International Journal of Computer Vision (2024) 132:776–804
https://doi.org/10.1007/s11263-023-01910-x

DeepFTSG: Multi-stream Asymmetric USE-Net Trellis Encoders with Shared Decoder Feature Fusion Architecture for Video Motion Segmentation

Gani Rahmon[1] · Kannappan Palaniappan[1] · Imad Eddine Toubal[1] · Filiz Bunyak[1] · Raghuveer Rao[2] · Guna Seetharaman[3]

Received: 3 March 2022 / Accepted: 14 September 2023 / Published online: 17 October 2023

Abstract

Discriminating salient moving objects against complex, cluttered backgrounds, with occlusions and challenging environmental conditions like weather and illumination, is essential for stateful scene perception in autonomous systems. We propose a novel deep architecture, named DeepFTSG, for robust moving object detection that incorporates single and multi-stream multi-channel *USE-Net trellis* asymmetric encoders extending U-Net with squeeze and excitation (SE) blocks and a single shared decoder network for fusing multiple motion and appearance cues. DeepFTSG is a deep learning based approach that builds upon our previous hand-engineered flux tensor split Gaussian (FTSG) change detection video analysis algorithm which won the CDNet CVPR Change Detection Workshop challenge competition. DeepFTSG generalizes much better than top-performing motion detection deep networks, such as the scene-dependent *ensemble-based* FgSegNet_v2, while using an order of magnitude fewer weights. Short-term motion and longer-term change cues are estimated using general-purpose unsupervised methods—flux tensor and multi-modal background subtraction, respectively. DeepFTSG was evaluated using the CDnet-2014 change detection challenge dataset, the largest change detection video sequence benchmark with 12.3 billion labeled pixels, and had an overall F-measure of 97%. We also evaluated the cross-dataset *generalization capability* of DeepFTSG trained solely on CDnet-2014 short video segments and then evaluated on unseen SBI-2015, LASIESTA and LaSOT benchmark videos. On the unseen SBI-2015 dataset, DeepFTSG had an F-measure accuracy of 87%, more than 30% higher compared to the top-performing deep network FgSegNet_v2 and outperforms the recently proposed KimHa method by 17%. On the unseen LASIESTA, DeepFTSG had an F-measure of 88% and outperformed the best recent deep learning method BSUV-Net2.0 by 3%. On the unseen LaSOT with axis-aligned bounding box ground-truth, network segmentation masks were converted to bounding boxes for evaluation, DeepFTSG had an F-Measure of 55%, outperforming KimHa method by 14% and FgSegNet_v2 by almost 1.5%. When a customized single DeepFTSG model is trained in a scene-dependent manner for comparison with state-of-the-art approaches, then DeepFTSG performs significantly better, reaching an F-Measure of 97% on SBI-2015 (+ 10%) and 99% on LASIESTA (+ 11%). The source code, pre-trained weights, and video demo for DeepFTSG are available at https://github.com/CIVA-Lab/DeepFTSG.

Keywords Change detection · Background subtraction · Flux tensor split Gaussian · U-Net · Squeeze and excitation deep network · XAI

Communicated by Matteo Poggi.

✉ Gani Rahmon
gani.rahmon@mail.missouri.edu

Kannappan Palaniappan
palaniappank@missouri.edu

Imad Eddine Toubal
itoubal@mail.missouri.edu

Filiz Bunyak
bunyak@missouri.edu

[1] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA

[2] U.S. DEVCOM Army Research Laboratory, Adelphi, MD, USA

[3] U.S. Naval Research Laboratory, Washington, DC, USA

1 Introduction

The world is in motion, and stateful dynamic perception that provides reactive information to the perceiver about the world is essential for the interpretation of visual motion across different time scales. The perception of moving objects is a hallmark of visual intelligence since autonomous systems need to interact with the world, not just perceive it. Gibson's *ecological optics* conceptualized vision as an active perceptual system in which space and motion perception are inseparable (Gibson, 1950). A stateful moving object detection stream focuses attention on visual perceptual processes such as tracking, recognition, avoidance, comprehension, interaction, behavior, etc. General motion detection and segmentation is a challenging task because of background clutter, distracting surfaces, occlusions, sporadic object motion, and changing environments such as camera motion, degraded imaging optics, weather, haze, fog, dust, smoke, dynamic background, illumination changes, specularities, shadows, repetitive textures or camouflage effects. Many approaches and pipelines have been proposed for moving object detection to tackle the challenges mentioned above (Barnich & Van Droogenbroeck, 2011; Bianco et al., 2017; Shervin et al., 2020). Earlier approaches typically consisted of hand-crafted solutions with limited adaptation to changing scenarios and often relied on a collection of special case procedures to handle challenging conditions and video categories. Recently, deep learning architectures have been developed for supervised learning-based moving object change detection. Transfer learning combined with many state-of-the-art CNN models like VGG-16 and ResNet-18 trained on large benchmark datasets provides suitable feature embeddings to be learned for new visual tasks with only minor modifications and limited training requirements. Autoencoders are a popular deep learning architecture for segmentation tasks. The features extracted in the encoder module, using a series of convolution and pooling layers, are upsampled by the decoder module to recover the original spatial resolution of the input image.

However, many current deep learning networks proposed for moving object detection rely on a single image using spatial-only appearance cues within an encoder-decoder framework and ignore the rich temporal dimension (Lim & Keles, 2018, 2020).

In this paper, we propose a novel hybrid moving object detection system, Deep Flux Tensor with Split Gaussian (DeepFTSG), which integrates a learned neural appearance model with FTSG motion and change cues using a single and multi-encoder with a shared decoder fusion network for robust moving object detection.

The proposed DeepFTSG networks extend our recent *single-stream* Motion U-Net (Rahmon et al., 2021) hybrid deep architecture for motion segmentation which augments deep appearance with shallow motion and change cues using early fusion. In our earlier work, the *unsupervised* Flux Tensor with Split Gaussian (FTSG) motion analysis algorithm (Wang et al., 2014a), which detects motion across multiple temporal scales, won the CVPR 2014 Change Detection Workshop challenge with an overall F-measure of 72.83% (Goyette et al., 2012; Wang et al., 2014b).

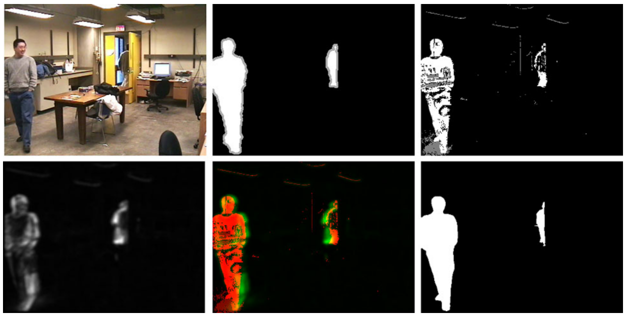


Fig. 1 DeepFTSG generalization result on a frame from unseen SBI-2015 video HumanBody2 (Fr = 730). First row: original image, ground truth mask, mixture of Gaussians (MoG). Second row: Flux, MoG Union Flux, our DeepFTSG mask

The proposed DeepFTSG networks with early and middle-fusion architectures consist of single and multi-stream encoder modules extended by squeeze and excitation blocks, followed by a shared decoder module after multiple bottleneck stages associated with each stream which can be viewed as a joint topological fused feature representation prior to the decoding stream. The squeeze and excitation blocks allow the network to perform feature recalibration by emphasizing informative features and suppressing less useful ones. Figure 1 shows sample moving object detection results using the proposed DeepFTSG network. Figures 2 and 5 provide an overview of the proposed architectures and squeeze and excitation blocks that will be described in detail in later sections. Two versions of DeepFTSG were tested—DeepFTSG-1 consists of a single-stream, where appearance-based and spatiotemporal features are fused early before being fed to the network. DeepFTSG-2 consists of two streams, where the first stream receives a three-channel RGB frame as input and extracts appearance-based, spatial-only features; the second stream receives pixel-level motion and change cues for the corresponding video frame and encodes spatiotemporal features. The feature maps generated by these multi-streams are then concatenated and processed through the network's decoder part resulting in a robust, multi-cue, moving object detection system. Pixel-level flux motion and background subtraction change cues are obtained using unsupervised hand-crafted approaches that do not require any training stage or labeled frames.

Robust multiscale object detection, image segmentation, and tracking tasks require object-level and pixel-level cues. The proposed DeepFTSG integrates pixel-level motion, and change cues efficiently computed using hand-crafted methods, with learned pixel and object-level appearance cues within a deep learning framework. The motion and change cues enable spatiotemporal reasoning, while the learned appearance features and feature fusion incorporate region and object-level information and semantic reasoning, significantly improving performance.

The main contributions of this paper are: (1) a robust moving object detection approach that integrates complementary appearance, motion, and change cues for spatiotemporal reasoning; (2) a novel multi-stream deep autoencoder network for fusing appearance-based and spatiotemporal information; (3) a hybrid, decoupled processing pipeline that takes advantage of hand-crafted pixel-level cues for reduced network complexity and labeled training data; and (4) the generalization capability of the proposed DeepFTSG to unseen videos, scenes and object categories compared to other approaches. The proposed system has been tested and evaluated on the comprehensive Change Detection 2014 Challenge dataset (Wang et al., 2014b).

2 Background and Related Work

Classical moving object detection approaches can be categorized into three broad classes; optical flow methods, temporal differencing, and background subtraction. Comprehensive reviews of these classical moving object detection methods can be found in Radke et al. (2005); Benezeth et al. (2008); Brutzer et al. (2011). Optical flow methods can be used with non-stationary cameras. However, reliable motion field computation under real-world conditions is challenging and computationally expensive, and these methods cannot deal with stopped objects. Temporal differencing-based methods are simple, fast, and can quickly adapt to different changes and thus are suitable for dynamic backgrounds, illumination changes, uncovered backgrounds by removed objects, etc. However, without an explicit background model, they cannot detect slow-moving or stopped objects, often resulting in foreground aperture problems and failing to detect parts of objects (particularly large objects with homogeneous interiors resulting in holes). Background subtraction-based methods that rely on change from an explicit background model are among the most popular moving object detection methods since they can handle slow-moving or stopped objects and do not suffer from foreground aperture problems. Sparse recovery methods for background subtraction are widely studied in the literature (Candes et al., 2011; Zhou et al., 2012; Liu et al., 2017; Xin et al., 2015; Liu et al., 2015). These methods identify moving objects by extracting sparse components from surveillance video frames, while low-rank components represent a background of stationary objects. However, background subtraction methods are sensitive to dynamic scene changes due to illumination changes, revealed background from moving objects, etc. Methods combining these approaches, such as Wang et al. (2014a), have produced better results.

The development of real-world computer vision systems has been revolutionized with the adoption of deep neural learning methods. Recent approaches for moving object detection explore deep learning architectures including convolutional neural networks (CNNs), generative adversarial networks (GANs), autoencoders (AE), recurrent neural networks (RNNs), multibranch networks trained with labeled data. DeepBS (Babaee et al., 2018) proposed a convolutional neural network trained using a combination of input frames and associated background images using the patch-based technique. The network is trained with randomly selected video frames (5% of the CDnet-2014 dataset) and associated ground truth masks. BSUV-net 2.0 (Tezcan et al., 2021) uses a fully convolutional neural network for background subtraction of unseen videos. The network input consists of a current frame and two background frames taken at different time points, along with their semantic segmentation results. A pre-trained DeepLabv3 is used to extract semantic segmentation results. BSGAN (Wenbo et al., 2020) uses median filtering for background estimation and then trains a Bayesian GAN to classify each pixel, to handle slow and sudden illumination changes, non-stationary backgrounds, and ghosting. Deep CNNs are adopted to construct the generator and the discriminator of Bayesian GAN. A 3D convolutional neural network with long short-term memory (LSTM) was proposed by Akilan et al. (2020) to incorporate temporal information in a deep learning framework for background subtraction. 3D convolutions manage the time-dependent video cues to capture the short temporal motions, and LSTM modules handle the long-short temporal motions during the down-sampling and up-sampling stages. Cascade CNN (Wang et al., 2017) is based on multi-resolution CNNs with a cascaded architecture. The network is trained with hand-picked frames that are made publicly available by the authors. FgSegNet (Lim & Keles, 2018) uses two encoder-decoder networks that produce multi-scale feature encodings. In the first model, three scales of inputs are given to an encoder. In the second model, a feature pooling module is included to extract multi-scale features. Both models use transposed CNNs on the decoder side. For training, 50 to 200 informative frames were manually selected with ground truth masks from the CDnet-2014 dataset. FgSegNet approach uses multiple networks that are optimized *per video-sequence*. FC-Siam (Caye Daudt et al., 2018) uses an encoder-decoder network with single and multiple streams to detect the change between two data images from large-scale Earth observation systems such as Copernicus or Landsat. Since two streams carry similar information, the authors used shared weights between them in the encoder part of the proposed fully connected siamese network.

Because many moving object detection benchmarks were established before the recent popularity of deep learning methods, no specific training and testing dataset partitions have been established in the benchmarks. That leads to different training and testing video frame partitioning schemes in various papers, making a fair comparison difficult. Consequently, as pointed out by Tezcan et al. (2021), most of the top-performing deep-moving object detection systems have been video frame- or video group-optimized and have never been tested on unseen videos, making it hard to judge their generalization capabilities. We address this limitation by using CDnet-2014 for training and validation, and SBI-15 (Maddalena & Petrosino, 2015) and LASIESTA (Carlos et al., 2016) as unseen testing videos.

Change detection can help us track and study the movement and behavior of arbitrary objects in a video sequence (Theau, 2008). Accurate video segmentation is, therefore, a crucial step in change detection. Moreover, video segmentation is an initial step of video object tracking. Hence, object-tracking datasets can also be used to evaluate the generalization capabilities of moving object detection methods. Many publicly available object tracking datasets, either single or multiple object tracking, could be used to address the generalization capability of the moving object detection methods. However, the evaluation result won't be that accurate since moving object detection detects moving objects in the scene, and there might be more than one object moving in that scene, but in the case of single object tracking, only one object (object of interest) in the scene would have a ground truth and the other object even if they are moving would be ignored. Therefore, we used some video sequences of LaSOT (Fan et al., 2019) single object tracking dataset as unseen test videos to evaluate the generalization capability of the proposed methods.

3 Change Detection Deep Learning Networks

We have designed a novel hybrid system to robustly detect moving foreground objects. The proposed system combines unsupervised computer vision methods for motion and change detection with deep learning-based semantic segmentation and fusion frameworks. This approach reduces architecture complexity and the need for extensive labeled training datasets by taking advantage of available handcrafted solutions that produce fast, reliable results. We built two deep networks to better analyze the contribution of motion and change cues to overall system performance. The first network, DeepFTSG-1 in Fig. 2, consists of a U-Net-like semantic segmentation architecture extended by squeeze and excitation blocks with single input streams, where the appearance-based and spatiotemporal information are fused early before being fed to the network. Our second network, DeepFTSG-2 in Fig. 5, extends DeepFTSG-1 by decoupling appearance-based information from spatiotemporal using multiple input streams. The first stream has appearance information, and the second stream incorporates spatiotemporal reasoning through motion and change cues. Finally, the two streams are combined after the joint topological representation through middle fusion.

