APSIPA Transactions on Signal and Information Processing, 2023, 12, e2

Overview Paper

A Survey of Efficient Deep Learning Models for Moving Object Segmentation

Bingxin Hou[1], Ying Liu[1], Nam Ling[1*], Yongxiong Ren[2] and Lingzhi Liu[2]

[1]*Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA 95053*
[2]*Kwai, Inc., Palo Alto, CA, USA 94306*

ABSTRACT

Moving object segmentation (MOS) is the process of identifying dynamic objects from video frames, such as moving vehicles or pedestrians, while discarding the background. It plays an essential role in many real-world applications such as autonomous driving, mobile robots, and surveillance systems. With the availability of a huge amount of data and the development of powerful computing infrastructure, deep learning-based methods have shown remarkable improvements in MOS tasks. However, as the dimension of data becomes higher and the network architecture becomes more complicated, deep learning-based MOS models are computationally intensive, which limits their deployment on resource-constrained devices and in delay-sensitive applications. Therefore, more research started to develop fast and lightweight models. This paper aims to provide a comprehensive review of deep learning-based MOS models, with a focus on efficient model design techniques. We summarize a variety of MOS datasets, and conduct a thorough review of segmentation accuracy metrics and model efficiency metrics. Most importantly, we compare the performance of efficient

[*] Corresponding author: Nam Ling, nling@scu.edu. This work was supported in part by Kwai, Inc. under the Kwai, Inc. grant and in part by the National Science Foundation under Grant ECCS-2138635.

Received 29 June 2022; Revised 22 September 2022
ISSN 2048-7703; DOI 10.1561/116.00000140

MOS models on popular datasets, identify competitive models and analyze their essential techniques. Finally, we point out existing challenges and present future research directions.

Keywords: Background subtraction, change detection, deep learning, efficient model design, lightweight model, moving object segmentation, real-time processing, video object segmentation.

1 Introduction

Moving object segmentation (MOS) [59] is a fundamental task in computer vision. It is the process of extracting dynamic foreground content from video frames, such as moving vehicles or pedestrians, while discarding the non-moving background. MOS is used as a critical video pre-processing step followed by higher-level tasks such as traffic monitoring [39], target re-identification [251], action recognition [261], human detection [244], and object tracking [256]. However, with the increasing amount of produced visual data and computation-resource-limited platforms such as self-driving cars, wireless surveillance cameras, and navigation robots, it becomes quite crucial and challenging to process a large amount of video data in a timely fashion. In recent years, the research attention has moved toward developing more cost-efficient MOS models [36] by reducing the model size and computational complexity, increasing the inference speed, while achieving acceptable segmentation accuracy.

1.1 Background and Scope

In general, MOS tasks can be categorized as object-level segmentation and instance-level segmentation [69]. As illustrated in Figure 1(a), the goal of object-level segmentation is to detect all moving objects as the foreground and to generate a pixel-level binary segmentation mask, no matter the video scene has a single instance or multiple instances. For example, a horse and the person riding the horse are both detected as foreground, without being distinguished as two different instances. In contrast, as illustrated in Figure 1(b), instance-level segmentation not only detects the foreground moving objects, but also assigns each instance a different label, which can be utilized for object identification and tracking over time. For example, a man and two dogs are all detected as foreground, and these three instances are each assigned a different label. This shall be differentiated from video semantic segmentation (VSS), where instances having the same semantic meaning are assigned the same class label. In that case, the two dogs in Figure 1(b) will be assigned a single class label.

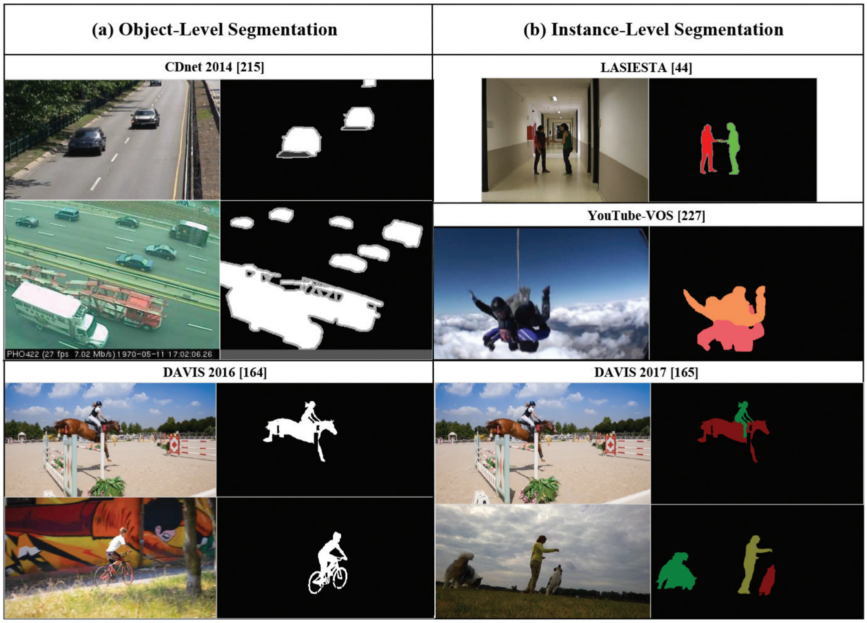

(a) Object-Level Segmentation
(b) Instance-Level Segmentation
CDnet 2014 [215]
LASIESTA [44]
YouTube-VOS [227]
PHO422 (27 fps 7.02 Mb/s) 1970-05-11 17:02:06.26
DAVIS 2016 [164]
DAVIS 2017 [165]

Figure 1: Sample frames and ground-truth segmentation masks of (a) object-level segmentation and (b) instance-level segmentation tasks. The frames are selected from popular datasets: CDnet 2014 [215], LASIESTA [44], YouTube-VOS [227], DAVIS 2016 [164], and DAVIS 2017 [165].

There are also several frequently used terms related to MOS: moving object detection (MOD), video object segmentation (VOS), and video salient object detection (VSOD). In the following, we will explain these terms, and define the discussion scope of MOS in this survey.

1.1.1 Moving Object Detection

Moving object detection (MOD) [6, 27, 91, 95, 169, 174, 235], also known as background subtraction [19, 59, 64, 89, 124, 180], is a traditional computer vision task that detects the changing foreground objects from static or dynamic background for video surveillance or anomaly detection purposes [59]. It can be viewed as an object-level segmentation task, which outputs a binary foreground mask without distinguishing different instances, as shown in Figure 1(a). Besides, dynamic elements in the background such as waving tree leaves, flowing water, snowing weather, are not the target objects to be detected.

1.1.2 Video Object Segmentation

In recent years, with the advances in deep learning and machine intelligence, the task of MOD is naturally extended to video object segmentation (VOS), to meet the requirement of emerging applications such as intelligence video editing, scene understanding, and autonomous driving. VOS refers to the task of segmenting main moving objects from a video sequence [165] which capture human attention. The early VOS task in the DAVIS 2016 challenge [164] is object-level segmentation, which only requires to segment one object or two spatially connected objects from a video sequence, generating a binary segmentation mask, as shown in Figure 1(a) DAVIS 2016 example. Later, multi-object VOS was introduced with the launch of the DAVIS 2017 challenge [165]. This task setting is more challenging as it is instance-level segmentation as shown in Figure 1(b), which requires not only separating the main moving objects from the background, but also discriminating different instances, generating a multi-class segmentation mask.

Depending on the level of human intervention during the segmentation process, VOS tasks can be divided into three categories: unsupervised VOS, semi-supervised VOS, and interactive VOS. In unsupervised VOS, human does not interact with the algorithm to obtain the segmentation results. In semi-supervised VOS [165], the algorithm is given a video sequence and the first-frame segmentation mask of the target objects, then the algorithm outputs the masks of those objects in remaining frames. Finally, interactive VOS assumes the user gives iterative refinement inputs to the algorithm, for example, in the form of a scribble, to segment the objects of interest. Methods have to produce a segmentation mask for that object in all the frames of a video sequence taking into account all the user interactions.

1.1.3 Video Salient Object Detection

A concept similar to unsupervised VOS is video salient object detection (VSOD) [67, 181]. VSOD aims also at finding objects in video frames that mostly attract human attention. However, it produces a sequence of probability maps that indicate the likelihood of each pixel belonging to most visually important objects. In contrast, unsupervised VOS outputs either a binary segmentation mask for single-instance case, or a multi-class segmentation mask for multi-instance case.

1.1.4 Moving Object Segmentation

In this survey, we mainly review state-of-the-art models for MOD, unsupervised VOS and semi-supervised VOS tasks. It is noteworthy that MOD can be considered as object-level unsupervised VOS. Since these three tasks all aim at segmenting main moving objects from videos, we adopt the term moving object segmentation (MOS), which differentiates our discussion scope from the original scope of VOS that also includes interactive VOS.

1.2 Overview of Existing MOS Approaches

Techniques used to segment video moving objects can be categorized as traditional approaches and deep learning-based approaches. Traditional approaches [11, 14, 15, 20, 28, 29, 39, 53, 68, 70, 72, 84–86, 117, 131, 145, 182, 185, 208, 211, 255, 256, 259, 262] do not require ground-truth labels for algorithm development. They usually include two components: background modeling and pixel classification. However, traditional methods meet difficulties when they are applied to complex scenarios, such as videos with dynamic backgrounds, shadows, illumination changes, and night scenes.

With the development of powerful computing infrastructure and the availability of a huge amount of data, deep neural networks (DNNs) have shown remarkable success in MOS tasks. Existing DNN-based MOS models are mostly supervised approaches based on 2D convolutional neural networks (CNNs) [8, 21, 35, 41, 42, 79–81, 96, 107–110, 133, 142, 160, 161, 172, 193, 194, 216, 248, 249], 3D CNNs [2, 58, 78, 127, 128, 171, 217], or generative adversarial networks (GANs) [9, 10, 158, 159, 186, 253, 254]. They demonstrated that DNNs can automatically extract spatial low-, mid-, and high-level features as well as temporal features, which turn out to be very helpful in MOS problems. Some methods combine both traditional methods and deep learning methods to get better performance such as RT-SBS [43], GraphMOS [63], and MotionRec [129].

Besides, semi-supervised video object segmentation (VOS) has emer-ged in recent years, in which the ground-truth segmentation mask of the first frame is provided during test time. This type of approach can be further categorized as (1) online fine-tuning-based methods, such as Meta-Learning [224], e-OSVOS [137], PreMOVS [123], etc.; (2) propagation-based methods, such as CTN [82], MaskTrack [163], FAVOS [37], AGSS [112], AGAME [88], DTN [245], SAT [32], Fasttmu [188], AOT-T [232], etc.; and (3) template-based methods, such as SwiftNet [206], STM [151], PLM [237], RANet [219], FRTM [170], TVOS [247], TTVOS [154], GC [105], LWL [13], MSN [222], RMNet [225], SiamMask [210], DDEAL [236], PiWiVOS [152], etc. The difference is that methods in category (1) need to re-train or fine-tune models online during the inference stage, and methods in categories (2) and (3) are usually used together and they do not need online model fine-tuning.

While existing DNN models provide superior MOS accuracy, they suffer from computationally expensive and memory-intensive problems. As the depth of neural network increases, the model size and computational complexity dramatically increase, making it challenging to apply these models to real-world scenarios, such as self-driving cars, robotics, and augmented reality. These tasks are typically deployed on mobile and embedded devices with limited memory and computing resources. Besides, they are usually latency-sensitive and need to be executed in a timely manner. High-complexity deep learning models cannot meet these requirements. Therefore, the research community now focuses more on designing cost-efficient deep MOS networks which have smaller model size, can achieve a faster inference speed, while maintaining high segmentation accuracy, so that they are suitable for mobile and embedded environments.

In this survey paper, we review the most recent advances in deep learning-based MOS techniques. We focus on fast and lightweight models and their performance on most popular MOS datasets. Besides, we provide a comprehensive list of existing MOS datasets and summarize performance evaluation metrics from both model accuracy and efficiency perspectives.

1.3 Previous Surveys

Table 1 lists existing surveys [6, 18, 19, 27, 59, 64, 89, 91, 95, 124, 132, 169, 174, 180, 213, 220, 234, 235] on MOS. Six of them [19, 27, 64, 132, 213, 234] reviewed recent deep learning-based MOS methods, and all the other surveys only focused on traditional methods.

Some surveys on deep learning-based MOS [19, 64] reviewed network architectures from background generation and background subtraction perspectives, and provided visual and quantitative performance evaluation on the CDnet 2014 dataset [215]. Methods about moving objects detection with a moving camera were also discussed [27], where a small number of deep learning models were briefly introduced. In [132], a detailed review of deep learning-based MOS model designs and evaluation settings is provided, such as different ways of splitting the training and test sets. It also provided performance evaluation on the CDnet 2014 dataset.

However, none of the aforementioned survey papers discussed techniques for efficient MOS model design for resource-constrained edge devices or delay-sensitive applications. They did not investigate fast or lightweight models, and did not compare inference speeds among different models. Besides, they did not discuss the newest MOS models developed in 2022, neither did they provide experimental studies on the recent popular MOS datasets such as DAVIS 2016 [164], DAVIS 2017 [165], and YouTube-VOS [227]. Moreover, the performance evaluation metrics introduced in existing surveys are limited. Recently, a comprehensive review on video object segmentation (VOS) models was conducted [213], including VOS datasets and VOS-oriented performance evaluation metrics. Nevertheless, it did not discuss fast and lightweight models, therefore it cannot provide insights for designing efficient

Table 1: The summary of previous reviews on MOS.

No.	Year	Title	Category	Author	Venue
1	2014	Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview [18]	Non-Deep Learning	Thierry Bouwmans	Compure Science Review
2	2014	A Comprehensive Review of Background Subtraction Algorithms Evaluated with Synthetic and Real Videos [180]	Non-Deep Learning	Andrews Sobrala, Antoine Vacavant	Computer Visoin Image Understanding
3	2015	Moving Object Detection: Review of Recent Research Trends [95]	Non-Deep Learning	Jaya S. Kulchandani, Kruti J. Dangarwala	International Conference on Pervasive Computing (ICPC)
4	2015	Review: Moving Object Detection Techniques [6]	Non-Deep Learning	Amandeep, Er. Monica Goyal	International Journal of Computer Science and Mobile Computing
5	2017	Review on Moving Object Detection in Video Surveillance [91]	Non-Deep Learning	Aqsa Khan, Nitin J. Janwe	International Journal of Advanced Research in Computer and Communication Engineering
6	2018	A Survey on Moving Object Detection and Tracking Based On Background Subtraction [174]	Non-Deep Learning	Rahul Sharma, Subham Gupta	The Oxford Journal of Intelligent Decision and Data Science

Table 1: Continued.

No.	Year	Title	Category	Author	Venue
7	2018	New Trends on Moving Object Detection in Video Images Captured by a Moving Camera: A Survey [235]	Non-Deep Learning	Mehran Yazdi, Thierry Bouwmans	Computer Science Review
8	2019	A Review on Moving Object Detection and Tracking Methods in Video [220]	Non-Deep Learning	Sarika S. Wangulkar, Roshani Talmale, Rajesh Babu	International Journal of Scientific Research in Science, Engineering and Technology IJSRSET
9	2019	Moving Object Detection Under Sudden Change of Illumination: A Review [169]	Non-Deep Learning	Rajib Debnath, Mrinal Kanti Bhowmik	International Journal of Computational Intelligence & IoT
10	2019	Background Subtraction for Moving Object Detection in RGBD data: A Survey [124]	Non-Deep Learning	Lucia Maddalena, Alfredo Petrosino	Journal of Imaging
11	2019	A Comprehensive Survey of Video Datasets for Background Subtraction [89]	Non-Deep Learning	Rudrika Kalsotra, Sakshi Arora	IEEE Access
12	2019	Background Subtraction in Real Applications: Challenges, Current Models and Future Directions [59]	Non-Deep Learning	Belmar Garcia-Garcia, Thierry Bouwmans, Alberto JorgeRosales Silva	Computer Science Review
13	2019	Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation [19]	**Deep Learning**	Thierry Bouwmans, Sajid Javed, Maryam Sultana, Soon Ki Jung	Neural Networks

Table 1: Continued.

No.	Year	Title	Category	Author	Venue
14	2020	Moving Objects Detection with a Moving Camera: A Comprehensive Review [27]	**Deep Learning**	Marie-Neige Chapel, Thierry Bouwmans	Computer Science Review
15	2020	Deep Learning Based Background Subtraction: A Systematic Survey [64]	**Deep Learning**	Jhony H. Giraldo, Huu Ton Le, Thierry Bouwmans	Handbook of Pattern Recognition and Computer Vision
16	2020	Video Object Segmentation and Tracking: A Survey [234]	**Deep Learning**	Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, Yong Zhou	ACM Trans. Intell. Syst. Technol
17	2021	An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs [132]	**Deep Learning**	Murari Mandal, Santosh Kumar Vipparthi	IEEE Transactions on Intelligent Transportation Systems
18	2021	A Survey on Deep Learning Technique for Video Segmentation [213]	**Deep Learning**	Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, Luc Van Gool	IEEE Transactions on Pattern Analysis and Machine Intelligence
19	**Ours**	**A Survey of Efficient Deep Learning Models for Moving Object Segmentation**	**Deep Learning**	**-**	**-**

models. Another survey on VOS [234] provided very limited discussion on efficient model design techniques. Besides, it did not present experimental results, and there were no comparison studies and analysis of model performance.

1.4 Our Contributions

In this work, we systematically review the most recent advances in deep learning-based MOS, with a focus on efficient (fast and lightweight) MOS models. Our major contributions are the following:

- For the first time in the literature, we provide a comprehensive review of efficient MOS model design techniques which play an important role in mobile and embedded device applications and in delay-sensitive scenarios.
- We discuss existing MOS datasets from the perspective of scene categories. We not only cover traditional MOD datasets such as CDnet 2014 and BMC [201] datasets, but also extend our discussion to newer VOS datasets such as DAVIS 2016, DAVIS 2017, and YouTube-VOS, which were designed for more complex scenes and higher resolutions.
- We conduct a thorough review of model performance evaluation metrics, including segmentation accuracy metrics for object-level and instance-level segmentation as well as for salient object segmentation, and model efficiency evaluation metrics such as inference speed, model size, trainable parameters, and computational complexity. In contrast, existing MOS survey papers only introduced a subset of them.
- We evaluate the performance of efficient MOS models on four most popular datasets: CDnet 2014, DAVIS 2016, DAVIS 2017, and YouTube-VOS. We identify models that are competitive in segmentation accuracy and inference speeds on these datasets, and analyze the essential techniques of these models.
- We identify existing challenges in MOS and provide insights into future research directions.

The remainder of the paper is organized as follows. In Section 2, we introduce existing MOS methods, with a brief overview of traditional approaches, and a more detailed discussion on deep learning-based approaches. In Section 3, we discuss efficient MOS model design techniques and summarize models that use these techniques. Section 4 introduces representative datasets for a variety of scene categories. Section 5 presents performance evaluation metrics including segmentation accuracy metrics and model efficiency metrics. In Section 6, we provide comparison studies of existing efficient MOS models for popular

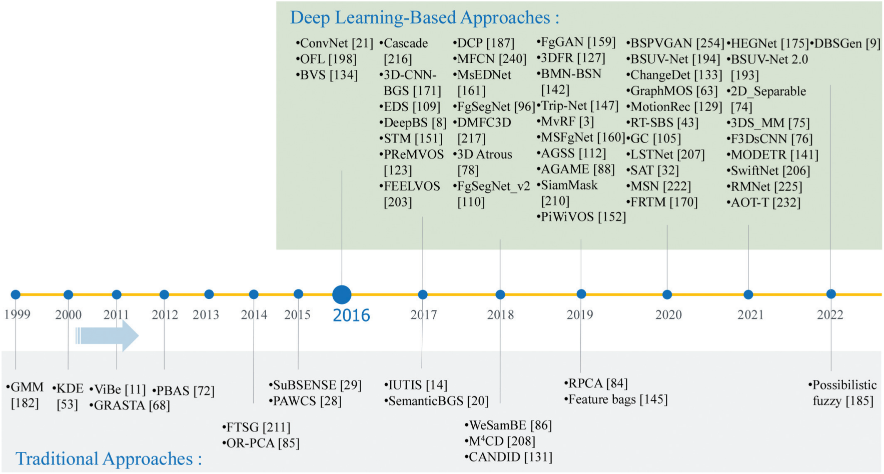


Figure 2: The timeline of MOS including traditional approaches [11, 14, 20, 28, 29, 53, 68, 72, 84–86, 131, 145, 182, 185, 208, 211] and deep learning approaches [3, 8, 9, 21, 32, 43, 63, 74–76, 78, 88, 96, 105, 109, 110, 112, 123, 127, 129, 133, 134, 141, 142, 147, 151, 152, 159–161, 170, 171, 175, 187, 193, 194, 198, 203, 206, 207, 210, 216, 217, 222, 225, 232, 240, 254].

datasets. Section 7 analyzes existing challenges in MOS and presents future research directions. Section 8 concludes the paper.

2 Existing MOS Approaches

MOS approaches can be categorized into traditional approaches and deep learning-based approaches. The chronological advancement in algorithms is depicted in Figure 2. Deep learning-based approaches are increasingly developed from 2016 and traditional approaches started in early 1999 and now are still being developed in parallel with deep learning-based approaches, and are even combined with deep learning approaches in a hybrid mode.

