored, e.g., predicting relative patch locations within an image [21], recovering part of the data [46], solving jigsaw puzzles [22], colorizing grayscale images [23], counting visual primitives [47] and predicting image rotations [48]. For videos, self-supervised signals come from motion consistency [24,25] and temporal continuity [49,50]. However, the pretext tasks based on videos estimate the optical flow as the clue of the moving object. The object of interest may be stationary and cannot be perceived through optical flow. We design a pretext task based on the sparse decomposition of object instances in videos, and use the encoder–decoder structures to reconstruct the original input.

3. Object Location Segmentation (OLSeg)

We propose a three-branch OLSeg model with bounding box prior, as shown in Figure 1. The model consists of a foreground branch, a background branch and a mask branch. For an input image from continuous video frames, the foreground and background branches construct the foreground and background information with an autoencoder [28], respectively. The output channels of encoder in the two autoencoders are different. The foreground branch contains more encoder channels than the background branch to represent more complex foreground information. The gradient loss in the background branch smooths the background and eliminates the influence of foreground. The mask branch with bounding box prior uses an U-Net [29] to generate the segmentation mask. The bounding box of the object is obtained from a pretrained object-detection network Yolov4 [30] acting on the input. The location information is used in an object loss to make the U-Net [29] focus more on the area where the foreground object appears. The closed loss and binary loss ensure that the generated mask is binary without concave holes. The mask is combined with the foreground and background to reconstruct the input via a reconstruction loss. The contributions of the three branches are complementary to each other for final decomposition of the object instances. For clarity, the pseudocode for OLSeg is shown in Algorithm 1.

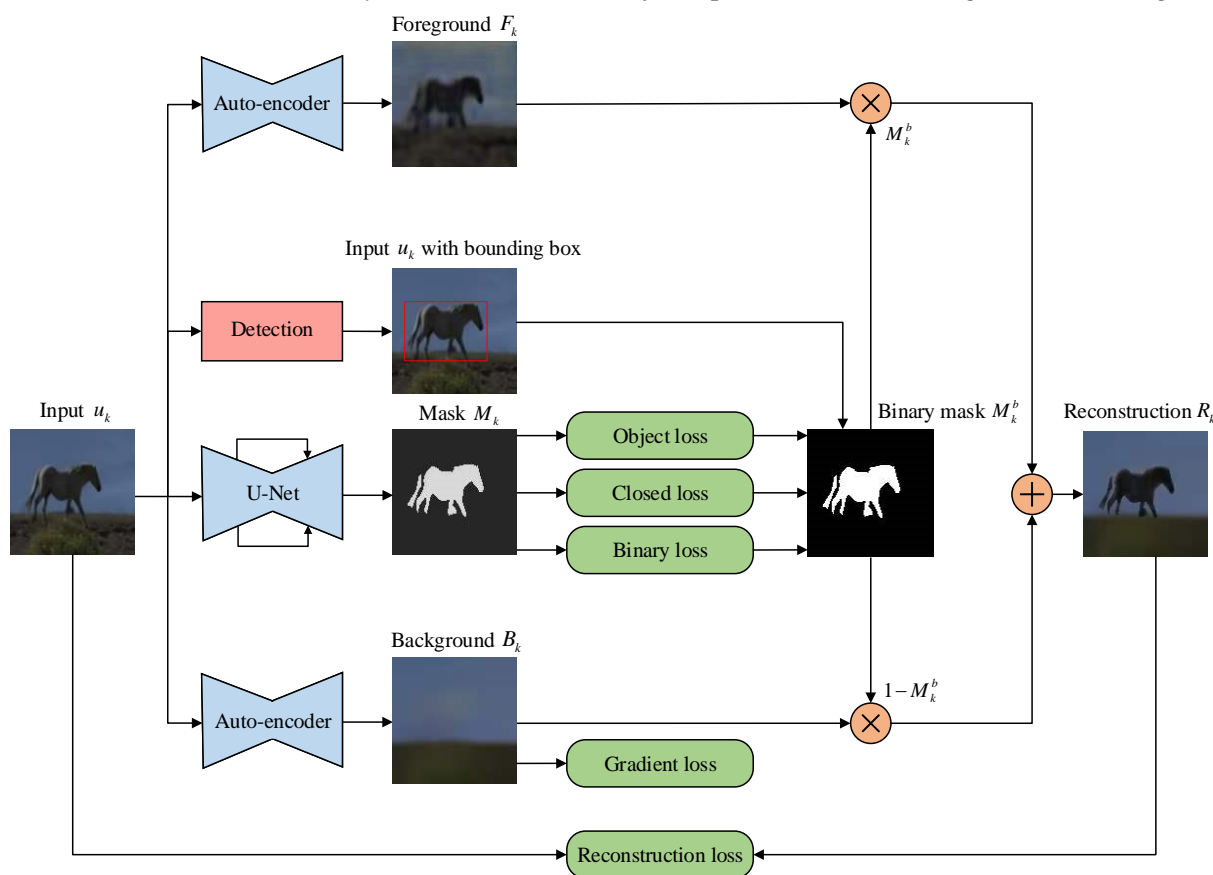


Figure 1. Overview of OLSeg. The model consists of the foreground and background branches with an autoencoder [28] respectively, and a mask branch with an U-Net [29]. The gradient loss in the background branch removes the foreground to great extent. The mask branch with bounding box prior uses an object loss, a closed loss and a binary loss. The outputs of the three branches work together to reconstruct the input.

3.1. The Foreground and Background Branches

The foreground and background branches aim to decompose the object instances into the foreground and background. The foreground branch consists of an autoencoder [28] f_{a_1} , and the background branch consists of an autoencoder [28] f_{a_2} . The two autoencoders [28] are trained separately. Let $U = (u_k; k \in (1, \dots, K))$ be continu-

ous video frames. Given an input image u_k , the outputs of f_{a_1} and f_{a_2} are obtained using Equations (1) and (2) respectively:

$$F_k = f_{a_1}(u_k), \tag{1}$$

$$B_k = f_{a_2}(u_k), \tag{2}$$

where F_k and B_k are the foreground and background of the input image u_k respectively.

Algorithm 1 Training OLSeg.

Input: An autoencoder f_{a_1} , an autoencoder f_{a_2} , an U-Net f_u and a pretrained Yolov4 f_p . Batch of continuous video frames $U = (u_k; k \in (1, \dots, K))$, coordinates (i, j) , numbers of coordinates N_{mk} and N_{ak} in M_k and A_k . The encoder channels C_f, C_b , and the loss balancing hyperparameters $\alpha, \beta_o, \beta_c, \beta_b$.