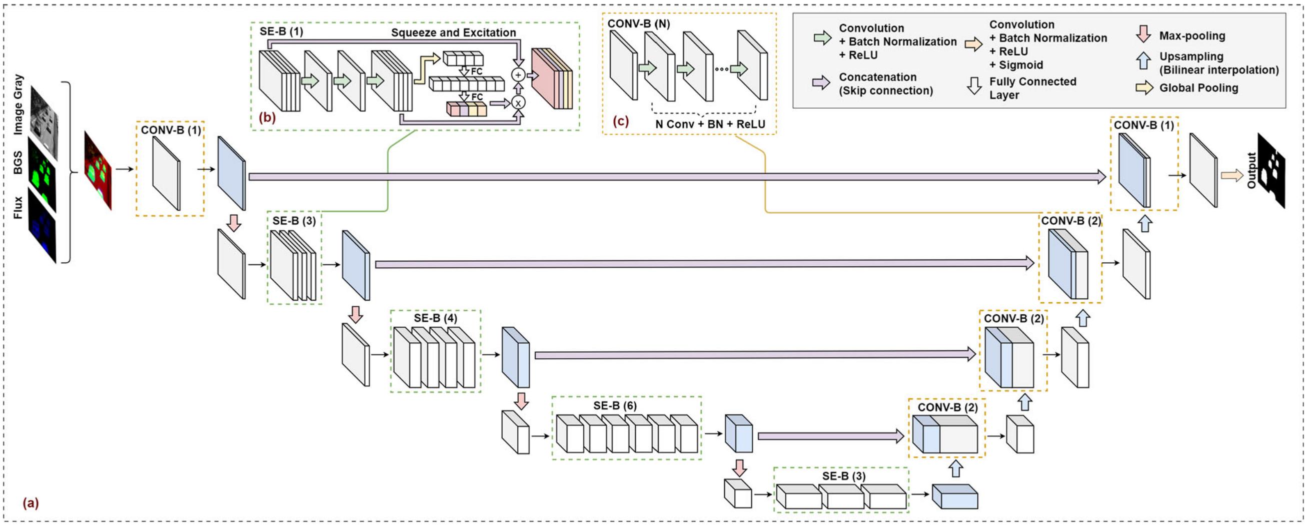


Fig. 2 The single-stream DeepFTSG-1 architecture with *early fusion*, and SE-ResNet-50 backbone, where each residual block uses a final squeeze and excitation. (a) shows the general architecture with 3-channel input (Gray, BGS, Flux), (b) shows the detailed conv block in the decoder, (c) shows the detailed conv block in the encoder

3.1 DeepFTSG-1: Single-Stream Early Fusion for Spatiotemporal Change Detection

Single frame object of interest detection and semantic segmentation tasks have been revolutionized by using recent deep learning architectures (Girshick, 2015; He et al., 2017; Redmon et al., 2016; Chen et al., 2018). While effective, single-frame detection networks similar to FgSegNet_v2 that rely only on appearance cues for change detection suffer from three main limitations: (1) they cannot detect untrained new (moving) objects, (2) they fail when appearance cues are limited (e.g. small targets when objects are far from the camera), and (3) they cannot differentiate between moving and stationary object instances. This brittleness leads to a lack of generalization power in such networks and likely will not scale well in real-world applications with unanticipated inputs (Yuille & Liu, 2020).

DeepFTSG augments appearance-based features with hand-engineered motion and change cues using fast unsupervised vision algorithms. The proposed single-stream moving object detection network, DeepFTSG-1, is an extension to our previous network named MU-Net2 (Rahmon et al., 2021), based on a SE-ResNet-50 (Hu et al., 2018) backbone instead of a normal U-Net encoder, where squeeze and excitation blocks are used after each residual block of the ResNet-50. That enables deeper layers without gradient degradation during network learning by using identity shortcut connections that skip one or more layers to facilitate deeper information propagation. In addition, the squeeze and excitation blocks allow the network to perform feature recalibration by emphasizing informative features and suppressing less useful ones. The proposed DeepFTSG-1 uses motion cues as input computed from multi-modal change detection and flux motion through our fast tensor-based motion estimation (Bunyak et al., 2007) and an adaptive split-gaussian multi-modal background subtraction model (Wang et al., 2014a; Zivkovic & van der Heijden, 2006; Zivkovic, 2004) respectively. DeepFTSG-1 incorporates a three-channel input processing stream, with the first (red) channel being the appearance (the three-channel RGB color input is converted to grayscale). The motion and change cues corresponding to the current frame computed using a background model based on past frames for the case of slower temporal change and a temporal sliding window of frames for the case of flux motion are assigned to the second (G) and third (B) channels.

Figure 2 shows the overall architecture of the proposed DeepFTSG-1 single-stream moving object detection network with an early fusion of motion cues. The overall network is similar to the U-Net architecture (Ronneberger et al., 2015) with skip connections after each block of SE-ResNet-50. The decoder part of the proposed network consists of four blocks, where the feature maps are upsampled and concatenated in each block to the feature maps from the corresponding SE-ResNet-50 block. Finally, a 1×1 convolution layer is applied to decrease the number of feature maps, and a final sigmoid activation layer produces the class label probabilities. Thresholding these probabilities leads to foreground/background segmentation masks. Table 1 provides detailed configuration and specifications of the proposed DeepFTSG-1.

We detail below how fast unsupervised scene-independent methods estimate change and motion cues.

3.1.1 Multi-modal Background Subtraction for Change Estimation

Change is estimated using a background subtraction (BGS) approach. There is extensive literature on estimating background subtraction models for identifying temporal change (Crivelli et al., 2011; Andrews & Antoine, 2014; Yizhe & Elgammal, 2017). To efficiently tackle multi-modal backgrounds, we use the adaptive mixture of Gaussians method described in Zivkovic and van der Heijden (2006); Zivkovic (2004) and implemented in OpenCV library (*BackgroundSubtractorMOG2*). The method supports a variable number of Gaussian models per pixel. The OpenCV implementation also enables shadow detection by default. The only parameter that we are setting is the variance threshold for the pixel-model matching (*setVarThreshold*), and it is empirically chosen as 16. Before feeding to the background subtraction module, the image sequence is smoothed using a 5×5 Gaussian filter. Foreground masks obtained from the background subtraction module are given to the DeepFTSG-1 and DeepFTSG-2 networks as input. The process returns information on longer-term change corresponding to moving objects, once moving but then become stopped objects and other long-term changes in the scene. Fig. 3 demonstrates the result of background subtraction for a single time-step.

3.1.2 Tensor-Based Motion Estimation

While background subtraction algorithms have advantages such as robustness to aperture problems and response to stopped objects, they are prone to dynamic background changes. Due to the recursive nature of these approaches, any error in background estimation also tends to persist for a long time. The temporal dynamics are slowly updated. To account for fast motion or short-term change, while being robust to dynamic background changes and background estimation errors, we use an explicit motion detection module. For fast and robust motion estimation, we use our efficient tensor-based motion computation scheme flux tensor (Bunyak et al., 2007) and build upon our previous experience in optimizing FTSG (Wang et al., 2014a). Optical flow-based motion estimation using deep and traditional methods can be

Table 1 Detailed configuration and specifications of the proposed DeepFTSG-1

Encoder			Decoder		
Layer	Output dimension	Block	Layer	Output dimension	Block
Input	320×480×3	–	Sigmoid	320×480×1	–
conv, 7×7, 64, stride 2	160×240×64	Down-sampling	Conv	320×480×1	
max pool, 3×3. stride 2	80×120×64	(CONV-B(1))	[Conv + BN + ReLu] × 2 Upsampling	320 × 480 × 16 320 × 480 × 32	Up-sampling
$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} \times 3$	80×120×256	Down-sampling (SE-B(3))	[Conv + BN + ReLU] × 2 Concatenation Upsampling	160 × 240 × 32 160 × 240 × 128 160 × 240 × 64	Up-sampling
$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} \times 5$	40×60×512	Down-sampling (SE-B(4))	[Conv + BN + ReLU] × 2 Concatenation Upsampling	80 × 120 × 64 80 × 120 × 384 80 × 120 × 128	Up-sampling
$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} \times 6$	20 × 30 × 1024	Down-sampling (SE-B(6))	[Conv + BN + ReLU] × 2 Concatenation Upsampling	40 × 60 × 128 40 × 60 × 768 40 × 60 × 256	Up-sampling
$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}, [128, 2048] \end{bmatrix} \times 3$	10 × 15 × 2048	Bottleneck SE-B(3)	[Conv + BN + ReLU] ×2 Concatenation Upsampling	20 × 30 × 256 20 × 30 × 3072 20 × 30 × 2048	Up-sampling

alternatively used (Dosovitskiy et al., 2015; Sun et al., 2018; Schuster et al., 2020). The flux tensor represents the temporal variation of the optical flow field within the local 3D spatiotemporal volume (Bunyak et al., 2007; Palaniappan et al., 2011; Wang et al., 2014a). In expanded matrix form, the flux tensor is defined as,

$$J_F = \begin{bmatrix} \int_\Omega \left\{\frac{\mathrm{d}^2 I}{\mathrm{d}x\mathrm{d}t}\right\}^2 \mathrm{d}y & \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}x\mathrm{d}t}\frac{\mathrm{d}^2 I}{\mathrm{d}y\mathrm{d}t}\mathrm{d}y & \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}x\mathrm{d}t}\frac{\mathrm{d}^2 I}{\mathrm{d}t^2}\mathrm{d}y \\ \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}y\mathrm{d}t}\frac{\mathrm{d}^2 I}{\mathrm{d}x\mathrm{d}t}\mathrm{d}y & \int_\Omega \left\{\frac{\mathrm{d}^2 I}{\mathrm{d}y\mathrm{d}t}\right\}^2 \mathrm{d}y & \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}y\mathrm{d}t}\frac{\mathrm{d}^2 I}{\mathrm{d}t^2}\mathrm{d}y \\ \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}t^2}\frac{\mathrm{d}^2 I}{\mathrm{d}x\mathrm{d}t}\mathrm{d}y & \int_\Omega \frac{\mathrm{d}^2 I}{\mathrm{d}t^2}\frac{\mathrm{d}^2 I}{\mathrm{d}y\mathrm{d}t}\mathrm{d}y & \int_\Omega \left\{\frac{\mathrm{d}^2 I}{\mathrm{d}t^2}\right\}^2 \mathrm{d}y \end{bmatrix} \quad (1)$$

where I is a spatiotemporal image volume and derivatives are calculated in x, y, t and integrated within the local area Ω. The elements of the flux tensor incorporate information about spatiotemporal gradient changes. By analyzing the changes in the gradient of the image intensity over time, the flux tensor can identify regions of the image that correspond to moving objects. This information can be used to segment the image into moving and stationary regions, allowing for efficient discrimination between the two. Sequential and parallel computations of the flux tensor matrix are described in Palaniappan et al. (2011). The trace of the flux tensor matrix can be compactly written,

$$trace(J_F) = \int_\Omega ||\frac{\mathrm{d}}{\mathrm{d}t} \nabla I||^2 \mathrm{d}y \quad (2)$$

and computed efficiently to classify moving and non-moving regions without expensive eigenvalue decompositions (Palaniappan et al., 2010; Dardo et al., 2016). There are four hyper-parameters that needs to be set, and we set them empirically as follows: *spatial filter size = 7, spatial averaging size = 5, temporal filter size = 7,* and *temporal averaging size = 5*. The output of the flux tensor is given directly to DeepFTSG-1 and DeepFTSG-2 networks as input. Figure 3 shows the sample result of flux analysis for a single frame of processed video. In Fig. 4 the spatio-temporal volumes of a video sequence are demonstrated and in *(b)* the flux is shown to visualize motion through time.

3.2 DeepFTSG-2: Multi-Stream Middle Spatiotemporal Fusion

The proposed DeepFTSG-2 extends DeepFTSG-1 by decoupling appearance-based information from spatiotemporal using multiple input streams. The first input stream receives three-channel RGB color input from the current frame, and the second input stream receives motion and change cues corresponding to the current frame, which is computed using a temporal sliding window of frames for the case of motion and using a background model computed from past frames for the case of change. The two input streams go through two parallel feature extraction modules. The first processing stream (appearance encoder) extracts spatial appearance features using the SE-ResNet-50 backbone, and the second processing stream (motion encoder) extracts spatiotemporal, motion, and change-based features using the ResNet-18 backbone. The feature maps generated by these two encoders are then fused and processed through the network's decoder. The motion and change cues are stacked channel-wise, where the red channel (R) corresponds to the background subtraction mask, the green channel (G) corresponds to the motion mask, and the blue channel (B) is set to 0. Three-channel input is used to comply with the ResNet-18 input format. DeepFTSG-2 does not share weights between the two streams and uses intermediate fusion since the input streams are of different phenomena, such as RGB and motion.

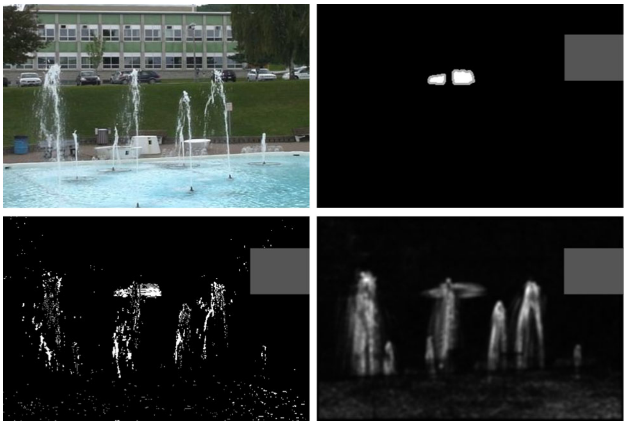


Fig. 3 BGS and flux result on a frame from CDnet-2014 video fountain01 (Fr = 730). First row: original image, ground truth. Second row: BGS, Flux

A deep encoder backbone is adopted for the appearance encoder as we take the raw image as input. The deep architecture (SE-ResNet-50), equipped with squeeze and excitation blocks, allows for deep feature extraction. A shallower backbone (ResNet-18) is used for the motion encoder since we use higher-level feature maps as input. Both streams have five spatial resolutions, each of which feeds into the corresponding block of the decoder through skip connections. Fig. 5 illustrates the architecture of the multi-stream DeepFTSG-2 with middle fusion along with the temporal domain. Table 2 provides detailed configuration and specifications of the proposed DeepFTSG-2. We provide a comprehensive set of results in Tables 16 and 17 to show the effect of different architectures/backbones on the quality of change detection.