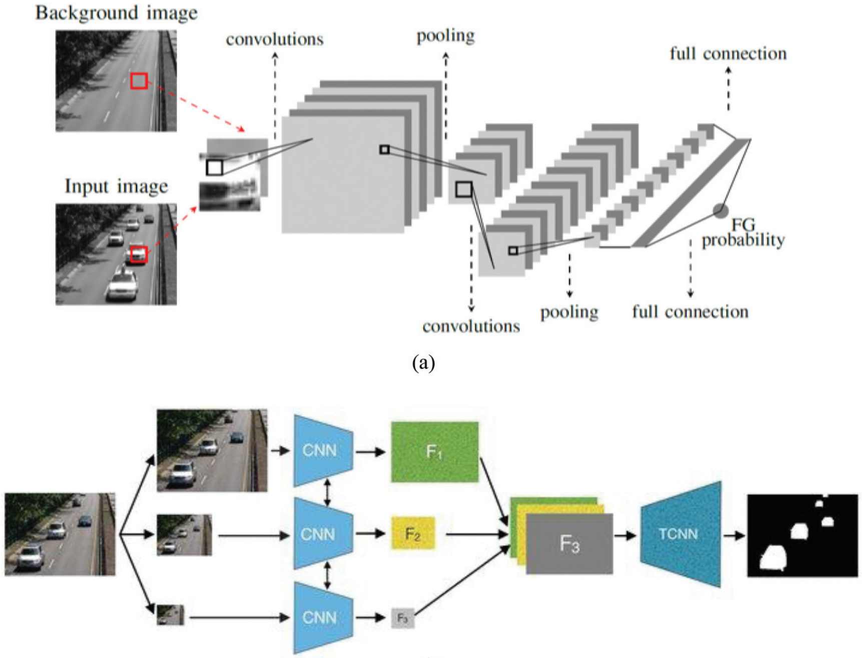
2.1 Traditional Approaches

Traditional approaches [11, 14, 15, 20, 28, 29, 39, 53, 68, 72, 84–86, 117, 131, 145, 182, 185, 208, 211, 255, 256, 262] do not require labeled ground-truth segmentation masks. They include two steps: background modeling and pixel classification. First, the background scene is initialized and updated over time, then each pixel is classified as foreground or background based on a threshold. Background modeling schemes can be parametric or non-parametric. Parametric approaches represent the background statistically using a probability density function (PDF) such as a single Gaussian or a mixture of Gaussians (GMM) [182, 262], and require to learn the parameters of these pdfs. Non-parametric approaches do not need to learn the parameters of probability distributions such as the kernel density estimation (KDE) method [53], or they do not have distribution assumptions for the background. There are also several sub-categories under the non-parametric approaches: filter-based, sample consensus, and principal-component-analysis (PCA) [116–118] methods. Examples of filter-based approaches are Kalman filtering [39], temporal median filtering [256], and running average filtering [255]. Examples of sample consensus methods are WeSamBE [86], ViBe [11], SuBSENSE [29], and PAWCS [28]. For instance, SuBSENSE uses a feedback system to adjust the background model automatically based on local binary similarity pattern (LBSP) features and pixel intensities [15]. PCA methods use eigenvalue decomposition for background modeling. To solve the camera motion problem, background subtraction based on robust principal-component analysis (RPCA) [84, 85] was also developed.

2.2 Deep Learning-Based Approaches

2.2.1 2D-CNN-based

Deep learning-based approaches have been recently proposed for MOS problems. The first deep learning-based work is ConvNet-GT [21] as shown in Figure 3(a), which performs change detection by replacing the pixel classification component with a well-defined neural network structure. The background is estimated with a temporal median filter, then it is stacked together with the original video frames to form the input of the CNN that outputs the binary masks. In a more advanced model MSFgNet [160], a motion-saliency network (MSNet) is used to estimate the background, which is then subtracted from the original frames, followed by a foreground extraction network (FgNet) that detects the moving objects. Another type of CNN is multi-scale feature learning-based CNN, which extracts multi-scale features to achieve better accuracy, such as MSCNN + Cascade [216], Guided Multi-Scale CNN [107], MCSCNN [108], MsEDNet [161] and VGG-16 [177] based networks FgSegNet_M [96] and FgSegNet_v2 [110]. In MSCNN+Cascade [216], segmentation results are generated pixel by pixel. Although it achieves good accuracy, the pixel-wise processing is very time consuming. In FgSegNet_M [96] as shown in Figure 3(b), a 2D CNN takes each video frame at three different resolution scales in parallel as the input of the encoding network.



(a)

(b)

Figure 3: 2D-CNN-based models: (a) ConvNet-GT (Copyright © 2016 IEEE [21]), (b) FgSegNet_M (Copyright © 2018 Elsevier B.V. [96]).

2.2.2 3D-CNN-based

The advantage of 3D convolution for MOS problems is that 3D convolution can utilize spatial-temporal information in visual data, which helps improve model accuracy. For example, in 3D CNN-LSTM [2] as shown in Figure 4(a), short-term temporal motions of a video sequence are captured by 3D convolutions, while the long-short term temporal motions are captured by 2D LSTM modules. Another example is 3DAtrous [78], which also captures long-term temporal information in the video data. It is trained based on an LSTM network with focal loss to tackle the class imbalance problem commonly seen in background subtraction. Besides, 3D-CNN-BGS [171] utilizes 3D convolution to track temporal changes in video sequences. It performs 3D convolution on 10 consecutive frames of the video, and upsamples the low-, mid-, and high-level feature layers of the network in a multi-scale approach to enhance segmentation accuracy. In [58], 3D CNN and a fully connected layer are adopted in a patch-wise method. In 3DS_MM [75] as shown in Figure 4(b), 3D separable convolutional neural network with a multi-input multi-output strategy is proposed, in which

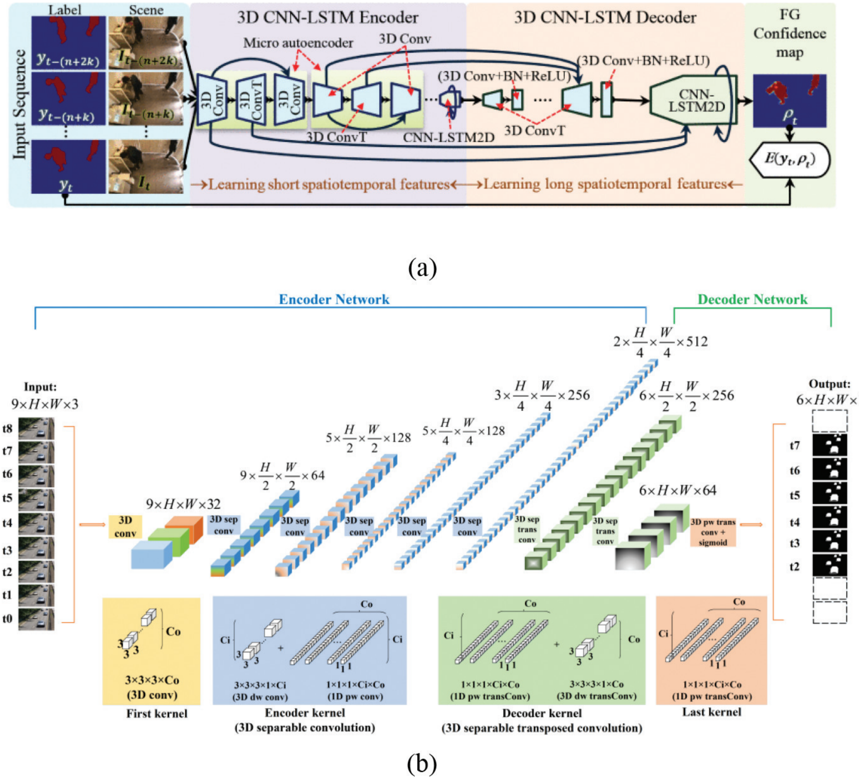


Figure 4: 3D-CNN-based models: (a) 3D CNN-LSTM model (Copyright © 2020 IEEE [2]), (b) 3DS_MM [75] model using 3D separable CNN.

the standard 3D convolution is decomposed into depthwise and pointwise convolutions, in order to reduce model size and computational complexity.

2.2.3 GAN-based

Generative adversarial networks (GAN) is also adopted in MOS problems. BScGAN [10] as shown in Figure 5(a) is based on conditional GAN (cGAN), in which the discriminator not only takes the foreground mask as the input, but also takes the stacked color image and its background as a conditional input to improve segmentation accuracy. FgGAN [159] shown in Figure 5(b) adopted GAN-based unpaired learning to solve challenging MOS problems such as dynamic background, bad weather, effect of shadow and irregular motion of objects. Since the training set has unpaired input and output images, this network is trained by adding a cycle-consistent loss in the traditional GAN loss. First, a video-wise background is estimated using GAN-based unpaired

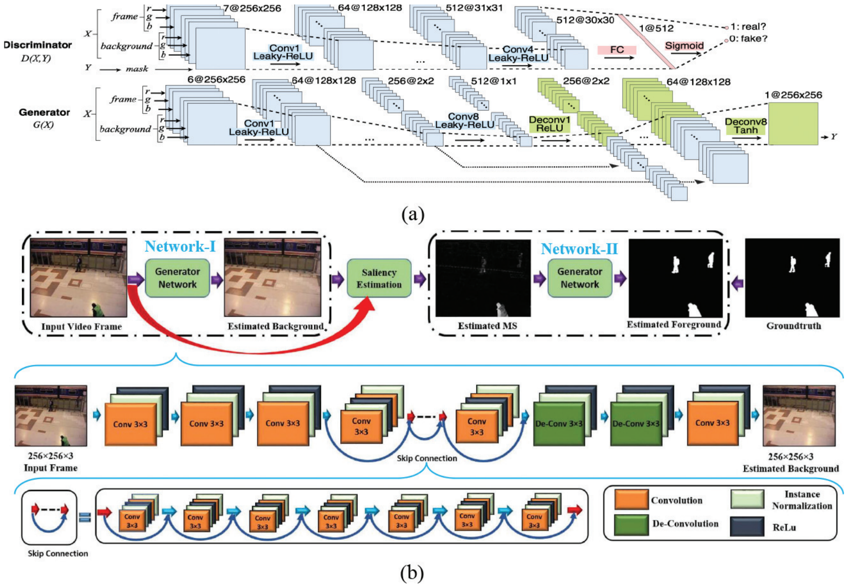


Figure 5: GAN-based models: (a) BScGAN (Copyright © 2018 IEEE [10]), (b) FgGAN (Copyright © 2019 IEEE [159]).

learning network (network-I). Then, to extract the motion information related to foreground, motion saliency is estimated using the estimated background and current video frame.

Further, the estimated motion saliency is given as the input to another GAN-based unpaired learning network (network-II) for foreground segmentation. In BSPVGAN [254], a median filter was used for background modeling and Bayesian GANs were adopted for pixel classification, because Bayesian GANs can address the problem of ghost, non-stationary background, and sudden illumination changes. Meanwhile, BSlsGAN [186] utilized conditional least squares adversarial networks, in which the generator loss function includes the L_1-loss and perceptual-loss between the generated segmentation mask and its respective ground truth to learn dynamic background variations. DBSGen [9] was also developed to address the dynamic background subtraction problem, which used two generative neural networks, one for dynamic motion removal and the other for background generation. The networks were optimized in an end-to-end fashion. Finally, the foreground moving objects were obtained by a pixel-wise distance threshold based on a dynamic entropy map.

2.2.4 Unsupervised VOS

Unsupervised video object segmentation (UVOS), also known as zero-shot VOS, automatically segments and tracks primary moving objects in a video sequence without any prior information. Early UVOS in DAVIS 2016 Challenge is object-level segmentation which generates binary segmentation masks and is relatively simple [51, 67, 104, 181, 195]. Besides, MOD tasks can also be considered as object-level unsupervised VOS.

With the launch of DAVIS 2019 Challenge, multi-object UVOS has become a hot topic. This task is more challenging as it is instance-level segmentation, which requires automatically discriminating different object instances and associating the same identities over time. An early approach RVOS [202] incorporated recurrent layers spatially and temporally to discover different object instances within a frame, and to keep the coherence of segmented objects along time. Since the network shared the encoder forward pass for all the objects in a frame, it achieved a fast overall runtime, although the segmentation accuracy is relatively low. More sophisticated models followed a two-stage paradigm: (1) detect object proposals using pre-trained Mask R-CNN, and (2) conduct generic feature matching for temporal association using re-identification techniques [212, 257, 263]. Although these approaches achieved higher accuracy, they are computationally expensive. For example, UnOVOST [263] not only requires Mask R-CNN for instance proposal generation, but also needs to compute optical flow for motion estimation. Complex post-processing and heuristics also make this method unsuitable for practical applications.

Recently, to strike a better balance between accuracy and efficiency, the model in [258] formulates instance proposal and foreground estimation in a unified framework, requiring much less time to generate instance proposals than Mask R-CNN based methods. Besides, it does not require additional post-processing components.

2.2.5 Semi-Supervised VOS

Semi-supervised VOS, also known as one-shot VOS, has emerged with the launch of DAVIS 2017 Challenge [165]. It incorporates human intervention during the inference stage. The most typical human intervention is to provide the ground-truth object mask of the first frame of the video. Many semi-supervised VOS models are trained and evaluated on the DAVIS 2016 dataset [164] designed for object-level segmentation, or on the DAVIS 2017 [165] and YouTube-VOS [227] datasets designed for instance-level segmentation.

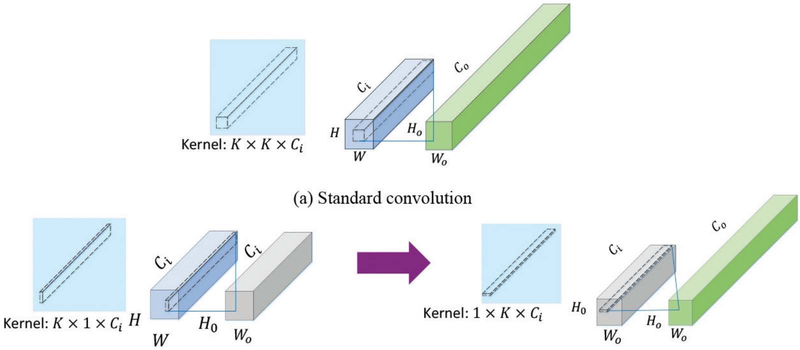
Deep learning models for semi-supervised VOS tasks can be categorized as online fine-tuning-based, template-based, and propagation-based methods. Online fine-tuning-based methods first learn general segmentation features offline from images and video sequences, then fine-tune the model at test time with the ground-truth object mask of the first frame. For example, PReMVOS [123] adopts online fine-tuning, which first generates a set of object segmentation mask proposals for each video frame, followed by selecting and merging these proposals into accurate and temporally consistent pixel-wise object tracks over a video sequence. It won both the 2018 DAVIS Challenge on VOS and the 2018 YouTube-VOS challenge.

On the other hand, template-based methods use the first frame with its ground-truth mask to extract object features as a template, then segment objects from subsequent frames by matching their features with the template. Examples of template-based methods are RANet [219] and TTVOS [154]. RANet applies a ranking system to the matching process between multiple templates and the input to extract reliable results. TTVOS combines short-term matching and long-term matching, in which short-term matching enhances target object localization and long-term matching improves fine details and handles object shape-changing. Propagation-based methods use the previous frame mask to infer the current frame mask. One example is MaskTrack [163], which proceeds on a per-frame basis, guided by the output of the previous frame towards the object of interest in the next frame.

2.2.6 Other Approaches

Some foreground and background segmentation models use data augmentation techniques to improve the performance of prior models. For example, data augmentation performed in [172] not only creates endless data on the fly, but also features semantic transformations of illumination, which enhances the generalization capability of the model. It successfully simulates flashes and shadows by applying the Euclidean distance transform over a randomly generated binary mask. Such data augmentation allows to effectively train an illumination-invariant deep learning model for background subtraction. Another example using data augmentation is [93]. Two data augmentation methods of adjusting background model images and past images are proposed and applied to their previously proposed foreground segmentation framework [92]. Through this method, the segmentation performance is improved in difficult areas such as static foreground and ghost objects, compared to previous studies.

Some other models combine traditional and deep learning approaches to leverage the strengths of both to obtain better performance such as RT-SBS [43], GraphMOS [63], MotionRec [129], and GraphBGS [62].


Kernel: $K \times K \times C_i$
C_i
H
W
H_o
W_o
C_o
Kernel: $K \times 1 \times C_i$
H
W
C_i
H_0
C_i
W_o
Kernel: $1 \times K \times C_i$
H_0
C_i
W_o
H_o
W_o
C_o

(a) Standard convolution

(b) Asymmetric convolution: $K \times 1$ kernel and $1 \times K$ kernel

Figure 6: Asymmetric convolution.

3 Efficient MOS Model Design

Although deep learning-based MOS models have achieved high accuracy in recent years, they come at a high computational cost and a slow inference speed due to complex network structures and intense convolution operations. The need for efficient models that require less inference time is vital, especially for applications such as autonomous driving. In the following, we summarize techniques used to design efficient deep neural network architectures to reduce the inference time, computational complexity and model size, while maintaining high segmentation accuracy. Existing models using these techniques are summarized in Table 2.

3.1 Asymmetric Convolution

Asymmetric convolution is to factorize a standard two-dimensional convolution kernel into two one-dimensional convolution kernels. As shown in Figure 6, a $K \times K$ convolution can be substituted with a $K \times 1$ convolution followed by a $1 \times K$ convolution. Such a scheme can effectively reduce network parameters and required calculations. For example, when the input channel C_i is the same as the output channel C_o, the number of parameters and computational cost are saved by 33% for a 3×3 kernel [189], and the performance degradation is often very small. A real-time foreground segmentation model DRSNet [205] uses asymmetric convolutions in its proposed MultiScaleSE Block, DoubleConv Block and NeckConv Block, to replace symmetric convolutions to reduce parameters and to ensure segmentation accuracy at the same time.

Table 2: Efficient MOS model design techniques.

Efficient Model Design Techniques			MOS Models
3.1	Asymmetric Convolution		DRSNet [205]
3.2	Dilated Convolution		DRSNet[205]; A Fast X-shaped with CompactASPP [241]; One-Shot Animal [228]; U^2- ONet [204]; Frame-Level Weakly Supervised [139]; Espnet [135]; PDB [181]; PyramidCSA [67]
3.3	Group Convolution and Depth-Wise Convolution		2D_Separable CNN [74]; One-Shot Animal [228]; 3DS_MM [75]; F3DsCNN [76]
3.4	Reduce Feature Map's Volume		ChangeDet [133]; 3DFR [127]; U^2-ONet [204]; AOT-T [232]; FRTM-fast [170]; FRTM [170]; G-FRTM-fast [155]; SiamMask [210]; DRSNet [205]; Adaptive Template [156]; MFCN [240]
3.5	Simplified Network Architecture		MSFgNet [160]; AGSS [112]; One-Shot Animal [228]; PiWiVOS-F [152]; FCESNet [168]; FAMINet [120]; Lightweight U-Net-like [111]; Guided Multi-Scale CNN [107]; MSCNN+Cascade [216]; Trip-Net [147]
3.6	Two-Branch Network		F3DsCNN [76]; ContextNet [166]
3.7	Channel Merging by Addition		DRSNet [205]; Lightweight U-Net-like [111]
3.8	Decoder Size Reduction		One-Shot Animal [228]; 2D_Separable CNN [74]; 3DS_MM [75]; F3DsCNN [76]
3.9	Quantization		T-RexNet [24]; S3-Net [38]
3.10	Small Convolution Kernel		A Fast X-Shaped with CompactASPP [241]; Frame-Level Weakly Supervised [139]; MSFgNet [160]
3.11	Skip Connection		Lightweight U-Net-like [111]; BMN-BSN [142]; BSUV-Net 2.0 [193]; 3D CNN-LSTM [2]; BScGAN [10]; U^2-ONet [204]; Edge Aggregation Network [157]; LSTNet [207]; Frame-Level Weakly Supervised [139]; MvRF [3]
3.12.1	Improving the Efficiency of Semi-Supervised VOS	Meta Learning	Meta-Learning [224]; e-OSVOS [137]

3.12.2		Template-Based and Propagation-Based Methods	SAT-Fast [32]; SAT [32]; Fasttmu [188]; SwiftNet [206]; RANet [219]; FRTM-fast [170];
			FRTM [170]; G-FRTM-fast [155]; TVOS [247]; TTVOS(HRNet) [154]; TTVOS(ResNet50) [154]; GC [105]; LWL [13]; MSN [222]; RMNet [225]; SiamMask [210];
			DDEAL(Res101) [236]; LSTNet [207]; Adaptive Template [156]; AGSS [112]

3.13	Multi-Input Multi-Output Strategy		3DS_MM [75]; F3DsCNN [76]; FCESNet [168]

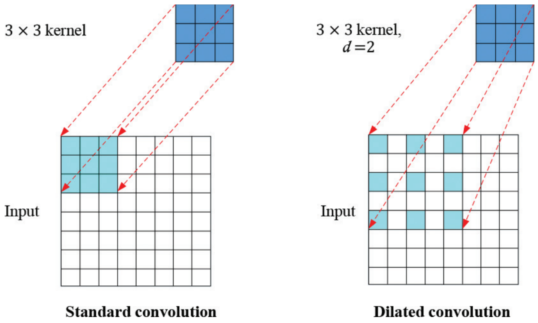


Figure 7: An illustration of the dilated convolution with kernel size $k = 3$ and dilation rate $d = 2$. Both the standard convolution and dilated convolution have the same number of parameters (i.e. kernel size is 3×3), whereas the dilated convolution has a larger 5×5 receptive field.

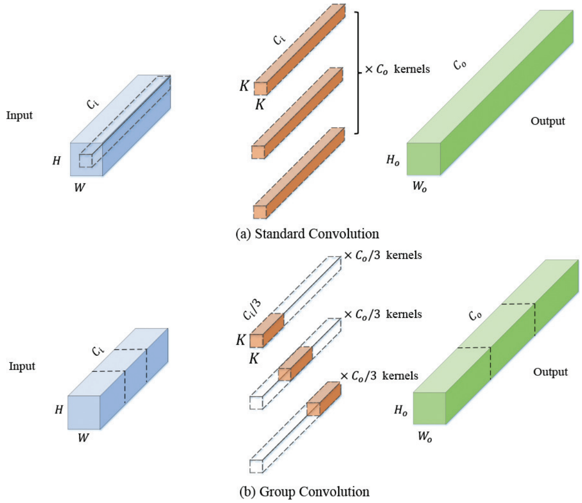
3.2 Dilated Convolution

Dilated convolution is adopted in MOS [204, 205, 228, 241] to allow larger receptive field with the same computation and number of model parameters (weights). In [228], dilated convolutions are implemented with different dilation rates to increase the receptive field for video segmentation. As illustrated in Figure 7, for a $k \times k$ dilated convolutional kernel with a dilation rate of $d > 1$, the effective size of the kernel is increased to $[(k-1)d+1]^2$. However, only $k \times k$ pixels participate in the convolutional operation, reducing the computational cost while increasing the effective kernel size and receptive field [135].

It is worth noting that dilated convolution is usually combined with multi-scale schemes to extract features. In PDB [181], multi-scale features are extracted by dilated convolution with different dilation rates to generate features with different receptive fields, which reduces complexity compared to using large kernel sizes. In PyramidCSA [67], multi-scale features are extracted by constrained self-attention with different attention window sizes and dilations in parallel branches, to capture motion cues of multi-scale objects and objects moving at various speeds. It has much less computation and memory usage than the non-local attention mechanism which extracts global context.