- 1: **for** $k = 1$ to K **do**
- 2: $F_k = f_{a_1}(u_k)$ ▷ Foreground F_k with the encoder channels C_f
- 3: $B_k = f_{a_2}(u_k)$ ▷ Background B_k with the encoder channels C_b
- 4: $M_k = f_u(u_k)$ ▷ Segmentation mask M_k
- 5: $A_k = f_p(u_k)$ ▷ Area of the foreground object A_k
- 6: $R_k = M_k^b \times F_k + (1 - M_k^b) \times B_k$ ▷ Reconstruction image R_k and binary mask M_k^b
- 7: **end for**
- 8: $L_g = \frac{1}{|U|} \sum_{k=1}^K \nabla B_k$ ▷ Gradient loss
- 9: $L_o = \frac{1}{|U|} \sum_{k=1}^K \sum_{(i,j) \notin A_k}^{N_{mk}-N_{rk}} |M_{k,(i,j)} - 0|^2$ ▷ Object loss
- 10: $L_c = \frac{1}{4|U|} \sum_{k=1}^K \sum_{(i,j) \in M_k}^{N_{mk}} \left[|M_{k,(i,j)} - M_{k,(i-1,j)}|^2 + |M_{k,(i,j)} - M_{k,(i+1,j)}|^2 + |M_{k,(i,j)} - M_{k,(i,j-1)}|^2 + |M_{k,(i,j)} - M_{k,(i,j+1)}|^2 \right]$ ▷ Closed loss
- 11: $L_b = -\frac{1}{|U|} \sum_{k=1}^K \sum_{(i,j) \in M_k}^{N_{mk}} |M_{k,(i,j)} - 0.5|^2$ ▷ Binary loss
- 12: $L_m = \beta_o L_o + \beta_c L_c + \beta_b L_b$ ▷ Loss of the mask branch
- 13: $L_r = \frac{1}{|U|} \sum_{k=1}^K |u_k - R_k|^2$ ▷ Reconstruction loss
- 14: $L = L_r + \alpha L_g + L_m$ ▷ Overall loss of OLSeg

Output: Segmentation network f_u .

The difference between f_{a_1} and f_{a_2} is that they have different output channels of encoders. Generally, the foreground information is more complex and variable than the background information. As shown in Figure 1, the horse is moving and the background is relatively simple in continuous video frames. Let C_f and C_b be the output channels of encoders in f_{a_1} and f_{a_2} , respectively. We use more output channels C_f to express the foreground information.

The key to realizing the foreground and background separation is to ensure that the background does not contain the foreground information. The powerful generation ability of autoencoders [28] leads to the appearance of foreground in the background. In order to ensure that the background is smooth and clean, we add a gradient loss to the background branch as Equation (3):

$$L_g = \frac{1}{|U|} \sum_{k=1}^K \nabla B_k, \tag{3}$$

where ∇ is the operation of minimizing the gradient.

Although the foreground and background extracted from f_{a_1} and f_{a_2} are blurred in Figure 1, the outline of the horse is visible in the foreground, and the sky and grass are

clean in the background. This proves the decomposition ability of the foreground and background branches to the object instances.

3.2. The Mask Branch

The mask branch consists of an U-Net [29] f_u to generate the segmentation mask. Multi-scale connections between the encoding and decoding paths of U-Net [29] ensure efficient integration of information in image segmentation. The output of U-Net [29] for the input u_k is represented as Equation (4):

$$M_k = f_u(u_k), \quad (4)$$

where M_k is the segmentation mask and the values range from 0 to 1.

It is challenging for the mask branch to distinguish the foreground and background of the input. The previous methods [24,25] use optical flow to treat the moving object as the foreground. However, the foreground may be relatively static in continuous video frames. We consider the location information of the foreground object to assist the mask branch to better discover the object. Our motivation is that the segmentation mask could be obtained if the object location is known. The operation of extracting object location information is shown in Figure 2. For the given input u_k , we select a Yolov4 [30] trained on the COCO dataset [3] to obtain the bounding box of the object area. The comparison of different pretrained object-detection networks is not considered because we only need to obtain the approximate location of the object. Then the corresponding location of the bounding box in the input is mapped to the segmentation mask. The object area in the bounding box for the segmentation mask is represented as A_k (A_k is their union if there are multiple object areas). Let the value of coordinates (i, j) in the segmentation mask M_k be $M_{k,(i,j)}$, and we introduce an object loss as Equation (5):

$$L_o = \frac{1}{|U|} \sum_{k=1}^K \sum_{(i,j) \notin A_k}^{N_{mk} - N_{rk}} |M_{k,(i,j)} - 0|^2, \quad (5)$$

where N_{mk} and N_{rk} represent the numbers of coordinates in M_k and A_k respectively. For the segmentation mask M_k , we set the values outside A_k to 0. The mask branch focuses on the object area inside the bounding box to more accurately obtain the mask.

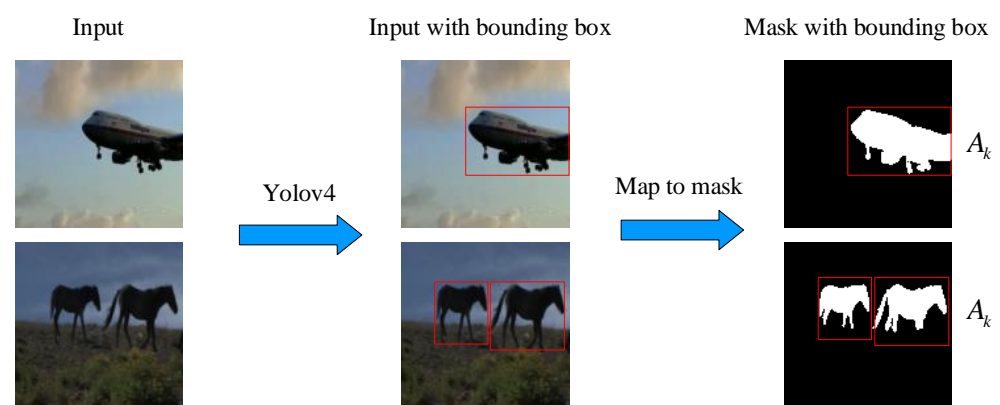


Figure 2. The operation of extracting object location information. We obtain the bounding box of the object from a pretrained Yolov4 [30]. The bounding box is then mapped into the segmentation mask, and the object area A_k is obtained.

