applied sciences

MDPI

Article

Deep Pixel-Level Matching via Attention for Video Co-Segmentation

Junliang Li [1], Hon-Cheng Wong [1,*], Shengfeng He [2], Sio-Long Lo [1], Guifang Zhang [1] and Wenxiao Wang [1]

[1] Faculty of Information Technology, Macau University of Science and Technology, Macao 999078, China; lijunliang_sherlock@hotmail.com (J.L.); sllo@must.edu.mo (S.-L.L.); 1909853vii30003@student.must.edu.mo (G.Z.); q2368584155@gmail.com (W.W.)

[2] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China; shengfenghe7@gmail.com

* Correspondence: hcwong@ieee.org

Received: 20 January 2020; Accepted: 07 March 2020; Published: 12 March 2020

Abstract: In video object co-segmentation, methods based on patch-level matching are widely leveraged to extract the similarity between video frames. However, these methods can easily lead to pixel misclassification because they reduce the precision of pixel localization; thus, the accuracies of the segmentation results of these methods are deducted. To address this problem, we propose a framework based on deep neural networks and equipped with a new attention module, which is designed for pixel-level matching to segment the object across video frames in this paper. In this attention module, the pixel-level matching step is able to compare the feature value of each pixel from one input frame with that of each pixel from another input frame for computing the similarity between two frames. Then a features fusion step is applied to efficiently fuse the feature maps of each frame with the similarity information for generating dense attention features. Finally, an up-sampling step refines the feature maps for obtaining high quality segmentation results by using these dense attention features. The ObMiC and DAVIS 2016 datasets were utilized to train and test our framework. Experimental results show that our framework achieves higher accuracy than those of other video segmentation methods that perform well in common information extraction.

Keywords: video co-segmentation; pixel-level matching; attention

1. Introduction

Video object co-segmentation refers to the process of jointly segmenting the common objects from two or more video frames. After the segmentation, a label is assigned to each pixel in these video frames to indicate whether a pixel belongs to the common foreground object or the background. Since video object co-segmentation methods need to extract common features between two frames, the step for similarity matching between frames plays a vital role in these methods. Although some models based on patch-level matching, such as [1], have been exploited to obtain the similarity between frames, they may reduce the precision of pixel localization, leading to the pixel misclassification problem. Therefore, a pixel-level similarity matching method should be adapted for obtaining the common information between frames during the video object co-segmentation process. Inspired by the ideas in [2,3], we consider that the attention mechanism is remarkable in refining the feature maps of video frames with their common information. The reshape and inflation operations are useful when computing the similarity between two frames since these operations are able to keep two feature maps respectively belonging to the two frames in the same size. Therefore, we apply these strategies to capture the common information between two frames' feature maps in pixel-level. In addition,

an attention mechanism that is capable of refining the video frames' feature maps with their common information is important. Thus, we design a new attention module with pixel-level matching for obtaining high quality similarity features between two video frames as well as generating accurate segmentation results.

There are some methods [4–11] proposed to separate the common objects from the background information among video frames for video co-segmentation; however, most of these methods are not good at feature representation. On the other hand, deep learning models [12] have the high learning capability in feature representation that is needed for feature extraction in video co-segmentation. The approach in [13] shows that deep models are able to extract reliable features for video co-segmentation, but it is weak in common information extraction. Inspired by the ideas in [2,3], we consider that the attention mechanism is good at refining the feature maps of the video frames with their common information. Therefore, we design a new attention module with pixel-level matching for obtaining high quality similarity features between two video frames. Our attention module includes three stages: pixel-level matching step, features fusion step, and up-sampling step. In the pixel-level matching step, the distance between any two feature values from two feature maps, respectively, belonging to two video frames is computed. For achieving this purpose, using the works in [2,3], we exploit the reshape and inflation operations to keep two feature maps, respectively, belonging to the two frames in the same size so that the similarity between two frames can be captured in pixel-level. In addition, the feature fusion step efficiently transforms the spatial attention maps containing the common information from the softmax layers to the dense attention maps by fusing the spatial attention maps with two video frames' feature maps. At last, the up-sampling step utilizes the dense attention maps to further refine the feature maps from the ResNet (baseline) [14] before our attention module in our network in order to obtain high quality segmentation results.

In this paper, we present a framework containing a new attention module with pixel-level matching for improving the accuracy of the segmentation results in a video co-segmentation problem. The contributions of this letter are twofold as follows. (1) We develop a new attention module with pixel-level matching so that the similarity information between video frames can be effectively utilized for increasing the accuracy in video co-segmentation; (2) we also build a deep learning framework that integrates the new attention module for extracting accurate features and generating reliable segmentation results.