3.3 Group Convolution and Depth-Wise Convolution

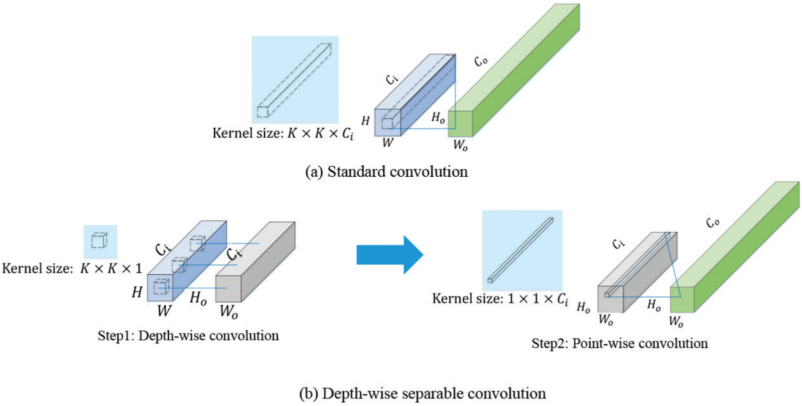
To reduce computational cost and model size, group convolution is adopted in [156]. The idea of group convolution was first introduced in AlexNet [94] to use the limited memory of two GPUs to train the model in parallel. Since then, it has been widely applied in computation-efficient network architecture designs. As shown in Figure 8, a group convolution splits the channels of the input feature maps into G mutually exclusive groups, then convolution is inde-



(a) Standard Convolution

(b) Group Convolution

Figure 8: An illustration of the group convolution with three groups $G = 3$. Each group of input is convolved with $(C_o/3)$ kernels of size $K \times K \times (C_i/3)$, to generate an output of size $H_o \times W_o \times (C_o/3)$. Three groups of outputs are concatenated to form the final result.


Kernel size: $K \times K \times C_i$
C_i
H
W
H_o
W_o
C_o
Kernel size: $K \times K \times 1$
C_i
C_i
H
W
H_o
W_o
Kernel size: $1 \times 1 \times C_i$
C_i
C_o
H_o
W_o
H_o
W_o

(a) Standard convolution

Step1: Depth-wise convolution

Step2: Point-wise convolution

(b) Depth-wise separable convolution

Figure 9: An illustration of the 2D depth-wise separable convolution.

pendently performed on each group, followed by output channel concatenation. Theoretically, it can reduce both the computational complexity and model parameters by a factor of G.

The method of depth-wise convolution was first proposed in MobileNet [77]. As demonstrated in Figure 9, it decomposes a standard 2D convolution into a depth-wise convolution (also known as spatial or channel-wise convolution) and a 1×1 point-wise convolution. While the depth-wise convolution applies an independent 2D filter for each input channel, the subsequent point-wise convolution performs 1D convolutions on the output of the depth-wise convolution along the channel direction. This separation can effectively reduce the amount of computation and model size. For example, when the filter size is $K \times K$, the computational cost of 2D separable convolution can be reduced to roughly $\frac{1}{K^2}$ of that of the standard 2D convolution. It can dramatically increase the inference speed while maintaining high detection accuracy. This idea was utilized in 2D_Separable CNN [74] and One-Shot Animal model [228] for video moving object segmentation.

To further exploit the temporal information in the video input, the 3DS_MM model was proposed [75], which adopts 3D convolution to extract spatio-temporal features in the video data and to improve segmentation accuracy. To reduce computational complexity and model size, the standard 3D convolution is decomposed into a depth-wise convolution and a point-wise convolution. While the depth-wise convolution performs a spatial-temporal convolution independently on each input channel, the subsequent point-wise convolution along the channel direction can effectively leverage channel correlations. When $K \times K \times K$ is the spatial-temporal filter size, the computational cost of such a 3D separable convolution can be reduced roughly to $\frac{1}{K^3}$ of that of the standard 3D convolution.

On the other hand, since it is better to apply nonlinear activations in a high-dimensional space than in a low-dimensional space to prevent information loss, MobileNetV2 [173] introduces the inverted residual bottleneck module. The input features with C_l channels are first expanded to a high-dimensional space with $C_h > C_l$ channels using a point-wise convolution. Subsequently, a 2D depth-wise convolution with nonlinear activations is performed on each of these C_h channels. Afterwards, another point-wise convolution with linear activatons projects the features back onto a low-dimensional space with C_l channels. To utilize spatio-temporal information in video data and to increase segmentation accuracy, the method proposed in F3DsCNN [76] replaces such 2D separable convolutions in the inverted residual bottleneck by 3D separable convolutions.

3.4 Reduce Feature Map's Volume

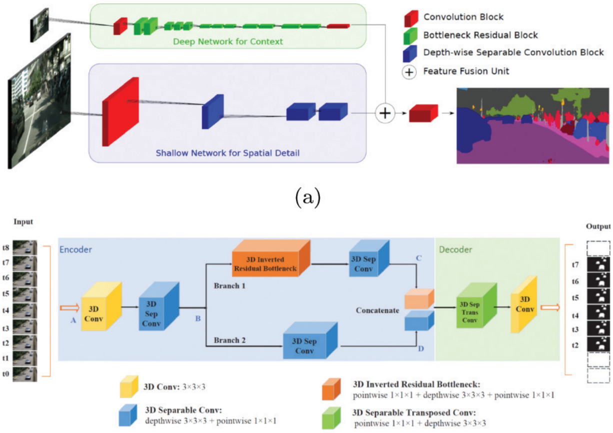
Convolution-based feature extraction has high computational complexity due to large image resolution and large number of channels. Methods are typically used in MOS to reduce the volume of feature maps. In particular, large-scale downsampling can be used to reduce the input image (or feature map) resolution before applying more convolutional layers, and point-wise convolution can be used to reduce the number of channels. Such techniques can be found in [127, 133, 155, 156, 170, 204, 205, 210, 232, 240]. For example, in U^2-ONet [204], OctConv (Octave Convolution) [34] is used to reduce spatial redundancy. It is well-known that natural images can be decomposed into a low and a high spatial frequency part. While the low-frequency part represents global structures, the high-frequency part contains fine details. Similarly, the feature maps of a convolution layer can also be factorized to low-frequency and high-frequency components. OctConv stores and processes the smoothly changing, low-frequency feature maps in a low-resolution tensor to reduce spatial redundancy, meanwhile reducing the memory and computation cost. Another example is a shallower network ChangeDet [133] which utilizes fewer number of kernels to reduce feature map channels, and uses max pooling to reduce feature map resolutions. This results in only 0.13 millions of trainable parameters with a model size of only 1.6 megabytes (MB).

3.5 Simplified Network Architecture

To reduce model size and computational cost, simplified network architectures are also used in MOS models. In FCESNet [168], there is no fully connected layers, and the network only contains 7 layers, which reduces the number of parameters and increases the inference speed. FAMINet [120] is proposed to include feature extractor, appearance network, motion network and integration network. The appearance network generates an initial segmentation, the motion network generates an optical flow, then the integration network takes these results and previous frames' predictions as its inputs, and outputs the final refined segmentation result. For efficiency, the motion network only has two convolutional layers and the integration network only has five convolutional layers. In the Lightweight U-Net-like model [111], thinner convolution layers are utilized to achieve an inference speed of 250 frames per second (fps) on a GTX Titan Xp GPU. The model size is only 435 kilobytes (KB). There are only 0.105 millions of model parameters, and only 0.13 billion floating-point operations (GFLOPs) are needed. In the One-Shot Animal Model [228], the proposed encoder module Xception-lite for video object segmentation is inspired by Xception-65 backbone [40]. While the original Xception-65 backbone has 65 layers to extract visual features which is time-consuming, the Xception-lite model only has 20 layers. To achieve the best trade-off between accuracy and speed, Xception-lite also incorporates residual connections and separable convolutions. Such simplified network architectures can also be found in Guided Multi-Scale CNN [107], MSCNN+Cascade [216], MSFgNet [160], Trip-Net [147], PiWiVOS-F [152], etc.

3.6 Two-Branch Network

Segmenting high-resolution inputs directly with classical frameworks like fully convolutional networks (FCN) [121] is time consuming. To overcome this shortcoming, the two-branch network scheme was proposed. The network consists of two branches, while one branch is shallow, captures spatial details and generates high-resolution feature representation, the other branch is deep


Deep Network for Context
Shallow Network for Spatial Detail
Convolution Block
Bottleneck Residual Block
Depth-wise Separable Convolution Block
Feature Fusion Unit
Input
Output
t8
t7
t6
t5
t4
t3
t2
t1
t0
Encoder
Decoder
3D Conv
3D Sep Conv
A
B
C
D
Branch 1
Branch 2
3D Inverted Residual Bottleneck
3D Sep Conv
Concatenate
3D Sep Trans Conv
3D Conv
3D Conv: 3×3×3
3D Separable Conv: depthwise 3×3×3 + pointwise 1×1×1
3D Inverted Residual Bottleneck: pointwise 1×1×1 + depthwise 3×3×3 + pointwise 1×1×1
3D Separable Transposed Conv: pointwise 1×1×1 + depthwise 3×3×3

(a)

(b)

Figure 10: Two-branch network: (a) ContextNet [166], (b) F3DsCNN (Copyright © 2021 IEEE [76]).

and obtains high-level semantic context. Finally, the two-branch output feature maps are fused to generate the segmentation result. For example, ContextNet [166] as shown in Figure 10(a) processes the video frame at two resolutions in two parallel branches for semantic segmentation. In F3DsCNN [76] as shown in Figure 10(b), the 3D convolution-based two-branch scheme is adopted to extract spatial-temporal information for video moving object segmentation. Moreover, F3DsCNN proposed to share the first few layers of the two branches such that the model size and complexity can be further reduced. Similar ideas were also adopted in ICNet [250] and FgSegNet_v2 [110], which extended the two-branch network to three-branch networks to extract low-, mid-, and high-level features.

Using such two-branch or multi-branch networks can get high quality segmentation results, since the high-resolution branch helps recover and refine the coarse prediction produced by the low-resolution branch. Although some details are missing and blurry boundaries are generated in the low-resolution branch, it already harvests most semantic parts.

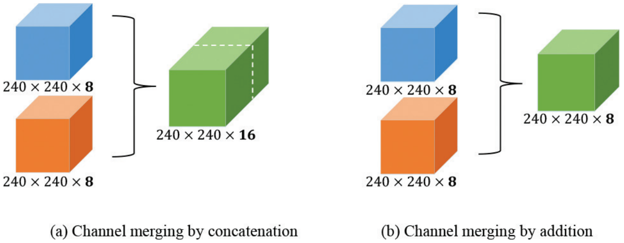
3.7 Channel Merging by Addition

To integrate shallow and deep semantic layers, usually channel concatenation is used as in Figure 11(a). However, it increases the computational cost of subsequent convolution layers since the number of channels increases. As illustrated in Figure 11(b), to reduce the computational cost, addition can be used to merge channels [205], [111]. In particular, the Lightweight U-Net-like model [111] adopts element-wise summing for feature fusion in the decoder, which stabilizes training convergence, and also achieves an inference speed of 250 fps on an Nvidia GTX Titan Xp GPU with a model size of 435 KB and 0.105 millions of trainable parameters.

3.8 Decoder Size Reduction

The encoder-decoder network is one of the most standard architectures of object segmentation. It is suggested that the architecture of an encoder–decoder model can be simplified by reducing the decoder's size, in order to save computational cost. In other words, we can adopt an asymmetric encoder-decoder architecture in which the encoder is larger than the decoder, instead of a symmetric encoder-decoder architecture. The rationale behind this idea is that the encoder should work in a similar fashion to original classification architectures, which extract deep features of smaller resolutions. On the contrary, the role of the decoder is to up-sample the output of the encoder, only enhancing its details. Therefore, reducing the decoder's size results in computational cost savings. Overall, this approach is appealing since most of the time the reduction in the decoder's size does not affect segmentation accuracy.

Decoder size reduction has been demonstrated in [74] and [75], in which the number of decoder layers is less than that of the encoder layers, which achieves faster inference speed without affecting the accuracy. In F3DsCNN [76], the last deconvolution layer in the decoder is replaced by upsampling to further increase the inference speed. In the One-Shot Animal model [228], the encoder is a deep network that generates five stages of shallow to deep feature maps. The feature maps are then upsampled and concatenated to form the final encoded feature map. In contrast, the decoder simply performs a 1 × 1 convolution to linearly fuse these feature maps to output a 1D probability map as the segmentation result, followed by post-processing to fine-tune the segmentation accuracy.



(a) Channel merging by concatenation

(b) Channel merging by addition

Figure 11: Channel merging by (a) concatenation, and (b) addition.

3.9 Quantization

The runtime of a network can be further reduced using quantization techniques. These techniques reduce the size/memory requirements of a network by encoding parameters in low-bit representations. In [38], quantization is adopted in both training and inference with 8-bit weights and features, which results in significant model compression and efficiency enhancement. In [24], the experiments were conducted with the half-precision floating point (FP16) format since this provides a good trade-off between accuracy and power consumption. The use of FP16 has achieved a good frame rate on the NVIDIA Jetson Nano edge-device, which shows its suitability for real-time applications [24].

3.10 Small Convolution Kernel

To benefit from rich contextual dependencies, an MOS model can use standard convolutions with large kernel size to enlarge the receptive field. However, it is harmful to use excessively large kernels, because they might lead to over-fitting [162], and increase the model size and complexity, especially when they are used in deep layers with a large number of input feature map channels. To address this problem, in [241], small kernel size is adopted along with dilated convolution (atrous convolution), which effectively enlarges the receptive field to capture long-range dependencies, meanwhile avoiding problems that may be caused by large kernels.

Although large kernels were adopted in [115] and [47] to close the performance gap between CNNs and transformers [48, 119], it is noteworthy that these methods [47, 115] adopt large kernels in depth-wise convolution to control the computational complexity. In particular, [115] also utilized inverted bottleneck, such that the depth-wise convolution with a large kernel is performed when the feature channels are small, followed by 1×1 point-wise convolution which raised the channel to a higher-dimensional space.

3.11 Skip Connection

Skip connection (or shortcut connection) is a technique that has demonstrated its effectiveness in maintaining high segmentation accuracy without adding network complexity, and has been adopted in classic lightweight models such as MobileNetV2 [77] and SqueezeNet [61].

In the Lightweight U-Net-like MOS model [111] as shown in Figure 12, long and short skip connections are proposed to facilitate data flow and maximize the usage of parameters in the model, enabling a lightweight design and leading to faster networks. As shown in Figure 12(a), long skip connections ship low-level features extracted by the second encoder layer directly across networks and share them to the corresponding decoding layer. Since the texture details are captured by low-level features and are usually lost in deep layers, long skip connections can provide decoders with such low-level features to more effectively infer the foreground masks. Compared to multi-scale feature extraction [216], long skip connections make one pass sufficient to pass richer information to the other end of the network [111]. On the other hand, short skip connections are adopted in bottleneck blocks of the network to enhance feature utilization rate, such that the bottleneck blocks in [111] can be designed as compact as possible. Figure 12(b) shows the detailed structure of the bottleneck block, which consists of four mini-blocks. Each mini-block is an extremely lightweight operation block that further consists of one convolutional layer, one instance normalization, and one parametric ReLU layer. As shown in Table 4, such efficient design of the Lightweight U-Net-like model achieved the highest F-measure (97.7%) with the fastest inference speed (250 fps) and required the fewest model parameters (0.1 M) among all models under the Titan Xp GPU group.

Skip connection is also used in BMN-BSN [142], BSUV-Net2.0 [193], 3D CNN-LSTM [2], BScGAN [10], U^2-ONet [204], Edge Aggregation Network [157], Long-Short Term Network (LSTNet) [207], Frame-Level Weakly Supervised Network [139], and MvRF-CNN [3] for MOS tasks. Besides, asymmetric skip connections [205] was also proposed, which can more effectively integrate information from different semantic layers than symmetric skip connections, while requiring smaller number of network parameters.

3.12 Improving the Efficiency of Semi-Supervised VOS

Recently, more and more semi-supervised VOS models (one-shot learning) have emerged, in which the object mask of the first frame is provided in the inference process, and the goal is to automatically segment the target object from the entire video sequence.

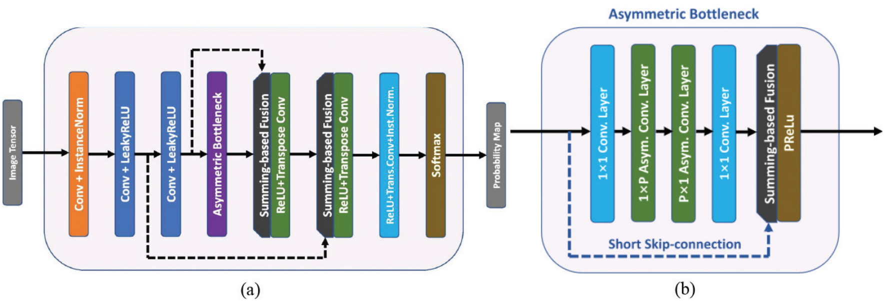


Figure 12: (a) Long and short skip connections in the Lightweight U-Net-like MOS model [111], and (b) the Asymmetric Bottleneck block of (a), using a short skip-connection.

There are three types of one-shot learning for video object segmentation: online fine-tuning, template-based, and propagation-based. Online fine-tuning refers to approaches that fine-tune a general-purpose segmentation model on the annotated first frame during test time using hundreds of iterations of gradient descent [230]. Due to the heavy computation burden of fine-tuning, the processing speed of such methods cannot satisfy the requirements for practical applications. To alleviate the complexity issue of online fine-tuning, meta learning can be used to optimize the online fine-tuning process [137, 224]. Or, template-based and propagation-based methods can be used to refrain from online fine-tuning.

3.12.1 Meta Learning

Meta learning, also known as "learning to learn", uses a bunch of similar learning tasks to train a meta-learner, such that it can adapt to a new task quickly with only a few training samples. In [224], a meta-learner is trained on multiple VOS tasks such that the meta model can capture their common knowledge and gains the ability to fast adapt the segmentation model to new video sequences. The e-OSVOS [137] approach meta learned the model initialization and learning rates for test time optimization by predicting individual learning rate at a neuron level. Furthermore, it applies an online adaptation to address the common performance degradation problem for future frames in the video sequence.

3.12.2 Template-Based and Propagation-Based Methods

In addition to meta learning, template-based and propagation-based methods (introduced in Section 2.2.5) are also proposed to alleviate the dependence on the online fine-tuning process, and they are usually combined in many fast VOS models.

However, template-based methods can suffer from memory issues because historical frames need to be memorized and updated, and propagation-based methods are vulnerable to temporal discontinuities like occlusions and rapid motion. To solve these problems, a real-time model SwiftNet [206] updates fewer frames in memory and adaptively selects incremental frames that have variations for memory update and ignores static ones. Besides, it abandons full-frame operations and incrementally processes with temporally varying pixels. Similarly, [156] adaptively updated the shape variation of target objects without heavy computation or additional memory. In [105], fixed-size feature representation is used to reduce memory which is needed in the template matching process. In the Regional Memory Network (RMNet) [225], local-to-local matching is performed between the current query frame and past frames. This effectively addresses the problems of mismatching to similar objects and high computational complexity caused by global-to-global matching. DDEAL [236] learns static cues from the labeled first frame and dynamically updates cues of subsequent frames for object segmentation. In [207], both template-based and propagation-based strategies were explored to match for pixel-level object segmentation and to handle the mismatching and drifting problem. In particular, the proposed long-term network exploits the object relationship between the current frame and the first frame, and the proposed short-term network explores immediate object variations.

Efficient design of template-based and propagation-based methods can also be found in SAT-fast [32], SAT [32], Fasttmu [188], RANet [219], FRTM-fast [170], FRTM [170], G-FRTM-fast [155], TVOS [247], TTVOS(HRNet) [154], TTVOS(ResNet50) [154], LWL [13], MSN [222], SiamMask [210], AGSS [112], etc.

3.13 Multi-Input Multi-Output Strategy

Another factor that affects the performance of a model is the input-output relationship. As shown in Figure 13, the input-output relationship of existing MOS networks has three types. The first type is single-input single-output (SISO), which is widely exploited in 2D CNNs such as FgSegNet_S [96] and 2D_Separable CNN [74]. The second type is multi-input single-output (MISO) which can be found in 3D CNNs such as 3D-CNN-BGS [171], 3DAtrous [78], and DMFC3D [217]. The disadvantage of SISO and MISO is that they result in a slow inference speed because only one frame output is predicted in every forward pass. The third type of input-output relation is multi-input multi-output (MIMO) such as FCESNet [168], which can take multiple input frames and output multiple frames of segmentation masks in each forward pass. It explores temporal correlations on a larger time span. Two recent models 3DS_MM [75] and F3DsCNN [76] combined the MIMO strategy with 3D separable CNN, which significantly increase the inference speed, so that they are suitable for computation-resource-limited and delay-sensitive applications.

3.14 Summary

Among the efficient techniques discussed above, the two-branch network [76, 166] and skip connection [111, 205] are very effective for scenarios such as MOS and semantic segmentation in which both high-level semantics and low-level details are needed. The two-branch network and skip connection extract high-level semantics by deep network layers to infer segmentation class labels, and extract low-level details by shallow network layers to recover segmentation boundaries. These two techniques also have the merit of reducing computational complexity, since only small-resolution features are processed by deep layers, while large-resolution features are processed by shallow layers.

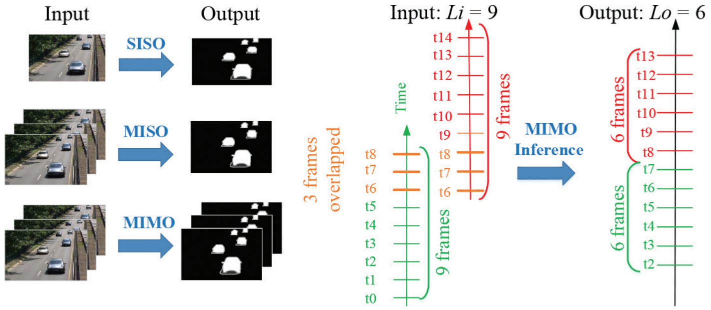


Figure 13: The input-output relationship of MOS models: single-input single-output (SISO), multi-input single-output (MISO), and multi-input multi-output (MIMO) [75].

Besides, channel merging by addition can be used after these two schemes to reduce the feature volume and computation.

When the dilated convolution is combined with multi-scale processing, it is very effective for dense prediction tasks [190], such as MOS, semantic segmentation, super-resolution, and image denoising. The reason is that multi-scale dilated convolution with different dilation rates can extract features from small to large receptive fields to model both local and global patterns. Meanwhile, it avoids increasing the kernel size.

The multi-input multi-output (MIMO) strategy is very effective in increasing the inference speed for MOS [73, 75, 76] and other video-related tasks, such as video frame prediction. Since multiple video frames can share the intermediate layers of the network, the computational complexity can be effectively reduced. On the other hand, decoder size reduction [74–76, 228] is useful to reduce the complexity of segmentation networks such as those for MOS and semantic segmentation. In these networks, a deep encoder is needed to extract high-level semantic features from pixels, while the decoder can be shallow, which only needs to generate the segmentation mask without recovering texture details.

Other techniques such as asymmetric convolution, depth-wise convolution, group convolution, reduce feature map's volume, simplified network architecture, quantization, and small convolution kernel can be used for not only MOS, but also general computer vision tasks, such as image classification and object detection. In particular, asymmetric convolution, depth-wise convolution, and group convolution are similar in the sense that they all decompose a standard convolution into several steps. Recent research [75, 76, 156, 205] shows that they effectively reduce model parameters (kernel weights) and computation for MOS tasks.

