electronics

MDPI

Article

Video Object Segmentation Using Multi-Scale Attention-Based Siamese Network

Zhiliang Zhu [1], Leiningxin Qiu [1], Jiaxin Wang [2], Jinquan Xiong [3,*] and Hua Peng [2]

[1] School of Software, East China Jiaotong University, Nanchang 330013, China; rj_zzl@ecjtu.edu.cn (Z.Z.); 2022218083500001@ecjtu.edu.cn (L.Q.)
[2] The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China; youdaoyzbx@163.com (J.W.); penghua@iscas.ac.cn (H.P.)
[3] Department of Mathematics and Computer Science, Nanchang Normal University, Nanchang 330032, China
* Correspondence: xjq_nnu@163.com

Abstract: Video target segmentation is a fundamental problem in computer vision that aims to segment targets from a background by learning their appearance information and movement information. In this study, a video target segmentation network based on the Siamese structure was proposed. This network has two inputs: the current video frame, used as the main input, and the adjacent frame, used as the auxiliary input. The processing modules for the inputs use the same structure, optimization strategy, and encoder weights. The input is encoded to obtain features with different resolutions, from which good target appearance features can be obtained. After processing using the encoding layer, the motion features of the target are learned using a multi-scale feature fusion decoder based on an attention mechanism. The final predicted segmentation results were calculated from a layer of decoded features. The video object segmentation framework proposed in this study achieved optimal results on CDNet2014 and FBMS-3D, with scores of 78.36 and 86.71, respectively. It outperformed the second-ranked method by 4.3 on the CDNet2014 dataset and by 0.77 on the FBMS-3D dataset. Suboptimal results were achieved on the video primary target segmentation datasets SegTrackV2 and DAVIS2016, with scores of 60.57 and 81.08, respectively.

Keywords: video object segmentation; object detection; deep learning; Siamese neural network; attention mechanism

Citation: Zhu, Z.; Qiu, L.; Wang, J.; Xiong, J.; Peng, H. Video Object Segmentation Using Multi-Scale Attention-Based Siamese Network. *Electronics* **2023**, *12*, 2890. https://doi.org/10.3390/electronics12132890

Academic Editors: Zhenghao Shi, Lifeng He, Miaohua Zhang, Jihua Zhu and Feng Zhao

Received: 17 May 2023
Revised: 20 June 2023
Accepted: 29 June 2023
Published: 30 June 2023

1. Introduction

Video object segmentation (VOS) is one of the most researched applications [1]. VOS aims to segment the foreground and background of each video image frame. It is widely used in several video-based applications [2]. The main challenges include attribute changes, occlusions, conflict between objects, background blurring, etc. [3]

In its early stages, owing to the successful research on image segmentation algorithms, video segmentation witnessed rapid development [1]. Several methods have been proposed to address the issue of object-level modeling, which can be broadly categorized into three types: background subtraction [4], motion segmentation [5], and trajectory segmentation [6]. The emergence of deep learning techniques has significantly enhanced the performance of video segmentation methods. Existing deep learning-based VOS methods can be grouped into four main types: unsupervised, semi-supervised, interactive, and language-guided supervised [3].

The fundamental architecture of VOS methods comprises two submodules, an encoder and a decoder, which perform the tasks of feature extraction and resolution restoration, respectively. Here, we use two adjacent frames as network inputs and propose a Siamese attention-based encoder–decoder network structure to extract and fuse movement and appearance features. In most cases, a multi-modal network performs worse than the best uni-modal network, owing to overfitting and a suboptimal optimization strategy [7].

Our multistream network can be viewed as a special multi-modal network with two similar inputs, outperforming the uni-modal network using a multiscale attention-based feature fusion module. The two encoders generate appearance features from the two inputs, and the decoder learns the movement features and fuses them. Without bells and whistles, our networks achieved superior performance over the state-of-the-art methods. The contributions of this study are as follows:

(1) We proposed an effective Siamese attention-based model that extracts and fuses appearance and movement features to generate foreground mask in an end-to-end manner without postprocessing steps.
(2) We demonstrated that using two adjacent frames can predict the foreground mask with higher accuracy than using optical flow as auxiliary inputs.
(3) We performed extensive and comprehensive experiments on the FBMS-3D dataset, and the experimental results confirm that the Siamese neural network and multiscale attention module function well. Moreover, the proposed methods can run in real time.
(4) The experimental results of the FBMS-3D, CDNet2014, SegTrackV2, DAVIS2016, and DAVIS2017 datasets show that our model outperforms the state-of-the-art model on the VOS dataset, and our model is comparable with the state-of-the-art model on the VOS dataset.

The remainder of this paper is organized as follows. In Section 2, related studies on video object segmentation are briefly introduced. We elaborate on the details of this method in Section 3. Section 4 provides a brief overview of the experimental datasets, evaluation metrics, and implementation details. We analyzed and compared the experimental results, and validated the effectiveness of each component of the method through ablation experiments. Finally, Section 5 summarizes this paper.

2. Related Work

2.1. Language-Guided Video Object Segmentation

Language-guided video object segmentation (LVOS) is a technique that performs VOS based on natural language expression [8]. Gavrilyuk et al. [9] segmented actors and their actions based on sentences. This was the first proposal for an LVOS [10]. Effectively integrating the feature information obtained from both sentences and videos is a key challenge in LVOS. RefVOS [11] was used to convert language features into linear projections and performed element-wise multiplication using visual features extracted by DeepLabV3. The authors of [12] proposed URVOS as a cross-modal attention module and constructed the first large-scale reference video object segmentation dataset, Refer-YouTube-VOS. Ye et al. [13] proposed three novel modules: cross-modal self-attention, gated multilevel fusion, and cross-frame self-attention. Ding et al. [14] proposed language-bridged duplex transfer to utilize language as an intermediary bridge to solve spatial misalignments or false distractors. Li et al. [15] proposed a meta-transfer module for transferring target information from the language domain to the image domain. Owing to the complexity of multi-modal tasks, an increasing number of researchers have adopted transformer-based approaches [10,16,17] for video object segmentation, which significantly reduce task complexity.