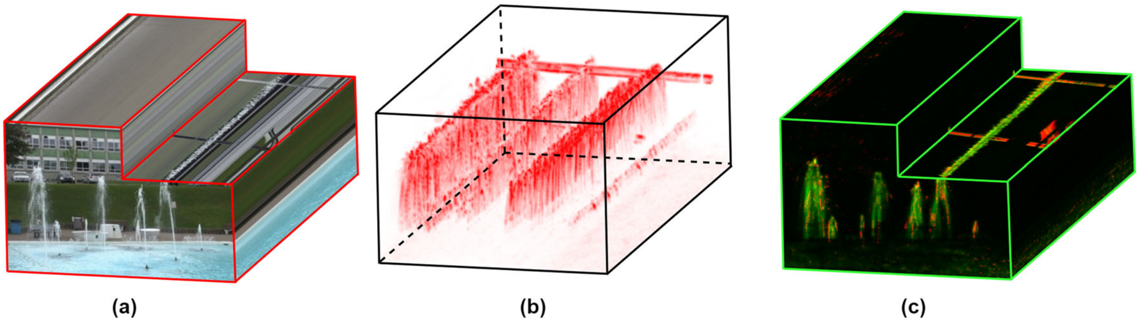


Fig. 4 Motion visualized as oriented energy fields. Spatio-temporal brightness volumes ($x - y - t$) of a video sequence fountain01 from CDnet-2014: **a** original input, **b** flux motion, **c** combination of motion and change (green channel is flux (fast) motion, red channel is multi-modal (persistent) change). In **c** a maximum intensity volume rendering is shown without alpha transparency for the black voxels

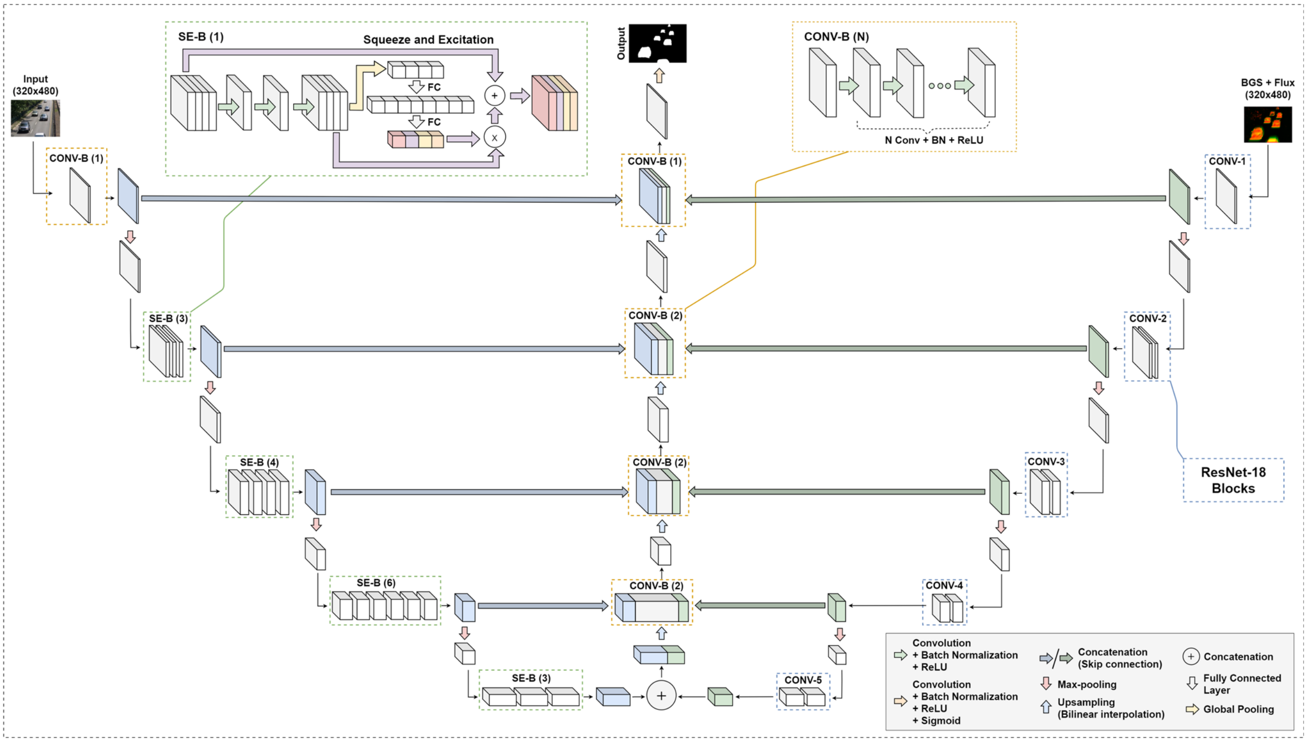


Fig. 5 The multi-stream DeepFTSG-2 *USE-Net trellis* architecture with *middle fusion* (or intermediate fusion), which includes appearance (RGB 3-channels) in the first stream, and multi-channel motion (flux) plus change cues (BGS 2-channels) in the second stream. The feature embedding backbones are fused in the decoder stage with shared forward connections

4 Experimental Results

In this section, we present details on test datasets, evaluation metrics, qualitative and quantitative results of the proposed DeepFTSG deep learning change detection architecture.

4.1 Benchmark Evaluation Datasets

We used four benchmark datasets to evaluate the proposed method, CDnet-2014 change detection challenge dataset (Wang et al., 2014b), SBI-2015 scene background initialization dataset (Maddalena & Petrosino, 2015), Labeled and Annotated Sequences for Integral Evaluation of SegmenTation Algorithms (LASIESTA) dataset (Carlos et al., 2016) and Large-scale Single Object Tracking (LaSOT) dataset (Fan et al., 2019). CDnet-2014 was developed to enable objective and precise quantitative comparison and ranking of change detection algorithms. It consists of nearly 159,278 frames, with 118,173 labeled from 53 video sequences, organized into 11 categories corresponding to realistic scenarios

Table 2 Detailed configuration and specifications of the proposed DeepFTSG-2

1st stream encoder		2nd stream encoder		Decoder	
Layer	Output dimension	Layer	Output dimension	Layer	Output d imension
Input	320×480×3	Input	320×480×3	Sigmoid	320×480×1
conv, 7×7, 64, stride 2	160×240×64	conv, 7×7, 64, stride 2	160×240×64	Conv	320×480x1
max pool, 3×3, stride 2	80×120×64	max pool, 3×3, stride 2	80×120×64	[Conv + BN + ReLU] ×2 Upsampling	320 × 480 × 16 320 × 480 × 32
$\begin{bmatrix} \text{conv}, 1 \times 1, 64 \\ \text{conv}, 3 \times 3, 64 \\ \text{conv}, 1 \times 1, 256 \\ \text{fc}, [16, 256] \end{bmatrix} \times 3$	80 × 120 × 256	$\begin{bmatrix} \text{conv}, 3 \times 3, 64 \\ \text{conv}, 3 \times 3, 64 \end{bmatrix} \times 2$	80 × 120 × 64	[Conv + BN + ReLU] ×2 Concatenation Upsampling	160 × 240 × 32 160 × 240 × 192 160 × 240 × 64
$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \\ \text{conv}, 1 \times 1, 512 \\ \text{fc}, [32, 512] \end{bmatrix} \times 4$	40 × 60 × 512	$\begin{bmatrix} \text{conv}, 3 \times 3, 128 \\ \text{conv}, 3 \times 3, 128 \end{bmatrix} \times 2$	40 × 60 × 128	[Conv + BN + ReLU] ×2 Concatenation Upsampling	80 × 120 × 64 80 × 120 × 448 80 × 120 × 128
$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \\ \text{conv}, 1 \times 1, 1024 \\ \text{fc}, [64, 1024] \end{bmatrix} \times 6$	20 × 30 × 1024	$\begin{bmatrix} \text{conv}, 3 \times 3, 256 \\ \text{conv}, 3 \times 3, 256 \end{bmatrix} \times 2$	20 × 30 × 256	[Conv + BN + ReLU] ×2 Concatenation Upsampling	40 × 60 × 128 40 × 60 × 896 40 × 60 × 256
$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \\ \text{conv}, 1 \times 1, 2048 \\ \text{fc}\,[\mathbf{128}, 2048] \end{bmatrix} \times 3$	10 × 15 × 2048	$\begin{bmatrix} \text{conv}, 3 \times 3, 512 \\ \text{conv}, 3 \times 3, 512 \end{bmatrix} \times 2$	10 × 15 × 512	[Conv + BN + ReLU] ×2 Concatenation Upsampling	20 × 30 × 256 20 × 30 × 3584 20 × 30 × 2560
				Concatenation	10 × 15 × 2560

Table 3 Distribution of major foreground object categories in each dataset collection indicating number of videos (vid) and total number of frames (fr)

Dataset	Person vid, fr	Vehicle vid, fr	Animal vid, fr	Synthetic vid, fr	Other vid, fr
CDnet-2014	30, 89459	31, 93800	0, 0	0, 0	4, 12200
SBI-2015	8, 2991	2, 933	0, 0	0, 0	1, 345
LASIESTA	43, 14860	5, 2975	0, 0	24, 8580	0, 0
LaSOT	4, 10090	2, 5060	4, 9224	0, 0	0, 0

and challenging conditions, including illumination change, bad weather, dynamic background, night videos, PTZ, thermal, etc. Spatial resolutions of the videos in the dataset vary from 320 × 240 to 720 × 576 and may include multiple moving objects. CDnet-2014 is the most comprehensive dataset for change and moving object detection, with continuously updated evaluations posted on the Change Detection Workshop website. We used the same approach as in FgSegNet (Lim & Keles, 2018) and (Wang et al., 2017) by selecting 200 frames from each video sequence within the labeled frames of the original CDnet-2014 dataset for training the proposed DeepFTSG networks. This split used only 10,600 CDnet frames for training, corresponding to approximately 6.6% of the whole dataset, with the remainder of the labeled frames used for testing, including hidden frames.

The Scene Background Initialization (SBI) 2015 dataset contains 14 video sequences with ground-truth labels provided by (Wang et al., 2017). We used 10 suitable video sequences from this dataset to evaluate our video segmentation models trained only on the CDnet-2014 dataset to see the generalization capability. However, the video sequences “Foilage”, “PeopleAndFoilage”, “Snellen” and “Toscana” were not used in the evaluation of our pre-trained model. The reason is that “Foilage”, “PeopleAndFoilage”, and “Snellen” eventually deal with the moving branch of a tree, which is not our object of interest, and “Toscana” has only six frames in total.

The Labeled and Annotated Sequences for Integral Evaluation of SegmenTation Algorithms (LASIESTA) dataset is composed of many real indoor and outdoor sequences organized in different categories, each of one covering a specific challenge in moving object detection strategies. Moreover, it contains sequences recorded with static and moving cameras and provides information about the moving objects remaining temporally static. LASIESTA dataset contains 26 indoor and 20 outdoor sequences (having 12 simulated motion sequences in both indoor/outdoor), and it is fully annotated at both pixel-level and object-level. We used all video sequences from this dataset to evaluate our video segmentation models trained only on the CDnet-2014 dataset to see the generalization capability.

There are few fully annotated video datasets for moving object detection with precise ground truth segmentation masks. However, there is an extensive collection of video datasets for object tracking with ground truth bounding boxes. Large-scale Single Object Tracking (LaSOT) is a large video collection for a single object tracking. LaSOT consists of 1550 video sequences with more than 3.87 million high-quality manually annotated frames incorporating careful inspection. We did an initial motion segmentation test using five object categories from LaSOT, including bicycle, car, dog, giraffe, and person, with two sample video sequences from each category to evaluate the generalization capacity of our video segmentation architecture models trained using only the CDnet-2014 dataset.

The Table 3 demonstrates the category distribution of each dataset used in this paper. The categories are person, vehicle (car, bus, track, etc.), animal (dog, giraffe, etc.), synthetic (simulated motion), and other. We demonstrate how many video sequences of each dataset have those categories and an approximate number of frames.

4.2 DeepFTSG Training Details

Weights for the ResNet-18 and SE-ResNet-50 modules used in DeepFTSG-1 and DeepFTSG-2 are initialized with pre-trained weights on ImageNet. The input image size to the deep networks is 320 × 480. Adam optimizer is used during training with an initial learning rate of 10^{-4} that is reduced by a factor of 10 after every 20 epochs. The CDnet-2014 training data is shuffled and split into 90% for training and 10% for validation per video basis, with 200×53=10,600 total frames in training set out of almost 160K in the total dataset. Since there is an imbalance between the foreground and background classes (i.e. in some frames foreground area constitutes less than 20% of the total image area), a combined loss function consisting of Dice loss (Eq. 3) (Sudre et al., 2017) and binary cross-entropy loss (Eq. 4) is used to train the network. The smoothed Dice loss is defined as,

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} p_i\, g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2 + \epsilon} \tag{3}$$

in which p_i are the predicted foreground probabilities and g_i is the corresponding ground-truth label, N is the total number of pixels in the mini-batch set of images, and ϵ is a small regularization value. The binary cross-entropy (BCE) loss is defined as,

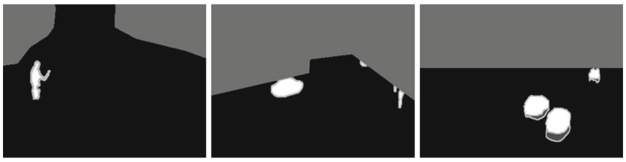


Fig. 6 Ground-truth images from CDnet-2014 dataset with ignored or don’t care regions shown in dark-gray color

$$L_{BCE} = \frac{1}{N} \sum_{i=1}^{N} \left[-g_i \log p_i + (1 - g_i) \log(1 - p_i) \right] \quad (4)$$

where g_i is the true label and p_i is the predicted probability as in L_{Dice}. The final combined loss to be minimized during training is given by,

$$Loss = \lambda * L_{BCE} + (1 - \lambda) * L_{Dice} \quad (5)$$

where λ is the weight parameter and is empirically chosen as 0.5. Using λ that changes with epoch number can lead to a small improvement in performance by emphasizing the segmentation boundary accuracy (Dice loss). In addition, since the labels provided by the CDnet-2014 dataset have ignored (masked out) regions, to avoid a penalty for those regions during training, we updated the loss function to ignore the regions that are not used and not penalize if the foreground is generated in these (don’t care) masked regions. Fig. 6 demonstrates the ground-truth images from CDnet-2014 dataset with ignore regions specified in dark-gray.

The proposed DeepFTSG-1 and DeepFTSG-2 models were implemented in PyTorch and trained for 50 epochs with a mini-batch size of 16. For each epoch, the training and validation samples are reshuffled.

It took ≈ 10 hours (each epoch takes about 12 min) on an NVIDIA GeForce GTX 1080 Ti GPU and ≈ 7.5 hours (each epoch takes about 9 min) on an NVIDIA Tesla V100 GPU to finish the whole training process for the single-stream DeepFTSG-1. For multi-stream DeepFTSG-2, it took ≈ 15 hours (each epoch takes about 18 min) on an NVIDIA GeForce GTX 1080 Ti GPU and ≈ 13.33 hours (each epoch takes about 16 min) to finish the whole training process on an NVIDIA Tesla V100 GPU.