4 Datasets

Table 3 provides numerous existing datasets for the training and evaluation of MOS models [4, 12, 16, 17, 22, 23, 44–46, 52, 66, 87, 90, 97–101, 103, 125, 126, 129, 130, 138, 143, 149, 150, 164, 165, 167, 178, 179, 184, 192, 196, 197, 200, 201, 209, 214, 215, 223, 227, 233, 238]. Most of these datasets are for object-level segmentation, such as CDnet 2014 [215], DAVIS 2016 [164], UCSD [126], etc., hence the ground-truth masks contain binary labels. Some of the datasets are for instance-level segmentation, such as DAVIS 2017 [165] and YouTube-VOS [227], in which the ground-truth masks contain multi-class segmentation labels. Videos in these datasets are captured by various types of cameras, including RGB cameras, thermal cameras, RGB-D cameras which capture RGB images and their corresponding depth images, and multi-spectral cameras. From a citation perspective, CDnet, DAVIS, and YouTube-VOS datasets have the most citation counts, followed by other popular datasets such as LASIESTA [44] and SABS [22].

In Table 3, each column lists the appearance year, dataset name, scene category, segmentation labels (binary or multi-class labels) provided, frame resolution, number of videos, provider, and the access link of the datasets. The datasets are grouped by different scene categories. Within each scene category, the datasets are separated by the type of segmentation labels. In the following, we will introduce some representative datasets.

4.1 Indoor & Outdoor Category

The biggest scene category is Indoor & Outdoor, which covers indoor scenes such as human activities and outdoor scenes such as traffic, sports, shadow, night videos, etc.

4.1.1 CDnet 2012 & CDnet 2014 Datasets

The CDnet 2012 [66] and CDnet 2014 datasets [215] provide realistic, camera-captured videos of diverse surveillance scenes. CDnet 2012 was developed as part of the CVPR 2012 Change Detection Workshop challenge. This dataset consists of 31 videos (∼70,000 frames) spanning 6 categories which include diverse change detection and motion detection challenges. CDnet 2014 was developed as part of the CVPR 2014 Change Detection Workshop challenge. It includes all the videos from CDnet 2012 and 22 additional videos (∼70,000 new frames) spanning 5 new scene categories that incorporate challenges not addressed in the CDnet 2012 dataset.

The complete CDnet dataset (2012 & 2014) contains 11 video categories: baseline, dynamic background, camera jitter, shadow, intermittent object motion, thermal, bad weather, low frame-rate, night scenes, PTZ (pan, tilt,

Table 3: The details of video datasets dedicated to MOS.

No.	Year	Dataset name	Scenes	Label	Resolution	Videos	Provider	Access link
1	1999	WALLFLOWER [197]	**Indoor & Outdoor**	binary	160 × 120	7	Microsoft Research	https://www.microsoft.com/en-us/download/details.aspx?id=54651
2	2004	PTIS/I2R [101]		binary	640 × 480 to 960 × 1280	10	Institute for Infocomm Research (IR), Singapore	https://www.a-star.edu.sg/i2r
3	2008–2014	LIMU [97]		binary	320 × 240	8	Kuyshu University, Japan	https://limu.ait.kyushu-u.ac.jp/dataset/en/
4	2011	SABS [22]		binary	800 × 600	9	Universitat Stuttgart, Germany	https://www.vis.uni-stuttgart.de/
5	2012	CDnet 2012 [66]		binary	320 × 240 to 720 × 575	31	Univ. Sherbrooke, Canada	http://changedetection.net/
6	2014	CDnet 2014 [215]		binary	320 × 240 to 720 × 576	22	Univ. Sherbrooke, Canada	http://changedetection.net/
7	2015	SBI2015 [125]		binary	146 × 150 to 2272 × 1704	14	National Research Council, Italy	https://sbmi2015.na.icar.cnr.it/SBIdataset.html
8	2016	DAVIS 2016 [164]		binary	720 × 480 to 3840 × 2159	50	ETH Zurich, Disney Research	https://davischallenge.org/davis2016/code.html
9	2017	CAMO-UOW [103]		binary	1600 × 1200 to 1920 × 1080	10	University of Wollongong, Australia	https://sites.google.com/view/wanqingli/data-sets/uow-camo
10	2017	SBMnet [87]		binary	240 × 240 to 800 × 600	79	University of Sherbrooke, Canada	http://pione.dinf.usherbrooke.ca/
11	2020	CDNet MotionRec [129]		binary	608 × 608	19	Malaviya National Institute of Technology Jaipur, INDIA	https://github.com/murari023/MotionRec

Table 3: Continued.

No.	Year	Dataset name	Scenes	Label	Resolution	Videos	Provider	Access link
12	2000–2017	PETS [238]		multi-class	-	-	University of Reading, UK	https://cs.binghamton.edu/~mrldata/pets2009
13	2014	FBMS-59 [149]		multi-class	640 × 480	59	University of Freiburg, Germany	https://lmb.informatik.uni-freiburg.de/resources/datasets/moseg.en.html
14	2016	LASIESTA [44]		multi-class	352 × 288	48	UPM, Madrid, Spain	https://www.gti.ssr.upm.es/data/lasiesta_database.html
15	2017	DAVIS 2017 [165]		multi-class	720 × 480 to 3840 × 2160	150	ETH Zurich, Disney Research	https://davischallenge.org/davis2017/code.html
16	2018–2021	YouTube-VOS [227]		multi-class	1920 × 1080	4000+	Youtube	https://youtube-vos.org/
17	2007	UCSD [126]	**Outdoor**	binary	242 × 156 to 468 × 348	18	University of California - San Diego, USA	http://www.svcl.ucsd.edu/projects/background_subtraction/ucsdbgsub_dataset.htm
18	2012	BMC [201]		binary	640 × 480	29	Univ. Puy en Velay, France, Université d'Auvergne	http://backgroundmodelschallenge.eu/
19	2013	Segtrack v2 [100]		binary	640 × 360	14	Ohio State University, USA	https://web.engr.oregonstate.edu/~lif/SegTrack2
20	2005	OSU Thermal Pedestrian [45]	**Thermal**	binary	360 × 240	10	Ohio State University, USA	https://vcipl-okstate.org/pbvs/bench/
21	2005	Terravic Motion IR [138]		binary	320 × 240	18	Ohio State University, USA	https://vcipl-okstate.org/pbvs/bench/

Table 3: Continued.

No.	Year	Dataset name	Scenes	Label	Resolution	Videos	Provider	Access link
22	2007	OSU color-thermal [46]		binary	320 × 240	6	Ohio State University, USA	https://vcipl-okstate.org/pbvs/bench/
23	2013	MOTIID [4]		binary	640 × 480	18	Ohio State University, USA	https://vcipl-okstate.org/pbvs/bench/
24	2014	Pedestrian Infared [16]		binary	480 × 360	4	Ohio State University, USA	https://vcipl-okstate.org/pbvs/bench/
25	2017	REMOTE SCENE IR [233]		binary	480 × 320	12	Guangle Yao, China	https://github.com/JerryYaoGl/BSEvaluationRemoteSceneIR
26	2017	GTFD [99]		binary	320 × 240 to 400 × 296	25	Anhui University, China	request
27	2019	TU-VDN [179]		binary	-	60	Tripura University, India	contact mrinalkantibhowmik@tripurauniv.ac.in, mkb_cse@yahoo.co.in and mkb.cse@gmail.com
28	2014	BU-TIV [223]		multi-class	512 × 512 to 1024 × 512	11	Boston University, USA	https://csr.bu.edu/BU-TIV/BUTIV.html
29	2017	GSM [143]	**RGB-D**	binary	640x480	7	Univ. de les Illes Balears, Spain	https://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html
30	2017	SBM-RGBD [23]		binary	640 × 480	33	National Research Council, Italy	https://rgbd2017.na.icar.cnr.it/SBM-RGBDdataset.html
31	2013	RGB-D RIGID MULTI-BODY [184]		multi-class	640 × 480	3	Univ. of Bonn, Germany	https://www.ais.uni-bonn.de/download/rigidmultibody/
32	2014	FLUXDATA FD-1665 [12]	**Multi-spectral**	binary	658 × 491	5	Université de Bourgogne, France	https://sites.google.com/view/ybenezeth/icra2014

Table 3: Continued.

No.	Year	Dataset name	Scenes	Label	Resolution	Videos	Provider	Access link
33	2010	MuHAvi [178]	**Human Action**	binary	720 × 576	1904	Kingston University Kingston upon Thames, UK	http://velastin.dynu.com/MuHAVi-MAS/
34	2008	CMU-MMAC [196]		multi-class	640 × 480 to 1024 × 768	1	CMU, USA	http://kitchen.cs.cmu.edu/
35	2006	iLids [192]	**Pedestrian**	multi-class	720 × 576	7	AVSS London	https://www.gov.uk/guidance/imagery-library-for- intelligent-detection-systems
36	2009	MIT traffic [214]	**Traffic**	binary	720 × 480	1	The Chinese University of Hong Kong, HongKong	https://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html
37	2003	ATON [167]	**Shadow**	binary	320 × 240	5	Univ. of California, USA	https://aimagelab.ing.unimore.it/visor/video_videosInCategory.asp?blist=1&idcategory=5&iStartFrom=0
38	2015	MARDCT [17]	**Maritime environment**	binary	800 × 240	2000	University of Rome, Italy	https://www.diag.uniroma1.it//~labrococo/MAR
39	2014	Fish4knowledge [90]	**Underwater**	binary	320x240	17	Univ. of Edinburgh, UK	https://homepages.inf.ed.ac.uk/rbf/Fish4Knowledge/
40	2019	UnderwaterCD [200]		binary	1920 × 1080	5	University of Rostock and Fraunhofer IGD Germany	http://underwaterchangedetection.eu/index.html
41	2020	MOR-UAV [130]	**Aerial**	binary	1280 × 720 to 1920 × 1080	30	Malaviya National Institute of Technology Jaipur, INDIA	https://visionintelligence.github.io/Datasets.html
42	2011	VIRAT [150]		multi-class	1920 x 1080	23	Kitware, USA	https://viratdata.org/

zoom), and air turbulence. Each category has four to six videos, resulting in a total of 53 videos. For example, the baseline category has sequences highway, office, pedestrians, and PETS2006. Each video has 900 to 7,000 frames. The videos were captured by different type of cameras ranging from low-resolution IP cameras, mid-resolution camcorders, PTZ cameras, to far- and near-infrared cameras. The spatial resolutions of these videos vary from 240 × 320 to 576 × 720 pixels.

4.1.2 DAVIS 2016 & DAVIS 2017 Datasets

Densely Annotated Video Segmentation (DAVIS) 2016 [164] and DAVIS 2017 [165] are benchmark datasets of video object segmentation. They contain 50 full high-definition (FHD) videos, featuring diverse types of object and camera motion. They include challenging examples with occlusion, motion blur and appearance changes. Accurate pixel-level annotations are provided for the moving objects in all video frames. In DAVIS 2016, binary segmentation labels are provided for each of the 20 videos, and in DAVIS 2017, multi-class segmentation labels are provided for each of the 30 videos.

4.1.3 YouTube-VOS Dataset

YouTube-VOS [227], first released in 2018 in conjunction with a workshop challenge, is the first large-scale benchmark that supports multiple video objects (instance-level) segmentation tasks. The dataset has a total of 197,272 object annotations which is 15 times more than those of DAVIS 2017. The object categories in this dataset include person, animals, vehicles, furniture, and other common categories.

The training set of YouTube-VOS has 3,471 videos with 65 unique object categories. The validation set used to evaluate model performance has 474 videos with 91 unique object categories. Among these 91 object categories, 65 are the “seen” categories which appear in the training set, and the remaining 26 are the “unseen” categories which do not exist in the training set, and can be used to evaluate the generalization capability of MOS algorithms.

4.2 Outdoor Category

4.2.1 BMC Dataset

The Background Models Challenge (BMC) dataset [201] is a benchmark dataset created in 2012 built from both synthetic and real videos. It focuses on outdoor scenarios with weather variations such as wind, sun or rain. The dataset is divided into learning and evaluation subsets. There are two scenes: a street and a rotary. For each scene, there are 5 event types: cloudy without acquisition noise, cloudy with noise, sunny with noise, foggy with noise, and windy with noise. The learning set has 10 synthetic videos, while the evaluation set has 10 synthetic videos and 9 real videos acquired from static cameras in video-surveillance contexts. This dataset can test the influence of some difficulties encountered during the object segmentation phase, such as casted shadows, the presence of a continuous car flow near the surveillance zone, fast light changes in the scene, and the presence of big objects.

4.2.2 UCSD Dataset

The UCSD background subtraction dataset [126] is used for background subtraction in highly dynamic scenes, which are extremely challenging problems. The video scenes are comprised of complicated moving backgrounds and camera motion. The dataset provides video frames in JPEG format and ground-truth masks in MATLAB array form, where 1's indicate foreground pixels and 0's indicate background pixels. Compared to the complete CDnet dataset which has 53 videos, UCSD is a small-scale dataset with only 18 videos. For some sequences, the frames of ground-truth masks are less than the frames in the original video sequences.

4.3 Thermal Category

4.3.1 TU-VDN Dataset

The TU-NVD dataset [179] is provided for MOS tasks in atmosphere-degraded outdoor scenes in night vision. The dataset consists of 60 video sequences under different atmospheric conditions, such as low light, dust, rain, and fog. 54 videos are taken with a forward-looking infrared (FLIR) camera mounted 90 degree alignments on a tripod stand by maintaining 200 meters to 2 kilometers distance from objects, and the other 6 videos are taken with a motion camera mounted on a moving vehicle (20∼30 kilometers/hour) where the objects, camera, and background are moving simultaneously. Each frame contains multiple types of moving objects, e.g., pedestrians, various types of vehicles, bicyclists, motorbikes, trains, and pets.

4.3.2 OSU Thermal Pedestrian Dataset

The OSU thermal pedestrian dataset [45] was captured by a Raytheon 300D thermal sensor at pedestrian intersection on the Ohio State University campus in order to detect persons in thermal imagery. The camera was mounted on the rooftop of an 8-story building. It has 10 video sequences in grayscale containing 284 images in a resolution of 360 × 240 pixels. For the ground truth data, only those people who were at least 50% visible in the image were selected (i.e., highly occluded people were not selected).

4.4 RGB-D Category

4.4.1 SBM-RGBD Dataset

The SBM-RGBD dataset [23] was created in order to evaluate and compare scene background modeling methods for MOS on RGBD videos. It has a diverse set of synchronized color and depth sequences acquired by the Microsoft Kinect sensors. The dataset consists of 33 videos (∼15,000 frames). The length of the videos varies from 70 to 1,400 frames, and the spatial resolution is 640 × 480. The depths are recorded at either 16 or 8 bits. The videos span 7 categories, selected to include diverse background modeling challenges related only to the RGB channels (RGB), related only to the depth channel (D), or related to all the channels (RGB+D). For example, the "Illumination Changes" category includes challenges related only to the RGB channels, the "Out of Sensor Range" category includes challenges related only to the depth channel, and the "Shadows" category includes challenges related to all the channels.

4.5 Multi-Spectral Category

4.5.1 FLUXDATA FD-1665 Dataset

Different from datasets captured by a normal camera sensor with RGB channels, FluxData FD-1665 dataset [12] was captured by the FD-1665 multi-spectral camera system. It has 1 indoor and 4 outdoor video sequences containing between 250 and 2,300 video frames. This dataset was created in order to investigate the use of multi-spectral videos of more than three channels (red, green, blue) for background subtraction. Multi-spectral imaging can allow the extraction of additional information which human vision cannot capture with just red, green, and blue receptors.

4.6 Underwater Category

4.6.1 UnderwaterCD Dataset

The Underwater Change Detection dataset [200] is a collection of 5 videos with pixel-level manually segmented ground-truth masks. This dataset was created to evaluate MOS algorithms against the difficulties of an underwater environment. The moving objects in the videos are always fish, which swim in swarms or separately. In the segmentation masks, fish are considered as foreground labeled as 0, all others are considered as background labeled as 255, and unsure classes are labeled as 120. Many special difficulties of an underwater environment are present in the videos, such as marine snow (challenging weather), illumination variations, dynamic background, strong shadows, camouflage, and bad lighting conditions.

There are also other datasets designed to be applied in specific application scenarios. For example, ATON [167] is a dataset with fast motion of cars on a highway and with strong shadows and small camera jitter. CAVIAR is a dataset [52] specifically for human detection in entrance lobby, and MarDCT (Maritime Detection, Classification, and Tracking) [17] is to evaluate different computer vision algorithms such as foreground segmentation and object detection in maritime environment. The foreground masks, object bounding boxes, and identification numbers are included as ground-truth annotations in the dataset.

5 Metrics

While existing MOS papers only introduced a limited number of performance evaluation metrics, in this section, we provide a comprehensive summary of both segmentation accuracy metrics and model efficiency evaluation metrics.

5.1 Segmentation Accuracy

5.1.1 Models Trained on Change Detection Datasets

Some MOS models are trained and evaluated on traditional change detection or background subtraction datasets, such as CDnet 2014 [215] and BMC [201]. The objective focuses on detecting all foreground moving objects from the background and generating binary segmentation masks. To evaluate the detection accuracy, the following metrics have been adopted.

1. **Precision** P and **Recall** R: They are defined based on true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) of the foreground segmentation algorithms. TP is the number of correctly detected foreground pixels, TN is the number of correctly detected background pixels, FP is the number of background pixels detected as foreground and FN is the number of foreground pixels detected as background. Given these four quantities calculated based on the full frames of ground-truth masks and full frames of predicted masks, the recall and precision are defined as $R = \frac{TP}{TP+FN}$ and $P = \frac{TP}{TP+FP}$, respectively. Recall R quantifies the model's capability of identifying ground-truth foreground pixels, while precision P quantifies the accuracy of predicted foreground pixels. A good MOS model should generate a high recall without sacrificing precision.

2. ***F*-measure:** In most existing works, the F-measure defined below is reported to represent a balance between the recall R and precision P:

$$F\text{-measure} = \frac{2 \times P \times R}{P + R} \tag{1}$$

or

$$F\text{-measure} = \frac{TP}{\frac{1}{2}(FN + FP) + TP}. \tag{2}$$

3. ***F*-measure Mean**: The F-measure Mean is obtained by taking the average of F-measure values over all frames in a video sequence.

4. ***E*-measure**: While the F-measure only quantifies pixel-level detection errors, cognitive vision studies have shown that human vision is highly sensitive to both global information and local details in scenes. To this end, the enhanced-alignment measure (E-measure) was recently proposed [56] for binary foreground map evaluation. It jointly captures image-level statistics and local pixel matching information with a newly defined alignment matrix.

   Let I be the ground-truth foreground map (GT) or the predicted foreground map (FM), that is, $I \in \{GT, FM\}$. $I(x, y)$ is the element of I at the (x, y) location, μ_I is the global mean of I. Then, the (x, y)-th element of the mean-shifted foreground map is

$$\varphi_I(x, y) = I(x, y) - \mu_I. \tag{3}$$

   An alignment matrix ξ_{FM} measures the similarity between φ_{GT} and φ_{FM}, and its (x, y)-th element is defined as

$$\xi_{FM}(x, y) = \frac{2\varphi_{GT}(x, y)\varphi_{FM}(x, y)}{\varphi^2_{GT}(x, y) + \varphi^2_{FM}(x, y)}. \tag{4}$$

   Since $\xi_{FM}(x, y)$ is the element-wise similarity between GT and FM, it captures local pixel matching information. Besides, $\varphi_{GT}(x, y)$ and $\varphi_{FM}(x, y)$ depend on global means μ_{GT} and μ_{FM}, which capture global statistics. Further, a convex function of $\xi_{FM}(x, y)$ is used to define an enhancement alignment matrix ϕ_{FM}, the (x, y)-th element of which is

$$\phi_{FM}(x, y) = \frac{1}{4}\big(1 + \xi_{FM}(x, y)\big)^2. \tag{5}$$

   Finally, the E-measure is defined as:

$$E\text{-measure} = \frac{1}{w \times h} \sum_{x=1}^{w} \sum_{y=1}^{h} \phi_{FM}(x, y), \tag{6}$$

where h and w are the height and width of the predicted foreground map FM.

5. S**-measure**: For salient object detection where the purpose is to accurately detect and segment the most attractive object in a scene, a new metric S-measure was proposed [55] to evaluate the similarity between the predicted saliency map (SM) and the ground-truth object mask (GT). The method first evaluates the region-aware structural similarity S_r and the object-aware structural similarity S_o between SM and GT, then the S-measure is calculated as

$$S\text{-measure} = \alpha \times S_o + (1 - \alpha) \times S_r, \tag{7}$$

where $\alpha \in [0, 1]$ is a weight to balance the contribution of S_o and S_r.

More specifically, the region-aware structural similarity S_r measures the similarity between SM and GT block by block. First, each of SM and GT is divided into K blocks, then the structural similarity measure (SSIM) [218] between the k-th block of SM and the k-th block of GT is calculated as SSIM(k), and S_r is defined as

$$S_r = \sum_{k=1}^{K} w_k \times \text{SSIM}(k). \tag{8}$$

The weight w_k assigned to the k-th block is proportional to the ground-truth foreground region this block covers.

In contrast, the object-aware structural similarity S_o measures the similarity of SM and GT holistically. Let x_{FG} represent the predicted foreground probability values in the ground-truth foreground region of SM, and calculate the mean and standard deviation of x_{FG} as $\bar{x}_{FG}$ and $\sigma_{x_{FG}}$, respectively. Then, the object-level foreground similarity between SM and GT is

$$S_{FG} = \frac{2\bar{x}_{FG}}{\bar{x}_{FG}^2 + 1 + 2\lambda \times \sigma_{x_{FG}}}, \tag{9}$$

where $\lambda > 0$ is a constant. Similarly, let x_{BG} represent the predicted background probability values in the ground-truth background region of SM, and calculate the mean and standard deviation of x_{BG} as $\bar{x}_{BG}$ and $\sigma_{x_{BG}}$, respectively. Then, the object-level background similarity between SM and GT is

$$S_{BG} = \frac{2\bar{x}_{BG}}{\bar{x}_{BG}^2 + 1 + 2\lambda \times \sigma_{x_{BG}}}. \tag{10}$$

Note that both S_{FG} and S_{BG} are between 0 and 1. A higher S_{FG} (S_{BG}) indicates SM and GT have more similar foregrounds (backgrounds).

Afterwards, the object-aware structural similarity S_o is calculated as

$$S_o = \mu \times S_{FG} + (1 - \mu) \times S_{BG}, \tag{11}$$

where μ is the ratio of foreground area in GT to image area (width × height).