2.2. Optical Flow-Based Methods

Optical flow assumes that the target object and background have different motion patterns. Because of its pixel-level motion estimation, it is widely used in VOS [3]. Tokmakov et al. [18] first proposed the use of optical flow features to determine whether an object is in motion, thereby enabling the segmentation of moving objects. Chen et al. [19] proposed SegFlow, which enables the bidirectional propagation of object segmentation and optical flow information within a unified framework. Jain et al. [20] extracted appearance information from RGB images and motion information from optical flow images, and achieved object segmentation by fusing this information using a fusion network. Although optical flow images can provide pixel-level information, their quality is difficult

to guarantee [3]. Researchers aim to explore information within optical flow maps as much as possible. Bao et al. [21] proposed a novel spatio-temporal Markov Random Field. Xiao et al. [22] further enhanced the representation of target frames by aligning and integrating neighboring frames. Zhou et al. [23] designed a motion-attentive transition that converts appearance information into a motion-attentive representation, resulting in a closer interaction between the two; however, for static objects, it is difficult to obtain their motion information through optical flow images.

2.3. Attention Mechanism

The attention mechanism is designed to simulate a human's natural ability to focus on salient regions in complex scenes [24]. It is widely used in computer vision tasks and has achieved significant success. Attention mechanisms can be categorized into the following types: channel attention, spatial attention, temporal attention, branch attention, channel and spatial attention, and spatial and temporal attention [25]. Channel attention facilitates the adaptive adjustment of the weights assigned to different channels, where each channel represents a different object or feature [26]. Spatial attention refers to the adaptive selection of spatial regions or areas. This enables the model to focus on specific spatial locations or regions of interest in the input data [27]. Temporal attention refers to the dynamic selection mechanism of focusing on specific time steps or frames in a sequence. This enables the model to emphasize size-relevant temporal information and capture temporal dependencies over time [28]. Branch attention refers to the dynamic selection mechanism for selecting specific branches or pathways in a neural network. This enables the model to allocate resources adaptively to different branches based on their relevance or importance for the task at hand [29].

3. Proposed Model

In this study, we aim to segment the main object in a video, which requires both the appearance and the movement information of the objects in the video, and generate a foreground mask for a given sequence of video frames. Semantic features can be learned from an image classification dataset, such as ImageNet [30], using image classification networks, and moving features can be learned from the foreground dataset using two-stream networks. The weights of the current and adjacent frames are shared to construct the Siamese network. As shown in Figure 1, our model is built upon multiple-level middle-fusion rather than the late-fusion encoder–decoder architecture to share and reduce weight. The encoder learns and generates features to predict the foreground mask while reducing resolution. The attention-based decoder fuses the features at multiple levels and generates a foreground mask while resuming the resolution for pixel-wise prediction. For the encoder, we applied widely used classification networks (i.e., MobileNet, VGG, and ResNet) [31–33] to generate the appearance features. The decoder compares the features from different stages of the encoders and different input sources to generate movement features, and fuses the multi-scale appearance and movement features using the attention layer to generate the final foreground mask.

3.1. Siamese Encoder

A Siamese neural network (SNN) is a network that shares the weight of two sub-networks that process different input data. Our network is a variation of the encoder–decoder network. The Siamese encoder is composed of one main encoder for the current frame and another auxiliary encoder for the adjacent frame. The encoders all apply widely used image classification structures, such as ResNet50. In addition, we removed the last pooling layer and all fully connected layers of the image classification network to obtain a skeleton network for image classification. Through migration learning, the skeleton network is able to reuse the weights and feature encoding capabilities learned on the ImageNet dataset. There are some video target segmentation networks pre-trained on the significant target segmentation dataset, but due to some differences between video

target segmentation and significant target segmentation, the model in this paper is only pre-trained on the image classification dataset. Image classification structures generate different resolution of features, $\mathbb{F}^{\mathrm{m}} = \{F_1^m, F_2^m, F_3^m, F_4^m, F_5^m\}$ and $\mathbb{F}^{\mathrm{a}} = \{F_1^a, F_2^a, F_3^a, F_4^a, F_5^a\}$, for the main input I^m and auxiliary input I^a; these features contain appearance information that is learned from classification tasks. For our video object segmentation, this appearance information can help to filter out background objects such as trees, mountains, and sky.

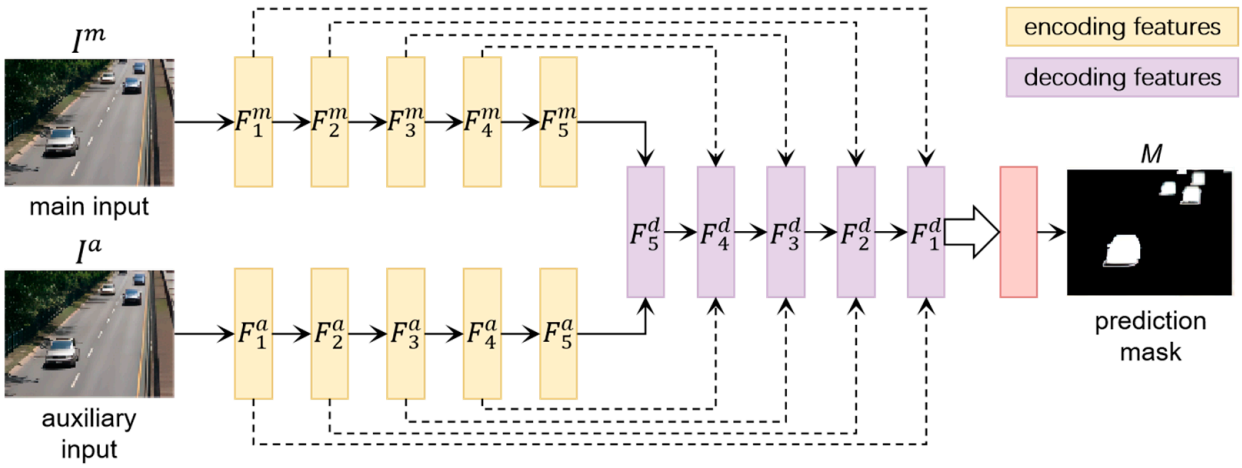

Figure 1. Our network structure when fusing from the 5th stage and upsampling at first stage. I^m and I^a are the main input (current video frame) and auxiliary input (adjacent video frame). M is the foreground mask. F^{m} and F^{a} are the encoding features for the main input and auxiliary input, and F^{d} represents the decoding features.