4.3 Evaluation Metrics

We evaluated the performance of the proposed DeepFTSG deep neural architecture on unseen frames in each video and compared it to the top-ranked methods listed in the Change Detection Workshop website using seven assessment metrics (Goyette et al., 2012). The seven metrics are: recall (Re), specificity (Sp), false-positive rate (FPR), false-negative rate (FNR), precision (P), F-Measure (F), and percentage of wrong classifications (PWC), and defined as,

$$Re = \frac{TP}{(TP + FN)}; \qquad Sp = \frac{TN}{(TN + FP)} \quad (6)$$

$$FPR = \frac{FP}{(FP + TN)}; \qquad FNR = \frac{FN}{(TP + FN)} \quad (7)$$

$$P = \frac{TP}{(TP + FP)}; \qquad F = \frac{2 \times P \times Re}{(P + Re)} \quad (8)$$

$$PWC = \frac{100 \times (FN + FP)}{(TP + TN + FP + FN)} \quad (9)$$

where TP (true positive) denotes the number of correctly labeled foreground pixels; TN (true negative) denotes the number of correctly labeled background pixels; FN (false negative) represents the number of wrongly classified foreground pixels, and FP (false positive) represents the number of wrongly classified background pixels. We have computed these metrics using the standardized assessment tool given by Goyette et al. (2012). Lower values indicate better performance for PWC, FNR, and FPR metrics, while higher values indicate better performance for Recall, Precision, and F-Measure metrics. Among these metrics, we use the F-Measure (F) also known as F_1 score, which is the harmonic mean of precision and recall, that is generally accepted as a good indicator of overall change detection performance, balancing precision and recall accuracy to reduce Type I (FP) and Type II (FN) errors.

4.4 Experiments on CDnet-2014 Benchmark Videos

Using the CDnet-2014 dataset, we trained two networks, DeepFTSG-1 and DeepFTSG-2. Quantitative evaluation results for the proposed DeepFTSG-1 and DeepFTSG-2 are shown in Table 4. The proposed DeepFTSG-1 network produced an overall F-measure of 0.9652, while the DeepFTSG-2 network produced a slightly better overall F-measure of 0.97. The lowest performance is in the difficult Night Videos category for both networks with an F-measure of 0.8481 for DeepFTSG-1 and 0.9023 for DeepFTSG-2. For reference, we include the results of our previous FTSG (Wang et al., 2014a) non-deep learning unsupervised approach that won the original CDnet-2014 challenge. The significant improvement (around 25%) of DeepFTSG compared to FTSG demonstrates that incorporating object appearance and fusing change semantics, results in better motion detection without necessarily requiring direct object recognition. DeepFTSG does not use an explicit object detection and classification network to learn object bounding box labels like vehicle, person, animal, bike, etc. though this could be incorporated in the future.

Table 4 Detailed evaluation of DeepFTSG-1 and DeepFTSG-2 on CDnet-2014 eleven video categories compared to Flux Tensor Split-Gaussian (FTSG) unsupervised results (Wang et al., 2014a)

Category	Recall (Re)		PWC		Precision (P)		F-Measure (F)		
	DeepFTSG-1	DeepFTSG-2	DeepFTSG-1	DeepFTSG-2	DeepFTSG-1	DeepFTSG-2	DeepFTSG-1	DeepFTSG-2	**FTSG (F)**
PTZ	0.9657	0.9754	0.0394	0.0303	0.9524	0.9693	0.9583	0.9721	0.3241
BadWeather	0.9592	0.9731	0.0829	0.0652	0.9825	0.9830	0.9706	0.9779	0.8228
Baseline	0.9934	0.9947	0.0247	0.0227	0.9952	0.9948	0.9943	0.9947	0.9330
CameraJit.	0.9903	0.9926	0.0714	0.0645	0.9925	0.9918	0.9914	0.9922	0.7513
DynamicBg.	0.9926	0.9926	0.0094	0.0089	0.9974	0.9973	0.9950	0.9950	0.8792
Intermitt	0.9850	0.9869	0.1077	0.1028	0.9989	0.9986	0.9919	0.9927	0.8927
LowFrameR.	0.9480	0.9494	0.0739	0.0836	0.9294	0.9212	0.9374	0.9328	0.6259
NightVid.	0.7753	0.8527	1.2387	0.6009	0.9569	0.9608	0.8481	0.9023	0.5130
Shadow	0.9933	0.9941	0.0541	0.0493	0.9947	0.9950	0.9940	0.9945	0.8535
Thermal	0.9866	0.9871	0.0633	0.0809	0.9963	0.9921	0.9914	0.9896	0.7768
Turbulence	0.9338	0.9303	0.0390	0.0473	0.9570	0.9273	0.9447	0.9259	0.7127
Overall	**0.9566**	**0.9663**	**0.1640**	**0.1051**	**0.9776**	**0.9756**	**0.9652**	**0.9700**	**0.7283**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

Table 5 Comparison using CDnet-2014 of DeepFTSG-1 and DeepFTSG-2 to top performing deep learning methods, including methods FgSegNet_v2 (Lim & Keles, 2020), FgSegNet_S (Lim & Keles, 2018), FgSegNet (Lim & Keles, 2018), BSPVGAN (Wenbo et al., 2020), MU-Net2 (Rahmon et al., 2021), BSGAN (Wenbo et al., 2020), FTSG (Wang et al., 2014a)

Methods	Overall				
	Rank	Re	PWC	P	F
FgSegNet_v2	1	**0.9891**	**0.0402**	**0.9823**	**0.9847**
FgSegNet_S	2	0.9896	0.0461	0.9751	0.9804
FgSegNet	3	0.9836	0.0559	0.9758	0.9770
BSPVGAN	4	0.9544	0.2272	0.9472	0.9501
MU-Net2	5	0.9454	0.2347	0.9407	0.9369
BSGAN	6	0.9476	0.3281	0.9232	0.9339
FTSG	22	0.7657	1.3763	0.7696	0.7283
M_FgSegNet_v2(50%)*		0.2705	2.9684	0.7182	0.3406
M_FgSegNet_v2*		0.8675	0.3751	0.9521	0.9078
M_KimHa*		0.9351	0.3331	0.9201	0.9275
DeepFTSG-1		0.9566	0.1640	0.9776	0.9652
DeepFTSG-2		0.9663	0.1051	0.9756	0.9700

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

M_FgSegNet_v2* and M_KimHa* with * indicating *single* network models trained using the same configuration used to train the DeepFTSG network. M_FgSegNet_v2(50%)* has *11 models*, one for each CDnet-2014 category, and the majority voting rule is used to get a final mask

We compare the performance of the proposed DeepFTSG-1 and DeepFTSG-2 methods with the top six state-of-the-art methods listed in Goyette et al. (2012) as shown in Table 5. Evaluations were done by uploading the results to the CDnet-2014 challenge website. Since the results are not published yet, they can be reached through these links, for DeepFTSG-1[1] and for DeepFTSG-2.[2] It can be seen from Table 5 that our proposed DeepFTSG is competitive with current state-of-the-art supervised methods and outperforms BSPVGAN and our previous work MU-Net2.

Since the authors of the current top-ranked method FgSegNet_v2 made their code publicly available on GitHub, we could run the code and produce the same results as the ones supplied by the authors. Table 6 summarizes the main differences between FgSegNet_v2 and our proposed DeepFTSG architectures.

One of the key differences is that FgSegNet_v2 trains a *separate* deep network for each video sequence, resulting in an *ensemble* of 53 distinctly parameterized networks for inference. The process takes considerable training time (29 days) to generate the ensemble of networks. On the other hand, our approach trains a *single* network, using only 200 frames per video, sufficient for inference across all 53 video sequences. Training takes a fraction of the time, 10 h for DeepFTSG-1 and 15 h for DeepFTSG-2. Compared to FgSegNet_v2, DeepFTSG requires less than 2.2% of the training time, and the inference is more efficient using less than 10% of the neural weights, robustly fuses appearance and motion, has competitive accuracy on CDnet-2014 and most importantly shows high generalizability across all objects and scenes using a single network. DeepFTSG can readily distinguish between moving and stationary states of the same object type, whereas FgSegNet_v2 and other similar architectures cannot.

Initially, to make a fair comparison between our proposed networks and top performing network FgSegNet_v2 on the CDnet-2014 dataset, we trained FgSegNet_v2 network for each category of the CDnet-2014 dataset instead of at the video level, resulting in 11 network models, instead of 53. Those 11 models were used to generate binary masks using an average voting procedure for each CDnet-2014 video frame. Usually, FgSegNet_v2 would require one trained network per video, which does not enable assessing the generalization capacity of FgSegNet_v2. Because of that, we used a voting-based approach to generate FgSegNet_v2 detection masks. Using the 11 FgSegNet_v2 masks, one from each CDnet-2014 category model, we used pixel-based averaging for each frame and applied a simple (majority) voting rule; if the pixel average is greater than 50% (e.g. sum greater than 5.5) then the pixel voting mask result will be true. We refer the result of this experiment as *M_FgSegNet_v2(50%)*. It can be seen from Table 5 that when the ensemble approach is used with 11 models instead of 53 models, the performance of the top-performing method decreases dramatically and has an F-Measure of 34% that is 63% lower than our proposed method DeepFTSG.

In terms of the original training regimen, FgSegNet_v2 trains 53 distinctively parameterized networks for inference while the recently proposed method by Kim and Ha (2020),

[1] http://jacarini.dinf.usherbrooke.ca/results2014/1192/

[2] http://jacarini.dinf.usherbrooke.ca/results2014/1193/

Table 6 Training duration and parameter sizes for FgSegNet_v2 and DeepFTSG using common hardware

Methods	# of models	GPU	Training time
FgSegNet_v2	53	GTX 1080 Ti	29 days
M_FgSegNet_v2*	1	GTX 1080 Ti	27 days
DeepFTSG-1	1	GTX 1080 Ti	10 h
DeepFTSG-2	1	GTX 1080 Ti	15 h

Methods	Network size (# parameters)
FgSegNet_v2	489 M (53 * 9,225,161)
M_FgSegNet_v2*	9 M (1 * 9,225,161)
DeepFTSG-1	35 M (1 * 35,037,041)
DeepFTSG-2	48 M (1 * 48,185,777)

M_FgSegNet_v2* with * indicating a single network model trained using the same configuration used to train the DeepFTSG network

which we refer to as KimHa, trains a single network using 61,593 images almost six times the number of images from CDnet-2014 used by FgSegNet and DeepFTSG, that also excludes four categories including camera jitter, PTZ, thermal and turbulence. For a fair comparison with FgSegNet_v2 and KimHa, we retrain both methods using the same configuration that we used to train DeepFTSG models, that is, selecting 200 frames from each video sequence (200 x 53 = 10,600 frames) within the labeled frames of the original CDnet-2014 dataset without excluding any categories. We refer to a trained *single model* corresponding version using all of the training video frames as *M_FgSegNet_v2* and *M_KimHa*, respectively. It can be observed from Table 5 that when a single model is trained for FgSegNet_v2, instead of separate 53 models, the performance of the top-performing method decreases significantly and has an F-Measure of 90.78% that is more than 6% lower than our proposed method DeepFTSG. Compared to *M_FgSegNet_v2*, DeepFTSG requires less than 2.4% of the training time. Fig. 7 demonstrates a qualitative comparison of methods on sample frames from various challenging categories of a CDnet-2014 dataset. Conventional background subtraction techniques such as SuBSENSE produce either ghost artifacts or fragmented foregrounds in the region of interest, as shown in Fig. 7.

4.5 Evaluation of Generalization Power

To assess the generalization or transfer learning capabilities of our proposed DeepFTSG and the contribution of the motion and change cues on unseen videos, we evaluated DeepFTSG-1 and DeepFTSG-2 trained only on CDnet-2014 and tested on the unseen SBI-2015 and LASIESTA videos. The DeepFTSG weights were frozen without additional training on any portion of the SBI-2015 or LASIESTA dataset.

We compare the classical algorithms and recently proposed methods with our proposed DeepFTSG networks on unseen videos from SBI-2015 as shown in Table 7 and Table 8. The results of the other methods are taken from the following papers (Mandal et al., 2021; Kim & Ha, 2020).

It can be observed from Table 7 that our proposed method provides better performance for the selected four video sequences (Candela, CAVIAR2, CaVignal, HighwayII) than two classical methods, existing state-of-the-art methods, and recently proposed methods even without retraining using new scenes in the SBI-2015 dataset. More specifically, our proposed method achieves an overall 44% performance improvement over the FgSegNet_v2 and 24% performance improvement over the recently proposed method 3DCD. Moreover, from Table 8, we can see that our method provides better performance than two classical methods and have a competitive result, less than 0.8% compared to the recently proposed method (Kim & Ha, 2020). However, when comparing with the recent proposed method (Kim & Ha, 2020), we need to consider also the fact that KimHa uses almost 61,593 images from CDnet-2014 for training the network model excluding categories such as camera jitter, PTZ, thermal and turbulence, but we use only 10,600 frames from CDnet-2014 for training the network model (stated in Sect. 4.2) without excluding any categories. To make a fair comparison with KimHa, we used a trained model referring as *M_KimHa* (stated in Sect. 4.4), that was trained on CDnet-2014 dataset and evaluated it on SBI-2015 dataset without any retraining. It can observed from Table 8 that the performance of the KimHa method drops around 18% (from KimHa (88%) to *M_KimHa* (70%)) when the same video sequences from CDnet-2014 dataset is used to train the method as DeepFTSG.