The segmentation mask sometimes shows unclosed concave holes. Small holes inside the object have an adverse impact on segmentation. We adopt a closed loss to produce a smooth mask without concave holes. The idea is that the value of the current coordinates

(i, j) in the mask M_k is consistent with the values of the surrounding adjacent coordinates. The closed loss L_c is calculated as Equation (6):

$$L_c = \frac{1}{4|U|} \sum_{k=1}^K \sum_{(i,j) \in M_k}^{N_{mk}} \left[|M_{k,(i,j)} - M_{k,(i-1,j)}|^2 + |M_{k,(i,j)} - M_{k,(i+1,j)}|^2 + |M_{k,(i,j)} - M_{k,(i,j-1)}|^2 + |M_{k,(i,j)} - M_{k,(i,j+1)}|^2 \right] \quad (6)$$

The reconstruction image is obtained by the combination of the foreground, background and mask. The values of the segmentation mask M_k range from 0 to 1. The low-entropy prediction of the mask branch can ensure that either the foreground or the background appears in the reconstruction image, rather than both of them appearing simultaneously. We minimize a binary loss to achieve entropy minimization as Equation (7):

$$L_b = -\frac{1}{|U|} \sum_{k=1}^K \sum_{(i,j) \in M_k}^{N_{mk}} |M_{k,(i,j)} - 0.5|^2, \quad (7)$$

where $M_{k,(i,j)}$ is constrained to be close to 0 or 1.

The loss of the mask branch is thus defined as Equation (8):

$$L_m = \beta_o L_o + \beta_c L_c + \beta_b L_b, \quad (8)$$

where β_o , β_c and β_b are hyperparameters to control the scales of L_o , L_c and L_b respectively.

3.3. The Overall Loss

We reconstruct the input image by combining the outputs of the foreground, background and mask branches. The reconstruction image R_k of u_k is represented as Equation (9):

$$R_k = M_k^b \times F_k + (1 - M_k^b) \times B_k, \quad (9)$$

where M_k^b is the binary mask obtained from the mask branch, and the values are close to 0 or 1.

We minimize the reconstruction loss using Equation (10):

$$L_r = \frac{1}{|U|} \sum_{k=1}^K |u_k - R_k|^2 \quad (10)$$

The overall loss L of OLSeg is described as Equation (11):

$$L = L_r + \alpha L_g + L_m, \quad (11)$$

where α is the hyperparameter to balance the loss.

4. Experiments

In this section, we conduct extensive experiments to verify the performance of the proposed OLSeg model. We first introduce the implementation details in Section 4.1. Subsequently, we study the effect of parameter selection in Section 4.2. Next, we evaluate OLSeg for the image segmentation tasks in Sections 4.3 and 4.4 respectively. Finally, we show the ablation study in Section 4.5.

4.1. Implementation Details

4.1.1. Datasets

- The YouTube Objects dataset [51] is a large-scale dataset that includes 10 types of objects (e.g., airplane, bird, boat, car, cat, cow, dog, horse, motorbike, and train) downloaded from the YouTube website. It contains 5484 videos and a total of 571,089 video frames. The video consists of the object entering and leaving the line of sight, the object being occluded and the significant changes in the object scale and visual angle. The dataset provides ground-truth bounding boxes on the object of interest in one frame for each of 1407 video shots as the validation set.
- The Internet dataset [13] is a commonly used object segmentation dataset. There are 15,000 images downloaded from the internet. The dataset contains 4542 images of airplanes, 4347 images of cars and 6381 images of horses with the high-quality annotation masks.
- The Microsoft Research Cambridge (MSRC) dataset [52] contains 14 object classes and about 420 images with accurate pixel-wise labels. Each object in the dataset has different background, illumination and pose. This is a real-world dataset, which is often used to evaluate the image segmentation tasks.

4.1.2. Training Details

The PyTorch framework is adopted on a single GPU machine with NVIDIA TITAN V in all experiments. We train our proposed model on the YouTube Objects dataset [51], and select the appropriate parameters on the validation set. The inputs of the training phase are continuous video frames to consider the motion clue. The foreground and background branches have the same network structure. We use ResNet18 [53] as the encoder, and 4 deconvolution and convolution operations followed by 3 convolution layers as the decoder [54]. For the mask branch, we use a 5-layer U-Net [29], and each layer contains 2 convolution operations. Table 1 shows the the hyperparameter settings on the YouTube Objects dataset [51]. The encoders output different channels in the foreground and background branches. The trainable parameters in the foreground, background and mask networks are about 1.8×10^7 , 1.7×10^7 and 1.3×10^7 , respectively. The floating point operations (FLOPs) of the proposed model are about 1.8×10^{10} . The total training time takes about 19 h.

Table 1. The hyperparameter settings on the YouTube Objects dataset.

Parameters	Values
Size of input	128×128
Size of output	128×128
Optimizer	Adam
Learning rate	0.00001
Epochs	10
Batch size	16
Output channels C_f of encoder in the foreground branch	128
Output channels C_b of encoder in the background branch	32
Hyperparameter α	1.5
Hyperparameter β_o	0.15
Hyperparameter β_c	0.05
Hyperparameter β_b	1

4.1.3. Evaluation Metrics

For the image segmentation task, we use the P and J metrics as in [13]. The P refers to the ratio of the correctly labeled pixels. The J is the Jaccard similarity, which represents the intersection over union of the prediction and the ground truth. Higher values of P and J indicate better model performance. We also adopt the correct localization (CorLoc) metric following previous image segmentation works [33,34], which measures the percentage of

images that are correctly localized according to the PASCAL criterion: the intersection over union (IoU) overlap ratio of the predicted box and the ground-truth box is greater than 0.5. These metrics are commonly used for image segmentation evaluation.

4.2. Parameter Selection

We conduct detailed parameter selection experiments on the validation set of the YouTube Objects dataset [51] to study the influence of hyperparameters.

4.2.1. Encoder Output Channels in the Foreground and Background Branches

We design different output channels for the encoders in the foreground and background branches based on the complexity of the foreground and background information. The foreground branch needs more encoder output channels to deal with the changeable foreground. The experimental results of different channel combinations in the foreground and background branches are shown in Figure 3. The recovered background still contains some foreground pixels when the output channels of both encoders are 64. We further increase the channels in the foreground branch, and reduce the channels in the background branch. The appropriate channel combination of $C_f = 128, C_b = 32$ obtains a clear and clean foreground and background.