3.2. Multi-Scale Attention-Based Decoder

Figure 2 shows the details of the decoding layer. The decoder compares and fuses the features from the main and auxiliary encoder, and generate features $\mathbb{F}^{\mathrm{d}} = \{F_1^d, F_2^d, F_3^d, F_4^d, F_5^d\}$ in multi-scale. Let $F_i^m \in \mathbb{F}^{\mathrm{m}}$ and $F_i^a \in \mathbb{F}^{\mathrm{a}}$ be the features from the main and auxiliary encoders, and $\mathrm{i} \in \{1, 2, 3, 4, 5\}$. As show in Equation (1), F_i^c is the merged features of the decoding layers in Figure 1. *Concat* is a function to concatenate two features, *Unsample* is a function to upsample feature maps to a target size through bilinear interpolation, and *Conv* is a function that uses convolutional layers to transform features. Due to *Concat*, the number of channels of decoding features will increase continuously. In order to reduce memory consumption and the number of parameters and alleviate overfitting problems, additional convolutions are added to reduce the number of channels of features. As shown in Equation (2), when the attention mechanism is not applied, the number of channels of the decoded features is reduced by a layer of convolution operation to match the number of channels of the encoded input features.

$$F_i^c = \begin{cases} Concat\left(F_i^m, F_i^a, Unsample\left(F_{i+1}^d\right)\right), i < 5 \\ Concat\left(F_i^m, F_i^a\right), i < 5 \end{cases} \tag{1}$$

$$F_i^d = Conv(F_i^c), i \in \{1, 2, 3, 4, 5\} \tag{2}$$

In this paper, we propose an attention module that is embedded as a network layer in the original Siamese network and trained end-to-end. The corresponding attention modules are available for features of different resolutions. The input features are selectively concatenated by the selector to obtain different intermediate features $F_i^{att-main}$ and $F_i^{att-aux}$, where $F_i^{att-main}$ is the attentional feature associated with the main input and $F_i^{att-aux}$ is the attentional feature associated with the auxiliary input. Finally, the final decoding feature F_i^d is obtained after attention decoding.

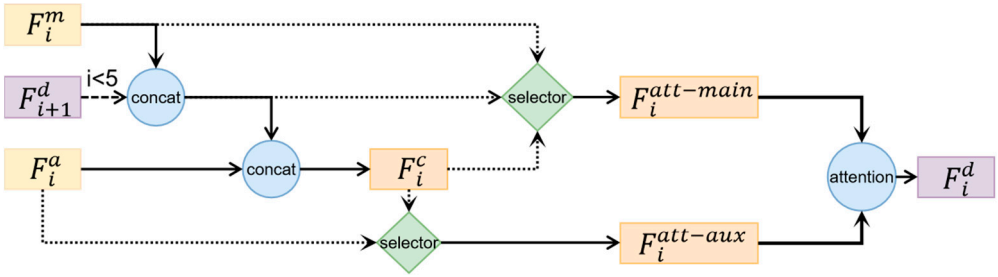

Figure 2. The details of decoding layers when the attention module is applied. The decoding layers take the main features and auxiliary features from two encoder layers, fuse features from the previous decoding layers with a different "attention" module, and output the fusion features for the next layer. The "selector" module selects one input as the output, as shown in Equations (3) and (4).

For the main input-related feature $F_i^{att-main}$, as shown in Equation (3), the selector can choose whether to use only the main feature or all the features.

$$F_i^{att-main} = \begin{cases} F_i^m, \text{use main feature and } i = 5 \\ concat\left(F_i^m, unsample\left(F_{i+1}^d\right)\right), & \text{use main feature and } i < 5 \\ F_i^c, \text{use all features} \end{cases} \tag{3}$$

For the auxiliary input-related feature $F_i^{att-aux}$, as shown in Equation (4), the selector has two options.

$$F_i^{att-aux} = \begin{cases} F_i^c, \text{use all features} \\ F_i^a, \text{use only auxiliary feature} \end{cases} \tag{4}$$

As shown in Equation (5), based on the attentional features $F_i^{att-main}$ and $F_i^{att-aux}$, the attention mechanism is applied to decode.

$$F_i^d = \varphi\left(F_i^{att-main}, F_i^{att-aux}\right) \tag{5}$$

As shown in Equation (6), the final predicted video target segmentation result can be calculated from a certain layer of decoding features F_i^d, where i $\in \{1, 2, 3, 4, 5\}$ are constants. When i is small, the corresponding decoded features have a higher resolution and contain more direct underlying features, and thus, are suitable for predicting local and fine targets. When i is larger, on the contrary, the network is more suitable for predicting global and huge targets.

$$M = f_1(I^m, I^a) = f_2\left(f_i^d\right) \tag{6}$$

3.3. Loss Function

We use softmax as the final activation function. The point-wise cross entropy loss $L_p(x, y)$ has the following format:

$$L_p(x, y) = -\log\left(\frac{\exp(p[j^*])}{\sum_j \exp(p[j])}\right) = -p[j^*] + \log\left(\sum_j \exp(p[j])\right) \tag{7}$$

where j^* represents the ground truth class index for point-wise prediction of the result. p, j^* means the ground truth is the background, and $j^* = 1$ means the ground truth is the object.

The average loss for the predicted mask loss is

$$L = \frac{1}{H \times W} \sum_{x,y}^{W,H} L_p(x, y) \tag{8}$$

where H and W are the height and width for the predicted mask; we ignore the mini-batch size here for simplicity.