To compare the performance of our proposed networks to the top performing network FgSegNet_v2 on unseen data, we trained FgSegNet_v2 network and referred it as *M_FgSegNet_v2* (described in Sect. 4.2), that was trained on CDnet-2014 dataset using a *single model*, and evaluate it on SBI-2014 dataset without any retraining. The Table 8 shows that the top performing method *M_FgSegNet_v2* has the worst generalization performance with an F-measure

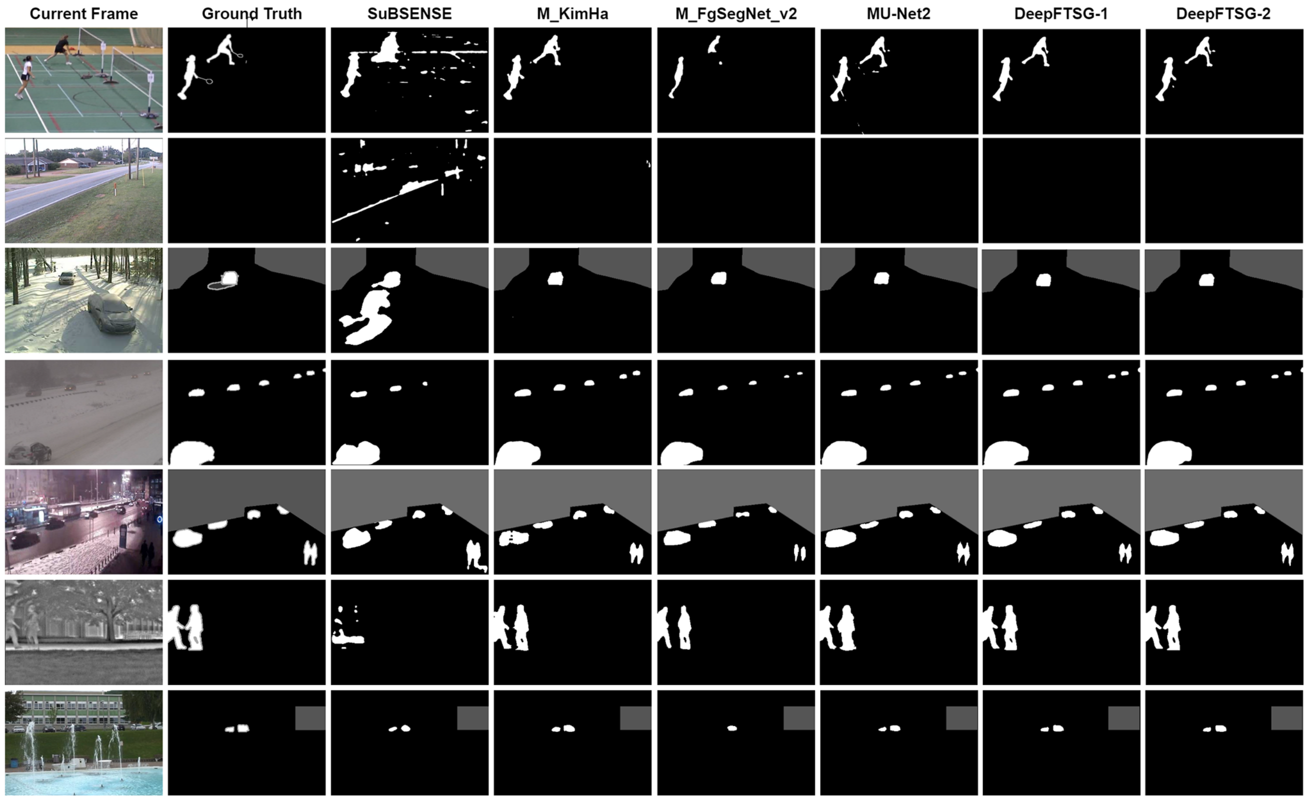


Fig. 7 Qualitative comparison of the proposed methods with top performing algorithms on sample frames from different categories of CDnet-2014. Column left to right: input images, ground truth, SuBSENSE, M_KimHa, M_FgSegNet_v2, MU-Net2, DeepFTSG-1, and DeepFTSG-2. Row top to bottom: cameraJitter/badminton, PTZ/continuousPan, intermittentObjectMotion/winterDriveway, badWeather/blizzard, nightVideos/tramStation, thermal/park, dynamicBackground/fountain01. The dark-gray areas represent pixels outside of CDnet-2014 regions of interest

Table 7 Comparative F-Score performance on a subset of the SBI-2015 dataset, including PAWCS (St-Charles et al., 2016), SuBSENSE (St-Charles et al., 2015), FgSegNet-S (Lim & Keles, 2018), FgSegnet-M (Lim & Keles, 2018), FgSegnet_v2 (Lim & Keles, 2020), 3DCD (Mandal et al., 2021), KimHa (Kim & Ha, 2020), FTSG (Wang et al., 2014a)

Method	Cand	CAV2	CaV	HigII	Avg
PAWCS	0.87	0.68	0.37	0.90	0.71
SuBSENSE	0.54	**0.87**	0.40	0.89	0.67
FgSegNet-S	0.23	0.11	0.68	0.24	0.32
FgSegNet-M	0.15	0.14	0.72	0.21	0.31
FgSegNet_v2	0.27	0.10	0.63	0.58	0.40
3DCD	0.67	0.62	0.53	0.59	0.60
KimHa	0.74	0.93	0.45	**0.96**	0.77
FTSG	**0.88**	0.58	0.69	0.77	0.73
DeepFTSG-1	0.67	0.76	0.96	0.91	0.82
DeepFTSG-2	0.72	0.77	**0.98**	0.91	**0.84**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

of 54% on an unseen data SBI-2015 when it has a *single model* or an F-Measure of 35% when it has 11 models, and majority voting is used. These results in Table 8 for both *M_FgSegNet_v2* and DeepFTSG support the observation that transfer learning to new unseen videos is more effective with better generalization capacity when a single network is used and with DeepFTSG when multi-cue video streams capturing motion, change and object appearance are fused using a unified network architecture. Fig. 8 demonstrates a qualitative comparison of methods on sample frames from various categories of the SBI-2015 dataset.

To further compare the generalization power of the proposed method, we evaluate it on the unseen LASIESTA dataset. Table 9 (per-video) and Table 10 (per-category) show a comparison of DeepFTSG with an unseen video performance of the algorithms reported in Mandal et al. (2021); Tezcan et al. (2021).

It can be observed from both Table 9 and Table 10 that DeepFTSG achieves significantly better results than state-of-the-art on an unseen video from LASIESTA. To make a fair comparison with KimHa and FgSegNet_v2, we used trained models referring as *M_KimHa* and *M_FgSegNet_v2* (stated in Sect. 4.4), that were trained on CDnet-2014 dataset, and evaluate them on LASIESTA dataset without any retraining. It can be observed from Table 9 that the performance

Table 8 Comparative F-Score performance on the full SBI-2015 dataset, including methods PAWCS (St-Charles et al., 2016), SuBSENSE (St-Charles et al., 2015), KimHa (Kim & Ha, 2020), FTSG (Wang et al., 2014a); CAV1: CAVIAR1, CAV2: CAVIAR2, CaV: Cavignal, Cand: Candela, HAM: HallAndMonitor, HigI: HighwayI, HigII: HighwayII, HB2: HumanBody2, IBt2: IBMtest2

Methods	Board	CAV1	CAV2	CaV	Cand	HAM	HigI	HigII	HB2	IBt2	Avg
SuBSENSE	0.578	0.914	0.871	0.398	0.536	0.776	0.552	0.894	0.835	0.939	0.729
PAWCS	0.780	0.859	0.677	0.370	0.873	0.741	0.702	0.903	0.701	0.939	0.754
KimHa	**0.929**	**0.966**	**0.927**	0.449	0.737	**0.974**	**0.927**	**0.963**	0.940	**0.979**	**0.879**
M_KimHa*	0.705	0.827	0.566	0.580	0.459	0.766	0.707	0.783	0.726	0.864	0.698
M_FgSegNet_v2*	0.477	0.846	0.121	0.895	0.386	0.762	0.531	0.144	0.646	0.611	0.542
M_FgSegNet_v2(50%)*	0.148	0.680	0.116	0.429	0.248	0.156	0.512	0.517	0.221	0.493	0.352
FTSG	0.796	0.652	0.577	0.687	**0.881**	0.727	0.438	0.769	0.782	0.841	0.715
DeepFTSG-1	0.760	0.916	0.759	0.956	0.674	0.909	0.833	0.909	**0.942**	0.947	0.861
DeepFTSG-2	0.814	0.923	0.766	**0.975**	0.720	0.877	0.892	0.905	0.902	0.940	**0.872**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

M_FgSegNet_v2* and M_KimHa* with * indicating single network models trained using the same configuration used to train the DeepFTSG network. M_FgSegNet_v2(50%)* with * has 11 models, one for each CDnet-2014 category, and the majority voting rule is used to get a final mask

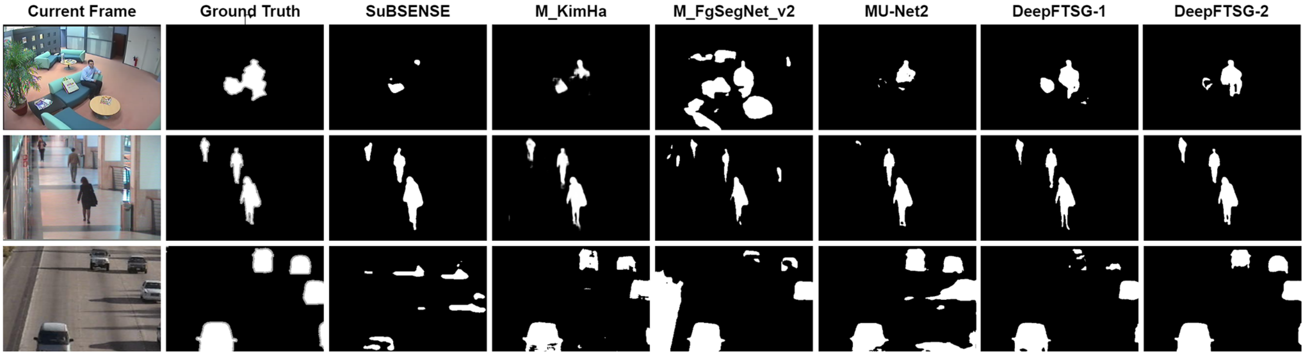


Fig. 8 Qualitative comparison of the proposed methods with top performing algorithms on sample frames from different categories of SBI-2015. Column left to right: input images, ground truth, SuBSENSE, M_KimHa, M_FgSegNet_v2, MU-Net2, DeepFTSG-1, and DeepFTSG-2. Row top to bottom: Candela_m1.10, CAVIAR1, HighwayI

of the KimHa method is comparable when the same video sequences from CDnet-2014 dataset is used to train the method as DeepFTSG, but DeepFTSG-2 still outperforms it by 5%. The top-performing method *M_FgSegNet_v2* has the worst generalization performance with an F-measure of 51% on an unseen data LASIESTA. Moreover, DeepFTSG-2 outperforms the BSUV-Net2.0 by 7%. In Table 9 the BSUV-Net2.0*, DeepFTSG-1*, and DeepFTSG-2* shows the result of using the scene dependent assessment strategy that is explained in Sect. 4.6. From Table 10 it can be seen that DeepFTSG-2 outperforms the BSUV-Net2.0 by 3% and *M_KimHa* by 9%. Many recent proposed methods ignored the simulated motion and moving camera for both indoor and outdoor sequences except BSUV-Net2.0. Since we run on every video of the LASIESTA dataset, we compare our performance on those videos with BSUV-Net2.0. The videos under moving camera and simulated motion categories in LASIESTA dataset are divided into four groups by the BSUV-Net2.0 authors (Tezcan et al., 2021), and the performance was evaluated with three different versions of BSUV-Net2.0, that is shown in Table 11. We also used trained models referred to as *M_KimHa* and *M_FgSegNet_v2* (stated in Sect. 4.4), which were trained on the CDnet-2014 dataset, and evaluated on LASIESTA dataset without any retraining. It can be seen from Table 11 that DeepFTSG outperforms all other methods in three video categories (Indoor pan & tilt, Indoor jitter, Outdoor jitter) and has a comparable result in the Outdoor pan & tilt category. DeepFTSG has low results in Outdoor pan & tilt category because in the Outdoor Moving Camera (OMC) sequences of the LASIESTA dataset, the camera is moving continually, and spatiotemporal information will not provide any reasonable information. The results from the generalization experiments show that the proposed appearance-based and spatiotemporal features fusing network DeepFTSG either using early or middle fusion is not specific to CDnet-2014 dataset, which it was trained on, and can be very effective on other datasets as well. Figure 9

Table 9 Comparative per-video F-Score performance on LASIESTA dataset, including methods Maddalena1 (Maddalena & Petrosino, 2008), Maddalena2 (Maddalena & Petrosino, 2012), Haines (Haines & Xiang, 2014), Cuevas (Daniel et al., 2018), FgSegNet-M (Lim & Keles, 2018), FgSegNet_v2 (Lim & Keles, 2020), 3DCD (Mandal et al., 2021), BSUV-Net2.0 (Tezcan et al., 2021)), SuBSENSE (St-Charles et al., 2015), FTSG (Wang et al., 2014a)

Method	ISI-2	ICA-2	IOC-2	IIL-2	IMB-2	IBS-2	OCL-2	ORA-2	OSN-2	OSU-2	Overall
Maddalena1	0.85	0.74	0.85	0.38	0.68	0.45	0.85	0.86	0.46	0.86	0.70
Maddalena2	**0.94**	**0.87**	**0.95**	0.23	0.85	0.40	0.88	0.86	0.71	0.88	0.76
Haines	0.81	**0.87**	**0.95**	0.81	0.71	0.73	**0.96**	0.90	0.04	0.90	0.77
Cuevas	0.76	0.63	0.88	0.79	0.68	0.66	0.90	0.87	0.09	0.81	0.71
FgSegNet-M	0.56	0.55	0.65	0.42	0.56	0.19	0.28	0.18	0.01	0.33	0.37
FgSegNet_v2	0.53	0.58	0.25	0.41	0.63	0.25	0.54	0.54	0.05	0.29	0.41
3DCD	0.86	0.49	0.93	0.85	0.79	**0.87**	0.87	0.87	0.49	0.83	0.79
BSUV-Net2.0	0.89	0.60	**0.95**	**0.89**	0.76	0.69	0.89	0.93	0.70	0.91	0.82
SuBSENSE	0.81	0.76	0.80	0.63	0.81	0.61	0.75	0.72	0.55	0.68	0.71
M_KimHa*	0.90	**0.87**	0.92	0.86	0.90	0.69	0.91	0.84	0.58	0.91	0.84
M_FgSegNet_v2*	0.70	0.72	0.69	0.37	0.70	0.18	0.74	0.91	0.02	0.05	0.51
FTSG	0.86	0.79	0.89	0.70	0.83	0.63	0.84	0.87	0.48	0.79	0.77
DeepFTSG-1	**0.94**	0.78	0.85	0.84	**0.91**	0.76	0.95	0.95	0.79	**0.95**	0.87
DeepFTSG-2	**0.94**	0.84	0.91	0.84	0.85	**0.87**	**0.96**	**0.96**	**0.81**	0.93	**0.89**
BSUV-Net2.0*	0.98	**0.99**	0.97	0.97	0.88	0.95	0.97	0.97	0.55	0.95	0.92
DeepFTSG-1*	0.98	**0.99**	**0.99**	**0.99**	**0.99**	**0.97**	**0.98**	**0.99**	0.97	0.97	0.98
DeepFTSG-2*	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.97**	**0.98**	**0.99**	**0.98**	**0.98**	**0.99**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

M_FgSegNet_v2* and M_KimHa* with * indicating single network models trained using the same configuration used to train the DeepFTSG network. BSUV-Net2.0*, DeepFTSG-1*, and DeepFTSG-2* with * indicating scene dependent assessment strategy that is explained in Sect. 4.6

demonstrates a qualitative comparison of methods on sample frames from various categories of the LASIESTA dataset.

Our next generalization experiment of the proposed methods involves using a single object tracking dataset named Large-scale Single Object Tracking (LaSOT) (Fan et al., 2019). To make a fair comparison of our proposed and recent methods along with FgSegNet_v2 on the LaSOT dataset, we used models trained on the CDnet-2014 dataset and ran inference on 5 categories of the LaSOT dataset. The categories are bicycle, car, dog, giraffe, and person. We select only two video sequences from each category. Those categories are selected on purpose to see how the methods would perform on the ones that they are familiar with, such as car, person, and somewhat bicycle, and the ones they are not familiar with at all, such as dog and giraffe. To evaluate the outputs of the proposed methods, since the LaSOT dataset ground truth is a bounding box and the output of the method is a segmentation mask, we convert the segmentation masks to bounding boxes and save them as binary image for both cases LaSOT ground truth and methods output, where foreground is a bounding box region and everything else is background (see Fig. 10). We use the same evaluation metrics stated in Sect. 4.3, especially recall, precision, and F-Measure. It can be seen from Table 12 that DeepFTSG-2 outperforms all other methods with an F-Measure of 55%. We can also observe that on new categories, dog and giraffe, DeepFTSG outperforms all other methods as well, having 74% F-Measure for a dog and 52% F-Measure for a giraffe. Since all of those video sequences in the LaSOT dataset involves a moving camera, many methods using background subtraction techniques such as SuBSENSE and *M_KimHa* fails to detect object properly. Hence *M_FgSegNet_v2* performs better than those methods having F-Measure of 53.6%. This evaluation method is inaccurate since we first convert the segmentation mask to a bounding box (making many false positive pixels). However, by using this strategy, we can test the proposed methods on different dataset types to see the generalization capabilities.