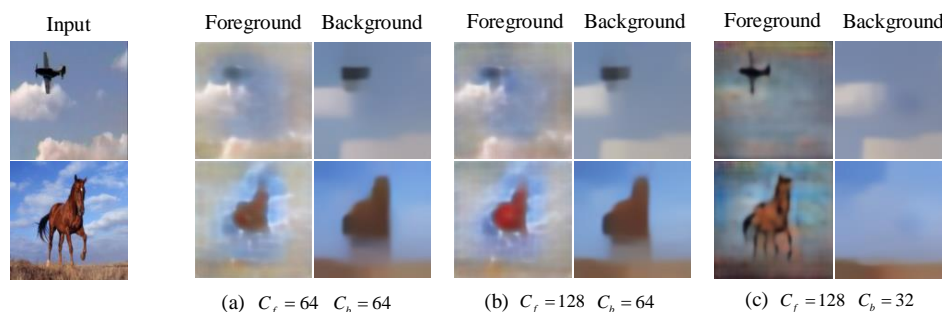


Figure 3. The experimental results of different channel combinations in the foreground and background branches. Appropriate channel combinations can achieve a cleaner foreground and background.

4.2.2. Hyperparameter α for the Gradient Loss

The separation of the foreground and background is important for reconstructing the input. The model is able to learn powerful feature representation only when the foreground and background are completely separated. We assign a hyperparameter α to balance the gradient loss in the background branch. Figure 4 shows the experimental results of different α for the gradient loss. The foreground information appears in the background when α is 0.5. We get a cleaner background with the increase of α , which is also conducive to improving the quality of segmentation mask. The hyperparameter $\alpha = 1.5$ is used for subsequent experiments.

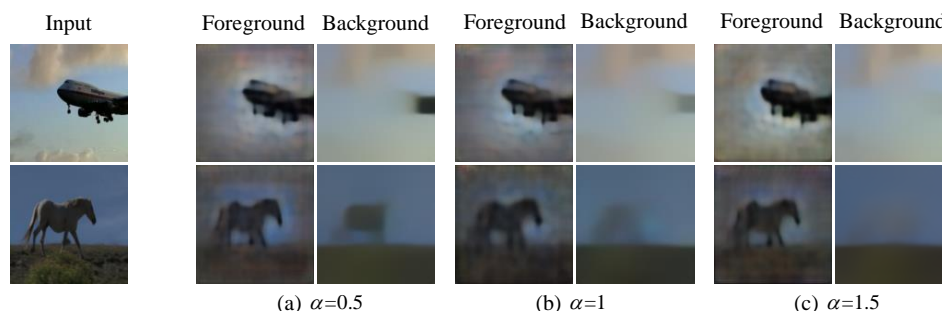


Figure 4. The experimental results of different α for the gradient loss. The foreground information is completely filtered out in the background with the increase of α .

4.2.3. Hyperparameter β_o for the Object Loss

The object loss uses a priori knowledge of object location in the mask branch. The hyperparameter β_o controls the scale of the object loss, and affects the quality of the segmentation mask. The experimental results of different β_o for the object loss are shown in Figure 5. For the mask, the object area in the red box is mapped from the original input. The information outside the red boxes cannot be completely removed when β_o is 0.05. However, a larger β_o may reduce the effect of other losses and lead to the unclear boundary of the object. We select β_o as 0.15 to obtain an accurate mask.

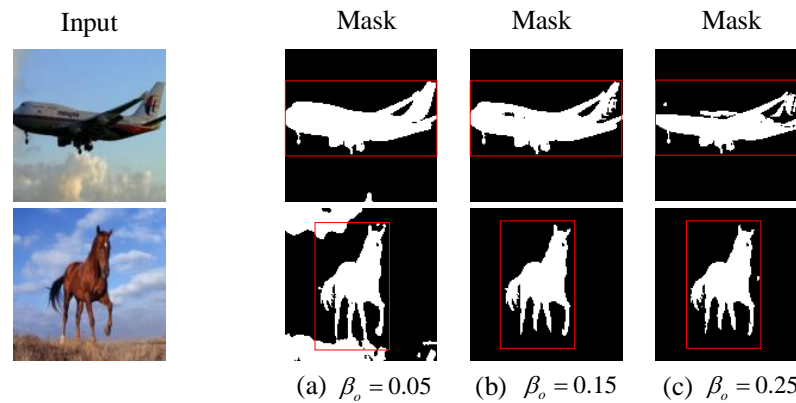


Figure 5. The experimental results of different β_o for the object loss. Appropriate β_o eliminates the information outside the red boxes and ensures the clear boundary of the object.

4.2.4. Hyperparameter β_c for the Closed Loss

The closed loss aims to promote the aggregation of the segmentation mask, and form a smooth object shape without concave holes. The hyperparameter β_c controls the proportion of the closed loss. The experimental results of different β_c for the closed loss are listed in Figure 6. The segmentation mask only retains the contour of the object and cannot form a closed area under a small value of β_c . The hyperparameter $\beta_c = 0.1$ ensures that the mask is closed and has a clear boundary.

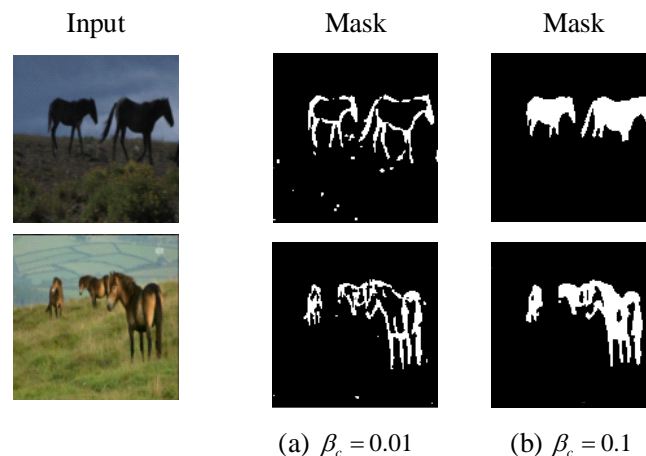


Figure 6. The experimental results of different β_c for the closed loss. The suitable β_c achieves a smooth mask without concave holes.

4.2.5. Hyperparameter β_b for the Binary Loss

The binary loss makes the segmentation mask binary to great extent. The pixels at each location of the foreground and background will contribute to the reconstruction image if the mask is not binary, which is contrary to the purpose of separation. Figure 7 shows the experimental results of different β_b for the binary loss. The experiment with $\beta_b = 1$ achieved better results.