4. Experiments

4.1. Datasets

As shown in Table 1, the segmentation datasets are labeled pixel-wise. For SegTrackV2, we split and sorted the dataset according to the video name order. We used the first nine videos to train and the remaining videos to validate (monkeydog, parachute, solider, and worm), but excluded penguin. Note that we removed "penguin" because it is not fully labeled. For CDNet2014, we use the first two videos to validate and the rest to train in each category. Finally, there were $2 \times 11 = 22$ videos to validate, and each category of video had at least two videos for training and two for testing. For example, we used the videos "blizzard" and "skating" to validate, and we used videos "snowFall" and "wetSnow" in the video category "badWeather" to train. For the other datasets, we followed the official dataset splitting strategy.

Table 1. Dataset overview. #GT: the total number of ground truth frames in the dataset. #Train and #Val: the number of videos for training and validation. Duration: the number of frames in videos. #Object: the total number of foreground object categories in the datasets. Input Shape: the frames' height and width in the datasets. Usage: FS means foreground segmentation, VOS means video object segmentation. Camera: "m" means moving and "f" means fixed.

Name	Year	#GT	#Train	#Val	Duration	#Object	Input Shape	Usage	Camera
SegTrackV2 [34]	2013	947	9	5	(21, 279)	24	(240×320, 360×640)	VOS	m + f
CDNet2014	2014	160,000	31	22	(900, 7000)	>4	(240×320, 486×720)	FS	f
FBMS-3D	2014	720	29	30	(19, 800)	>12	(228×350, 540×960)	FS	m
DAVIS2016	2016	3455	30	20	(25, 127)	50	(480×854, 480×1301)	VOS	m
DAVIS2017	2017	10,459	60	30	(25, 127)	376	(480×854, 480×1301)	VOS	m

SegTrackV2: The SegTrackV2 dataset [34] is a dataset containing both fixed and moving camera videos. It contains 14 video sequences with 24 objects, with 947 frames labeled. It contains the challenges of "motion blur", "appearance change", "complex deformation", "slow motion", "occlusion", and "multiple adjacent/interacting objects".

CDNet2014: The ChangeDetection.net (CDNet) dataset [35] is a fixed camera dataset expanded from CDNet2012 [36]. CDNet2014 contains 53 videos and nearly 160,000 pixel-wise annotated frames. These videos belong to 11 categories of different video object segmentation challenges. For example, videos from the thermal and turbulence categories are not captured using RGB cameras; in particular, videos in the thermal category are captured using infrared cameras. Videos from the cameraJitter and PTZ categories have small camera changes (jitter or zoom in/out). Though the CDNet2014 dataset has nearly 160,000 annotated frames, many frames have background objects only; foreground objects are rare and mainly feature humans in this dataset.

FBMS and FBMS-3D: The Freiburg–Berkeley motion segmentation dataset (FBMS) [37] is an extended moving camera dataset from the BMS dataset [38], containing 59 video sequences, with 29 for training and 30 for validation. Only some of the video frames are annotated (from 3 to 41 frames) compare to the video sequences (from 19 to 800 frames); we used FBMS-3D in our experiment (original FBMS with partially new segmentations to fix some errors [39,40]).

DAVIS2016 and DAVIS2017: The Densely Annotated Video Segmentation (DAVIS) dataset [41,42] was designed for main video object segmentation. DAVIS2016 contains either one single main object or two spatially connected objects with mask annotation for each video sequence at the pixel level, with 50 video sequences and 3455 annotated frames in total. DAVIS2017 contains 150 video sequences, with all their frames annotated with multiple object masks: 60 for training, 30 for validation, 30 for normal testing, and

the remaining 30 for specific challenge testing. It consists of 10,459 annotated frames and 376 objects.

4.2. Evaluation Metrics

We used the means of the F-Measures in three experiments as the benchmark metrics. Let V_t^{TP}, V_t^{TN}, and V_t^{FP} be the number of true positives, true negatives, and false positives for the foreground mask M_t at time t. Then, we can obtain the precision (P), recall (R), balanced F-Measure (F or F_1), and F-Measure (F_β) as show in Equation (9):

$$\begin{aligned} V^{TP} = \sum_t^N V_t^{TP} \quad V^{TN} = \sum_t^N V_t^{TN} \quad V^{FP} = \sum_t^N V_t^{FP} \\ P = \frac{V^{TP}}{V^{TP}+V^{FP}} \quad R = \frac{V^{TP}}{V^{TP}+V^{FN}} \\ F = F_1 = \frac{2P \times R}{P+R} \quad F_\beta = \frac{\left(1+\beta^2\right)P \times R}{\beta^2 P+R} \end{aligned} \tag{9}$$

4.3. Implementation Details

We used PyTorch as our deep learning framework. We adopted the Adam optimizer and set the learning rate to 10^{-4}. We trained the dataset for 30 epochs with batch sizes equal to 4. We chose ResNet50 as the default main and auxiliary encoders for our network and adopted weights pre-trained on ImageNet dataset. The first fusion stage resolution was 5 and the upsampling had the same resolution as the features in first stage.

For data processing, we resized each video frame to 224 × 224 pixels, and normalize the pixel value x to $[-1, 1]$ using the linear mapping function $x^* = \frac{2x}{255} - 1, x \in [0, 255]$. For the segmentation labels of images, we remapped the background and foreground such that 0 stood for the background and 1 stood for the foreground (we ignored the unlabeled area in the ground truth). For the DAVIS2016, DAVIS2017, FBMS-3D, and SegTrackV2 datasets, there were only background, foreground, and unlabeled area labels, but for the CDNet2014 dataset, the pixels were labeled with Static, Shadow, Non-ROI, Unknown, and Motion. Following the official standard process, we viewed Static and Shadow as the background, Unknown and Motion as the foreground, and ignores the loss for Non-ROI. We set the frame index gap to 5 for each video sequence, and generated an adjacent frame for the main frame.