The rest of this paper is organized as follows. We review the related work in Section 2. Our framework and attention module are described in Section 3. In Section 4, we discuss the experimental results of the ablation study as well as the comparisons with state-of-the-art methods. Conclusions are given in Section 5.

2. Related Work

Video co-segmentation. Some proposal-based methods, such as the models in [6,9,15], extract the common information among the proposals from video frames, but proposal-based methods easily lead to a pixel misclassification problem. Therefore, we need to extract the common information between frames in pixel-level. On the other hand, the algorithm in [4] extracts the color, texture, and relative motion features from video frames and then applies the co-feature Gaussian mixture models to capture the common feature between video frames. A generative multi-video model [5] and a trainable apparent model [7] were presented for video co-segmentation. In addition, Wang et al. [8] proposed a framework containing clustering and Markov random field model to separate the common objects across videos; the weakly supervised VODC approach in [10] was developed for video object discovery and co-segmentation; a co-saliency strategy was applied to obtain the common objects from videos in [11]. However, all these methods cannot generate reliable features, because these methods do not have high ability in feature learning, they may not accurately capture the features belonging to the foreground object when the object's properties (i.e., size, pose and viewpoint) change in the video. Although the model in [13] applies FCN [16] to extract features from video frames, the common information is not refined by the deep model after being captured by a spatial-temporal

graphical model and the clustering algorithm; thus, the common information cannot be obtained accurately and utilized efficiently. Therefore, different from these methods, our network is built for extracting the common information in pixel-level within the deep learning architecture so that the objects' features can be refined with their common information by the convolutional layers in the deep model to enhance the accuracy of segmentation.

Attention modules. Because attention modules do well in information matching, they are widely used in image co-segmentation [17], video segmentation [3], and scene segmentation [2]. Specifically, the model in [17] adopts the attention mechanism to refine the feature maps; the DANet in [2] leverages the attention modules in two branches architecture, including a position attention module and channel attention module to segment each object in the street scene; the COSNet in [3] also presents a co-attention model within a deep model for video segmentation, but this architecture is too complex to implement since it needs a lot of memory space. Therefore, we try to develop a new deep learning architecture that includes an attention module for common information extraction.

3. Our Framework

In this section, we present the pipeline of our framework first and then show the details of our new attention module.

3.1. The Pipeline of Our Framework

Our framework is depicted in Figure 1, F_a denotes the input frame from one video, and F_b the input frame from anther video; f_a is the feature map of F_a and f_b is the feature map of F_b; S_a and S_b stand for segmentation results corresponding to F_a and F_b, respectively. As is shown in Figure 1: two frames $\{F_a, F_b\}$ from different videos are concatenated in the concat layer at first; then the concatenated data are received by the ResNet [14] as the baseline in our network. We selected the ResNet as the baseline because the networks of which baselines are ResNets, such as PSPNet [18], have achieved high accuracy in image segmentation; we selected the ResNet with 101 convolutional layers (ResNet-101) as the baseline in our network as it does not require large memory space. In the ResNet, the last convolutional layer generates a concatenated binary mask. In this binary mask, the feature map in one channel contains the features of the foreground objects, and the one in another channel contains the features of the background information. Because of the limitation of memory, we only selected the feature map with the foreground objects' features to our attention module for common information extraction; then, this feature map is split by a slice layer to generate two feature maps $\{f_a, f_b\}$ before being received by our attention module. However, the feature map with the features of the background information is sent to the silence layer so that it is not utilized by our framework anymore. At last, the slice layer, which follows our attention module, splits the concatenated segmentation result from our attention module into two segmentation results $\{S_a, S_b\}$.

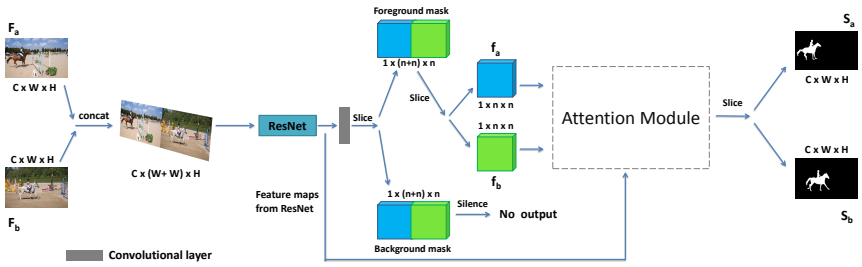

Figure 1. The pipeline of our framework with a new attention module for video co-segmentation. Two video frames $\{F_a, F_b\}$ are concatenated together in a concat layer before processing by the convolutional neural network; the concatenated data are then fed into the ResNet as the baseline for feature extraction. Then, two one-channel feature maps $\{f_a, f_b\}$ from the baseline are fed into our attention module for common information extraction. Finally, two segmentation results $\{S_a, S_b\}$ belonging to the two input frames $\{F_a, F_b\}$, respectively, are outputted.