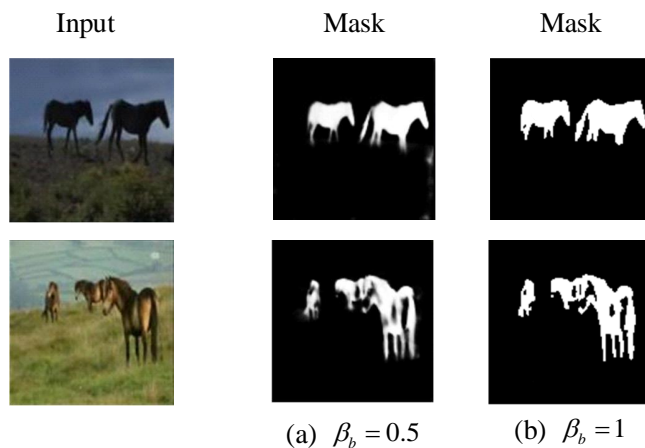


Figure 7. The experimental results of different β_b for the binary loss. The binary mask is obtained when β_b is 1.

4.2.6. Results and Visualization

We evaluated OLSeg on the validation set of the YouTube Object dataset [51]. The dataset provides ground-truth bounding boxes on the object of interest. We used the metric CorLoc as in [33,34] for quantitative analysis. For the purpose of this evaluation, we automatically fitted a bounding box to the largest connected component in the pixel-level segmentation output by our model. We compared the proposed model with the object segmentation methods [33–35] in Table 2. OLSeg outperformed the others in 6 out of 10 classes and achieved the best overall performance. However, the results for birds showed a poor performance due to the lack of constraints to account for background complexity. The foreground we defined may also contain multiple instances. For motorbikes, masks of all instances in the images were obtained, such as humans and motorbikes. In addition, we show the test complexity of these methods. The proposed model processed each image in 0.03 s, which is faster than other methods. The reason is that our model only needs to forward the U-Net [29] to obtain the segmentation mask, without relying on multiple images or complex operations. Our model takes much longer during its training phase and requires a large amount of training data.

Table 2. Comparisons of CorLoc (%) on the YouTube Object dataset. Time (s) denotes the processing speed of each image.

Methods	Airplane	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Avg	Time (s)
Stretcu et al. [33]	38.3	62.5	51.1	54.9	64.3	52.9	44.3	43.8	41.9	45.8	49.9	6.9
Papazoglou et al. [34]	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1	4
Koh et al. [35]	64.3	63.2	73.3	68.9	44.4	62.5	71.4	52.3	78.6	23.1	60.2	N/A
OLSeg	70.9	60.8	74.7	61.3	65.5	64.1	66.3	56.7	45.1	48.3	61.4	0.03

Figure 8 shows the visualization results on the validation set of the YouTube Objects dataset [51]. OLSeg extracted the segmentation mask, foreground and background from 10 classes. The generated background is smooth and contains little foreground information. We acquired the clear segmentation boundary of the mask. Multiple objects could also be accurately segmented, such as the cows and horses. However, as shown in the third row on the right of Figure 8, it is difficult to extract the mask when the color of the horse is similar to the background color. The proposed model produces high-quality segmentation results in multiple classes.

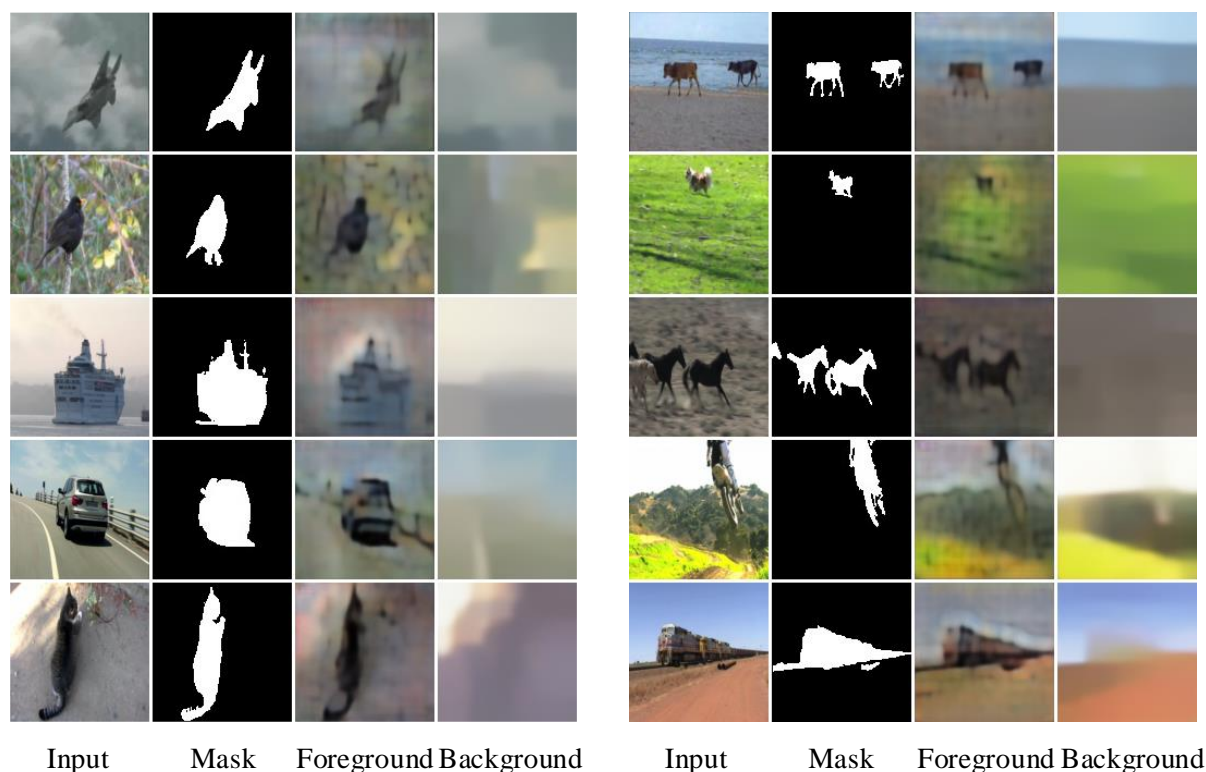


Figure 8. The visualization results on the validation set of the YouTube Objects dataset. For each class, we show the segmentation mask, foreground and background extracted by OLSeg.

4.3. Evaluation on the Internet Dataset

We ran the trained OLSeg to evaluate the performance of image segmentation on the Internet dataset [13]. We used two metrics, P (precision) and J (Jaccard similarity), and compared our results with five previously proposed co-segmentation methods [12–16]. The results of different methods are shown in Table 3. The performance on airplanes was slightly better than cars and horses. For the P of airplanes, cars and horses, OLSeg is 0.87%, 1.73% and 0.26% higher than the method in [16], respectively. The J obtained by OLSeg is also greatly improved. The comparison results show that our proposed model outperformed other methods on all three object classes.