4.4. Comparison with the State-of-the-Art Methods

As shown in Table 2, we used ResNet50 as the same encoder for all networks. Yakubovskiy and Pavel [43] implemented uni-modal networks (UNet, FPN, PAN, PSPNet, LinkNet, and DeepLabV3Plus). Additionally, the multi-modal version of DeepLabV3Plus fused two frames via concatenation at the first layer.

Table 2. Benchmarks. UNet [44], FPN [45], PSPNet [46], LinkNet [47], PAN [48], D3+(DeepLabV3Plus) [49], and ChangeNet [50]; "*" means that the multi-modal structure of DeepLabV3Plus is not pre-trained on the ImageNet dataset.

Network	UNet	FPN	PAN	PSPNet	LinkNet	D3+	D3+	ChangeNet	Ours	Ours-Attention
Multi-modal							√	√	√	√
SegTrackV2	55.69	63.00	58.25	36.91	37.66	60.55	39.61 *	55.41	59.82	60.57
CDNet2014	74.06	73.43	72.44	66.06	57.31	72.00	19.87 *	72.51	72.65	78.36
FBMS-3D	85.14	83.60	84.21	73.78	85.33	85.93	69.02 *	83.68	86.13	86.71
DAVIS2016	78.98	80.04	79.86	65.36	63.83	81.19	60.59 *	78.00	80.96	81.08
DAVIS2017	75.45	75.97	74.77	63.25	63.29	76.51	59.81 *	75.46	77.48	76.11

4.4.1. Performance on SegTrackV2

The SegTrackV2 dataset contains rare objects from the ImageNet dataset and both fixed and moving camera videos. It is designed for video object segmentation; therefore, certain secondary foreground objects are not labeled, which hinders the learning of moving

features. The worst and best state-of-the-art methods were PSPNet with $F_1 = 36.91\%$ and FPN with $F_1 = 63.00\%$. Our basic Siamese neural network structure achieved an upper-middle result, with $F_1 = 59.82\%$, whereas our multiscale attention-based network achieved a suboptimal result, with $F_1 = 60.87\%$.

4.4.2. Performance on CDNet2014

The main foreground targets in the CDNet2014 dataset are people and cars; however, there exist numerous possibilities for the distribution of foreground targets in space, and the scale of the foreground targets varies substantially, as shown in Figure 3. Larger foreground targets occupy 1/3 of the entire image, whereas smaller foreground targets are near points in the image. In the majority of cases, our approach yields results that are consistent with the labeling results, and the primary issue is misclassifying the video background as the foreground. This indicates that our attention approach is not simply limited to modeling and predicting people and cars, but can also learn information regarding changes in the scene and misclassify changes in the scene as changes in the foreground.

Figure 3. The prediction masks for the CDNet2014 dataset. The order of the images is as follows: main input image, our attention-based network's prediction masks, and ground truth.

4.4.3. Performance on FBMS-3D

As shown in Figure 4, the results of our attention prediction model are basically consistent with the labeled results. However, for large targets in the third video in the first row, the prediction result for vehicles is incomplete. In the second video in row 8, the prediction for people is inconsistent, which indicates that the features learned by the model lack global information and it is difficult to obtain segmentation of the whole foreground target via local motion of the non-rigid body.

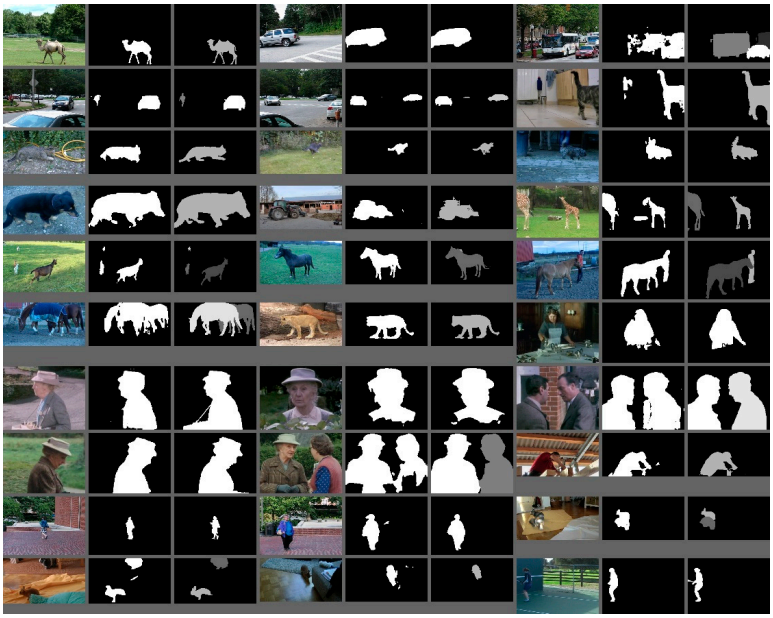

Figure 4. The prediction masks for the FBMS-3D dataset. The order of the images is as follows: main input image, our attention-based network's prediction masks, and ground truth.

4.4.4. Performance on DAVIS

The DAVIS2016 and DAVIS2017 datasets are designed for video object segmentation and they focus on salient main objects in videos. Our methods are slightly worse than the best state-of-the-art method, DeepLabV3Plus ($F_1 = 81.08\%$ compared to $F_1 = 81.19\%$ on DAVIS2016, and $F_1 = 76.11\%$ compared to $F_1 = 76.51\%$ on DAVIS2017), because our attention-based network cannot distinguish secondary objects from foreground objects well. As shown in Figure 5, the secondary objects in the video "pigs" are labeled as foreground, while they are labeled as background in the videos "car-roundabout" and "camel". The inconsistency in the datasets makes it hard to learn the fusion of appearance and movement features.

Figure 5. The failure case in the DAVIS2017 dataset. The order of the images is as follows: main input image, our attention-based network's prediction masks, and ground truth.