3.2. Our Attention Module

The pipeline of our attention module is shown in Figure 2. $v_{a,1}$ and $v_{b,1}$, respectively, denote the feature maps that are reshaped and inflated from f_a and f_b; $v_{a,2}$ and $v_{b,2}$, respectively, denote the feature maps that are only inflated in channel-level from f_a and f_b. D is the spatial attention map that is concatenated after the softmax layers, V is the feature maps that is generated by concatenating $v_{a,2}$ and $v_{b,2}$ and V' is the dense attention feature which is obtained by fusing D and V. Our module receives two feature maps (f_a and f_b) belonging to two input frames and outputs the segmentation results of these two frames. Feature maps f_a and f_b contain the foreground objects' features.

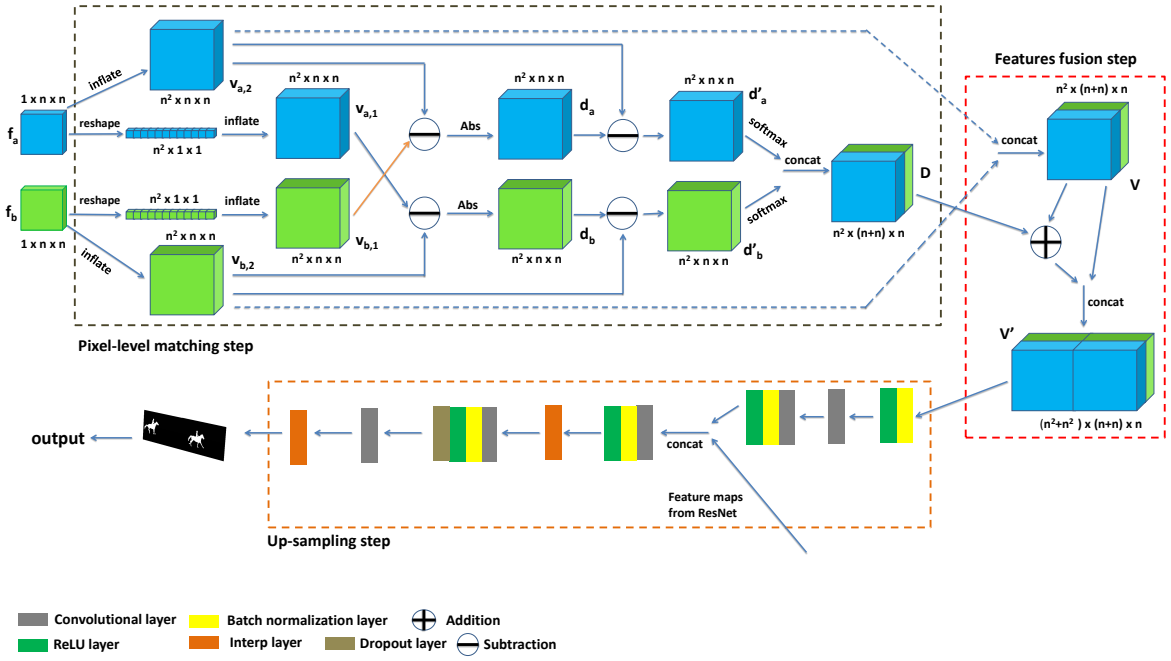

Figure 2. The pipeline of our attention module for pixel-level similarity computation. The module receives a pair of feature maps $\{f_a, f_b\}$ containing the foregrounds' features to generate feature maps $v_{a,1}$, $v_{a,2}$, $v_{b,1}$, and $v_{b,2}$ through the reshape and inflation operations. Next, a pixel-level matching step is leveraged to capture the common information from $v_{a,1}$, $v_{a,2}$, $v_{b,1}$, and $v_{b,2}$ and generate the feature map D, which is used to fuse with the concatenated feature map V and calculate the high quality dense feature V' in the features fusion step. Then, V' is fed into the up-sampling step to refine the feature maps from the baseline. Finally, the segmentation result is outputted.