Table 3. Comparisons of P and J (%) on the Internet dataset. The P and J denote the precision and Jaccard similarity respectively.

Methods	Airplane		Car		Horse	
	P	J	P	J	P	J
Joulin et al. [14]	49.25	15.36	58.70	37.15	63.84	30.16
Joulin et al. [15]	47.48	11.72	59.20	35.15	64.22	29.53
Kim et al. [12]	80.20	7.90	68.85	0.04	75.12	6.43
Rubinstein et al. [13]	88.04	55.81	85.38	64.42	82.81	51.65
Quan et al. [16]	91.00	56.30	88.50	66.80	89.30	58.10
OLSeg	91.87	61.50	90.23	68.45	89.56	58.72

We compared the proposed model with the methods in [12–15] to visualize the segmentation performance. The qualitative results of different methods on the Internet dataset [13] are shown in Figure 9. Compared with other methods, OLSeg achieves a clearer segmentation boundary. When the segmentation mask cannot be obtained by Kim et al. [12], the proposed model also acquires a more realistic mask. The proposed model has a powerful ability to adapt to the size of the foreground object. The airplane image with a complex back-

ground could also be well segmented. The visualization results show the effect of OLSeg.

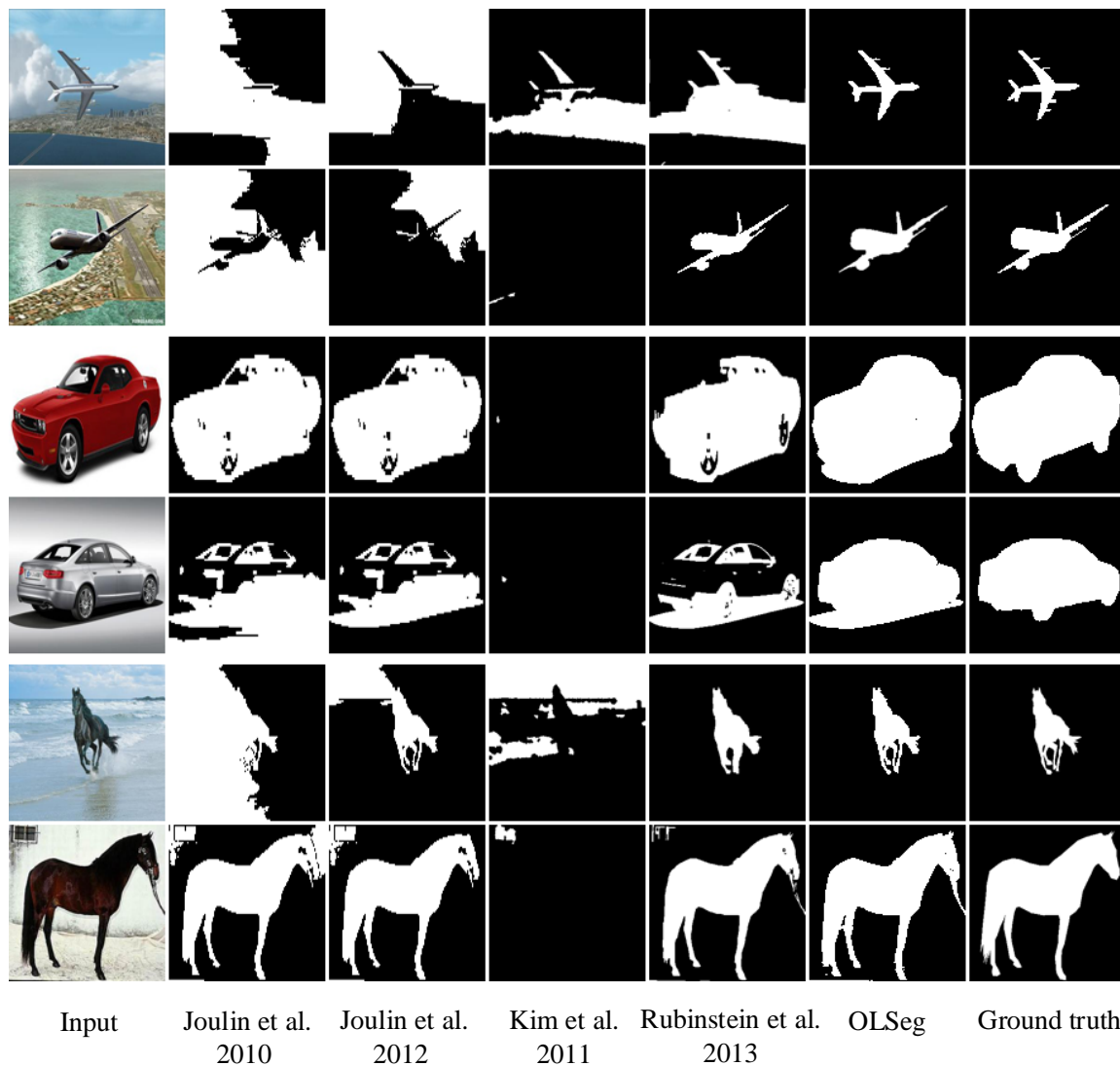


Figure 9. The qualitative results of different methods on the Internet dataset. OLSeg produces improved results compared to other methods in [12–15].

4.4. Evaluation on the MSRC Dataset

We evaluated the proposed model on the MSRC dataset [52]. We show the metrics \bar{P} (average precision) and \bar{J} (average Jaccard similarity) of our model as well as five related segmentation methods [12–16] in Table 4. OLSeg achieves a \bar{P} of 87.56% and a \bar{J} of 65.85%, which are 1.35% and 2.53% better than the method in [16]. Our overall \bar{P} and \bar{J} have higher values than the compared methods. The comparisons from Table 4 demonstrate that the proposed model produces good segmentation results on multiple object classes.

Table 4. Comparisons of \bar{P} and \bar{J} (%) on the MSRC dataset. The \bar{P} and \bar{J} denote the average precision and average Jaccard similarity, respectively.

MSRC	Joulin et al. [14]	Joulin et al. [15]	Kim et al. [12]	Rubinstein et al. [13]	Quan et al. [16]	OLSeg
\bar{P}	71.53	77.01	61.34	78.31	86.21	87.56
\bar{J}	45.27	50.97	34.48	56.69	63.32	65.85

We further report the qualitative results of different methods on the MSRC dataset [52] in Figure 10. The proposed model is able to accurately segment the foreground object despite the large variation in style, color, texture, scale and position. For the complex objects such as bikes, OLSeg can also segment their finer contours. The tree is not very distinctive from the background in terms of color, but it is still successfully segmented. The text inside the sign is well removed due to the closed loss. There is a clear visual improvement for OLSeg over other compared methods. The experimental results have demonstrated the generalization of the proposed model.