The proposed DeepFTSG network uses a generalized multi-stream architecture that can be readily extended to support additional multimodal stream cues with varying fusion stages. To demonstrate it, we ran an additional experiment where we extended DeepFTSG-2 with an additional streaming cue having infrared information and named it DeepFTSG-3. Instead of two streams, DeepFTSG-3 has three streams, where the first stream input is an RGB frame (VIS), the second stream is infrared (IR) information of that frame, where we used SE-ResNet-50 backbone, and the third stream is a combination of BGS and flux for both RGB and

Table 10 Comparative per-category F-Score performance on LASIESTA dataset with stationary camera video subset, including methods Maddalena1 (Maddalena & Petrosino, 2008), Maddalena2 (Maddalena & Petrosino, 2012), Haines (Haines & Xiang, 2014), Cuevas (Daniel et al., 2018), BSUV-Net2.0 (Tezcan et al., 2021), SuBSENSE (St-Charles et al., 2015), FTSG (Wang et al., 2014a)

Method	ISI	ICA	IOC	IIL	IMB	IBS	OCL	ORA	OSN	OSU	Overall
Maddalena1	0.87	0.85	0.91	0.61	0.76	0.42	0.88	0.84	0.58	0.80	0.75
Maddalena2	**0.95**	0.86	0.95	0.21	0.91	0.40	**0.97**	0.90	0.81	0.88	0.78
Haines	0.89	**0.89**	0.92	0.85	0.84	0.68	0.83	0.89	0.17	0.86	0.78
Cuevas	0.88	0.84	0.78	0.65	**0.93**	0.66	0.93	0.87	0.78	0.72	0.80
BSUV-Net2.0	0.92	0.68	**0.96**	**0.88**	0.81	0.77	0.93	0.94	0.84	0.79	0.85
SuBSENSE	0.80	0.83	0.79	0.43	0.86	0.58	0.83	0.78	0.71	0.69	0.73
M_KimHa*	0.91	**0.89**	0.93	0.67	0.90	0.69	0.80	0.68	0.73	0.72	0.79
M_FgSegNet_v2*	0.80	0.77	0.41	0.64	0.74	0.51	0.46	0.49	0.07	0.11	0.50
FTSG	0.87	0.78	0.85	0.70	0.89	0.64	0.88	0.87	0.67	0.81	0.80
DeepFTSG-1	0.92	0.82	0.90	0.82	0.91	0.84	0.85	**0.95**	**0.88**	**0.93**	**0.88**
DeepFTSG-2	**0.95**	**0.89**	0.93	**0.88**	0.89	**0.90**	0.81	0.81	0.86	0.92	**0.88**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

M_FgSegNet_v2* and M_KimHa* with * indicating single network models trained using the same configuration used to train the DeepFTSG network

Table 11 F-Score comparison on LASIESTA dataset subset with moving camera and simulated motion sequences combined

Categories	BSUV-Net2.0-v1	BSUV-Net2.0-v2	BSUV-Net2.0-v3	SuBSENSE	M_KimHa	M_FgSegNet_v2	DeepFTSG-1	DeepFTSG-2
Indoor pan & tilt	0.48	0.52	0.58	0.42	0.36	0.66	0.70	**0.88**
Outdoor pan & tilt	0.56	0.42	0.58	0.20	0.21	**0.61**	0.56	0.56
Indoor jitter	0.81	0.88	0.84	0.71	0.70	0.72	0.87	**0.92**
Outdoor jitter	0.75	0.85	0.50	0.63	0.53	0.7	0.81	**0.89**

The definition for the significance of the bold in Table is that themethods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

Data augmentation as used by BSUVNet includes a combination of SAC: spatially-aligned crop, RSC: randomly-shifted crop, and PTZ: PTZ camera crop; BSUV-Net2.0-v1: SAC, BSUV-Net2.0-v2: SAC+RSC, BSUV-Net2.0-v3: SAC+PTZ. The other methods did not use any augmentation

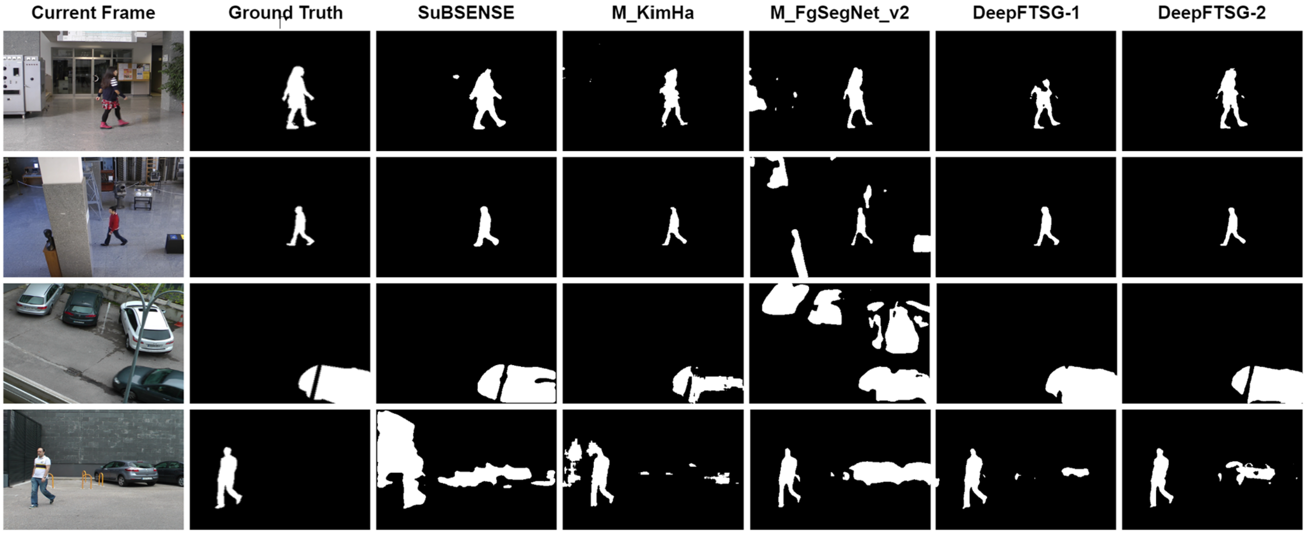


Fig. 9 Qualitative comparison of the proposed methods with top performing algorithms on sample frames from different categories of LASIESTA. Column left to right: input images, ground truth, SuBSENSE, M_KimHa, M_FgSegNet_v2, DeepFTSG-1, and DeepFTSG-2. Row top to bottom: I_MC_02, I_OC_1, O_CL_01, O_MC_01

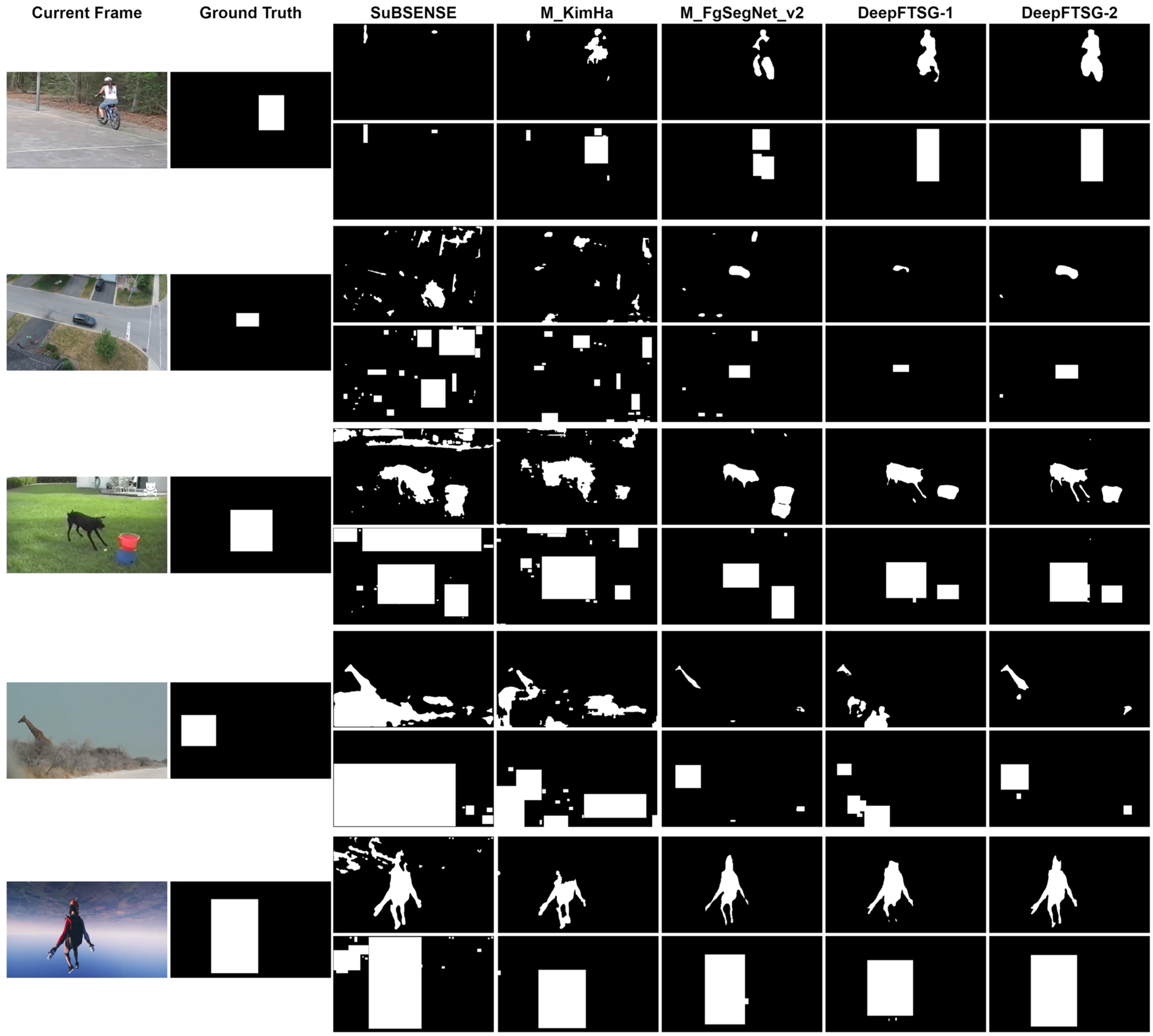


Fig. 10 Qualitative comparison on LaSOT (bounding box single object tracking as masks) of the proposed methods with top performing algorithms on sample frames from different video categories. Column left to right: input images, ground truth, SuBSENSE, M_KimHa, M_FgSegNet_v2, DeepFTSG-1, and DeepFTSG-2. Row top to bottom: bicycle/bicycle-12, car/car-11, dog/dog-8, giraffe/giraffe-16, person/person-10. The first row on each category represents the output mask from each method, and the second row is a bounding box obtained from the first-row mask

infrared cues, making the third stream having four channels as input (1st-channel: BGS of RGB frame, 2nd-channel: flux of RGB frame, 3rd-channel: BGS of the infrared frame, 4th-channel: flux of infrared frame). In infrared, the non-visible heat radiation emitted or reflected by all objects, regardless of lighting conditions, can be imaged. Hence, infrared information provides a superior advantage in challenging conditions, such as low light, night-time, shadows, visual obstructions, degraded visual environments, and camouflaging foliage. Figure 11, illustrates the generalization of the DeepFTSG network architecture to support scalable multi-stream learning and inference. The 2- and 3-stream architectures correspond to DeepFTSG-2 and DeepFTSG-3. The 4-stream network is a future extension to support optical flow or stereo-based depth information as additional streams.

For this experiment, we used the Grayscale-Thermal Foreground Detection (GTFD) dataset (Li et al., 2017) that includes 25 aligned grayscale-thermal video pairs with high diversity and has a segmentation mask as a ground truth. We used 21 video sequences to train and 2 to test our proposed networks DeepFTSG-2 and DeepFTSG-3 using the same training strategy explained in Sect. 4.2. The only dif-

Table 12 Detailed evaluation of DeepFTSG with other methods on LaSOT dataset using bounding boxes; Cat: Category, SuB: SuBSENSE, UNet: M_KimHa, FgSeg: M_FgSegNet_v2, DF-1: DeepFTSG-1, DF-2: DeepFTSG-2

Cat.	Recall (Re)						Precision (P)						F-Measure (F)					
	SuB	UNet	FgSeg	FTSG	DF-1	DF-2	SuB	UNet	FgSeg	FTSG	DF-1	DF-2	SuB	UNet	FgSeg	FTSG	**DF-1**	**DF-2**
bicycle	**0.719**	0.566	0.449	0.622	0.419	0.626	0.391	0.367	**0.582**	0.393	0.464	0.476	0.378	0.427	**0.507**	0.414	0.410	0.498
car	0.301	0.314	**0.828**	0.365	0.287	0.647	0.230	0.229	0.500	0.198	0.428	**0.534**	0.168	0.215	**0.589**	0.246	0.279	0.536
dog	0.696	0.834	0.708	0.674	0.779	**0.835**	0.361	0.507	**0.692**	0.400	0.716	0.668	0.408	0.619	0.700	0.484	**0.746**	0.742
giraffe	**0.572**	0.571	0.364	0.496	0.203	0.460	0.456	0.437	**0.644**	0.449	0.488	0.624	0.506	0.494	0.464	0.454	0.280	**0.523**
person	0.584	0.473	0.436	**0.867**	0.389	0.570	0.263	0.213	**0.485**	0.228	0.413	0.435	0.332	0.273	0.421	0.301	0.391	**0.450**
Overall	0.575	0.552	0.557	0.605	0.415	**0.628**	0.340	0.351	0.580	0.334	0.502	**0.548**	0.358	0.406	0.536	0.380	0.421	**0.550**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

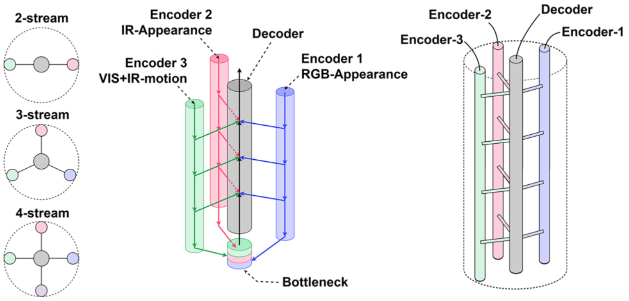


Fig. 11 Schematic view of how the DeepFTSG *USE-Net trellis* network architecture can be extended to support multi-stream architectures in a scalable way. The 2-stream network architecture is DeepFTSG-2 with RGB appearance and 2-channel motion streams. The middle diagram shows a sample 3-stream network DeepFTSG-3 incorporating RGB-Appearance (blue color), Infrared-Appearance (red color), and RGB+IR 4-channel motion (green color) encoder streams with a single decoder stream (gray color) for fusion (Color figure online)

Table 13 Comparison of FTSG, DeepFTSG-2 and DeepFTSG-3 on four unseen 4-channel (RGB+IR) video sequences from GTFD (2) and FPSS (2) dataset

Methods	GTFD dataset Overall			
	Re	PWC	*P*	F
FTSG	0.5344	0.4321	0.3950	0.4513
DeepFTSG-2	0.6799	0.2743	0.5859	0.6241
DeepFTSG-3	0.7512	0.1543	0.7884	0.7691
Methods	FPSS dataset Overall			
	Re	PWC	*P*	F
FTSG	0.2550	0.6754	0.7054	0.3740
DeepFTSG-2	0.4374	0.6205	0.6670	0.5176
DeepFTSG-3	0.4415	0.5619	0.7688	0.5424

ference is that we used 852 frames for training and 46 frames for validation. For DeepFTSG-2, we used appearance and motion cues as input to the streams, and for DeepFTSG-3, we used appearance, infrared, and motion cues as input to the streams. We used two unseen video sequences (movingClouds, pedestrian7) from the GTFD dataset that was not included in the training to evaluate the models; note that GTFD uses a subset of 24 and 30 frames, respectively, in both of these sequences that are from the OSU dataset. The same evaluation metrics described in Sect. 4.3 were used.