Pixel-level matching and features fusion. In order to compute the difference between the feature values in f_a with the feature values in f_b, we apply the reshape layers and tile layers. These two layers transform the size of the feature maps from $1 \times n \times n$ to $n^2 \times n \times n$. Considering the case of f_a in Figure 2 as an example, its size is $1 \times n \times n$, and it is first reshaped into a new map with n^2 channels by a reshape layer; in this new map, each channel only contains one feature value from f_a. Then, the new map is inflated along the width and height into the new feature map $v_{a,1}$ with size $n^2 \times n \times n$ by two tile layers. In $v_{a,1}$, all elements' values in the matrix in the same channel are the same. Similarly, feature map $v_{b,1}$ is also generated from f_b in the same way. In addition, f_a and f_b are inflated in channel-level to generate $v_{a,2}$ and $v_{b,2}$, respectively. Therefore, $v_{a,2}$ and $v_{b,2}$ have the size as that of $v_{a,1}$ and $v_{b,1}$. With $v_{a,1}$, $v_{b,1}$, $v_{a,2}$, and $v_{b,2}$, the following equation is utilized to compute the difference between two feature maps in pixel-level:

$$\begin{cases} d_a = |v_{a,2} - v_{b,1}| \\ d_b = |v_{b,2} - v_{a,1}| \end{cases} \quad (1)$$

d_a denotes the feature maps representing the pixel-level difference between $v_{a,2}$ and $v_{b,1}$; d_b stands for the feature maps representing the pixel-level difference between $v_{b,2}$ and $v_{a,1}$. In Figure 2, we can observe that the Eltwise layers are set after $v_{a,1}$ and $v_{b,1}$ to complete the subtraction operation for computing the difference. Followed by the Eltwise layers, the AbsVal layers are used to obtain the absolute values of the results from the Eltwise layers to keep the values of these results positive. d_a and d_b are outputted from the two AbsVal layers, respectively. In d_a and d_b, the feature values representing the common feature between two objects are smaller than the ones representing the dissimilar feature since the result from the subtraction operation between two similar features is smaller than that between two dissimilar features. However, the convolutional kernel usually considers the high feature values as the features belonging to the foreground objects. To ensure that the values belonging to the common feature are higher than the ones belonging to the dissimilar feature, we generate feature maps d'_a by subtracting d_a from $v_{a,2}$ and d'_b by subtracting d_b from $v_{b,2}$. d'_a and d'_b are two feature maps that satisfy the requirement of the convolutional computation.

In order to obtain a spatial attention map, d'_a and d'_b are fed into two softmax layers, respectively, and the results from these two softmax layers are concatenated in a concat layer with a rectified linear unit (ReLU) layer [19] to generate the concatenated spatial attention map D; $v_{a,2}$ and $v_{b,2}$ are also concatenated in a concat layer to obtain the concatenated feature map V. First, D and V have an additional operation using Equation (2) and the result is then concatenated with V for further refining to generate the dense attention feature V'. This dense attention feature V' is outputted as the result of feature fusion.

$$V' = Concat(V, V + D) \tag{2}$$

Up-sampling for refining features. After receiving the dense attention feature V', following the idea in [20], we set a batch normalization (BN) layer [21] with a ReLU layer to normalize the feature values in V' for reducing internal covariate shift so that the loss function can converge easily during the training step. The result from the BN layer is fed into a convolutional layer and a convolutional block, and the high quality feature maps containing the common information outputted from the convolutional block are used to refine the feature maps from our baseline (ResNet-101). The feature maps in which the channel-size is m_1 before processing by the last convolutional layer in our baseline are exported to concatenate with the feature maps in which the channel-size is m_2 from the convolutional block in the concat layer. Through the channel-level concatenating operation, which is the same operation of concatenating V and the result of D+V in Figure 2, this concat layer provides the feature maps in which the channel-size is m_1 + m_2 to the convolutional block that follows this concat layer for refining features. Followed by the convolutional block for feature refining, an Interp layer up-samples the size of the feature maps to $N/8$ (N is the original size of the input frame), and another convolutional block with dropout layer also receives the results from the Interp layer for generating the feature maps, which are then fed into the last convolutional layer of our framework to obtain segmentation results. At last, the Interp layer that follows the last convolutional layer recovers the segmentation results into their original sizes. Two Interp layers in this step up-sample the feature maps via bilinear interpolation.

Tile layer for inflation. The tile layer is a type of layer in the Caffe framework [22] for inflating the size of feature maps. In our attention module in Figure 2, we can observe that the $n^2 \times 1 \times 1$ feature map reshaped from f_a is inflated by two tile layers to form $v_{a,1}$. Taking this process as an example, we show the details in Figure 3 and explain how the tile layer works. In the $n^2 \times 1 \times 1$ feature map, each channel only contains one element with one value (we assume that this value is x). Thus, an array ($n \times 1$) in which the elements' values are x is generated when the tile layer inflates each channel's element along the height (the step of inflating in height-level in Figure 3); then a matrix ($n \times n$) whose elements' values are x is generated when the tile layer inflates each channel's array along the width (the step of inflating in width-level in Figure 3). Moreover, when the tile layer inflates f_a ($1 \times n \times n$)

to form $v_{a,2}$ ($n^2 \times n \times n$) in channel level, it directly replicates the matrix in only one channel in f_a to all channels in $v_{a,2}$ (the step of inflating in channel-level in Figure 3).