4.5. Ablation Study

4.5.1. Siamese Neural Network

As shown in Equation (10), the encoder f shares the weight and processes of the main input I^m and auxiliary input I^a, which enables a reduction in the model parameters. It facilitates the fast learning and optimization of the model, allowing the network to learn the generic encoding features of the input and improving the generalization ability of the model. Meanwhile, a pseudo-Siamese neural network (PSNN) does not share the weight or use different neural network structures for two inputs, which increases the parameters and requires learning how to align and fuse two different features. We changed the encoder f with MobileNetV2 [51], VGG16 [33], VGG19 [33], and ResNet50 [31] and ran the model three times for each experiment, and reported the mean metric F_1. The results are shown in Table 3. The Siamese neural network can increase F_1 for MobileNetV2 and ResNet50; however, it decreases it for VGG16 and VGG19. The standard deviation $std(F_1)$ is smaller

when using the Siamese structure because the network has fewer parameters. But $std(F_1)$ for MobileNetV2 and ResNet50 is smaller due to the improved network structure.

$$\aleph_{SNN} = (I^m, I^a) = g(f(I^m), f(I^a)) \; \aleph_{PSNN} = (I^m, I^a) = g(f_1(I^m), f_2(I^a)) \tag{10}$$

Table 3. Siamese neural network experiment on the FBMS-3D validation dataset. SNN: using or not using the Siamese neural network structure. $std(F_1)$: the standard deviation for balancing the F-Measure F_1.

Encoders	MobileNetV2	MobileNetV2	VGG16	VGG16	VGG19	VGG19	ResNet50	ResNet50
SNN		√		√		√		√
F_1 (%)	79.69	82.00 (+2.31)	83.61	82.81 (−0.80)	84.26	83.72 (−0.54)	84.47	86.14 (+1.67)
$std(F_1)$	1.46	0.31	1.46	1.38	1.25	0.80	1.19	0.29

4.5.2. Uni-Modal vs. Multi-Modal

As shown in Equation (11), the uni-modal network has only one input, while the multi-modal network takes more than one input. As shown in Table 4, this experiment analyzed the uni-modal input structure, the bimodal structure based on adjacent frames, and the bimodal structure based on optical flow. We chose ResNet50 as the encoder f for the uni-modal network or the main encoder f_1 for the multi-modal network. The main input I^m was the current frame in the video sequences, and the auxiliary input I^a could be an adjacent frame or optical flow. We used LiteFlowNet [52] to generate the optical flow, and the auxiliary encoder for optical flow was not pre-trained. We propose using light-weight network structures like MobileNetV2 and VGG [33] to learn optical flow features, as heavy-weight network structures like ResNet50 cannot improve performance. This experiment shows that uni-modal is better than multi-modal with simple fusion strategies (e.g., sum, mean and concatenation) in most cases, except when using an adjacent frame as the auxiliary input and VGG16 or VGG19 as the auxiliary encoder. For different encoders, using an adjacent frame is always better than using optical flow, except for ResNet50.

$$\aleph_{uni-modal} = g(f(I^m)) \; \aleph_{multi-modal} = g(f_1(I^m), f_2(I^a)) \tag{11}$$

Table 4. F1 multi-modal network experiment on the FBMS-3D validation dataset. a: adjacent frame. o: optical flow. -: uni-modal, without auxiliary encoder and input.

Auxiliary Encoder	MobileNetV2		VGG11		VGG16		VGG19		ResNet50		-
Auxiliary Input	a	o	a	o	a	o	a	o	a	o	-
F_1 (%)	85.62	85.54	85.78	85.53	86.49	86.06	86.59	86.12	84.47	85.03	86.34

4.5.3. Attention Experiment

In this paper, we conducted different multi-scale fusion experiments together with attentional mechanism experiments. Due to memory limitations, the use of dual attention (d) and position attention (p) could only be applied on the lowest-resolution feature maps. Both self-attention and collaborative-attention modules could improve the segmentation results, such as channel attention [53,54], spatial attention [54,55], global attention [56,57], position attention [53], and dual attention [53]. The fusion stage also influenced the results. Early-fusion is usually worse than late-fusion in accuracy. However, for video object segmentation, channel attention is better than other types of attention, as shown in Table 5. The reason may be that the channel attention module tends to learn the complementary dictionary weight ($C \times 1$) for the feature dictionary ($H \times W \times 1$), while other attention modules tend to learn the mask weight ($H \times W$), which is redundant for segmentation tasks. Note that the dual attention module is a combination of position attention and channel attention (c2), whose F_1 was 85.69%, which is less than that of position attention

(p: 86.33%), channel attention (c2: 86.71%), and no attention (n: 86.14%). We also experimented on the features to generate attention weight. When we used auxiliary features only to generate attention weights, we obtained the best result, with $F_1 = 86.71\%$, and when we used all the features, the result was $F_1 = 85.83\%$.

Table 5. Attention experiment on the FBMS-3D validation dataset. Attention type (d: dual attention, s: spatial attention, g1/g2: global attention, n: no attention, p: position attention, c1/c2: channel attention). Fusion stage (ALL: fuse features at all stages, LR: fuse features at low-resolution stage, HR: fuse features at high-resolution stage). Attention feature (ALL: focus on all features, Main: focus on main features, do not focus on auxiliary features).

Attention Type	**d [48]**	**S [49]**	**G2 [51]**	**n**	**G1 [52]**	**P [48]**	**C1 [49]**	**C2 [48]**	**C2**	**C1**	**C1**	**C1**
Fusion Stage	LR	ALL	ALL	-	ALL	LR	ALL	LR	LR	ALL	LR	HR
Attention Feature	ALL	ALL	ALL	-	ALL	ALL	ALL	ALL	Main	Main	ALL	ALL
F_1 (%)	85.69	85.71	85.85	86.14	86.32	86.33	86.48	86.71	86.57	86.20	86.41	86.69

4.5.4. Speed and Accuracy Trade-Off

As shown in Table 6, the most accurate model is ResNet50 + Siamese neural network + channel attention (c2, $F_1 = 86.71\%$), the fastest model is VGG16 + pseudo-Siamese neural network (FPS = 231), and the smallest model is MobileNetV2 + Siamese neural network (training parameters = 51 MB). MobileNetV2 has fewer training parameters (51–53 MB, 64 FPS), while its speed is slower than that of VGG16 (55–59 MB, 215–131 FPS), as depth-wise convolution and point-wise convolution cost more time for the GPU (graphics processing unit) compared to normal convolution. Note that c2 channel attention has a negligible influence on model size and inference speed, but improves performance.