5.1.2 Models Trained on DAVIS and YouTube-VOS Datasets

Many recent MOS models are trained and evaluated on the DAVIS 2016 [164], DAVIS 2017 [165], and YouTube-VOS [227] datasets. The goal is to perform object-level or instance-level segmentation from a visual recognition or object tracking perspective. The segmentation accuracy is evaluated by the Jaccard index $\mathcal{J}$, contour accuracy $\mathcal{F}$, and the $\mathcal{J}\&\mathcal{F}$ score.

1. **Jaccard Index** $\mathcal{J}$: It is also called the intersection over union (IoU) of the predicted object mask and the ground-truth object mask. The Jaccard index $\mathcal{J}$ has been widely adopted to measure the region similarity between ground-truth objects and segmented objects since its first appearance in PASCAL VOC2008 [54]. For a specific instance class, given the predicted segmentation mask M and the ground-truth object mask G, $\mathcal{J}$ is defined as

$$\mathcal{J} = \frac{|M \cap G|}{|M \cup G|}. \tag{12}$$

With TP, FN, and FP, $\mathcal{J}$ can also be defined as

$$\mathcal{J} = \frac{TP}{FN + FP + TP}. \tag{13}$$

2. **Contour Accuracy** $\mathcal{F}$: It is not measured based on the region of objects, but based on the boundary of objects. For a specific instance class, the boundary recall R_c and boundary precision P_c are first calculated based on the boundaries of objects in the ground-truth masks and those in the predicted masks, then the contour accuracy $\mathcal{F}$ is defined as:

$$\text{Contour Accuracy } \mathcal{F} = \frac{2 \times P_c \times R_c}{P_c + R_c}. \tag{14}$$

3. $\mathcal{J}\&\mathcal{F}$: The overall score $\mathcal{J}\&\mathcal{F}$ is calculated as the average of $\mathcal{J}$ and $\mathcal{F}$ by

$$\mathcal{J}\&\mathcal{F} = \frac{\mathcal{J} + \mathcal{F}}{2}. \tag{15}$$

4. $\mathcal{J}$ **Mean**, $\mathcal{F}$ **Mean**, and $\mathcal{J}\&\mathcal{F}$ **Mean**: For a specific instance class, these three metrics are calculated as the average Jaccard index $\mathcal{J}$, the average contour accuracy $\mathcal{F}$, and the average $\mathcal{J}\&\mathcal{F}$ score over all frames in a video sequence, respectively.

   Besides, for multiple-instance segmentation, the per-class (per-instance) Jaccard index $\mathcal{J}$, contour accuracy $\mathcal{F}$, and $\mathcal{J}\&\mathcal{F}$ score are first calculated, then each of these metrics is averaged over all instance classes. Afterwards, the $\mathcal{J}$ Mean, $\mathcal{F}$ Mean, and $\mathcal{J}\&\mathcal{F}$ Mean are calculated as the average scores over all frames in a video sequence.

5.2 Model Efficiency

Model efficiency is a critical aspect to consider when we develop and deploy MOS models on resource-constrained devices and for real-time applications. We can evaluate the efficiency of a model from the perspectives of model size, inference speed, number of model parameters, and computational complexity.

1. **Model Size**: Model size is measured in megabyte (MB) or gigabyte (GB), which is to quantify the storage space required to store the model in memory or in hard disks. Embedded and mobile devices usually have limited storage space, therefore it is important to design lightweight models to be deployed on these devices.

2. **Inference Speed**: Inference speed is one of the most informative indicators to compare the efficiency of different MOS models. It is measured in frames per second (fps), which refers to the number of video frames that are processed within a second during the inference stage.

3. **Number of Model Parameters**: The number of model parameters is measured in millions (M). It is used as an indirect indicator of model complexity and memory usage (during the inference stage). Trainable parameters are network weights updated by back-propagation during the training process, and they contribute to the prediction power of an MOS model. Parameters that are not updated during the training process are called non-trainable parameters, such as pre-trained weights, pre-defined fixed filters, the number of hidden layers, and the number of nodes in the network.

4. **Computational Complexity**: Computational complexity is measured in floating-point operations (FLOPs). It refers to the number of floating-point multiplication-and-addition operations to run a single instance of a given MOS model. Gigaflops (GFLOPs) is used in this paper, and one gigaflop has one billion FLOPs.

6 Performance Comparison of Efficient MOS Models

Tables 4 to 7 compare the performance of state-of-the-art MOS models in terms of segmentation accuracy and model efficiency on CDnet 2014, DAVIS 2016, DAVIS 2017, and YouTube-VOS datasets. Each column lists the index number, year of publication, method name, GPU, accuracy, inference speed in fps, number of trainable parameters (M), and model size (MB).

The methods in each table are grouped by the GPU models used and GPU models are listed in a descending order of their inference efficiency scores according to the deep learning hardware ranking [1]. Within each GPU group, the methods are listed in a descending order of inference speed. The test images of CDnet 2014 (Table 4), DAVIS 2016 (Table 5), DAVIS 2017 (Table 6), and YouTube-VOS (Table 7) are resized to 240×320, 480×864, 480×864, and 256×448, respectively.

Table 4 shows the performance comparison on the CDnet 2014 dataset. For this dataset, the objective focuses on detecting moving objects as the foreground from the video frames, hence the segmentation accuracy is evaluated by F-measure.

For Table 4, we will discuss the segmentation accuracy, model efficiency (inference speed, trainable parameters, and model size), and their trade-off for different groups of GPUs. For the Titan Xp GPU group, the Lightweight U-Net-like network [111] achieved the highest accuracy (97.7% in F-measure), since the design of long and short skip connections effectively ship the low- and high-level features across the network. Meanwhile, this model achieved the fastest speed of 250 fps and is extremely lightweight, because the short skip connections allow the bottleneck blocks to be very compact and to have very few convolution layers. Models using MIMO and/or separable convolution techniques such as 3DS_MM [75], F3DsCNN [76] and 2D_Separable CNN [74] also have fast inference speeds, few trainable parameters, and small model sizes, while achieving high segmentation accuracy. ChangeDet [133] is a shallow network with reduced feature map volumes, while 3DFR [127] uses only a few 3D filters and most filters are 2D or 1D, therefore these two models are extremely lightweight. However, such lightweight designs led to accuracy degradation, and their multi-input single-output (MISO) strategy reduced the inference speeds.

For the GTX 1080Ti GPU group of Table 4, Frame-Level Weakly Supervised model [139] has a relatively deeper network, hence its trainable parameters are more and model size is bigger. However, the network adopts a single path and two-channel input (a gray-scale frame and estimated background) and only small convolutional kernels (3×3) are used, so it achieved a fast inference speed of 134 fps. Although MvRF-CNN [3], Guided Multi-Scale CNN [107], and 3D CNN-LSTM [2] have few parameters and small model size, their inference speeds are slow, since MvRF-CNN adopts multi-scale processing with larger

Table 4: The segmentation accuracy and efficiency of lightweight MOS methods on CDnet 2014 dataset.

No.	Year	Method	GPU	*F*-measure Mean (%) ↑	Inference Speed (fps) ↑	# Param (M) ↓	Model Size (MB) ↓
1	2020	Edge Aggregation Network [157]	Tesla V100	96.9	19.6	20	240
2	2021	Lightweight U-Net-like [111]	Titan Xp	97.7	250	0.1	0.4
3	2021	3DS_MM [75]		95.2	151	0.4	1.5
4	2020	2D_Separable CNN [74]		91.5	149	1	3.8
5	2021	F3DsCNN [76]		95.9	120	4.3	5
6	2022	ChangeDet [133]		88.0	58.8	0.1	1.6
7	2019	3DFR [127]		86.0	33.3	0.1	2.8
8	2019	Frame-Level Weakly Supervised [139]	GTX 1080Ti	84.8	134	3.7	14.8
9	2019	MvRF-CNN [3]		95.1	42	0.9	4.0
10	2018	Guided Multi-Scale CNN [107]		75.9	28	0.3	1.3
11	2019	3D CNN-LSTM [2]		95.7	24	0.2	0.9
12	2021	BSUV-Net 2.0 [193]	Tesla P100	81.0	29	15.9	110
13	2018	BScGAN [10]	GTX 1080	97.1	400 (BMC dataset)	-	-
14	2019	FCESNet [168]	Titan X	86.0	112	-	-
15	2018	MFCN [240]	GTX 1060	98.7	27	20.8	50
16	2018	FgSegNet_M [96]	GTX 970	98.8	18	6.5	60.4
17	2016	MSCNN+Cascade [216]		95.0	13	0.5	3.8
18	2019	Trip-Net [147]	Titan Black	84.2	282	0.3	2.5
19	2019	BMN-BSN [142]	-	80.0	48	-	-
20	2020	RT-SBS [43]	-	82.8	25	0.6	3.0

kernel sizes such as 5 × 5 and 9 × 9, Guided Multi-Scale CNN is pixel-wise processing, and 3D CNN-LSTM has a 16-channel input (four RGB frames) and relies on 3D convolution, which has high computational complexity. In terms of accuracy, multi-scale processing with larger kernel sizes (MvRF-CNN) and spatio-temporal processing (3D CNN-LSTM) achieved higher F-measure scores, while very shallow network architecture (Guided Multi-Scale CNN) led to much lower accuracy.

For the GTX 970 GPU group of Table 4, FgSegNet_M [96] has more parameters and a larger model size since it's a deeper network with multi-scale processing. The MSCNN+Cascade [216] network is extremely lightweight due to its simple network architecture. However, since it generates segmentation results pixel-by-pixel, the inference speed is slower than FgSegNet_M. Besides, the multi-scale deeper network FgSegNet_M achieved much higher accuracy than the shallower MSCNN+Cascade network, at the expense of more trainable parameters and a larger model size.

In terms of overall performance, the Lightweight U-Net-like model [111], 3DS_MM [75], 2D_Separable CNN [74], F3DsCNN [76], MvRF-CNN [3], and 3D CNN-LSTM [2] achieved relatively higher detection accuracy, faster inference speeds, fewer trainable parameters and smaller model sizes, which demonstrates their potential of performing MOS tasks in delay-sensitive environments and resource-constrained devices.

For DAVIS 2016, DAVIS 2017, and YouTube-VOS datasets, the goal is to segment objects in video frames from a visual recognition or object tracking perspective, therefore the segmentation accuracy is evaluated by the Jaccard index $\mathcal{J}$, contour accuracy $\mathcal{F}$, and the $\mathcal{J}\&\mathcal{F}$ score.

Table 5 shows the object segmentation performance comparison on the DAVIS 2016 dataset. Again, we analyze the trade-off between segmentation accuracy and model efficiency for different groups of GPUs. For Tesla V100, models with faster inference speeds tend to have fewer trainable parameters and smaller model sizes. However, larger models do not always generate higher segmentation accuracy. For example, AOT-T [232] utilizes a long-term attention for matching with the first frame's embedding and a short-term attention for matching with several nearby frames' embeddings, therefore it achieved quite high segmentation accuracy. Meanwhile, it has light-weight backbone encoder and decoder, adopts only 1 layer of the proposed long short-term transformer (LSTT) block, hence its inference speed is also quite fast and it has few parameters and small model size. In contrast, RMNet [225] has a larger model but its accuracy is not as competitive as AOT-T [232].

For Titan Xp in Table 5, again the faster model RANet [219] has fewer trainable parameters and smaller model size. Although the model size of RANet is smaller than that of DDEAL (Res101) [236], its segmentation accuracy is on par with that of DDEAL (Res101), since RANet applies a ranking system to the

Table 5: The segmentation accuracy and efficiency of lightweight MOS methods on DAVIS 2016 dataset.

No.	Year	Method	GPU	*J&F* Mean (%) ↑	*J* Mean (%) ↑	*F* Mean (%) ↑	Inference Speed (fps) ↑	# Param (M) ↓	Model Size (MB) ↓
1	2021	AOT-T [232]	Tesla V100	86.8	86.1	87.4	51.4	5.7	23.3
2	2020	FRTM-fast (ResNet-18) [170]		78.5	-	-	41.3	13.2	53.9
3	2020	FRTM (Resnet-101) [170]		83.5	-	-	22	46.9	187.6
4	2021	RMNet [225]		81.5	80.6	82.3	12	53	202
5	2020	SAT [32]	RTX 2080 Ti	83.1	82.6	83.6	39	72	288
6	2019	RANet [219]	Titan Xp	85.5	85.5	85.4	30.3	61.5	246
7	2021	DDEAL(Res101) [236]		85.4	85.1	85.7	25	125	536
8	2019	DTN [245]	GTX 1080Ti	83.6	83.7	83.5	14.3	-	-
9	2020	Fasttmu [188]		78.9	77.5	80.3	11**	5.2	37.8
10	2019	SiamMask(ResNet-50) [210]	RTX 2080	69.75	71.7	67.8	55	27	105.9
11	2021	SwiftNet(ResNet-18) [206]	Tesla P100	90.1	90.3	89.9	70**	32.5	130
12	2021	SwiftNet(ResNet-50) [206]		90.4	90.5	90.3	25**	37.4	149
13	2020	GC [105]	Tesla P40	86.6	87.6	85.7	25	45.9	175
14	2021	TTVOS (HRNet) [154]	-	81.1	-	-	78.3	1.61	6
15	2021	G-FRTM-fast (RN18, $\tau = 1$) [155]	-	80.9	-	-	37.6	-	-
16	2020	MSN [222]	-	84.35	83.8	84.9	10	-	-

Note: **The inference speeds of these models were evaluated on the DAVIS 2017 dataset in the original papers.

matching process between multiple templates and the input to extract reliable segmentation results. While DDEAL (Res101) learns static cues from the labeled first frame and dynamically updates cues of the subsequent frames for good object segmentation accuracy, it adopts a heavier backbone ResNet-101, therefore the model size and parameters are quite big.

For Tesla P100 in Table 5, SwiftNet (ResNet-18) has faster speed, fewer parameters and smaller model size than SwiftNet (ResNet-50) [206] since it adopts a lighter backbone ResNet-18. Besides, SwiftNet achieved the highest accuracy among all models in Table 5, and it provides quite high inference speed (considering that it was run on a less powerful GPU). The reason is, as a real-time template-based VOS model, SwiftNet updates fewer frames in memory and adaptively selects incremental frames that have variations for memory update and ignores static ones. Besides, it abandons full-frame operations and incrementally processes with temporally varying pixels.

In Table 5, another model which also achieved a good trade-off between accuracy and model efficiency is TTVOS (HRNet) [154]. It is a template-based approach with a lightweight backbone network HRNet. To improve the segmentation accuracy, it utilized short-term matching to enhance target object localization, and utilized long-term matching to improve fine details and handle object shape-changing. Moreover, it proposed a new temporal consistency loss for better temporal coherence between neighboring frames.

Table 6 evaluates model performance on the DAVIS 2017 dataset. We also analyze the trade-off between segmentation accuracy and model efficiency for models in different GPU groups. For Tesla V100, AOT-T [232] has faster inference speed, fewer trainable parameters and smaller model size than RMNet [225], while RMNet offers higher accuracy (83.5% in $\mathcal{J}\&\mathcal{F}$ Mean) at the expense of worse model efficiency.

For RTX 2080Ti in Table 6, SAT-fast [32] achieved faster inference speed, has fewer parameters and smaller model size than SAT [32], but SAT achieved higher segmentation accuracy (72.3% in $\mathcal{J}\&\mathcal{F}$ Mean).

For Titan Xp in Table 6, the inference speeds among AGSS [112], TVOS [247], and PiWiVOS-F [152] are increasing, while their trainable parameters and model sizes are decreasing. Although TVOS [247] is not the largest model among the three, it offers the highest segmentation accuracy (72.3% in $\mathcal{J}\&\mathcal{F}$ Mean), because it proposed a label propagation approach which attempts to capture information all the way from the first frame to the frame preceding the current frame, therefore its segmentation accuracy is enhanced. To limit the computational overhead, TVOS [247] performed sampling densely within the recent history and sparsely in the more distant history, yielding a model that accounts for object appearance variation while reducing temporal redundancy. SwiftNet (ResNet-50) and SwiftNet (ResNet-18) [206] in the Tesla P100 GPU group were already analyzed during the discussion of Table 5 models.

Table 6: The segmentation accuracy and efficiency of lightweight MOS methods on DAVIS 2017 dataset.

No.	Year	Method	GPU	*J&F* Mean (%) ↑	*J* Mean (%) ↑	*F* Mean (%) ↑	Inference Speed (fps) ↑	# Param (M) ↓	Model Size (MB) ↓
1	2021	AOT-T [232]	Tesla V100	72	68.3	75.7	51.4	5.7	23.3
2	2020	LWL [13]		70.8	68.2	73.5	14	-	-
3	2021	RMNet [225]		83.5	81	86	12*	53	202
4	2020	SAT-fast [32]	RTX 2080Ti	69.5	65.4	73.6	60	16	64
5	2020	SAT [32]		72.3	68.6	76	39	72	288
6	2019	PiWiVOS-F [152]	Titan Xp	54.9	55.7	54	85	8	31.3
7	2020	TVOS [247]		72.3	69.9	74.7	37	12.7	50
8	2019	RANet [219]		65.7	63.2	68.2	30*	61.5	246
9	2019	AGSS [112]		67.4	64.9	69.9	10	48	197
10	2019	DTN [245]	GTX 1080Ti	67.4	64.2	70.6	14.3*	-	-
11	2020	Fasttmu [188]		70.6	69.1	72.1	11	5.2	37.8
12	2019	SiamMask(ResNet-50) [210]	RTX 2080	56.4	54.3	58.5	55	27	105.9
13	2021	SwiftNet (ResNet-18) [206]	Tesla P100	77.8	75.7	79.9	70	32.5	130
14	2021	SwiftNet (ResNet-50) [206]		81.1	78.3	83.9	25	37.4	149
15	2020	GC [105]	Tesla P40	71.4	69.3	73.5	25*	45.9	175
16	2021	TTVOS(HRNet) [154]	-	62.1	-	-	78.3*	1.61	6
17	2021	TTVOS(ResNet-50) [154]	-	69.5	-	-	37.7*	14.8	159
18	2021	G-FRTM-fast (RN18, $\tau = 1$) [155]	-	71.7	-	-	37.6*	-	-
19	2020	MSN [222]	-	74.1	71.4	76.8	10	-	-

Note: *The inference speeds of these models were evaluated on the DAVIS 2016 dataset in the original papers.

Table 7: The segmentation accuracy and efficiency of lightweight MOS methods on YouTube-VOS dataset.

No.	Year	Method	GPU	J&F Mean (%) ↑	Seen J Mean (%) ↑	Seen F Mean (%) ↑	Unseen J Mean (%) ↑	Unseen F Mean %) ↑	Inference Speed (fps) ↑	# Param (M) ↓	Model Size (MB) ↓
1	2020	FRTM-fast (ResNet-18) [170]	Tesla V100	65.7	68.6	71.3	58.4	64.5	41.3*	13.2	53.9
2	2021	AOT-T [232]		79.7	79.6	83.8	73.7	81.8	41	5.7	23.3
3	2020	FRTM [170]		72.1	72.3	76.2	65.9	74.1	21.9*	46.9	187.6
4	2021	RMNet [225]		81.5	82.1	85.7	75.7	82.4	12*	53	202
5	2020	SAT [32]	RTX 2080 Ti	63.6	67.1	55.3	70.2	61.7	39*	72	288
6	2021	TVOS [247]	Titan Xp	67.8	67.1	69.4	63	71.6	37**	12.7	50
7	2019	AGSS [112]		71.3	71.3	75.2	65.5	73.1	12.5	48	197
8	2019	SiamMask(ResNet-50) [210]	RTX 2080	52.8	60.2	58.2	45.1	47.7	55*	27	105.9
9	2021	SwiftNet(ResNet-18) [206]	Tesla P100	73.2	73.3	76.3	68.1	75	70**	32.5	130
10	2021	SwiftNet(ResNet-50) [206]		77.8	77.8	81.8	72.3	79.5	25**	37.4	149
11	2020	LSTNet [207]	Titan X Pascal	71.8	70.9	74.9	66.8	74.8	12.8	-	-
12	2020	GC [105]	Tesla P40	73.2	72.6	75.6	68.9	75.7	25*	45.9	175
13	2021	G-FRTM-fast (RN18, $\tau = 1$) [155]	-	60.9	65.1	66.7	53	58.8	37.6*	-	-
14	2020	MSN [222]	-	71.1	72.4	75.2	65.4	71.4	10*	-	-

Note: *The inference speeds of these models were evaluated on the DAVIS 2016 dataset in the original papers. ** The inference speeds of these models were evaluated on the DAVIS 2017 dataset in the original papers.

Table 7 evaluates model performance on the YouTube-VOS dataset. As introduced earlier in Section 4, the validation set of YouTube-VOS has 65 "seen" object categories which appeared in the training set, and 26 "unseen" object categories which did not exist in the training set. To evaluate the segmentation accuracy, the Jaccard index $\mathcal{J}$ and contour accuracy $\mathcal{F}$ are each calculated independently for the "seen" and "unseen" category, then the four resultant metrics are averaged as the overall $\mathcal{J}\&\mathcal{F}$ score.

For Table 7, we again analyze the trade-off between model accuracy and efficiency for models in different groups of GPU. For Tesla V100, the inference speeds of RMNet [225], FRTM [170], and FRTM-fast (ResNet-18) [170] are increasing, while their trainable parameters and model size are decreasing. Besides, the larger model RMNet [225] achieved much higher accuracy (81.5% in $\mathcal{J}\&\mathcal{F}$ Mean) than smaller models FRTM and FRTM-fast (ResNet-18). The reason that FRTM and FRTM-fast (ResNet-18) [170] are lightweight and faster is, these methods take previous segmentation masks as training data to learn a lightweight target appearance model during the inference stage using fast optimization.

For Titan Xp in Table 7, AGSS [112] has more parameters and larger model size than TVOS [247], correspondingly, its segmentation accuracy is higher (71.3% in $\mathcal{J}\&\mathcal{F}$ Mean). The SwiftNet models [206] in Tesla P100 GPU group were already discussed earlier.

In terms of the overall performance of models in Table 7, AOT-T [232], SwiftNet (ResNet-18) and SwiftNet (ResNet-50) [206] offer good trade-off among segmentation accuracy (79.7%, 73.2%, 77.8% in $\mathcal{J}\&\mathcal{F}$ Mean, respectively), trainable parameters (5.7 M, 32.5 M, 37.4 M, respectively), and model sizes (23.3 MB, 130 MB, 149 MB, respectively).

We also provide a comprehensive list of the access links of lightweight and non-lightweight MOS models with implementation languages in Tables 8 and 9.

7 Challenges and Future Directions

In this section, we identify existing challenges in the field of MOS indicated by the deep learning-based methods reviewed in this paper, and present future research directions.

7.1 Automatic Neural Architecture Search

From the analysis of efficient model design techniques in Section 3, we observe that well-designed network architectures can improve the efficiency while maintaining the accuracy of MOS models, such as carefully selected number of network layers and feature map channels, two-branch network, using smaller decoder size than the encoder size, multi-input multi-output structures, etc.