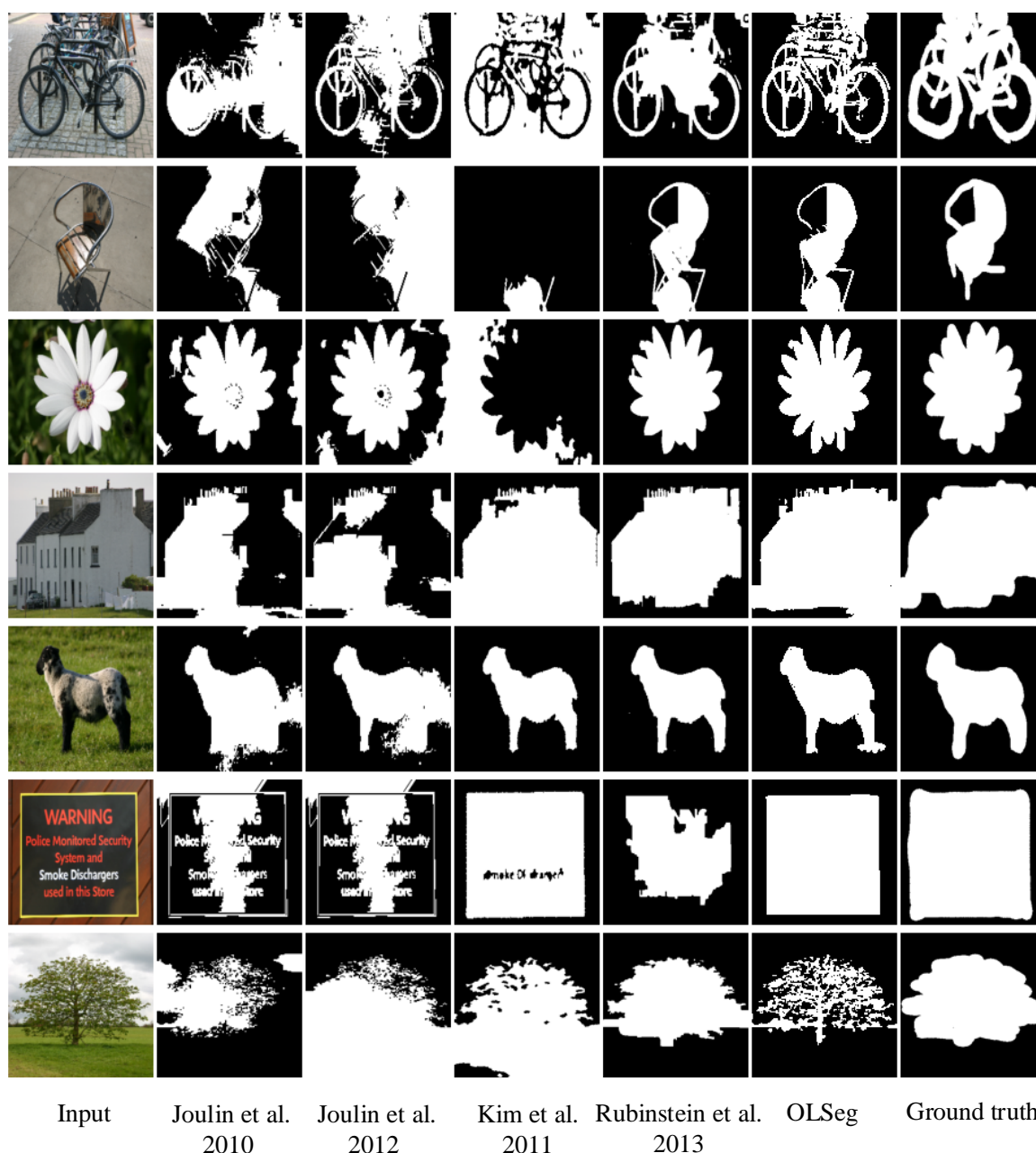


Figure 10. The qualitative results of different methods on the MSRC dataset. OLSeg achieves a clear improvement over other compared methods in [12–15].

4.5. Ablation Study

We performed an ablation study of the proposed OLSeg on the Internet and MSRC datasets, and the results of \bar{P} and \bar{J} are shown in Table 5. Specifically, the contributions of different components in OLSeg were investigated. We did not analyze the contribution

of the object loss because it is decisive in mask generation. Without the object loss, the segmentation mask will be cluttered and show a poor performance to distinguish the foreground from background. We removed the gradient loss, the closed loss and the binary loss, respectively, and observed the effect on the final results. We can see that the gradient loss contributes most to the performance. The results are reasonable since a clean background cannot be recovered without the gradient loss, which affects the quality of the mask. The closed loss plays an important role in preventing the generation of the mask with concave holes and noise. The binary loss also contributes to the final results. The ablation study from Table 5 demonstrates the effectiveness of the proposed OLSeg model.

Table 5. Ablation study of OLSeg on the Internet and MSRC datasets.

Methods	Internet		MSRC	
	\bar{P}	\bar{J}	\bar{P}	\bar{J}
OLSeg	90.55	62.89	87.56	65.85
Without the gradient loss	83.12	54.67	78.73	56.29
Without the closed loss	84.62	56.02	80.84	58.23
Without the binary loss	86.45	59.13	84.01	62.72

5. Conclusions

In this paper, we designed a pretext task of decomposing object instances in video for image segmentation, and propose an OLSeg model of three branches with bounding box prior. The pretext task benefits from the sparsity of image instances and the inter-frame structure of videos. The proposed model is trained from video and is able to capture the foreground, background and segmentation mask in a single image. The constraints in the foreground and background branches ensure the generation of the foreground and background. The mask branch uses the bounding box prior, and consists of multiple losses to produce an accurate segmentation mask for the object of interest. This is consistent with the assumption that segmentation mask could be obtained if the object location is known. The experimental results show our model achieves better segmentation performance than compared methods on various image segmentation datasets.

The parameters in OLSeg are roughly estimated by grid search in this work. In future work, how to use a more adaptive method for parameter optimization is worthy of further study. In addition, we will also explore more efficient networks and other constraints to obtain more robust mask for the object of interest in unlabeled data.