Table 13 shows the result of DeepFTSG-2 and DeepFTSG-3 on two unseen video sequences of the GTFD dataset. It can be observed that by adding infrared information along with motion cues from infrared, our accuracy improved from 62.4% (DeepFTSG-2) to 76.9% (DeepFTSG-3), which is around 14.5% improvement in F-Measure. Figure 12, shows the qualitative result of DeepFTSG-2 and DeepFTSG-3 on unseen video sequences of the GTFD dataset; note that in the Ground Truth segmentation image only those people moving within the short video segment are marked. In Fig. 13 the spatio-temporal volumes of a video sequence of GTFD (OSU video) dataset are demonstrated to visualize the motion and change through time in both VIS and IR.

To further test the generalization ability of DeepFTSG-3, we run another experiment using Force Protection Surveillance System (FPSS) dataset (Chan, 2009). The FPSS dataset consists of 53 pairs of color and FLIR video sequences collected at the Adelphi Laboratory Center (ALC) of ARL between Nov 2004 and Jan 2005. All video sequences consist of 640 × 480 pixel images collected using a thermal vision sentry personnel observation device (POD) manufactured by FLIR Systems. The primary moving objects selected were people and vehicles. Classes and centroids of the moving objects in the video sequences were provided. Using those centroids, we created bounding boxes for each moving object manually as ground truth for the first two sequences (rf20041120_161701fc, rf20041216_143701fc). To test the generalization, we used DeepFTSG-2 and DeepFTSG-3 trained on the GTFD dataset and tested it on these two sequences of the FPSS dataset that we created a bounding box manually without any additional training.

Table 13 shows the result of DeepFTSG-2 and DeepFTSG-3 on two new video sequences of the FPSS dataset. We can observe that by adding infrared information along with motion cues from infrared, our accuracy improved from 51.8% (DeepFTSG-2) to 54.2% (DeepFTSG-3), which is around 2.4% improvement in F-Measure. Figure 14, shows the qualitative result of DeepFTSG-2 and DeepFTSG-3 on new video sequences of the FPSS dataset. This experiment demonstrates the extension ability of the proposed network DeepFTSG to support additional streaming cues.

4.6 Scene Dependent Assessment

Recently proposed methods also evaluate their methods using scene dependent assessment (SDA) strategy, where some frames from the test videos are also used for training (fine tuning or transfer learning) the deep network. That is consistent with current approaches that train one network per video, one per class of similar videos, or one per video collection. However, such networks need to be trained for each real-world scenario and are often brittle to environmental changes. Moreover, the SDA approach is also not ideal for evaluating the generalization capacity of deep learning mod-

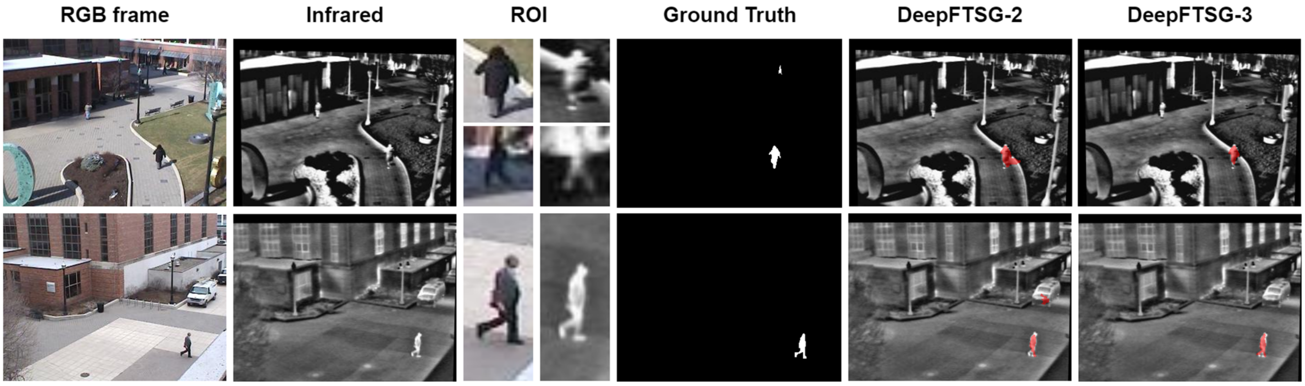
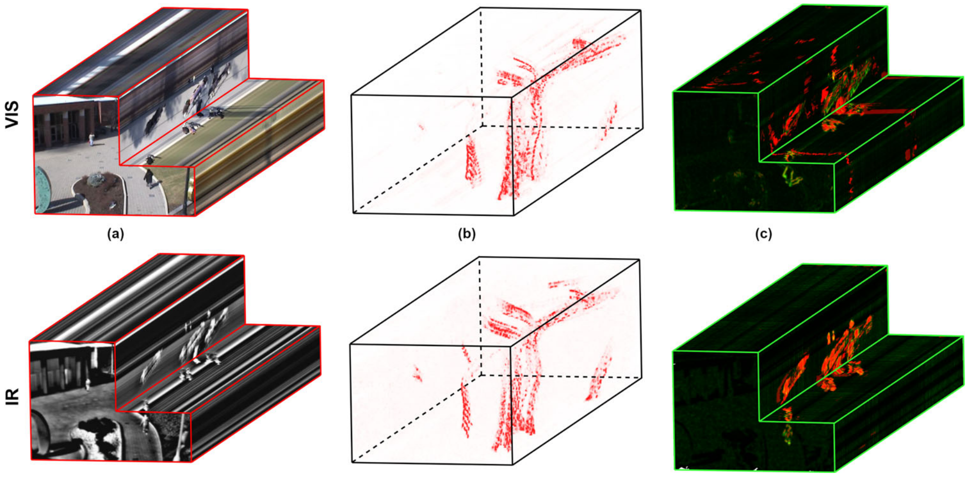


Fig. 12 Qualitative result of DeepFTSG-2 and DeepFTSG-3 on two unseen video sequences of GTFD (OSU videos) dataset

Fig. 13 Spatio-temporal ($x - y - t$) volumes of a video sequence of GTFD (OSU video) dataset for VIS (Row 1) and IR (Row 2) channels. **a** original input, **b** flux motion, **c** combination of change and motion (green ← flux, red ← change) (Color figure online)



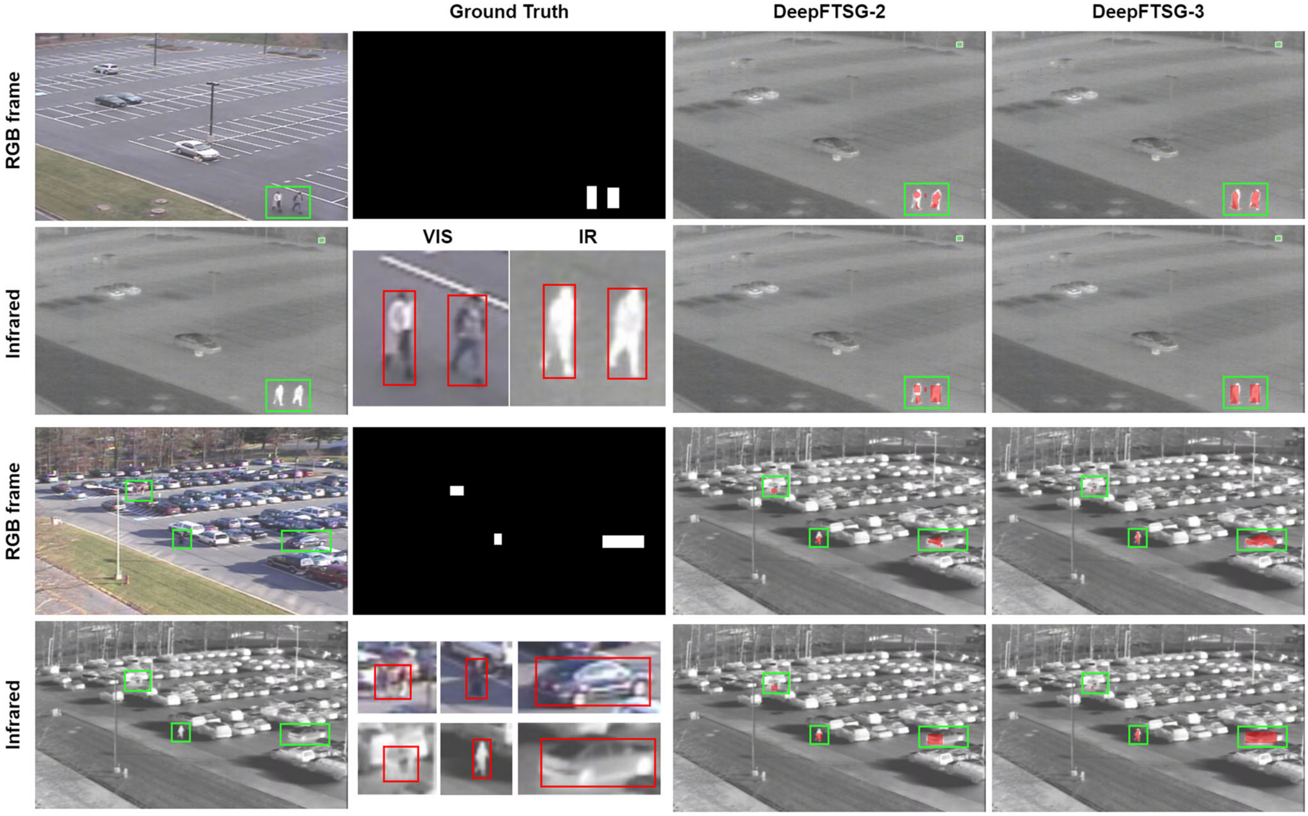


Fig. 14 Qualitative result of DeepFTSG-2 and DeepFTSG-3 on two new video sequences of FPSS dataset

Table 14 Comparative F-Score performance using scene dependent assessment (SDA) training strategy on SBI-2015 dataset, including methods FgSegNet-S (Lim & Keles, 2018), FgSegNet-M (Lim & Keles, 2018), FgSegNet_v2 (Lim & Keles, 2020), 3DCD (Mandal et al., 2021); Cand: Candela, CAV1: CAVIAR1, CAV2: CAVIAR2, CaV: CaVignal, Fol: Foilage, HAM: HallAndMonitor, HigI: HighwayI, HigII: HighwayII, HB2: HumanBody2, IBt2: IBMtest2, PAF: PeopleAndFoilage, Snel: Snellen

Methods	Board	Cand	CAV1	CAV2	CaV	Fol	HAM	HigI	HigII	HB2	IBt2	PAF	Snel	Avg
FgSegNet-S	0.88	0.25	0.67	0.04	0.52	0.68	0.62	0.83	0.42	0.78	0.72	0.88	0.22	0.58
FgSegNet-M	0.89	0.27	0.74	0.19	0.61	0.60	0.67	0.73	0.36	0.79	0.78	0.87	0.42	0.61
FgSegNet_v2	0.89	0.25	0.55	0.10	0.65	0.86	0.46	0.82	0.59	0.63	0.57	0.88	0.68	0.61
3DCD	0.85	0.31	0.81	0.58	0.55	0.66	0.63	0.73	0.79	0.67	0.74	0.80	0.74	0.68
DeepFTSG-1	**0.99**	0.92	0.98	**0.94**	**0.99**	**0.99**	**0.97**	0.98	0.98	**0.98**	**0.98**	**0.99**	0.85	**0.97**
DeepFTSG-2	**0.99**	**0.98**	**0.99**	**0.94**	**0.99**	**0.99**	**0.97**	**0.99**	**0.99**	**0.98**	**0.98**	**0.99**	**0.86**	**0.97**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

els to detect and segment motion across different temporal scales. However, to make a fair comparative analysis of our proposed networks with existing deep learning methods, we also did experiments following the SDA approach. Using the SDA strategy, our proposed methods are trained similarly to the 3DCD method (Mandal et al., 2021), such that for both SBI-2015 and LASIESTA datasets, the training is performed on 50% of the frames (for SBI-2015 50% of frames = 2380 frames, for LASIESTA 50% of frames = 4010 frames) and evaluation is performed using the complete 100% of frames.

The accuracy on the SBI-2015 dataset using the SDA training strategy is shown in Table 14. The overall F-Measure of the proposed methods are 97% that is almost 30% higher than the recently proposed 3DCD method (68%) and 36% higher than the final version of FgSegNet (61%).