Table 6. Speed and accuracy trade-off on the FBMS-3D validation dataset for different attention types and encoders. Enc: encoders for main and auxiliary encoders (M: MobileNetV2, V16: VGG16, V19:VGG19). SNN: Siamese neural network. Att: Attention type (d: dual attention, s: spatial attention, g1/g2: global attention, n: no attention, p: position attention, c1/c2: channel attention). FPS: frames per second with batch size of 4. #Par: training parameters for network (MB).

Enc	**M**	**M**	**V16**	**V16**	**V19**	**V19**	**ResNet50**								
SNN	-	√	-	√	-	√	-	√	√	√	√	√	√	√	√
Att	n	n	n	n	n	n	n	n	d	s	G1	G2	p	C1	C2
F_1 (%)	79.69	82.00	83.61	82.81	84.26	83.72	84.47	86.14	85.69	85.71	86.32	85.85	86.33	86.48	86.71
FPS	64	64	215	231	198	210	57	62	62	59	57	54	62	55	62
#Par	53	51	69	55	79	60	177	153	173	154	184	157	173	156	153

5. Conclusions

In this study, we proposed a multiscale attention-based Siamese model to learn object segmentation in videos. The multiscale attention module in our networks can learn and fuse appearance and movement features more effectively than a simple feature fusion strategy. In addition, the parameter size of the model can be effectively reduced by sharing weights. Our model can run on both fixed and moving camera videos in the wild, and our experiments demonstrate that our method achieves state-of-the-art performance on video object segmentation datasets. Our method demonstrates superior performance on fixed-camera videos compared with moving-camera videos. However, the use of current and adjacent frames as inputs has limitations. When this method is applied to segment large objects, there may be instances in which the segmentation is incomplete, resulting in missing parts of the target. Furthermore, accurately segmenting the main objects in multi-object videos can pose a challenge. In the future, we aim to explore the use of three or more video frames as inputs, employ advanced methods for motion feature extraction and fusion, and combine multi-modal input frameworks with a neural architecture search. These

approaches can enhance the performance of video segmentation methods and improve their adaptability to various video scenarios.

Author Contributions: Conceptualization, J.W. and Z.Z.; methodology, J.W.; software, H.P.; validation, J.X., H.P. and Z.Z.; writing—original draft preparation, J.W. and L.Q.; writing—review and editing, L.Q. and Z.Z.; project administration, Z.Z.; funding acquisition, J.X. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the State Key Laboratory of Computer Science Open Subject Fund (Grant No. SYSKF2102), the Key Research Project of the Education Department of Jiangxi Province (Grant No. GJJ212602), the Natural Science Foundation of Jiangxi Province (Grant No. 20224BAB202016), and the General Program of Humanities and Social Science Research of the Universities of Jiangxi Province (Grant No. TQ22104).

Data Availability Statement: Not applicable.

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

References

1. Zhou, T.F.; Porikli, F.; Crandall, D.; Gool, L.V.; Wang, W.G. A survey on deep learning technique for video segmentation. *arXiv* **2021**, arXiv:2107.01153.
2. Hou, W.J.; Qin, Z.Y.; Xi, X.M.; Lu, X.K.; Yin, Y.L. Learning disentangled representation for self-supervised video object segmentation. *Neurocomputing* **2022**, *481*, 270–280. [CrossRef]
3. Gao, M.Q.; Zheng, F.; Yu, J.J.Q.; Shan, C.F.; Ding, G.G.; Han, J.G. Deep learning for video object segmentation: A review. *Artif. Intell. Rev.* **2023**, *56*, 457–531. [CrossRef]
4. Farin, D.; de With, P.H.N.; Effelsberg, W.A. Video-object segmentation using multi-sprite background subtraction. In Proceedings of the IEEE International Conference on Multimedia and Expo, Taipei, Taiwan, 27–30 June 2004; pp. 343–346.
5. Zhuo, T.; Cheng, Z.Y.; Zhang, P.; Wong, Y.K.; Kankanhalli, M. Unsupervised online video object segmentation with motion property understanding. *IEEE Trans. Image Process.* **2019**, *29*, 237–249. [CrossRef] [PubMed]
6. Wang, W.G.; Shen, J.B.; Porikli, F.; Yang, R.G. Semi-supervised video object segmentation with super-trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 985–998. [CrossRef]
7. Wang, W.Y.; Tran, D.; Feiszli, M. What makes training multi-modal classification networks hard? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14—19 June 2020; pp. 12695–12705.
8. Luo, Z.Y.; Xiao, Y.C.; Liu, Y.; Li, S.Y.; Wang, Y.T.; Tang, Y.S.; Li, X.; Yang, Y.J. SOC: Semantic-Assisted Object Cluster for Referring Video Object Segmentation. *arXiv* **2023**, arXiv:2305.17011.
9. Gavrilyuk, K.; Ghodrati, A.; Li, Z.Y.; Snoek, G.M.C. Actor and action video segmentation from a sentence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5958–5966.
10. Liang, C.; Wang, W.G.; Zhou, T.F.; Miao, J.X.; Luo, Y.W.; Yang, Y. Local-global context aware transformer for language-guided video segmentation. *arXiv* **2022**, arXiv:2203.09773.
11. Bellver, M.; Ventura, C.; Silberer, C.; Kazakos, I.; Torres, J.; Giro-i-Nieto, X. Refvos: A closer look at referring expressions for video object segmentation. *arXiv* **2020**, arXiv:2010.00263.
12. Seo, S.; Lee, J.Y.; Han, B. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020; pp. 208–223.
13. Ye, L.W.; Rochan, M.; Liu, Z.; Zhang, X.Q.; Wang, Y. Referring segmentation in images and videos with cross-modal self-attention network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3719–3732.
14. Ding, Z.H.; Hui, T.R.; Huang, J.S.; Wei, J.Z.; Han, J.Z.; Liu, S. Language-bridged spatial-temporal interaction for referring video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4964–4973.
15. Li, D.Z.; Li, R.Q.; Wang, L.J.; Wang, Y.F.; Qi, J.Q.; Zhang, L.; Liu, T.; Xu, Q.Q.; Lu, H.C. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2022; pp. 1297–1305.
16. Botach, A.; Zheltonozhskii, E.; Baskin, C. End-to-end referring video object segmentation with multimodal transformers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4985–4995.
17. Wu, J.N.; Jiang, Y.; Sun, P.Z.; Yuan, Z.H.; Luo, P. Language as queries for referring video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4974–4984.
18. Tokmakov, P.; Alahari, K.; Schmid, C. Learning motion patterns in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3386–3394.
19. Cheng, J.C.; Tsai, Y.H.; Wang, S.J.; Yang, M.H. Segflow: Joint learning for video object segmentation and optical flow. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 686–695.