Table 8: The access links of lightweight MOS models.

No.	Year	Model	Author	Implementation	Code and Result Link
1	2016	MSCNN+Cascade [216]	Yi Wang *et al.*	Matlab	https://github.com/zhimingluo/MovingObjectSegmentation
2	2018	FgSegNet_M [96]	Long Ang Lim *et al.*	Keras	https://github.com/lim-anggun/FgSegNet
3	2018	FgSegNet_v2 [110]	Long Ang Lim *et al.*	Keras	https://github.com/lim-anggun/FgSegNetv2
4	2019	3D CNN-LSTM [2]	Akilan *et al.*	Keras	https://github.com/nalika/A-3D-CNN-LSTM-Based-Image-to-Image-Foreground-Segmentation
5	2019	AGSS [112]	Huaijia Lin *et al.*	Pytorch	https://github.com/Jia-Research-Lab/AGSS-VOS
6	2019	AGAME [88]	Joakim Johnander *et al.*	Pytorch	https://github.com/joakimjohnander/agame-vos
7	2019	RANet [219]	Ziqin Wang *et al.*	Pytorch	https://github.com/Storife/RANet/
8	2019	SiamMask(ResNet-50) [210]	Qiang Wang *et al.*	Pytorch	https://github.com/foolwood/SiamMask
9	2019	MvRF-CNN [3]	Akilan *et al.*	Keras	https://github.com/taimurhassan/inc-inst-seg
10	2020	SAT-Fast [32]	Xi Chen *et al.*	Pytorch	https://github.com/MegviiDetection/video_analyst
11	2020	SAT [32]	Xi Chen *et al.*	Pytorch	https://github.com/MegviiDetection/video_analyst
12	2020	FRTM-fast [170]	Andreas Robinson *et al.*	Pytorch	https://github.com/andr345/frtm-vos
13	2020	Fasttmu [188]	Mingjie Sun *et al.*	Pytorch	https://github.com/insomnia94/FTMU
14	2020	TVOS [247]	Yizhuo Zhang *et al.*	Pytorch	https://github.com/microsoft/transductive-vos.pytorch

Table 8: Continued.

No.	Year	Model	Author	Implementation	Code and Result Link
15	2020	LWL [13]	Goutam Bhat *et al.*	Pytorch	https://github.com/visionml/pytracking
16	2020	MSN [222]	Ruizheng Wu *et al.*	Pytorch	https://github.com/dvlab-research/MSN
17	2020	RT-SBS [43]	Anthony Cioppa *et al.*	Pytorch	https://github.com/cioppaanthony/rt-sbs
18	2020	2D_Separable CNN [74]	Bingxin Hou *et al.*	Tensorflow	https://github.com/houbingxin/2DsepMOD
19	2021	BSUV-Net 2.0 [193]	Ozan Tezcan *et al.*	Pytorch	https://github.com/ozantezcan/BSUV-Net-2.0
20	2021	AOT-T [232]	Zongxin Yang *et al.*	Pytorch	https://github.com/z-x-yang/AOT
21	2021	SwiftNet (ResNet-18) [206]	Haochen Wang *et al.*	Pytorch	https://github.com/haochenheheda/SwiftNet
22	2021	SwiftNet (ResNet-50) [206]	Haochen Wang *et al.*	Pytorch	https://github.com/haochenheheda/SwiftNet
23	2021	G-FRTM [155]	Andreas Robinson *et al.*	Pytorch	https://github.com/HYOJINPARK/Reuse_VOS
24	2021	TTVOS(HRNet) [154]	Hyojin Park *et al.*	Pytorch	https://github.com/HYOJINPARK/TTVOS
25	2021	TTVOS(ResNet-50) [154]	Hyojin Park *et al.*	Pytorch	https://github.com/HYOJINPARK/TTVOS
26	2021	RMNet [225]	Haozhe Xie *et al.*	Pytorch	https://github.com/hzxie/RMNet
27	2021	DDEAL(Res101) [236]	Yingjie Yin *et al.*	Pytorch	https://github.com/YingjieYin/Directional-Deep-Embedding-and-Appearance-Learning-for-Fast-Video-Object-Segmentation
28	2021	3DS_MM [75]	Bingxin Hou *et al.*	Pytorch	https://github.com/houbingxin/3DSMM
29	2021	F3DsCNN [76]	Bingxin Hou *et al.*	Pytorch	https://github.com/houbingxin/F3DsCNN

Table 9: The access links of non-lightweight MOS models.

No.	Year	Model	Author	Implementation	Code and Result Link
1	1999	GMM [182]	Chris Stauffer *et al.*	Matlab	https://github.com/SEHAIRIKamal/A-Matlab-Background-Subtraction-Library
2	2004	GMM Zivkovic [262]	Z. Zivkovic *et al.*	Matlab	https://github.com/SEHAIRIKamal/A-Matlab-Background-Subtraction-Library
3	2011	DECOLOR [259]	Xiaowei Zhou *et al.*	Matlab	https://github.com/GreenTeaHua/DECOLOR-/blob/master/decolor.zip
4	2011	ViBe [11]	O. Barnich *et al.*	C++	http://www.telecom.ulg.ac.be/research/vibe/
5	2014	SuBSENSE [29]	St-Charles, P.-L. *et al.*	C++	https://github.com/ethereon/subsense
6	2014	OR-RPCA [85]	Sajid Javed *et al.*	Matlab	https://github.com/andrewssobral/lrslibrary
7	2015	PAWCS [28]	Pierre-Luc St-Charles *et al.*	C++	http://bitbucket.org/pierrelucstcharles/pawcs
8	2017	Gaussian Models [70]	Peter Henderson *et al.*	C++	https://github.com/Breakend/MotionDetection
9	2017	IUTIS-5 [14]	S. Bianco *et al.*	-	http://jacarini.dinf.usherbrooke.ca/method/227/
10	2017	WeSamBE [86]	S Jiang *et al.*	C++	https://github.com/aimeng100/WeSamBE
11	2017	DeepBS [8]	M. Babaee *et al.*	C++	https://github.com/Babaee/DeepBS
12	2017	Modified ConvNet [42]	Lucas P. Cinelli,	Pytorch	https://github.com/lpcinelli/foreground-segmentation

Table 9: Continued.

No.	Year	Model	Author	Implementation	Code and Result Link
13	2017	FusionSeg [51]	Suyog Dutt Jain *et al.*	Matlab	https://github.com/suyogduttjain/fusionseg
14	2017	CTN [82]	W.-D. Jang *et al.*	Matlab	http://mcl.korea.ac.kr/~dotol1216/CVPR2017_CTN
15	2017	MaskTrack [163]	F. Perazzi *et al.*	-	https://davischallenge.org/results/2016/msk.zip
16	2017	PLM [237]	J. Shin Yoon *et al.*	Matlab	https://jsyoon4325.wixsite.com/pix-matching
17	2018	3DAtrous [78]	Zhihang Hu *et al.*	Tensorflow	https://github.com/lthngan/3D_Atrous_Ressidual_Network
18	2018	BScGAN [10]	M. C. Bakkay *et al.*	-	http://jacarini.dinf.usherbrooke.ca/method/486/
19	2018	PReMVOS [123]	J. Luiten, P. *et al.*	Pytorch	https://github.com/JonathonLuiten/PReMVOS
20	2018	FAVOS [37]	J. Cheng *et al.*	Caffe	https://github.com/JingchunCheng/FAVOS
21	2019	BSUV-Net [194]	Ozan Tezcan *et al.*	Pytorch	https://github.com/ozantezcan/BSUV-Net-inference
22	2019	Illuminination BGS [172]	Dimitrios Sakkos *et al.*	Keras	https://github.com/dksakkos/illumination_augmentation
23	2019	STM [151]	Seoung Wug Oh *et al.*	Pytorch	https://github.com/seoungwugoh/STM
24	2020	GraphMOS [63]	Jhony H Giraldo *et al.*	Pytorch	https://github.com/jhonygiraldo/GraphMOS
25	2020	MotionRec [129]	Murari Mandal *et al.*	Keras	https://github.com/lav-kush/MotionRec
26	2020	GraphBGS [62]	Jhony H. Giraldo *et al.*	Matlab	https://github.com/jhonygiraldo/GraphBGS

However, it is difficult to select the best network architecture purely by experience.

Neural Architecture Search (NAS) designed by Google is currently a very hot sub-topic of Automated Machine Learning (AutoML), aiming at automatically designing neural network architectures, hence minimizing the reliance on expert experience and knowledge. It has been applied to object detection and semantic segmentation, such as the Auto-DeepLab [114] and the Customizable Architecture Search (CAS) [246]. Besides, it has been utilized to find lightweight semantic image segmentation networks [113, 146]. However, rare effort has been put in MOS yet. Therefore, NAS-based methods would be a valuable future research direction in the area of MOS.

7.2 Transformer-Based MOS Models

Among the deep-learning-based MOS methods reviewed in this paper, most are using CNN structures. Although CNN is successful in extracting local features, it is weak at capturing long-term temporal dependencies. While the recurrent neural network (RNN) is able to explore long-term correlations, it suffers from long back-propagation process due to the seriality of recurrent structures.

In recent years, the Transformer [148] has emerged as a popular architecture to explore the global correlations among a sequence of inputs. The potential of transformers for video object segmentation has been recently studied [50, 136, 232], but it is not thoroughly investigated. Besides, it is prohibitively expensive to scale transformers to long sequences, because self-attention mechanism has quadratic time and memory complexities with respect to the input sequence length [260]. More and more research focuses on how to reduce the complexity of transformer-based models. Some transformer-based models have been proved to outperform previous models in both accuracy and inference speed in semantic segmentation. For example, SETR [252] deploys a pure transformer to encode an image as a sequence of patches with a simple decoder. Swin Transformer [119] adopts a shifted windowing scheme, which brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. Seg-B/L [183] is built on the Vision Transformer (ViT) with a point-wise linear decoder. Due to these successful examples, developing transformer-based MOS models along with complexity reduction schemes would be a promising future research direction.

7.3 The Generalization Capability of MOS Models

Existing MOS algorithms have achieved good segmentation accuracy when the training and test sets have the same or similar distribution. However, the performance could severely deteriorate when the test set has unknown distribution shifts, which is the domain shift problem. This was demonstrated in Table 7, where the Jaccard Index $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ of the "Unseen" scenario are significantly worse than the "Seen" scenario. In future research, it is important to find solutions to improve the generalization capability of MOS models, which refers to the models' ability to adapt properly to new, previously unseen data [60]. The following are three methods that can be considered.

7.3.1 Domain Adaptation

Domain adaptation (DA) is the technique to adapt a model learned from training data (source domain) to test data (target domain). It can reduce performance degradation caused by domain shift. DA techniques have been applied to tasks such as image classification [57, 199] and semantic segmentation [7, 83, 106], but it was less frequently used in VOS tasks, due to the challenge of high complexity of video data. DAVOS [242] applied adversarial DA techniques to the VOS task with a domain confusion loss for unsupervised training in the target domain. Another example is VOSTR [33], which proposed a self-learning framework to segment objects in unseen videos. It consists of three steps: (1) refining responses of the trained source model, (2) selecting object-like proposals via a segment mining module, and (3) learning a CNN model with a transferable module for adapting seen categories in the source domain to the unseen target video. In this way, existing annotations in source images are exploited and visual information is transferred to segment videos with unseen object categories, without using any annotations in the target video. In [239], conditional GAN (cGAN) and DA were utilized to adapt a background subtraction model trained on the CDnet 2014 natural images to the target domain of very high resolution (VHR) optical remote sensing videos. It significantly improved the F-measure of the segmented foreground mask for the target domain.

7.3.2 Continual Learning

Another way to improve models' transferability from old data to new data is continual learning, in which machine learning models are adapted to continuous streams of information [153]. It has been applied to semantic segmentation [49] and object detection [176, 243], where old models are updated by sequentially adding new classes [49]. For example, a deep model consolidation (DMC) module is proposed in [243] to tackle continual learning of image classification and object detection with a distillation-based method. Besides, these three works [49, 176, 243] also properly address the "catastrophic forgetting" problem commonly seen in continual learning, which refers to an abrupt degradation of performance on the original dataset, when the training objective is adapted to the new data.

Nevertheless, continual learning has not been applied to MOS problems yet, hence it would be an interesting future research direction. For applications on mobile and embedded devices, in particular, the memory to store data is limited, hence continual learning becomes quite important because it can learn from only the new data while the old data can be discarded. Besides, model design is important to continual learning as well, because the choice of architecture can significantly impact the continual learning performance, and different architectures lead to different trade-offs between the ability to remember previous tasks and learning new ones [140].

7.3.3 General Dataset

In the future, more general video datasets are also desired to improve the robustness and generalization capability of trained models, and to facilitate fair performance comparison among different models. For example, the training set is desired to have diverse scene content and various conditions such as noisy and compressed frames, diverse appearance and trajectory of the moving objects. When models are trained on a dataset with a specific video scene, such as dynamic background or extreme weather, this would limit the generalization capability of the trained model, unless the model is purposely designed for specific use. Besides, for performance evaluation and comparison, it is desired to create benchmark training sets such that different models can be trained on the same data, resulting in a fair performance comparison among different models. Although some existing datasets are collections of assembled public datasets, such as SBI2015 [125] and SBM-RGBD [23], in the future, more general video datasets are still needed.

7.4 Unsupervised Learning on Unlabeled Data

Most existing deep learning-based MOS methods are supervised learning, which requires extensive amounts of annotated data to improve the performance and to avoid over-fitting. However, obtaining pixel-wise segmentation labels is labor-intensive and expensive. Therefore, unsupervised learning is desired to address this problem. It is noteworthy that unsupervised learning here means that ground-truth segmentation masks are not available for training, while the unsupervised VOS discussed in Sections 1 and 2 refers to the lack of human intervention or manual annotation at test time, but it does require ground-truth labels to train the model.

Recently, a few unsupervised learning methods have emerged for MOS. For example, MuG [122] models video object patterns by comprehensively exploring supervision signals from different granularities of unlabeled videos. The effectiveness of this approach was demonstrated in both unsupervised VOS and semi-supervised VOS. The adversarial contextual model [231] achieved better or similar performance compared to prior unsupervised learning methods, and even edged out methods that rely on supervised pre-training. Nevertheless, compared to supervised learning, its segmentation accuracy is still not competitive. Also, it was only applied to single-instance MOS datasets, such as DAVIS 2016, and how it will perform on multi-instance segmentation scenarios is unknown. Such kind of unsupervised learning is a future direction for MOS research.

7.5 Knowledge Distillation

Knowledge distillation [71] is a well-studied model compression technique. It learns a small "student" model to mimic a large "teacher" model and to leverage the knowledge source of the teacher to obtain similar or higher prediction accuracy. The small student model can achieve a faster inference speed and be deployed on devices with limited resources, e.g., mobile phones and embedded devices. Knowledge distillation has been extensively applied to image classification [31, 226, 243], object detection [30, 243], and semantic segmentation [229]. For example, knowledge distillation is leveraged in an incremental learning framework in [243] for image classification and object detection. The proposed deep model consolidation (DMC) module adopts a novel double distillation training loss to allow the final student model to learn from two teacher models simultaneously.

Several works also adopted it for video-related tasks. JITNet [144] employed knowledge distillation for video semantic segmentation. This method trained the student network in an online fashion on the live video, intermittently running the teacher network to provide a target for learning. Such online distillation yields semantic segmentation models that closely approximate their Mask R-CNN teacher with 7 to 17× lower inference runtime cost. In [191], a lightweight network tailored for video salient object detection (VSOD) through spatiotemporal knowledge distillation is proposed. It achieved competitive performance against prior works and the runtime of this lightweight model is very fast with 0.01 s per frame.

Nevertheless, knowledge distillation has not been extensively applied to MOS problems, which is a promising future research direction.

7.6 Model Deployment and Performance Evaluation on Edge Devices

Although many fast and lightweight MOS models have been developed, currently most inference tasks are conducted on GPU severs. In future research, it is desirable to deploy efficient MOS models and evaluate their performance on commercially available edge devices, such as the edge TPU developed by

Google [26], the Neural Compute 2 developed by Intel [65], Jetson Nano, Jetson TX1, and Jetson AGX Xavier developed by Nvidia [25], and AI Edge developed by Xilinx [221]. It is important to compare the accuracy and efficiency of various models on these devices. The test results can provide necessary insights to advance technologies in edge AI applications such as autonomous vehicles [5] and mobile robots [102].

8 Conclusion

In this paper, we presented a comprehensive review of deep learning-based MOS algorithms. Under the motivation of delay-sensitive MOS scenarios and applications on mobile and embedded devices, we summarized efficient MOS models in detail. In particular, we introduced 13 efficient MOS model design techniques, summarized a variety of MOS datasets, thoroughly reviewed performance evaluation metrics including accuracy metrics and efficiency metrics, compared model performance on popular MOS datasets and analyzed essential techniques of competitive models. The access links to each model and each dataset were also provided. Last but not least, we pointed out existing challenges in MOS and present future research directions from the perspectives of automatic network architecture search, leveraging transformer approaches, improving models' generalization capabilities, unsupervised learning, model compression through knowledge distillation, and model deployment and evaluation on edge devices. We believe this review brings rich information about different aspects of deep learning-based MOS and provides readers with useful insights into future research endeavors.

Biographies

Bingxin Hou received the M.S. degree in Imaging Science from the Rochester Institute of Technology in Rochester, New York in 2010. She worked with advisor Dr. Roy. S. Berns in the Munsell Color Science Laboratory. She worked in Hewlett Packard Company (HP) as Color Imaging Scientist from 2011 to 2017. Currently, she is a Ph.D. candidate in Computer Science and Engineering at Santa Clara University. She is working with advisors Dr. Nam Ling and Dr. Ying Liu on efficient deep network for moving object detection, and video coding for machine vision. Her research interests include deep learning, computer vision, video compression and camera image processing.

Ying Liu received the B.S. degree in communications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2006, the M.S. and Ph.D. degrees in Electrical Engineering from The State University of New

York at Buffalo, NY, USA, in 2008 and 2012, respectively. She currently is an Assistant Professor in the Department of Computer Science and Engineering at Santa Clara University, Santa Clara, CA, USA. She serves as an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology. Her main research interests are in image and video processing, deep learning, and computer vision.

Nam Ling received the B. Eng degree in electrical engineering from the National University of Singapore and the M.S. and Ph.D. degrees in computer engineering from the University of Louisiana at Lafayette, USA. He is currently the Wilmot J. Nicholson Family Chair Professor of Santa Clara University, California, USA, and the Chair of its Department of Computer Science and Engineering, since 2010. From 2002 to 2010, he was an Associate Dean for its School of Engineering. He has been an IEEE Fellow since 2008. His main research interests are in video/image coding, 3D/stereoscopic video/image, rate control, and the use of deep learning in image/video.

Yongxiong Ren obtained his Ph.D. (2016) degree in Electrical Engineering at the University of Southern California, Los Angeles. He received his Master (2011) and Bachelor (2008) degrees from Peking University, China and Beijing University of Posts and Telecommunications, China, respectively. He is currently a Video Algorithm Architect at Kwai Inc, Palo Alto, focusing on optimization and acceleration of deep learning models. He has more than 130 publications with over 9000 Google scholar citations. His publication lists contain two book chapters, 5 U.S. patents, more than 60 journal papers, and more than 70 conference papers and invited presentations. He is a recipient of Best Paper Award at IEEE GLOBECOM 2014. His research interests include deep learning, digital signal processing, image processing, and hardware architecture.

Lingzhi Liu is the Location Manager of US R&D Center, the Head and Chief Architect of Heterogeneous Computing Group of Kuaishou Technology in Palo Alto, CA since 2018. Before joining Kuaishou, he held several manager and professional positions in Alibaba-inc, Realtek USA, Intel Corp. and Futurewei Technologies around the Silicon Valley, California. He also worked in Fujitsu 2005 and Midea Corp from 1998 to 2000. He was a Postdoctoral Researcher in EE Dept. of University of Washington from 2005 to 2008. He was an adjunct Professor of Wuhan University, China from 2015 to 2018. He received the B.S. degree from Xi'an Jiaotong University and the Ph.D. degree from Shanghai Jiaotong University, China, in 1998 and 2004 respectively. His general interests include neural network algorithm and architecture, multimedia algorithm and implementation, VLSI system, ASIC and FPGA design and channel coding theory. Dr. Liu has published more than 80 patents, 40 papers and 100 proposals to international standards. Dr. Liu was the Panel/Keynote Chair of

ICME2013, Track Chair of APSIPA ASC 2010, Review Committee member of ISCAS2010, Session Chair of ASICON 2007. He served the TPC of conferences including VLSI-SOC 2014, APSIPA ASC 2011&2010, PICom 2009 and etc. He is a Senior Member of IEEE since 2008.

References

[1] AI Benchmark. Deep Learning Hardware Ranking, https://ai-benchmark.com/ranking_deeplearning.html.

[2] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-Based Image-to-Image Foreground Segmentation", *IEEE Transactions on Intelligent Transportation Systems*, 21(3), 2020, 959–71.

[3] T. Akilan, Q. J. Wu, and W. Zhang, "Video Foreground Extraction Using Multi-View Receptive Field and Encoder–Decoder DCNN for Traffic and Surveillance Applications", *IEEE Transactions on Vehicular Technology*, 68(10), 2019, 9478–93.

[4] A. Akula, R. Ghosh, S. Kumar, and H. K. Sardana, "Moving Target Detection in Thermal Infrared Imagery using Spatiotemporal Information", *J. Opt. Soc. Am. A*, 30(8), 2013, 1492–501.

[5] A. Amanatiadis, E. Karakasis, L. Bampis, S. Ploumpis, and A. Gasteratos, "ViPED: On-Road Vehicle Passenger Detection for Autonomous Vehicles", *Robotics and Autonomous Systems*, 112, 2019, 282–90.

[6] Amandeep and E. M. Goyal, "Review: Moving Object Detection Techniques", *International Journal of Computer Science and Mobile Computing*, 4(9), 2015, 345–9.

[7] N. Araslanov and S. Roth, "Self-Supervised Augmentation Consistency for Adapting Semantic Segmentation", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 15379–89, DOI: 10.1109/CVPR46437.2021.01513.

[8] M. Babaee, D. Dinh, and G. Rigoll, "A Deep Convolutional Neural Network for Background Subtraction", *ArXiv: 1702.01731*, 2017.

[9] F. Bahri and N. Ray, "Dynamic Background Subtraction by Generative Neural Networks", *ArXiv: 2202.05336*, 2022.

[10] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "BScGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks", in *25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, 2018, 4018–22.

[11] O. Barnich and M. Van Droogenbroeck, "ViBe: A Universal Background Subtraction Algorithm for Video Sequences", *IEEE Transactions on Image Processing*, 20(6), 2011, 1709–24.