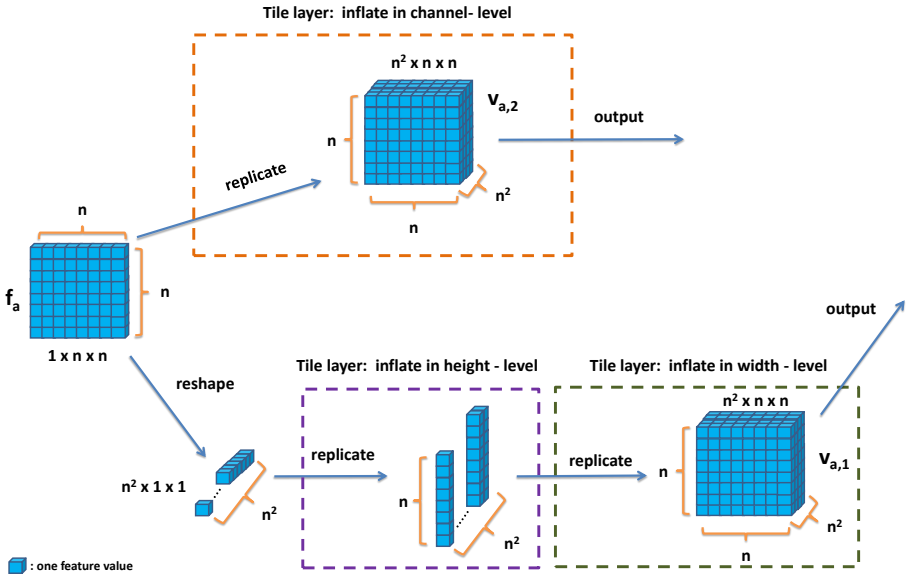

Figure 3. The process of transforming f_a to $v_{a,1}$ and $v_{a,2}$ in the tile layers.

4. Experiments

In our evaluation experiments, we evaluated our framework on two video co-segmentation datasets: ObMiC dataset [15] for ablation study and DAVIS 2016 dataset [23] for state-of-the-art methods comparison. We implemented our network in the Caffe framework [22]. In the training step, we set the learning rate to 0.001. The momentum and the weight decay are set to 0.9 and 0.0005. And the batch size is set to three during the training step. We also used the weight pre-trained by ImageNet [24] as the initial weight of our baseline during the training step in all experiments in this paper. In the testing step, two frames in an input frame-pair should come from different video sequences, but the length of each video sequence in a video-pair is different. Thus, we randomly selected several frames from a shorter video in the video and each selected frame forms frame-pairs with two or three frames obtained from the longer video. In this way, we can make sure that two input frames are from different videos. Each selected frame in the short video has several binary masks as the results, and the results of the same selected frame took an average operation to generate the final result of this selected frame.

Evaluation metric. In this paper, we employ two kinds of metrics to evaluate the segmentation performance of our network. One metrics is the pixel-wise accuracy (Pixel Acc.), the other is the intersection-over-union (IoU) accuracy [5]. On the one hand, the pixel-wise accuracy is computed in the accuracy layer in the Caffe framework, and it is defined as

$$Pixel\ Acc. = \frac{n_{Predict}}{N_{All}} \tag{3}$$

where $n_{Predict}$ denotes the number of pixels, which are assigned by the correct labels in the segmentation result, and N_{All} denotes the number of all pixels in this segmentation result. On the other hand, the IoU accuracy is a metric that is widely employed to evaluate the performance of segmentation approaches, and it is defined as

$$IoU = \frac{Seg \bigcap GT}{Seg \bigcup GT} \tag{4}$$

where *Seg* denotes the segmentation result, and *GT* denotes ground truth segmentation. mIoU denotes the average IoU scores.

4.1. Ablation Study

We conducted an ablation study on the ObMiC dataset [15] to evaluate the performance of our framework with or without the attention module. This dataset contains eight videos with 206 frames, so these eight videos are utilized for training, validation and testing.

In this study, we selected one object in each video sequence in the ObMiC dataset as our framework mainly deals with the single-object segmentation. Due to the fact that the length of the videos in the dataset is short, we selected the first frame in each video to generate the proposals containing the target as the training data and the second frame in each video to generate the proposals containing the target as the validation data; the rest of the frames in each video are considered as the testing data.