Author Contributions: Conceptualization, Y.W., H.Z. and B.Z.; methodology, Y.W., C.L. and B.D.; software, B.D. and C.L.; validation, Y.W.; formal analysis, Y.W., B.D. and C.L.; investigation, Y.W.; resources, Y.W. and C.L.; data curation, B.D.; writing—original draft preparation, Y.W.; writing—review and editing, H.Z., L.C. and Y.W.; visualization, B.D. and C.L.; supervision, H.Z. and B.Z.; project administration, Y.W.; funding acquisition, L.C., H.Z. and B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant No. 62001267, the Natural Science Foundation of Shandong Province under Grant No. ZR2020QF013, the Project funded by China Postdoctoral Science Foundation (2021M691955), and the Future Plan for Young Scholars of Shandong University, and the Fundamental of Shandong University under Grant No. 2020HW017.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [[CrossRef](#)]
2. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* **2019**, *39*, 1856–1867. [[CrossRef](#)] [[PubMed](#)]

3. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
4. Zhou, T.; Li, L.; Li, X.; Feng, C.M.; Li, J.; Shao, L. Group-wise learning for weakly supervised semantic segmentation. *IEEE Trans. Image Process.* **2021**, *31*, 799–811. [[CrossRef](#)] [[PubMed](#)]
5. Wei, Y.; Xiao, H.; Shi, H.; Jie, Z.; Feng, J.; Huang, T.S. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Juan, PR, USA, 17–19 June 2018; pp. 7268–7277.
6. Lee, J.; Kim, E.; Lee, S.; Lee, J.; Yoon, S. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5267–5276.
7. Faktor, A.; Irani, M. Co-segmentation by composition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Washington, DC, USA, 1–8 December 2013; pp. 1297–1304.
8. Fan, J.; Zhang, Z.; Song, C.; Tan, T. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Kyoto, Japan, 13–19 June 2020; pp. 4283–4292.
9. Hochbaum, D.S.; Singh, V. An efficient algorithm for co-segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 269–276.
10. Mukherjee, L.; Singh, V.; Dyer, C.R. Half-integrality based algorithms for cosegmentation of images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 2028–2035.
11. Vicente, S.; Kolmogorov, V.; Rother, C. Cosegmentation revisited: Models and optimization. In Proceedings of the European Conference on Computer Vision (ECCV), Heraklion, Greece, 5–11 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 465–479.
12. Kim, G.; Xing, E.P.; Fei-Fei, L.; Kanade, T. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 169–176.
13. Rubinstein, M.; Joulin, A.; Kopf, J.; Liu, C. Unsupervised joint object discovery and segmentation in internet images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 1939–1946.
14. Joulin, A.; Bach, F.; Ponce, J. Discriminative clustering for image co-segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1943–1950.
15. Joulin, A.; Bach, F.; Ponce, J. Multi-class cosegmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 542–549.
16. Quan, R.; Han, J.; Zhang, D.; Nie, F. Object co-segmentation via graph optimized-flexible manifold ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 687–695.
17. Zhao, D.; Ding, B.Q.; Wu, Y.L.; Chen, L.; Zhou, H.C. Unsupervised learning from videos for object discovery in single images. *Symmetry* **2021**, *13*, 38.
18. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Hypercolumns for object segmentation and fine-grained localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 447–456. [[CrossRef](#)]
19. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929.
20. Zhou, Z.H. Learnware: On the future of machine learning. *Front. Comput. Sci.* **2016**, *10*, 589–590.
21. Doersch, C.; Gupta, A.; Efros, A.A. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1422–1430. [[CrossRef](#)]
22. Noroozi, M.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 69–84.
23. Larsson, G.; Maire, M.; Shakhnarovich, G. Learning representations for automatic colorization. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 577–593.
24. Pathak, D.; Girshick, R.; Dollár, P.; Darrell, T.; Hariharan, B. Learning features by watching objects move. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2701–2710.
25. Mahendran, A.; Thevlis, J.; Vedaldi, A. Cross pixel optical-flow similarity for self-supervised learning. In *Asian Conference on Computer Vision (ACCV)*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 99–116.
26. Dosovitskiy, A.; Springenberg, J.T.; Riedmiller, M.; Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 766–774.
27. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434. [[CrossRef](#)] [[PubMed](#)]

28. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241. [[CrossRef](#)]
30. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
31. Batra, D.; Kowdle, A.; Parikh, D.; Luo, J.; Chen, T. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, 13–18 June 2010; pp. 3169–3176.
32. Jerripothula, K.R.; Cai, J.; Yuan, J. Image co-segmentation via saliency co-fusion. *IEEE Trans. Multimed.* **2016**, *18*, 1896–1909.
33. Stretcu, O.; Leordeanu, M. Multiple frames matching for object discovery in video. In *Proceedings of the British Machine Vision Conference (BMVC)*, Swansea, UK, 7–10 September 2015; Volume 1, p. 3. [[CrossRef](#)]
34. Papazoglou, A.; Ferrari, V. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Sydney, Australia, 1–8 December 2013; pp. 1777–1784.
35. Koh, Y.J.; Jang, W.D.; Kim, C.S. POD: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 1068–1076.
36. Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; Van Gool, L. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Montreal, BC, Canada, 11–17 October 2021; pp. 7303–7313.
37. Zhou, T.; Wang, S.; Zhou, Y.; Yao, Y.; Li, J.; Shao, L. Motion-attentive transition for zero-shot video object segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, 7–12 February 2020; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 13066–13073.
38. Rother, C.; Minka, T.; Blake, A.; Kolmogorov, V. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, NY, USA, 17–22 June 2006; Volume 1, pp. 993–1000.
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
40. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
41. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 8–14 September 2018; pp. 181–196.
42. Wei, X.S.; Luo, J.H.; Wu, J.; Zhou, Z.H. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Trans. Image Process.* **2017**, *26*, 2868–2881.
43. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning (ICML)*, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103. [[CrossRef](#)] [[PubMed](#)]
44. Le, Q.V. Building high-level features using large scale unsupervised learning. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 26–31 May 2013; pp. 8595–8598.
45. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661.
46. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
47. Noroozi, M.; Pirsiavash, H.; Favaro, P. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October 2017; pp. 5898–5906.
48. Gidaris, S.; Singh, P.; Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv* **2018**, arXiv:1803.07728.
49. Lee, H.Y.; Huang, J.B.; Singh, M.; Yang, M.H. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October 2017; pp. 667–676.
50. Wei, D.; Lim, J.J.; Zisserman, A.; Freeman, W.T. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8052–8060.
51. Prest, A.; Leistner, C.; Civera, J.; Schmid, C.; Ferrari, V. Learning object class detectors from weakly annotated video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 16–21 June 2012; pp. 3282–3289.
52. Shotton, J.; Winn, J.; Rother, C.; Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–15.

-
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
 54. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and checkerboard artifacts. *Distill* **2016**, *1*, e3.