[12] Y. Benezeth, D. Sidibé, and J. B. Thomas, “Background Subtraction with Multispectral Video Sequences”, in *IEEE Int. Conf. Robot. Autom. Workshop Non-Classical Cameras, Camera Netw. Omnidirec- tional Vis. (OMNIVIS)*, Hong Kong, China, 2014, 6.

[13] G. Bhat, F. J. Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. V. Gool, and R. Timofte, “Learning What to Learn for Video Object Segmentation”, *ArXiv: 2003.11540*, 2020.

[14] S. Bianco, G. Ciocca, and R. Schettini, “Combination of Video Change Detection Algorithms by Genetic Programming”, *IEEE Transactions on Evolutionary Computation*, 21(6), 2017, 914–28.

[15] G. Bilodeau, J. Jodoin, and N. Saunier, “Change Detection in Feature Space Using Local Binary Similarity Patterns”, in *International Conference on Computer and Robot Vision*, Regina, SK, Canada, 2013, 106–12.

[16] G. Bilodeau, A. Torabi, P.-L. St-Charles, and D. Riahi, “Thermal–Visible Registration of Human Silhouettes: A Similarity Measure Performance Evaluation”, *Infrared Physics & Technology*, 64, 2014, 79–86.

[17] D. D. Bloisi, L. Iocchi, A. Pennisi, and L. Tombolini, “ARGOS-Venice Boat Classification”, in *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Karlsruhe, Germany, 2015, 1–6.

[18] T. Bouwmans, “Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview”, *Computer Science Review*, 11-12, 2014, 31–66.

[19] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep Neural Network Concepts for Background Subtraction: A Systematic Review and Comparative Evaluation”, *Neural Networks*, 117, 2019, 8–66.

[20] M. Braham, S. Piérard, and M. Van Droogenbroeck, “Semantic Background Subtraction”, in *IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, 4552–6.

[21] M. Braham and M. Van Droogenbroeck, “Deep Background Subtraction with Scene-Specific Convolutional Neural Networks”, in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, Bratislava, Slovakia, 2016, 1–4.

[22] S. Brutzer, B. Höferlin, and G. Heidemann, “Evaluation of Background Subtraction Techniques for Video Surveillance”, in *CVPR 2011*, Colorado Springs, CO, USA, 2011, 1937–44.

[23] M. Camplani, L. Maddalena, G. Moyá Alcover, A. Petrosino, and L. Salgado, “A Benchmarking Framework for Background Subtraction in RGBD Videos”, in *New Trends in Image Analysis and Processing – ICIAP 2017*, Depok, West Java, Indonesia, 2017, 219–29.

[24] A. Canepa, E. Ragusa, R. Zunino, and P. Gastaldo, "T-RexNet:A Hardware-Aware Neural Network for Real-Time Detection of Small Moving Objects", *Sensors*, 21(4), 2021, 1252.

[25] S. Cass, "Nvidia Makes It Easy to Embed AI: The Jetson Nano Packs a Lot of Machine-Learning Power into DIY Projects-[Hands On]", *IEEE Spectrum*, 57(7), 2020, 14–6.

[26] S. Cass, "Taking AI to the Edge: Google's TPU Now Comes in a Maker-Friendly Package", *IEEE Spectrum*, 56(5), 2019, 16–7.

[27] M.-N. Chapel and T. Bouwmans, "Moving Objects Detection with a Moving Camera: A Comprehensive Review", *Computer Science Review*, 38, 2020, 100310, https://www.sciencedirect.com/science/article/pii/S157401372030410X.

[28] P. St-Charles, G. Bilodeau, and R. Bergevin, "A Self-Adjusting Approach to Change Detection Based on Background Word Consensus", in *IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, 2015, 990–7.

[29] P. St-Charles, G. Bilodeau, and R. Bergevin, "SuBSENSE: A Universal Change Detection Method with Local Adaptive Sensitivity", *IEEE Transactions on Image Processing*, 24(1), 2015, 359–73.

[30] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, "Data-Free Knowledge Distillation for Object Detection", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Virtual, 2021, 3289–98.

[31] W.-C. Chen, C.-C. Chang, and C.-R. Lee, "Knowledge Distillation with Feature Maps for Image Classification", in *Asian Conference on Computer Vision*, Springer, Perth, Australia, 2018, 200–15.

[32] X. Chen, Z. Li, Y. Yuan, G. Yu, J. Shen, and D. Qi, "State-Aware Tracker for Real-Time Video Object Segmentation", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, 9381–90.

[33] Y.-W. Chen, Y.-H. Tsai, Y.-Y. Lin, and M.-H. Yang, "VOSTR: Video Object Segmentation via Transferable Representations", *International Journal of Computer Vision*, 128, 2020, 931–49.

[34] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an Octave: Reducing Spatial Redundancy in Convolutional Neural Networks With Octave Convolution", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, October 2019.

[35] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise Deep Sequence Learning for Moving Object Detection", *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9), 2019, 2567–79.

[36] Y. Chen and Q. Yu, “Efficient Moving Object Segmentation Algorithm Based on the Improvement of Generalized Geodesic Active Contour Model”, in *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, China, 2016, 630–5.

[37] J. Cheng, Y. Tsai, W. Hung, S. Wang, and M. Yang, “Fast and Accurate Online Video Object Segmentation via Tracking Parts”, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, June 2018, 7415–24.

[38] Y. Cheng, Y. Yang, H.-B. Chen, N. Wong, and H. Yu, “S3-Net: A Fast and Lightweight Video Scene Understanding Network by Single-Shot Segmentation”, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA, 2021, 3329–37.

[39] S. C. S. Cheung and C. Kamath, “Robust Techniques for Background Subtraction in Urban Traffic Video”, in *Visual Communications and Image Processing. San Jose, California, USA,* Vol. 5308, 2004, 881–92.

[40] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, July 2017, 1800–7.

[41] S. Choo, W. Seo, D. Jeong, and N. I. Cho, “Multi-Scale Recurrent Encoder-Decoder Network for Dense Temporal Classification”, in *24th International Conference on Pattern Recognition (ICPR)*, Beijing , China, 2018, 103–8.

[42] L. P. Cinelli, L. A. Thomaz, A. F. da Silva, E. A. da Silva, and S. L. Netto, “Foreground Segmentation for Anomaly Detection in Surveillance Videos Using Deep Residual Networks”, *Proc. 35th Simpósio Brasileiro De Telecomunicações E Processamento De Sinais*, 2017, 3–6.

[43] A. Cioppa, M. V. Droogenbroeck, and M. Braham, “Real-Time Semantic Background Subtraction”, in *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 2020, 3214–8.

[44] C. Cuevas, E. M. Yáñez, and N. García, “Labeled Dataset for Integral Evaluation of Moving Object Detection Algorithms: LASIESTA”, *Computer Vision and Image Understanding*, 152, 2016, 103–17.

[45] J. W. Davis and M. A. Keck, “A Two-Stage Template Approach to Person Detection in Thermal Imagery”, in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION’05)*, Vol. 1, Washington DC, USA, 2005, 364–9.

[46] J. W. Davis and V. Sharma, “Background-Subtraction Using Contour-Based Fusion of Thermal and Visible Imagery”, *Computer Vision and Image Understanding*, 106(2), 2007, 162–82, Special issue on Advances in Vision Algorithms and Systems beyond the Visible Spectrum.

[47] X. Ding, X. Zhang, J. Han, and G. Ding, “Scaling Up Your Kernels to 31 × 31: Revisiting Large Kernel Design in CNNs”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 11963–75.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale”, *arXiv preprint arXiv:2010.11929*, 2020.

[49] A. Douillard, Y. Chen, A. Dapogny, and M. Cord, “PLOP: Learning Without Forgetting for Continual Semantic Segmentation”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, 4040–50.

[50] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, “SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 5912–21.

[51] S. Dutt Jain, B. Xiong, and K. Grauman, “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*, 2017, 3664–73.

[52] “EC Funded CAVIAR project/IST 2001 37540”, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[53] A. Elgammal, D. Harwood, and L. Davis, “Non-Parametric Model for Background Subtraction”, in *Proceedings of the 6th European Conference on Computer Vision-Part II (ECCV)*, No. 17, Dublin, Ireland, 2000, 751–67.

[54] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge”, *International Journal of Computer Vision*, 88, 2009, 303–38.

[55] D. Fan, M. Cheng, Y. Liu, T. Li, and A. Borji, “Structure-Measure: A New Way to Evaluate Foreground Maps”, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, October 2017, 4558–67.

[56] D. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, “Enhanced-Alignment Measure for Binary Foreground Map Evaluation”, in *International Joint Conferences on Artificial Intelligence*, Stockholm, Sweden, 2018, 698–704.

[57] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-Adversarial Training of Neural Networks”, *J. Mach. Learn. Res.*, 17(1), 2016, 2096–30.

[58] Y. Gao, H. Cai, X. Zhang, L. Lan, and Z. Luo, “Background Subtraction via 3D Convolutional Neural Networks”, in *International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018, 1271–6.

[59] B. Garcia-Garcia, T. Bouwmans, and R. A. Jorge, "Background Subtraction in Real Applications: Challenges, Current Models and Future Directions", *Computer Science Review*, 35, 2020, 100204.

[60] "Generalization", https://developers.google.com/machine-learning/crash-course/generalization/video-lecture.

[61] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, "SqueezeNext: Hardware-Aware Neural Network Design", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Los Alamitos, CA, USA: IEEE Computer Society, June 2018.

[62] J. H. Giraldo and T. Bouwmans, "GraphBGS: Background Subtraction via Recovery of Graph Signals", 2020, eprint: ArXiv:2001.06404.

[63] J. H. Giraldo, S. Javed, and T. Bouwmans, "Graph Moving Object Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 1–1.

[64] J. H. Giraldo, H. T. Le, and T. Bouwmans, "Deep Learning Based Background Subtraction: A Systematic Survey", in *Handbook of Pattern Recognition and Computer Vision*, 2020, chap. 1.3, 51–73.

[65] Y. Gorbachev, M. Fedorov, I. Slavutin, A. Tugarev, M. Fatekhov, and Y. Tarkan, "Openvino Deep Learning Workbench: Comprehensive Analysis and Tuning of Neural Networks Inference", in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[66] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A New Change Detection Benchmark Dataset", in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, 2012, 1–8.

[67] Y. Gu, L. Wang, Z. Wang, Y. Liu, M.-M. Cheng, and S.-P. Lu, "Pyramid Constrained Self-Attention Network for Fast Video Salient Object Detection", in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, New York, USA, 2020, 10869–76.

[68] J. He, L. Balzano, and J. C. S. Lui, "Online Robust Subspace Tracking from Partial Information", *ArXiv: 1109.3827*, 2011.

[69] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN", in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, 2980–8.

[70] P. Henderson and M. Vertescher, "An Analysis of Parallelized Motion Masking Using Dual-Mode Single Gaussian Models", *ArXiv: 1702.05156*, 2017.

[71] G. Hinton, O. Vinyals, J. Dean, *et al.*, "Distilling the Knowledge in a Neural Network", *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[72] M. Hofmann, P. Tiefenbacher, and G. Rigoll, “Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter”, in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, 38–43.

[73] B. Hou, “Deep Learning-Based Low Complexity and High Efficiency Moving Object Detection Methods”, *PhD thesis*, Santa Clara University, Department of Computer Science and Engineering, March 2022, https://www.proquest.com/openview/cce577ab1c91e6f9692fa827211f7233/1?pq-origsite=gscholar&cbl=18750&diss=y.

[74] B. Hou, Y. Liu, and N. Ling, “A Super-Fast Deep Network for Moving Object Detection”, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, 1–5.

[75] B. Hou, Y. Liu, N. Ling, L. Liu, and Y. Ren, “A Fast Lightweight 3D Separable Convolutional Neural Network With Multi-Input Multi-Output for Moving Object Detection”, *IEEE Access*, 9, 2021, 148433–48.

[76] B. Hou, Y. Liu, N. Ling, L. Liu, Y. Ren, and M. K. Hsu, “F3DsCNN: A Fast Two-Branch 3D Separable CNN for Moving Object Detection”, in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 2021, 1–5.

[77] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, 2017, eprint: ArXiv:1704.04861.

[78] Z. Hu, T. Turki, N. Phan, and J. T. L. Wang, “A 3D Atrous Convolutional Long Short-Term Memory Network for Background Subtraction”, *IEEE Access*, 6, 2018, 43450–9.

[79] A. Ignatov, R. Timofte, S. Ko, S. Kim, K. Uhm, S. Ji, S. Cho, J. Hong, K. Mei, J. Li, *et al.*, “AIM 2019 Challenge on RAW to RGB Mapping: Methods and Results”, in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea, 2019, 3584–90.

[80] A. Ignatov, R. Timofte, Z. Zhang, M. Liu, H. Wang, W. Zuo, J. Zhang, R. Zhang, Z. Peng, S. Ren, *et al.*, “AIM 2020 Challenge on Learned Image Signal Processing Pipeline”, 2020, eprint: ArXiv:2011.04994.

[81] S. Jain, B. Xiong, and K. Grauman, “FusionSeg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos”, *ArXiv: 1701.05384*, 2017.

[82] W. Jang and C. Kim, “Online Video Object Segmentation via Convolutional Trident Network”, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, 7474–83.

[83] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Perez, "xMUDA: Cross-Modal Unsupervised Domain Adaptation for 3D Semantic Segmentation", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[84] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. K. Jung, "Moving Object Detection in Complex Scene Using Spatiotemporal Structured-Sparse RPCA", *IEEE Transactions on Image Processing*, 28(2), 2019, 1007–22.

[85] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung, "OR-PCA with MRF for Robust Foreground Detection in Highly Dynamic Backgrounds", in *Computer Vision – ACCV 2014*, Cham: Springer International Publishing, 2015, 284–99.

[86] S. Jiang and X. Lu, "WeSamBE: A Weight-Sample-Based Method for Background Subtraction", *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9), 2018, 2105–15.

[87] P.-M. Jodoin, L. Maddalena, A. Petrosino, and Y. Wang, "Extensive Benchmark and Survey of Modeling Methods for Scene Background Initialization", *IEEE Transactions on Image Processing*, 26(11), 2017, 5244–56.

[88] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg, "A Generative Appearance Model for End-To-End Video Object Segmentation", in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, 8945–54.

[89] R. Kalsotra and S. Arora, "A Comprehensive Survey of Video Datasets for Background Subtraction", *IEEE Access*, 7, 2019, 59143–71.

[90] I. Kavasidis, S. Palazzo, R. D. Salvo, D. Giordano, and C. Spampinato, "An Innovative Web-Based Collaborative Platform for Video Annotation", *Multimedia Tools and Applications*, 70, 2013, 413–32.

[91] A. Khan and N. J. Janwe, "Review on Moving Object Detection in Video Surveillance", *International Journal of Advanced Research in Computer and Communication Engineering*, 6, 2017, 664–70.

[92] J.-Y. Kim and J.-E. Ha, "Foreg.round Objects Detection Using a Fully Convolutional Network With a Background Model Image and Multiple Original Images", *IEEE Access*, 8, 2020, 159864–78.

[93] J.-Y. Kim and J.-E. Ha, "Spatio-Temporal Data Augmentation for Visual Surveillance", *IEEE Access*, 9, 2021, 165014–33.

[94] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in *Advances in Neural Information Processing Systems*, Vol. 25, Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf, 2012.

[95] J. S. Kulchandani and K. J. Dangarwala, “Moving Object Detection: Review of Recent Research Trends”, in *2015 International Conference on Pervasive Computing (ICPC)*, Pune, India, 2015, 1–5.

[96] L. A. Lim and H. Yalim Keles, “Foreground Segmentation Using Convolutional Neural Networks for Multiscale Feature Encoding”, *Pattern Recognition Letters*, 112, 2018, 256–62.

[97] “Laboratory for Image and Media Understanding, LIMU dataset”, https://limu.ait.kyushu-u.ac.jp/dataset/en/.

[98] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, “MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking”, *ArXiv: 1504.01942*, 2015.

[99] C. Li, X. Wang, L. Zhang, J. Tang, H. Wu, and L. Lin, “Weighted Low-Rank Decomposition for Robust Grayscale-Thermal Foreground Detection”, *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4), 2017, 725–38.

[100] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, “Video Segmentation by Tracking Many Figure-Ground Segments”, in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia, 2013, 2192–9.

[101] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical Modeling of Complex Backgrounds for Foreground Object Detection”, *IEEE Transactions on Image Processing*, 13(11), 2004, 1459–72.

[102] Q. Li, Y. Fu, P. Q. J, G. T. Nguyen, T. Hannu, Z. Zou, and W. Tomi, “Edge Computing for Mobile Robots: Multi-Robot Feature-Based Lidar Odometry with FPGAs”, in *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, IEEE, 2019, 1–2.

[103] S. Li, D. A. F. Florêncio, W. Li, Y. Zhao, and C. Cook, “A Fusion Framework for Camouflaged Moving Foreground Detection in the Wavelet Domain”, *IEEE Transactions on Image Processing*, 27, 2018, 3918–30.

[104] S. Li, B. Seybold, A. Vorobyov, A. Fathi, Q. Huang, and C.-C. J. Kuo, “Instance Embedding Transfer to Unsupervised Video Object Segmentation”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 6526–35.

[105] Y. Li, Z. Shen, and Y. Shan, “Fast Video Object Segmentation Using the Global Context Module”, in *Computer Vision – ECCV 2020: 16th European Conference*, Glasgow, United Kingdom, Springer-Verlag, 2020, 735–50.

[106] Y. Li, L. Yuan, and N. Vasconcelos, “Bidirectional Learning for Domain Adaptation of Semantic Segmentation”, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 6929–38, DOI: 10.1109/CVPR.2019.00710.

[107] X. Liang, S. Liao, X. Wang, W. Liu, Y. Chen, and S. Z. Li, "Deep Background Subtraction with Guided Learning", in *IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, USA, 2018, 1–6.

[108] J. Liao, G. Guo, Y. Yan, and H. Wang, "Multiscale Cascaded Scene-Specific Convolutional Neural Networks for Background Subtraction", in *19th Pacific-Rim Conference on Multimedia,* Hefei, China, September 2018, 524–33.

[109] K. Lim, W. Jang, and C. Kim, "Background Subtraction Using Encoder-Decoder Structured Convolutional Neural Network", in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 2017, 1–6.

[110] L. A. Lim and H. Yalim Keles, "Learning Multi-Scale Features for Foreground Segmentation", *Pattern Analysis and Applications*, 23(3), 2019, 1369–80.

[111] C. Lin, S. Zhang, S. You, X. Liu, and Z. Zhu, "Real-Time Foreground Object Segmentation Networks Using Long and Short Skip Connections", *Information Sciences*, 571, 2021, 543–59.

[112] H. Lin, X. Qi, and J. Jia, "AGSS-VOS: Attention Guided Single-Shot Video Object Segmentation", in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, 3948–56.

[113] P. Lin P.and Sun, G. Cheng, S. Xie, X. Li, and J. Shi, "Graph-guided Architecture Search for Real-time Semantic Segmentation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 4203–12.

[114] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and F.-F. Li, "Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, 82–92.

[115] H. Liu Z.and Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A Convnet for the 2020s", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 11976–86.

[116] Y. Liu, Z. Bellay, P. Bradsky, G. Chandler, and B. Craig, "Edge-to-Fog Computing for Color-Assisted Moving Object Detection", in *Big Data: Learning, Analytics, and Applications*, Vol. 10989, SPIE, 2019, 9–17.

[117] Y. Liu and D. A. Pados, "Compressed-Sensed-Domain L1-PCA Video Surveillance", *IEEE Transactions on Multimedia*, 18(3), 2016, 351–63.

[118] Y. Liu, K. Tountas, D. A. Pados, S. N. Batalama, and M. J. Medley, "L1-Subspace Tracking for Streaming Data", *Pattern Recognition*, 97, 2020, 106992.

[119] Z. Liu, Y. Lin, Y. Cao, Y. Hu H.and Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, 10012–22.

[120] Z. Liu, J. Liu, W. Chen, X. Wu, and Z. Li, "FAMINet: Learning Real-Time Semisupervised Video Object Segmentation With Steepest Optimized Optical Flow", *IEEE Transactions on Instrumentation and Measurement*, 71, 2022, 1–16.

[121] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[122] X. Lu, W. Wang, J. Shen, Y.-W. Tai, D. J. Crandall, and S. C. H. Hoi, "Learning Video Object Segmentation From Unlabeled Videos", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[123] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-Generation, Refinement and Merging for Video Object Segmentation", in *Computer Vision – ACCV 2018*, Cham: Springer International Publishing, 2019, 565–80.

[124] L. Maddalena and A. Petrosino, "Background Subtraction for Moving Object Detection in RGBD Data: A Survey", *Journal of Imaging*, 4, 2018, 71.

[125] L. Maddalena and A. Petrosino, "Towards Benchmarking Scene Background Initialization", in *International Conference Image Anal. Process. Cham, Switzerland: Springer,* Vol. 9281, 2015, 469–76.

[126] V. Mahadevan and N. Vasconcelos, "Spatiotemporal Saliency in Dynamic Scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), 2010, 171–7.

[127] M. Mandal, V. Dhar, A. Mishra, and S. K. Vipparthi, "3DFR: A Swift 3D Feature Reductionist Framework for Scene Independent Change Detection", *IEEE Signal Processing Letters*, 26(12), 2019, 1882–6.

[128] M. Mandal, V. Dhar, A. Mishra, S. K. Vipparthi, and M. Abdel-Mottaleb, "3DCD: Scene Independent End-to-End Spatiotemporal Feature Learning Framework for Change Detection in Unseen Videos", *IEEE Transactions on Image Processing*, 30, 2021, 546–58.

[129] M. Mandal, L. K. Kumar, M. Singh Saran, and S. K. Vipparthi, "MotionRec: A Unified Deep Framework for Moving Object Recognition", in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA, 2020, 2723–32.

[130] M. Mandal, L. K. Kumar, and S. K. Vipparthi, "MOR-UAV: A Benchmark Dataset and Baselines for Moving Object Recognition in UAV Videos", in *New York, NY, USA, Proceedings of the 28th ACM Interna-*

tional Conference on Multimedia, Association for Computing Machinery, 2020, 2626–35.

[131] M. Mandal, P. Saxena, S. Vipparthi, and S. Murala, "CANDID: Robust Change Dynamics and Deterministic Update Policy for Dynamic Background Subtraction", in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, 2468–73.

[132] M. Mandal and S. K. Vipparthi, "An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs", *IEEE Transactions on Intelligent Transportation Systems*, 2021, 1–22.

[133] M. Mandal and S. K. Vipparthi, "Scene Independency Matters: An Empirical Study of Scene Dependent and Scene Independent Evaluation for CNN-Based Change Detection", *IEEE Transactions on Intelligent Transportation Systems*, 23(3), 2022, 2031–44, DOI: 10.1109/TITS.2020.3030801.