In Table 1, "first frame w/o attention" and "second frame w/o attention" indicate that the ResNet was applied as the baseline to receive two input frames and generate the segmentation results without our attention module. On the other hand, "first frame with attention" and "second frame with attention" indicate that the segmentation results belonging to the two input frames were obtained from our whole framework containing the attention module. After each training epoch, our framework performed the validation for computing the pixel-wise accuracy. The pixel-wise accuracy in each validation is computed by averaging the pixel-wise accuracies of all validation data, and we got the pixel-wise accuracy (Pixel Acc.) in Table 1 by averaging the pixel-wise accuracies obtained in the validations, which were performed after the training loss was converged. When the training step was finished, the testing data in ObMiC dataset were input into our framework for getting the segmentation results, we compared each testing frame's segmentation result with the GroundTruth to gain its own IoU score, and the mIoU accuracy was computed by averaging all testing frames' IoU scores. Both pixel-wise accuracy and mIoU accuracy in Table 1 show that the performance of our framework with the attention module is better than that of our baseline without the attention module. In other words, our attention module is able to increase the accuracy of segmentation results from the deep learning model.

Table 1. Comparison of the accuracies from our framework without (w/o) and with the attention module on the ObMiC validation dataset in pixel-wise accuracy (Pixel Acc.) and on the ObMiC testing dataset in mIoU accuracy (mIoU). Numbers in bold are the best performance.

Method	Pixel Acc. (%)	mIoU (%)
First frame w/o attention	95.99	61.20
Second frame w/o attention	97.11	
First frame with attention	**97.09**	**70.99**
Second frame with attention	**97.92**	

4.2. Comparisons with the State-of-the-Art Methods

We also compared the performance of our framework with those of the state-of-the-art methods on the DAVIS 2016 dataset [23] to prove that our framework can achieve higher accuracy in video co-segmentation.

Since the objects in an input video-pair should belong to the same category or at least have similar features so that the common information between two objects can be captured, we selected 32 videos including 2238 frames from the training data and testing data in the DAVIS 2016 dataset to form 16 video-pairs that satisfy the requirements in our evaluation experiment. Because the lengths of the videos in the DAVIS 2016 dataset are so long and the properties such as sizes and viewpoints of the objects usually change, we randomly chose six frames, which include the same object in different sizes and viewpoints in each video sequence to generate proposals containing the target as the training data. In the training step, we still used the weight pre-trained by ImageNet as the initial weight for

our baseline. In the testing step, we input the whole video sequence into our framework and the state-of-the-art methods to generate results.

In Table 2, we compare the results of our framework with those from unsupervised learning methods and supervised learning methods. The unsupervised learning methods we used for comparison include VOS [25], FOS [26], BVS [27], and DVCS [28], they can efficiently utilize the common information in the video segmentation problem; the supervised learning methods include the ResNet (the baseline in our framework), and it was also trained by the same training data that were used to train our framework. In Table 2, we can observe that the accuracies of our framework are higher than those of the unsupervised learning methods except for the testing on the videos of *Blackswan*, *Dance-Jump*, *Horsejump-High*, *Kite-Surf*, *Kite-Walk*, *Libby*, *Paragliding-Launch*, and *Stroller*. As it is shown in Figure 4, some background information in the segmentation results of the frames in the videos of *Blackswan*, *Dance-Jump*, *Horsejump-High*, and *Stroller* was recognized as the foreground object since our network is still weak in extracting the semantic information to distinguish the foreground and background. Moreover, our network is still not remarkable in capturing features of the small parts, for example, the lines in the videos of *Kite-Surf* and *Kite-Walk*, the lines and the head of the human in the video of *Paragliding-Launch*, and the legs of the dog in the video of *Libby* are not segmented in the segmentation results in Figure 4. On the contrary, compared with the ResNet (the baseline), the accuracies of our framework are better than those of the ResNet in all videos. The examples of comparing the segmentation results from our baseline (ResNet) with those from our network are shown in Figure 5, we can observe that the edges in our network's results are smoother and more details, such as the leg of the camel in the video frame, can be segmented by our network with the attention module. Overall, the mIoU score (Avg.) of our framework is higher than those of the other methods, proving that our framework can improve the accuracy of video co-segmentation.

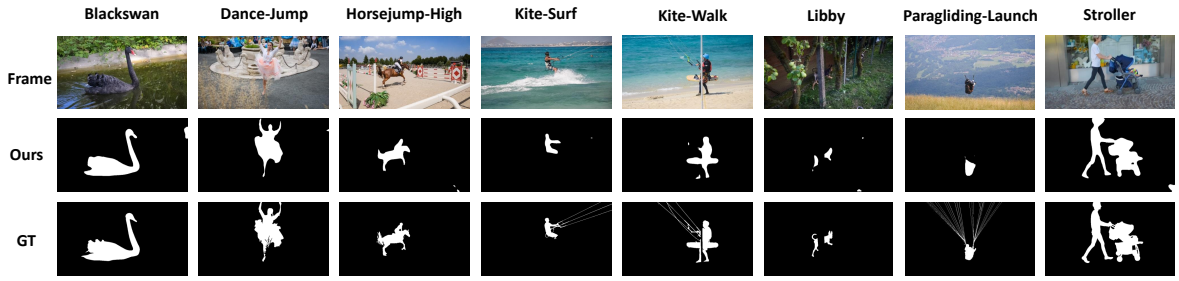

Figure 4. The cases of failure from our network.