20. Dutt, J.S.; Xiong, B.; Grauman, K. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3664–3673.
21. Xiao, H.X.; Feng, J.S.; Lin, G.S.; Liu, Y.; Zhang, M.J. Monet: Deep motion exploitation for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1140–1148.
22. Bao, L.C.; Wu, B.Y.; Liu, W. CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5977–5986.
23. Zhou, T.F.; Wang, S.Z.; Zhou, Y.; Yao, Y.Z.; Li, J.W.; 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; pp. 13066–13073.
24. de Santana Correia, A.; Colombini, E.L. Attention, please! A survey of neural attention models in deep learning. *Artif. Intell. Rev.* **2022**, *55*, 6037–6124. [CrossRef]
25. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
26. Qin, Z.Q.; Zhang, P.Y.; Wu, F.; Li, X. Fcanet: Frequency channel attention networks. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 783–792.
27. Chu, X.X.; Tian, Z.; Wang, Y.Q.; Zhang, B.; Ren, H.B.; Wei, X.L.; Xia, H.X.; Shen, C.H. Twins: Revisiting the design of spatial attention in vision transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 9355–9366.
28. Coull, J.T. fMRI studies of temporal attention: Allocating attention within, or towards, time. *Cogn. Brain Res.* **2004**, *21*, 216–226. [CrossRef]
29. Shi, X.M.; Qi, H.; Shen, Y.M.; Wu, G.Z.; Yin, B.C. A spatial–temporal attention approach for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4909–4918. [CrossRef]
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.H.; Karpahty, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
31. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
32. Howard, A.G.; Zhu, M.L.; Chen, B.; Kalenichenko, D.; Wang, W.J.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Li, F.X.; Kim, T.; Humayun, A.; Tsai, D.; Rehg, J.M. Video segmentation by tracking many figure-ground segments. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2192–2199.
35. Wang, Y.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 387–394.
36. Goyette, N.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Ishwar, P. Changedetection. net: A new change detection benchmark dataset. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 1–8.
37. Ochs, P.; Malik, J.; Brox, T. Segmentation of moving objects by long term video analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1187–1200. [CrossRef]
38. Brox, T.; Malik, J. Object segmentation by long term analysis of point trajectories. In Proceedings of the European Conference on Computer Vision, Heraklion, Greece, 5–11 September 2010; pp. 282–295.
39. Bideau, P.; Learned-Miller, E. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 10–16 October 2016; pp. 433–449.
40. Bideau, P.; Learned-Miller, E. A detailed rubric for motion segmentation. *arXiv* **2016**, arXiv:1610.10033.
41. Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Gool, L.V.; Gross, M.; Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 724–732.
42. Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbelaez, P.; Sorkine-Hornung, A.; Fool, L.V. The 2017 davis challenge on video object segmentation. *arXiv* **2017**, arXiv:1704.00675.
43. Segmentation Models Pytorch. 2020. Available online: https://github.com/qubvel/segmentation_model (accessed on 16 December 2020).
44. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
45. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
46. Zhao, H.S.; Shi, J.P.; Qi, X.J.; Wang, X.G.; Jia, J.Y. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.

47. Chaurasia, A.; Culurciello, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In Proceedings of the IEEE Visual Communications and Image Processing, Petersburg, VA, USA, 10–13 December 2017; pp. 1–4.
48. Li, H.C.; Xiong, P.F.; An, J.; Wang, L.X. Pyramid attention network for semantic segmentation. *arXiv* **2018**, arXiv:1805.10180.
49. Chen, L.C.; Zhu, Y.K.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
50. Varghese, A.; Gubbi, J.; Ramaswamy, A.; Balamuralidhar, P. ChangeNet: A deep learning architecture for visual change detection. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
51. Sandler, M.; Howard, A.; Zhu, M.L.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
52. Hui, T.W.; Tang, X.; Loy, C.C. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8981–8989.
53. Fu, J.; Liu, J.; Tian, H.J.; Li, Y.; Bao, Y.J.; Fang, Z.W.; Lu, H.Q. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
54. Chen, L.; Zhang, H.W.; Xiao, J.; Nie, L.Q.; Shao, J.; Liu, W.; Chua, T.S. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5659–5667.
55. Zhao, H.S.; Zhang, Y.; Liu, S.; Shi, J.P.; Loy, C.C.; Lin, D.H.; Jia, J.Y. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 267–283.
56. Liu, M.; Yin, H. Cross attention network for semantic segmentation. In Proceedings of the IEEE International Conference on Image Processing, Taipei, Taiwan, 22–25 September 2019; pp. 2434–2438.
57. Liu, W.; Rabinovich, A.; Berg, A.C. Parsenet: Looking wider to see better. *arXiv* **2015**, arXiv:1506.04579.