[134] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, "Bilateral Space Video Segmentation", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, 743–51.

[135] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation", in *Proceedings of the european conference on computer vision (ECCV)*, 2018, 552–68.

[136] J. Mei, M. Wang, Y. Lin, and Y. Yuan Y.and Liu, "TransVOS: Video Object Segmentation with Transformers", *arXiv preprint arXiv:2106.00588*, 2021.

[137] T. Meinhardt and L. Leal-Taixé, "Make One-Shot Video Object Segmentation Efficient Again", *ArXiv: 2012.01866*, 2020.

[138] R. Miezianko, "Terravic Research Infrared Database", https://github.com/nkbenamara/Terravic-Facial-IR-Database-Annotations-.

[139] T. Minematsu, A. Shimada, and R. Taniguchi, "Simple Background Subtraction Constraint for Weakly Supervised Background Subtraction Network", in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Taipei, Taiwan, 2019, 1–8.

[140] S. I. Mirzadeh, A. Chaudhry, D. Yin, T. Nguyen, R. Pascanu, D. Gorur, and M. Farajtabar, "Architecture Matters in Continual Learning", *ArXiv:2202.00275*, 2022.

[141] E. Mohamed and A. E. Sallab, "MODETR: Moving Object Detection with Transformers", *ArXiv: 2106.11422*, 2021.

[142] V. Mondéjar-Guerra, J. Rouco, J. Novo, and M. Ortega, "An End-to-End Deep Learning Approach for Simultaneous Background Modeling and Subtraction", in *British Machine Vision Conference*, 2019, 266.

[143] G. Moyà-Alcover, A. Elgammal, A. Jaume-i-Capó, and J. Varona, "Modeling Depth for Nonparametric Foreground Segmentation Using RGBD Devices", *Pattern Recognition Letters*, 96, 2017, 76–85, Scene Background Modeling and Initialization.

[144] R. T. Mullapudi, S. Chen, K. Zhang, D. Ramanan, and K. Fatahalian, "Online Model Distillation for Efficient Video Inference", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 3573–82.

[145] S. Muralikrishna, B. Muniyal, and U. Dinesh Acharya, "Adaptive Cluster Based Model for Fast Video Background Subtraction", English, *International Journal of Advanced Computer Science and Applications*, 10(12), 2019, 689–96.

[146] V. Nekrasov, H. Chen, C. Shen, and I. Reid, "Fast Neural Architecture Search of Compact Semantic Segmentation Models via Auxiliary Cells", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, 9126–35.

[147] T. P. Nguyen, C. C. Pham, S. V.-U. Ha, and J. W. Jeon, "Change Detection by Training a Triplet Network for Motion Feature Extraction", *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2), 2019, 433–46.

[148] C. Nicolas, M. Francisco, S. Gabriel, U. Nicolas, K. Alexander, and Z. Sergey, "End-to-End Object Detection with Transformers", in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, 2020, 23–8.

[149] P. Ochs, J. Malik, and T. Brox, "Segmentation of Moving Objects by Long Term Video Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6), 2014, 1187–200.

[150] S. W. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A Large-Scale Benchmark Dataset for Event Recognition in Surveillance Video", in *CVPR 2011*, 2011, 3153–60.

[151] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video Object Segmentation Using Space-Time Memory Networks", in *IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, 9225–34.

[152] R. Palliser Sans, "Fast Video Object Segmentation by Pixel-Wise Feature Comparison", *PhD thesis*, UPC, Centre de Formació Interdisciplinària Superior, Departament de Teoria del Senyal i Comunicacions, May 2019, http://hdl.handle.net/2117/169370.

[153] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual Lifelong Learning with Neural Networks: A Review", *Neural Networks*, 113, 2019, 54–71.

[154] H. Park, G. Venkatesh, and N. Kwak, "TTVOS: Lightweight Video Object Segmentation with Adaptive Template Attention Module and Temporal Consistency Loss", *ArXiv: 2011.04445*, 2020.

[155] H. Park, J. Yoo, S. Jeong, G. Venkatesh, and N. Kwak, "Learning Dynamic Network Using a Reuse Gate Function in Semi-supervised Video Object Segmentation", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, 8401–10.

[156] H. Park, J. Yoo, G. Venkatesh, and N. Kwak, "Adaptive Template and Transition Map for Real-Time Video Object Segmentation", *IEEE Access*, 9, 2021, 116914–26.

[157] P. W. Patil, K. M. Biradar, A. Dudhane, and S. Murala, "An End-to-End Edge Aggregation Network for Moving Object Segmentation", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, 8146–55.

[158] P. W. Patil, A. Dudhane, and S. Murala, "End-to-End Recurrent Generative Adversarial Network for Traffic and Surveillance Applications", *IEEE Transactions on Vehicular Technology*, 69(12), 2020, 14550–62.

[159] P. W. Patil and S. Murala, "FgGAN: A Cascaded Unpaired Learning for Background Estimation and Foreground Segmentation", in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa Village, HI, USA, 2019, 1770–8.

[160] P. W. Patil and S. Murala, "MSFgNet: A Novel Compact End-to-End Deep Network for Moving Object Detection", *IEEE Transactions on Intelligent Transportation Systems*, 20(11), 2019, 4066–77.

[161] P. W. Patil, S. Murala, A. Dhall, and S. Chaudhary, "MsEDNet: Multi-Scale Deep Saliency Learning for Moving Object Detection", in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, 2018, 1670–5.

[162] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large Kernel Matters–Improve Semantic Segmentation by Global Convolutional Network", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 4353–61.

[163] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning Video Object Segmentation from Static Images", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, 3491–500.

[164] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation", in *IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 724–32.

[165] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS Challenge on Video Object Segmentation", *ArXiv: 1704.00675*, 2017.

[166] R. P. K. Poudel, U. Bonde, S. Liwicki, and C. Zach, "ContextNet: Exploring Context and Detail for Semantic Segmentation in Real-Time", in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, BMVA Press, 2018, 146.

[167] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting Moving Shadows: Algorithms and Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), 2003, 918–23.

[168] M. Qiu and X. Li, "A Fully Convolutional Encoder–Decoder Spatial–Temporal Network for Real-Time Background Subtraction", *IEEE Access*, 7, 2019, 85949–58.

[169] M. K. B. R. Debnath, "Moving Object Detection Under Sudden Change of Illumination: A Review", *International Journal of Computational Intelligence & IoT*, 2(3), 2019, Available at SSRN: https://ssrn.com/abstract=3358306.

[170] A. Robinson, F. Järemo Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, "Learning Fast and Robust Target Models for Video Object Segmentation", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, 7404–13.

[171] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-End Video Background Subtraction with 3D Convolutional Neural Networks", *Multimedia Tools and Applications*, 77, 2017, 23023–41.

[172] D. Sakkos, H. P. H. Shum, and E. S. L. Ho, "Illumination-Based Data Augmentation for Robust Background Subtraction", 2019, 1–8.

[173] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, 4510–20.

[174] R. Sharma and S. Gupta, "A Survey on Moving Object Detection and Tracking Based On Background Subtraction", *The Oxford Journal of Intelligent Decision and Data Science*, 2018, 2018, 55–62.

[175] C.-H. Shih and W.-J. Tsai, "Hierarchical Embedding Guided Network for Video Object Segmentation", in *IEEE International Conference on Image Processing (ICIP)*, 2021, 1124–8.

[176] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental Learning of Object Detectors Without Catastrophic Forgetting", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017, 3420–9.

[177] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", in *May 7 - 9, International Conference on Learning Representations*, 2015.

[178] S. Singh, S. A. Velastin, and H. Ragheb, "MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods", in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Boston, Massachusetts, USA, 2010, 48–55.

[179] A. Singha and M. K. Bhowmik, "TU-VDN: Tripura University Video Dataset at Night Time in Degraded Atmospheric Outdoor Conditions for Moving Object Detection", in *2019 IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, 2019, 2936–40.

[180] A. Sobral and A. Vacavant, "A Comprehensive Review of Background Subtraction Algorithms Evaluated with Synthetic and Real Videos", *Computer Vision and Image Understanding*, 122, 2014, 4–21.

[181] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, "Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection", in *Computer Vision – ECCV 2018*, Munich, Germany, 2018, 744–60.

[182] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mmixture Models for Real-Time Tracking", in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Vol. 2, Fort Collins, Colorado, 1999, 246–52.

[183] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for Semantic Segmentation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, 7262–72.

[184] J. Stückler and S. Behnke, "Efficient Dense Rigid-Body Motion Segmentation and Estimation in RGB-D Video.", *International Journal of Computer Vision*, 113(3), 2015, 233–45.

[185] B. N. Subudhi, M. K. Panda, T. Veerakumar, V. Jakhetiya, and S. Esakkirajan, "Kernel-Induced Possibilistic Fuzzy Associate Background Subtraction for Video Scene", *IEEE Transactions on Computational Social Systems*, 2022, 1–12.

[186] M. Sultana, A. Mahmood, T. Bouwmans, and S. K. Jung, "Dynamic Background Subtraction Using Least Square Adversarial Learning", in *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 2020, 3204–8.

[187] M. Sultana, A. Mahmood, S. Javed, and S. K. Jung, "Unsupervised Deep Context Prediction for Background Foreground Separation", *Machine Vision and Applications*, 30, 2018, 375–95.

[188] M. Sun, J. Xiao, E. G. Lim, B. Zhang, and Y. Zhao, "Fast Template Matching and Update for Video Object Tracking and Segmentation", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, 10788–96.

[189] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, 2818–26.

[190] N. Takahashi and Y. Mitsufuji, "Densely Connected Multi-Dilated Convolutional Networks for Dense Prediction Tasks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 993–1002.

[191] Y. Tang, Y. Li, and W. Zou, "Fast Video Salient Object Detection via Spatiotemporal Knowledge Distillation", *arXiv preprint arXiv:2010.10027*, 2020.

[192] i-LIDS Team, "Imagery Library for Intelligent Detection Systems (i-LIDS); A Standard for Testing Video Based Detection Systems", in *Proceedings 40th Annual 2006 International Carnahan Conference on Security Technology*, Lexington, KY, USA, 2006, 75–80.

[193] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net 2.0: Spatio-Temporal Data Augmentations for Video-Agnostic Supervised Background Subtraction", *IEEE Access*, 9, 2021, 53849–60.

[194] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A Fully-Convolutional Neural Network for Background Subtraction of Unseen Videos", in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA, 2020, 2763–72.

[195] P. Tokmakov, K. Alahari, and C. Schmid, "Learning Motion Patterns in Videos", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 3386–94.

[196] F. D. L. Torre, J. K. Hodgins, A. W. Bargteil, X. M. Artal, J. C. Macey, A. C. I. Castells, and J. Beltran, "Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database", *tech. rep.* No. CMU-RI-TR-08-22, Pittsburgh, PA: Carnegie Mellon University, April 2008.

[197] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance", in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 1, Corfu, Greece, 1999, 255–261 vol.1.

[198] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video Segmentation via Object Flow", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, 3899–908.

[199] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial Discriminative Domain Adaptation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[200] "Underwater Change Detection Dataset. Accessed: Jun. 25, 2020", [Online].%20Available:%20http://underwaterchangedetection.%20eu/index.html.

[201] A. Vacavant, T. Chateau, and L. Wilhelm A.and Lequièvre, "A Benchmark Dataset for Outdoor Foreground/Background Extraction", in *Proceedings of the 11th International Conference on Computer Vision - Volume Part I*, *ACCV'12*, Berlin, Heidelberg: Springer-Verlag, 2012, 291–300.

[202] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i-Nieto, "RVOS: End-to-End Recurrent Network for Video Object Segmentation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 5277–86.

[203] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, "FEELVOS: Fast End-To-End Embedding Learning for Video Object Segmentation", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, 9473–82.

[204] C. Wang, C. Li, J. Liu, B. Luo, X. Su, Y. Wang, and Y. Gao, "U2-ONet: A Two-Level Nested Octave U-Structure Network with a Multi-scale Attention Mechanism for Moving Object Segmentation", *Remote Sensing*, 13(1), 2020, 60.

[205] F. Wang and Y. Zhang, "A De-Raining Semantic Segmentation Network for Real-Time Foreground Segmentation", *Journal of Real-Time Image Processing*, 18(3), 2021, 873–87.

[206] H. Wang, X. Jiang, H. Ren, Y. Hu, and S. Bai, "SwiftNet: Real-Time Video Object Segmentation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual, June 2021, 1296–305.

[207] J. Wang, Z. Teng, B. Zhang, and J. Fan, "Integrating Long-Short Term Network for Efficient Video Object Segmentation", in *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*, BMVA Press, 2020, https://www.bmvc2020-conference.com/assets/papers/0167.pdf.

[208] K. Wang, C. Gou, and F.-Y. Wang, "$\mathrm{M}^4\mathrm{CD}$: A Robust Change Detection Method for Intelligent Visual Surveillance", *IEEE Access*, 6, 2018, 15505–20.

[209] M. Wang, W. Li, and X. Wang, "Transferring a Generic Pedestrian Detector towards Specific Scenes", in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, 2012, 3274–81.

[210] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast Online Object Tracking and Segmentation: A Unifying Approach”, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, 1328–38.

[211] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models”, in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, 420–4.

[212] W. Wang, J. Shen, X. Lu, S. C. Hoi, and H. Ling, “Paying Attention to Video Object Pattern Understanding”, *IEEE Transactions on Pattern analysis and Machine Intelligence*, 43(7), 2020, 2413–28.

[213] W. Wang, T. Zhou, F. M. Porikli, D. J. Crandall, and L. V. Gool, “A Survey on Deep Learning Technique for Video Segmentation”, *ArXiv: 2107.01153*, 2021.

[214] X. Wang, X. Ma, and W. E. L. Grimson, “Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3), 2009, 539–55.

[215] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDnet 2014: An Expanded Change Detection Benchmark Dataset”, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, USA, 2014, 393–400.

[216] Y. Wang, Z. Luo, and P.-M. Jodoin, “Interactive Deep Learning Method for Segmenting Moving Objects”, *Pattern Recognition Letters*, 96, 2017, 66–75.

[217] Y. Wang, Z. Yu, and L. Zhu, “Foreground Detection with Deeply Learned Multi-Scale Spatial-Temporal Features”, *Sensors (Basel, Switzerland)*, 18, 2018.

[218] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: from Error Visibility to Structural Similarity”, *IEEE Transactions on Image Processing*, 13(4), 2004, 600–12.

[219] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao, “RANet: Ranking Attention Network for Fast Video Object Segmentation”, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, 3977–86.

[220] S. S. Wangulkar, R. Talmale, and R. Babu, “A Review on Object Detection and Tracking in Video”, *International Journal of Scientific Research in Science, Engineering and Technology IJSRSET*, 8(2), 2019.

[221] D. Wu, Y. Zhang, X. Jia, L. Tian, T. Li, L. Sui, D. Xie, and Y. Shan, “A High-Performance CNN Processor Based on FPGA for MobileNets”, in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, Barcelona, Spain, 2019, 136–43.

[222] R. Wu, H. Lin, X. Qi, and J. Jia, "Memory Selection Network for Video Propagation", in *Computer Vision – ECCV 2020*, Cham: Springer International Publishing, 2020, 175–90.

[223] Z. Wu, N. Fuller, D. Theriault, and M. Betke, "A Thermal Infrared Video Benchmark for Visual Analysis", in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Washington, DC, USA, 2014, 201–8.

[224] H. Xiao, B. Kang, Y. Liu, M. Zhang, and J. Feng, "Online Meta Adaptation for Fast Video Object Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(5), 2019, 1205–17.

[225] H. Xie, H.Yao, S. Zhou, S. Zhang, and W. Sun, "Efficient Regional Memory Network for Video Object Segmentation", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, 1286–95.

[226] K. Xu, L. Rui, Y. Li, and L. Gu, "Feature Normalized Knowledge Distillation for Image Classification", in *European Conference on Computer Vision*, Springer, Glasgow, UK, 2020, 664–80.

[227] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark", 2018, eprint: ArXiv:1809.03327.

[228] T. Xue, Y. Qiao, H. Kong, D. Su, S. Pan, K. Rafique, and S. Sukkarieh, "One-Shot Learning-Based Animal Video Segmentation", *IEEE Transactions on Industrial Informatics*, 18(6), 2022, 3799–807.

[229] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-Image Relational Knowledge Distillation for Semantic Segmentation", *arXiv preprint arXiv:2204.06986*, 2022.

[230] L. Yang, X. Wang Y.and Xiong, J. Yang, and A. K. Katsaggelos, "Efficient Video Object Segmentation via Network Modulation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, 6499–507.

[231] Y. Yang, A. Loquercio, D. Scaramuzza, and S. Soatto, "Unsupervised Moving Object Detection via Contextual Information Separation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019.

[232] Z. Yang, Y. Wei, and Y. Yang, "Associating Objects with Transformers for Video Object Segmentation", *ArXiv: 2106.02638*, 2021.

[233] G. Yao, T. Lei, J. Zhong, P. Jiang, and W. Jia, "Comparative Evaluation of Background Subtraction Algorithms in Remote Scene Videos Captured by MWIR Sensors", *Sensors*, 17(9), 2017.

[234] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video Object Segmentation and Tracking: A Survey", *ACM Trans. Intell. Syst. Technol.*, 11(4), 2020.

[235] M. Yazdi and T. Bouwmans, "New Trends on Moving Object Detection in Video Images Captured by a Moving Camera: A Survey", *Computer Science Review*, 28, 2018, 157–77.

[236] Y. Yin, D. Xu, X. Wang, and L. Zhang, "Directional Deep Embedding and Appearance Learning for Fast Video Object Segmentation", *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 1–11.

[237] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I.-S. Kweon, "Pixel-Level Matching for Video Object Segmentation Using Convolutional Neural Networks", in *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, 2186–95.

[238] D. P. Young and J. M. Ferryman, "PETS Metrics: On-Line Performance Evaluation Service", in *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 2005, 317–24.

[239] W. Yu, J. Bai, and L. Jiao, "Background Subtraction Based on GAN and Domain Adaptation for VHR Optical Remote Sensing Videos", *IEEE Access*, 8, 2020, 119144–57, DOI: 10.1109/ACCESS.2020.3004495.

[240] D. Zeng and M. Zhu, "Multiscale Fully Convolutional Network for Foreground Object Detection in Infrared Videos", *IEEE Geoscience and Remote Sensing Letters*, 15(4), 2018, 617–21.

[241] J. Zhang, S. Wang, J. Qiu, X. Pan, J. Zou, Z. Duan Y.and Pan, and Y. Li, "A Fast X-Shaped Foreground Segmentation Network with CompactASPP", *Engineering Applications of Artificial Intelligence*, 97, 2021, 104077.

[242] J. Zhang, Z. Wang, S. Zhang, and G. Wei, "DAVOS: Semi-Supervised Video Object Segmentation via Adversarial Domain Adaptation", *ArXiv*, abs/2105.10201, 2021.

[243] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-Incremental Learning via Deep Model Consolidation", in, Snowmass Village, CO, USA, 2020, 1120–9.

[244] L. Zhang and Y. Liang, "Motion Human Detection Based on Background Subtraction", in *2010 Second International Workshop on Education Technology and Computer Science*, Vol. 1, Wuhan, Hubei, China, 2010, 284–7.

[245] L. Zhang, Z. Lin, J. Zhang, H. Lu, and Y. He, "Fast Video Object Segmentation via Dynamic Targeting Network", in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 2019, 5581–90.

[246] Y. Zhang, Z. Qiu, J. Liu, T. Yao, and T. Liu D.and Mei, "Customizable architecture search for semantic segmentation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, 11641–50.

[247] Y. Zhang, Z. Wu, H. Peng, and S. Lin, “A Transductive Approach for Video Object Segmentation”, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, 6947–56.

[248] C. Zhao and A. Basu, “Dynamic Deep Pixel Distribution Learning for Background Subtraction”, *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11), 2020, 4192–206.

[249] C. Zhao, T. Cham, X. Ren, J. Cai, and H. Zhu, “Background Subtraction Based on Deep Pixel Distribution Learning”, in *IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, USA, 2018, 1–6.

[250] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, “ICNet for Real-Time Semantic Segmentation on High-Resolution Images”, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, 405–20.

[251] L. Zheng, Y. Yang, and A. Hauptmann, “Person Re-identification: Past, Present and Future”, *ArXiv:1610.02984*, 2016.

[252] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, “Rethinking Semantic Segmentation From a Sequence-to-Sequence Perspective With Transformers”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, 6881–90.

[253] W. Zheng and K. Wang, “Background Subtraction Algorithm With Bayesian Generative Adversarial Networks”, *Zidonghua Xuebao/Acta Automatica Sinica*, 44, 2018, –.

[254] W. Zheng, K. Wang, and F.-Y. Wang, “A Novel Background Subtraction Algorithm Based on Parallel Vision and Bayesian GANs”, *Neurocomputing*, 394, 2020, 178–200.

[255] Y. Zheng and L. Fan, “Moving Object Detection Based on Running Average Background and Temporal Difference”, in *IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Hangzhou, China, 2010, 270–2.

[256] Q. Zhou and J. K. Aggarwal, “Tracking and Classifying Moving Objects Using Single or Multiple Cameras”, in *Handbook of Pattern Recognition and Computer Vision*, World Scientific Publishing Company, 499–524.

[257] T. Zhou, J. Li, S. Wang, R. Tao, and J. Shen, “MATNet: Motion-Attentive Transition Network for Zero-Shot Video Object Segmentation”, *IEEE Transactions on Image Processing*, 29, 2020, 8326–38.

[258] T. Zhou, W. Wang, Y. Yao, and J. Shen, “Target-Aware Adaptive Tracking for Unsupervised Video Object Segmentation”, in *The 2020 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, Vol. 3, Seattle, WA, USA, 2020.

[259] X. Zhou, C. Yang, and W. Yu, “Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3), 2013, 597–610.

[260] C. Zhu, W. Ping, C. Xiao, M. Shoeybi, T. Goldstein, A. Anandkumar, and B. Catanzaro, “Long-Short Transformer: Efficient Transformers for Language and Vision”, in *Advances in Neural Information Processing Systems*, virtual, 2021.

[261] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li, “A Comprehensive Study of Deep Video Action Recognition”, *ArXiv:2012.06567*, 2020.

[262] Z. Zivkovic, “Improved Adaptive Gaussian Mixture Model for Background Subtraction”, in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, Vol. 2, Cambridge, UK, 2004, 28–31.

[263] I. E. Zulfikar, J. Luiten, and B. Leibe, “UnOVOST: Unsupervised Offline Video Object Segmentation and Tracking for the 2019 Unsupervised Davis Challenge”, in *Proceedings of the 2019 DAVIS Challenge on Video Object Segmentation-CVPR Workshops*, Vol. 3, Long Beach, CA, 2019.