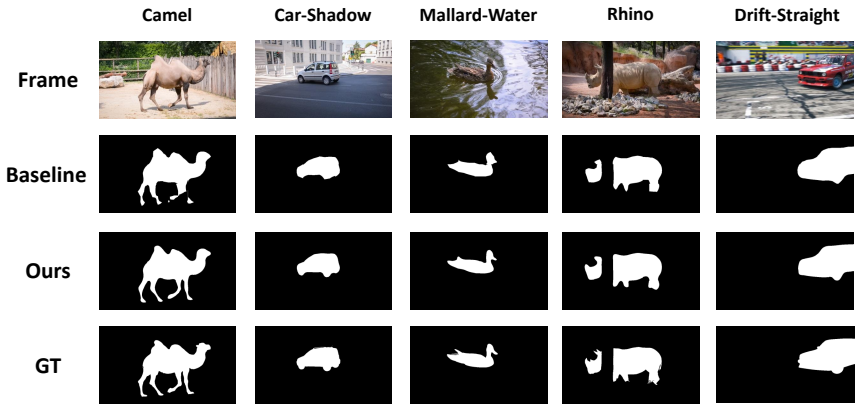

Figure 5. The examples of the segmentation results from our baseline and network.

Table 2. The average intersection-over-union (IoU) scores of our framework, baseline (ResNet), and four video object segmentation methods on the DAVIS dataset. Four video object segmentation methods are unsupervised learning methods; the baseline and our framework are supervised learning methods. Numbers in bold are the best performance of each row.

	Unsupervised Methods				**Supervised Methods**	
Video	VOS	FOS	BVS	DVCS	Baseline	Ours
Blackswan	84.2	73.2	**94.3**	91.5	87.8	92.9
Bmx-Bumps	30.9	24.1	43.4	45.2	41.6	**47.4**
Bmx-Trees	19.3	18.0	38.2	41.1	45.1	**50.9**
Breakdance	54.9	46.7	50.0	52.9	81.0	**84.4**
Breakdance-Flare	55.9	61.6	72.7	60.2	84.0	**87.9**
Camel	57.9	56.2	66.9	82.7	85.8	**91.9**
Car-Roundabout	64.0	80.8	85.1	75.2	88.7	**91.6**
Car-Shadow	58.9	69.8	57.8	75.9	92.4	**93.6**
Cows	33.7	79.1	89.5	88.7	88.1	**92.2**
Dance-Jump	74.8	59.8	**74.5**	64.2	66.6	69.9
Dance-Twirl	38.0	45.3	49.2	60.6	77.3	**81.6**
Dog	69.2	70.8	72.3	86.4	91.3	**93.6**
Drift-Chiance	18.8	66.7	3.3	71.5	79.9	**81.6**
Drift-Straight	19.4	68.3	40.2	66.6	89.4	**91.7**
Goat	70.5	55.4	66.1	79.4	85.6	**87.7**
Horsejump-High	37.0	57.8	80.1	**80.9**	71.2	79.0
Horsejump-Low	63.0	52.6	60.1	75.8	77.1	**82.2**
Kite-Surf	58.5	27.2	42.5	**68.7**	54.5	63.5
Kite-Walk	19.7	64.9	**87.0**	71.6	71.3	75.5
Libby	61.1	50.7	77.6	**79.9**	64.9	72.2
Mallard-Water	78.5	8.7	90.7	74.6	89.6	**92.3**
Motocross-Bumps	68.9	61.7	40.1	83.3	83.5	**88.1**
Motocross-Jump	28.8	60.2	34.1	68.6	86.0	**89.3**
Paragliding	86.1	72.5	87.5	90.2	87.4	**91.6**
Paragliding-Launch	55.9	50.6	**64.0**	60.0	57.0	59.7
Parkour	41.0	45.8	75.6	77.9	81.0	**84.7**
Rhino	67.5	77.6	78.2	83.8	92.6	**94.7**
Rollerblade	51.0	31.8	58.8	77.0	81.0	**85.0**
Scooter-Black	50.2	52.2	33.7	44.5	85.6	**87.2**
Scooter-Gray	36.3	32.5	50.8	66.1	80.7	**84.3**
Soapbox	75.7	41.0	78.9	79.4	81.2	**85.6**
Stroller	75.9	58.0	76.7	**87.8**	78.5	84.9
Avg.	53.3	53.8	63.1	72.3	78.4	**82.5**

5. Conclusions

In this paper, we present a framework containing a new attention module for video co-segmentation. The attention module is designed to obtain the similarity information in pixel-level from two input frames for refining the feature maps of each frame through an attention mechanism in order to generate accurate segmentation results. We conducted an ablation study on the ObMiC dataset and compared the results of our framework with those of other video segmentation methods that perform well in common information extraction on the DAVIS 2016 dataset. Experimental results show that our framework achieves higher accuracies.

Author Contributions: J.L. contributed the idea and methodology and performed the experiments; J.L., H.-C.W., S.H., and S.-L.L. wrote the original draft; G.Z. and W.W. performed some experiments; H.-C.W. supervised the project and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Development Fund of Macao SAR (0027/2018/A1 and 0016/2019/A1) and the Faculty Research Grant of Macau University of Science and Technology.

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

References

1. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2462–2470.
2. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 3146-3154.
3. Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; Porikli, F. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3623–3632.
4. Chen, D.; Chen, H.; Chang, L. Video object cosegmentation. In Proceedings of the 20th ACM International Conference on Multimedia, Nara, Japan, 29 October–2 November 2012; pp. 805–808.
5. Chiu, W.C.; Fritz, M. Multi-class video co-segmentation with a generative multi-video model. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition 2013, Portland, OR, USA, 23–28 June 2013; pp. 321–328.
6. Lou, Z.; Gevers, T. Extracting primary objects by video co-segmentation. *IEEE Trans. Multimed.* **2014**, *16*, 2110–2117. [CrossRef]
7. Rubio, J.C.; Serrat, J.; López, A. Video co-segmentation. In Proceedings of the 2nd Asian Conference on Computer Vision, Okinawa, Japan, 5–8 November 2013; pp. 13–24.
8. Wang, C.; Guo, Y.; Zhu, J.; Wang, L.; Wang, W. Video object cosegmentation via subspace clustering and quadratic pseudo-boolean optimization in an MRF framework. *IEEE Trans. Multimed.* **2014**, *16*, 903–916. [CrossRef]
9. Wang, W.; Shen, J.; Li, X.; Porikli, F. Robust video object cosegmentation. *IEEE Trans. Image Proc.* **2015**, *24*, 3137–3148. [CrossRef] [PubMed]
10. Wang, L.; Hua, G.; Sukthankar, R.; Xue, J.; Niu, Z.; Zheng, N. Video object discovery and co-segmentation with extremely weak supervision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2074–2088. [CrossRef] [PubMed]
11. Wang, W.; Shen, J.; Sun, H.; Shao, L. Video co-saliency guided co-segmentation. *IEEE Trans. Circ. Syst. Video Technol.* **2017**, *28*, 1727–1736. [CrossRef]
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2019**, *521*, 436–555. [CrossRef] [PubMed]
13. Tsai, Y. H.; Zhong, G.; Yang, M. H. Semantic co-segmentation in videos. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 760–775.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
15. Fu, H.; Xu, D.; Zhang, B.; Lin, S.; Ward, R.K. Object-based multiple foreground video co-segmentation via multi-state selection graph. *IEEE Trans. Image Proc.* **2015**, *24*, 3415–3424. [CrossRef] [PubMed]
16. Long, L.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 International Conference on Computer Vision, Araucano Park, Las Condes, Chile, 11–18 December 2015; pp. 3431–3440.
17. Zhang, C.; Lin, F.; Yao, R.; Shen. C. CAN Net: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In In Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5127–5226.
18. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia. J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
19. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Palermo, Italy, 3–5 June 2011; pp. 315–323.
20. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
21. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.

22. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM International Conference Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
23. Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 724–732.
24. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
25. Lee, Y.J.; Kim, J.; Grauman, K. Key-segments for video object segmentation. In Proceedings of the 13th International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1995–2002.
26. Papazoglou, A.; Ferrari, V. Fast object segmentation in unconstrained video. In Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition 2013, Portland, OR, USA, 23–28 June 2013; pp. 1777–1784.
27. Marki, N.; Perazzi, F.; Wang, O.; Sorkine-Hornung, A. Bilateral space video segmentation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 743–751.
28. Liu, Z.; Wang, L.; Hua, G.; Zhang, Q.; Niu, Z.; Wu, Y.; Zheng, N. Joint video object discovery and segmentation by coupled dynamic Markov networks. *IEEE Trans. Image Proc.* **2018**, *27*, 5840–5853.