The comparison of the proposed methods with the existing methods in terms of an average F-Measure in each video category of the LASIESTA, except the simulated motion and moving camera sequences, are demonstrated in the Table 15. From quantitative analysis on Table 15, it can be seen that the proposed DeepFTSG outperforms in all ten categories

Table 15 Comparative F-Score performance of SDA strategy on LASIESTA dataset, including methods FgSegNet-S (Lim & Keles, 2018), FgSegNet-M (Lim & Keles, 2018), FgSegNet_v2 (Lim & Keles, 2020), 3DCD (Mandal et al., 2021)

Method	ISI	ICA	IOC	IIL	IMB	IBS	OCL	ORA	OSN	OSU	Overall
FgSegNet-S	0.32	0.57	0.37	0.33	0.64	0.21	0.17	0.10	0.08	0.27	0.31
FgSegNet-M	0.44	0.71	0.29	0.32	0.68	0.27	0.24	0.17	0.18	0.21	0.35
FgSegNet_v2	0.44	0.60	0.30	0.32	0.50	0.22	0.31	0.24	0.28	0.38	0.36
3DCD	0.91	0.76	0.90	0.90	0.90	0.81	0.89	0.89	0.72	0.85	0.85
DeepFTSG-1	**0.99**	**0.99**	0.98	**0.99**	**0.99**	0.97	0.98	**0.99**	0.98	0.97	0.98
DeepFTSG-2	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.98**	**0.99**	**0.99**	**0.99**	**0.98**	**0.99**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

Table 16 Ablation study using different inputs and losses for DeepFTSG-1 on different datasets

Exp	Input				Ignore Mask		CDnet-2014			SBI-2015			LASIESTA		
	RGB	Gray	Flux	BGS	No	Yes	Re	P	F	Re	P	F	Re	P	F
Exp 1	✓				✓		0.9460	0.9762	0.9593	0.8900	0.5862	0.6763	**0.9376**	0.4556	0.5398
Exp 2		✓		✓	✓		0.9380	0.9786	0.9550	0.8442	0.8848	0.8594	0.8538	0.8077	0.8135
Exp 3		✓	✓		✓		0.9378	0.9767	0.9554	0.8573	0.8458	0.8415	0.7214	0.7214	0.7204
Exp 4		✓	✓	✓	✓		0.9413	**0.9799**	0.9578	**0.8389**	**0.9070**	**0.8632**	0.8542	0.8231	0.8261
Exp 5		✓	✓	✓		✓	**0.9566**	0.9776	**0.9652**	0.8378	0.9021	0.8605	0.8853	**0.8334**	**0.8465**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

Table 17 Ablation study using different inputs, losses and backbones for DeepFTSG-2 on different datasets

Exp	Input			2nd Stream B-bone		Ignore Mask		CDnet-2014			SBI-2015			LASIESTA		
	RGB	Flux	BGS	SE-RN-50	RN-18	No	Yes	Re	P	F	Re	P	F	Re	P	F
Exp 1	✓	✓	✓	✓		✓		0.8938	**0.9887**	0.9345	0.7688	**0.9271**	0.8270	0.8665	0.8156	0.8137
Exp 2	✓	✓	✓		✓	✓		0.9472	0.9777	0.9608	0.7784	0.9099	0.8132	0.8608	0.8255	0.8169
Exp 3	✓	✓	✓	✓			✓	0.9299	0.984	0.9514	0.8181	0.8931	0.8394	**0.9205**	0.8198	0.8503
Exp 4	✓		✓		✓		✓	0.9619	0.9698	0.9642	0.8578	0.7451	0.7923	0.8988	0.4960	0.5755
Exp 5	✓	✓			✓		✓	0.9559	0.9812	0.9677	0.8316	0.8307	0.8164	0.9030	0.6706	0.7112
Exp 6	✓	✓	✓		✓		✓	**0.9663**	0.9756	**0.9700**	**0.8689**	0.8844	**0.8715**	0.9130	**0.8343**	**0.8510**

The definition for the significance of the bold in Table is that the methods proposed in this paper are highlighted in bold and the best result in each column is highlighted in bold

of the LASIESTA dataset. Moreover, the DeepFTSG performance 63% higher than FgSegNet_v2 (36%) and 14% higher than 3DCD (85%). From Table 9 it can be observed that DeepFTSG outperforms BSUV-Net2.0 (92%) by 7% in overall F-Measure. The results of the other methods are taken from 3DCD paper (Mandal et al., 2021), which has the same SDA strategy for training and evaluation as the proposed methods. Those experiments demonstrate that the proposed DeepFTSG networks are not specific to the CDnet-2014 dataset and can be very effective on other datasets.

4.7 Ablation Study

To understand the impact of fusing appearance-based features with spatiotemporal features of the proposed DeepFTSG for moving object detection, we did an ablation study by performing different experiments with different combinations of appearance-based features and spatiotemporal features. Table 16 demonstrates an ablation study performed using DeepFTSG-1, with different inputs and including ignore region in the loss function. For all experiments, there is not a significant improvement in the CDnet-2014 dataset, because all of those experiments include sequences from the CDnet-2014 dataset for training and validation of a network. To make a better assumption of how each piece of information improves the accuracy of the network, we need to look at the results of unseen datasets, SBI-2015 and LASIESTA. From *Experiment 1*, it can be observed that if we had only appearance-based information, we obtained an F-Score of 67.6% for SBI-2015 and 54% for LASIESTA, even though we had an F-Score of almost 96% for CDnet-2014. In *Experiments 2 and 3*, we fused spatiotemporal information

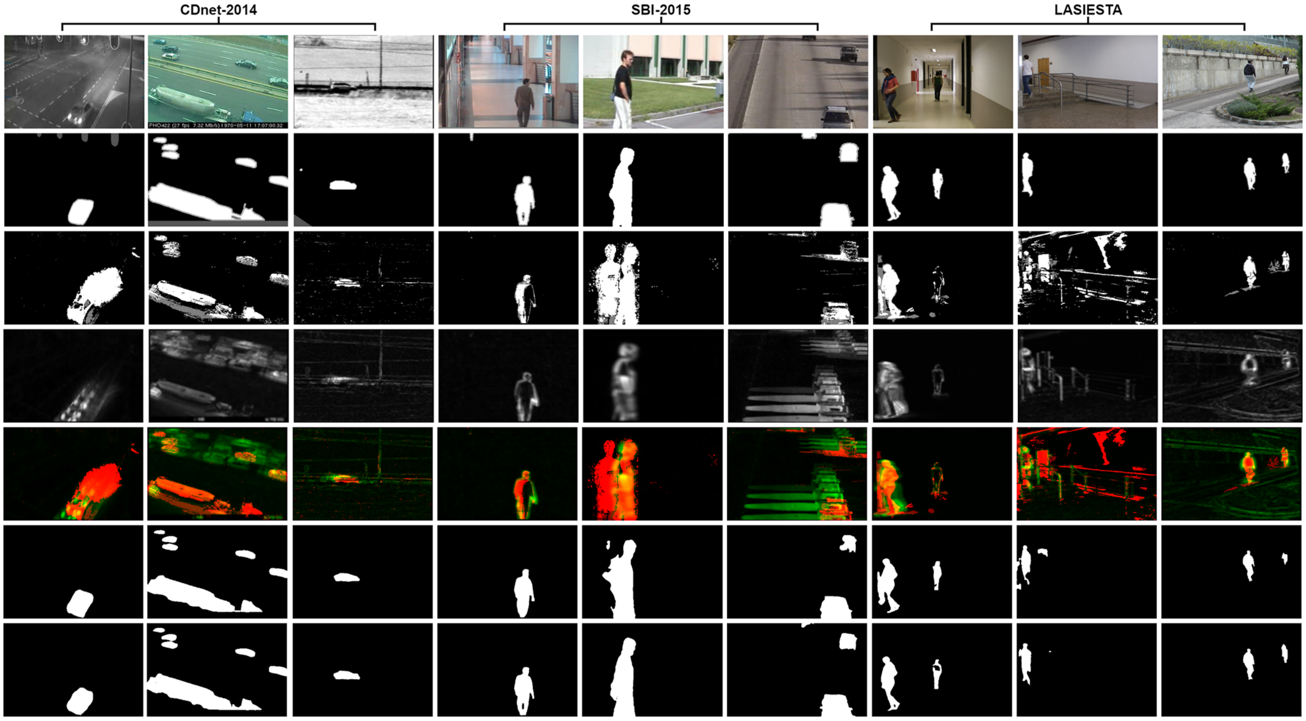


Fig. 15 Sample results and comparative analysis of detection performance. Rows top to bottom: input images, ground truth masks, change mask, flux motion, flux motion (green channel) and change (red channel), DeepFTSG-1, and DeepFTSG-2 masks. Columns left to right: first three columns from CDnet-2014 (nightVideos/streetCornerAtNight, lowFramerate/turnpike, turbulence/turbulence3); the next three columns from SBI-2015 (CAVIAR1, CaVignal, HighwayI); the last three from LASIESTA (IndoorBootstrap (I_BS_01), IndoorMovingCamera (I_MC_01), OutdoorSimulatedMotion (O_SM_04)) (Color figure online)

with appearance-based information by separating motion and change cues to observe how they will impact separately. In *Experiment 2*, only the change cue (BGS) was fused with appearance, and in *Experiment 3*, only the motion cue (Flux) was fused with appearance. We can observe that only fusing change cue increase the accuracy of the network for unseen SBI-2015 almost by 19% and 27% for LASIESTA. Fusing only motion cues also increases the network's accuracy but not as much as change cues. However, fusing motion and change cues with appearance-based information (*Experiment 4*) improves the accuracy of the network by almost 1%, with respect to *Experiment 2*. By not penalizing regions labeled as *ignore mask* in CDNet-2014 ground-truth, in the loss function, we slightly improve the accuracy on unseen video, which is demonstrated by *Experiment 5* in Table 16 (2% on LASIESTA)

The ablation study performed using DeepFTSG-2, with different inputs, backbone in the second stream, and including ignoring region in the loss function, is shown in Table 17. The difference between *Experiments 1 and 2* is the backbone used for the second stream of the network. Using ResNet-18 instead of SE-ResNet-50 for spatiotemporal feature extraction improved the network accuracy on CDnet-2014 by almost 2.5%, and there was no significant change in unseen videos. Conducting the same experiments, but this time not penalizing the *ignore mask* regions in CDNet-2014 in the loss function (*Experiment 2 and 6*), increased the accuracy of the network by nearly 1% and significantly more for unseen videos (5.8% and 3.4% for SBI-2015 and LASIESTA respectively). To understand how motion and change cues impact the network accuracy, we did two more experiments (*Experiment 4 and 5*), where we give only change cue (BGS) in the second stream (*Experiment 4*), and in *Experiment 5* only motion cue (Flux) is given. Even so, there was no significant change in the accuracy on CDnet-2014, we can observe that having motion cue increase the accuracy on unseen videos SBI-2015 by more than 2% and almost 14% for LASIESTA, with respect to having change cue (*Experiment 4*). Having both motion and change cues in the second stream (*Experiment 6*), we can observe that for the case of unseen videos, the accuracy improved by almost 5.5% in SBI-2015 and almost 14% in LASIESTA, with respect to having only motion cue (*Experiment 5*). The best accuracy for seen and unseen datasets was achieved in *Experiment 6*, having appearance-based information in the first stream and motion and change cues in the second stream, and using ResNet-18 backbone to extract spatiotemporal features, and not penalizing the ignore regions in the loss function during training and validation, has the accuracy of 97% in CDnet-2014, 87.2% in unseen SBI-2015 and 85.1% in unseen LASIESTA. From those experiments, we can observe that motion and change cues, when given along with appearance-based features, improve the performance of the moving object detection networks significantly on unseen videos, and the best results are achieved when they are fused instead of used separately. Qualitative results for one frame in 9 videos across three datasets are shown in Fig. 15, illustrating the improved accuracy of the proposed DeepFTSG networks. The video demo is available for the results of DeepFTSG-2[3] on CDnet-2014, SBI-2015 and LASIESTA datasets, where SBI-2015 and LASIESTA are completely unseen datasets that are used for generalization purposes.

5 Conclusions

We developed DeepFTSG, a deep convolutional neural network, to robustly detect moving objects in videos. DeepFTSG consists of a novel U-Net encoder-decoder structure that integrates object appearance cues with hand-crafted motion and change cues, using an early or middle fusion of single or multiple streams. Unsupervised tensor-based motion estimation and an unsupervised mixture of Gaussian background subtraction cues are used as part of the input stream to DeepFTSG, incorporating intrinsic temporal dynamics for accurate change detection. Decoupling pixel-level motion and change estimation from the network and assigning them to hand-crafted methods greatly reduces network complexity, training times, and most important amount of training data. DeepFTSG can learn object-level modeling, spatiotemporal fusion, and semantic change analysis using just 200 frames per video from the CDnet-2014 collection of 53 video sequences. The performance difference between DeepFTSG and FTSG helps to quantify the benefit of appearance information and visual cue fusion. DeepFTSG, which incorporates learned object appearance models along with a supervised fusion of visual motion cues, improves accuracy by about 25% over FTSG, which uses only motion cues without supervised learning. Using motion and change cues along with appearance generates accurate detection on unseen video sequences when fused altogether, instead of fusing just appearance with motion or change cues separately. The decoupled structure of DeepFTSG improves the adaptability of the proposed system to new domains using transfer learning. Compared to the top ranking FgSegNet_v2, and recently proposed methods, such as 3DCD, KimHa or BSUV-Net2.0, the DeepFTSG multi-cue multi-stream network, produces considerably more accurate detection performance on unseen video sequences. The proposed DeepFTSG network uses a generalized multi-stream architecture that can be readily extended to support additional streaming cues with varying fusion stages. Like ventral visual stream processing in the human visual system, DeepFTSG is a multi-stream multi-cue fusion framework that can be generalized to more than two streams; for example incorporating infrared cue, depth cue and optical flow streams.

Acknowledgements Research partially supported by U.S. Army Research Laboratory cooperative agreement W911NF1820285 and Army Research Office DURIP W911NF-1910181. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U. S. Government or agency.

Appendix A Ablation Experiments

More ablation experiments are demonstrated in this appendix to evaluate the contribution of each component in the proposed method (Table 18). Most of the experiment involves different input streams, either one, two, or three streams, to the proposed method. Various combinations of input information were tested as well. The best result on all three datasets was achieved when the inputs were Grayscale, BGS, and flux for DeepFTSG-1 and RGB, BGS and flux for DeepFTSG-2 (explained in the main section of the paper). However, instead of using BGS, if the SuBSENSE background model is used, there is a slight improvement in F-Measure for the CDnet-2014 dataset, but for the LASIESTA dataset, the F-Measure drops significantly (76.45%) for DeepFTSG-2. Moreover, instead of using Binary Cross Entropy loss, using Focal Loss did not significantly improve. Using the SE-ResNet-50 backbone in the encoder part instead of the ordinary U-Net encoder significantly improved F-Measure in all three datasets.

[3] https://youtu.be/kdDxea5xalU

Table 18 Ablation study to evaluate the contribution of each component in the proposed method; U-Net*: Kim&Ha method, Gr: RGB converted to grayscale, SuB: SuBSENSE background model, T: time, st: stream, ER: Early, MD: Middle, RN: ResNet, ICL: Input Cross Link, FPM: Feature Pooling Module, TWE: Twersky Loss, DICE: Dice Loss, BCE: Binary Cross Entropy Loss, FL: Focal Loss, CD: CDnet-2014, SBI: SBI-2015, LAS: LASIESTA, F: F-Measure

Methods	Input	Fusion	Stream	Encoder	Operators	Loss	CD (F)	SBI (F)	LAS (F)
U-Net	RGB	–	One	U-Net	–	DICE + BCE	0.9241	0.3753	0.2892
U-Net	Gr, BGS, Flux	ER	One	U-Net	–	DICE + BCE	0.9328	0.7877	0.7455
U-Net*	Gr(T), Gr(T-25), Gr(T-50), Gr(T-75), Gr(T-100), SuB	ER	One	U-Net	–	DICE + BCE	0.9328	0.6983	0.7151
MU-Net1	RGB	–	One	RN-18	RN, ICL	TWE + BCE	0.9147	0.3785	–
MU-Net1	RGB	–	One	RN-18	RN	TWE + BCE	0.9135	0.3734	–
MU-Net2	Gr, BGS, Flux	ER	One	RN-18	RN, ICL	TWE + BCE	0.9369	0.7625	–
MU-Net2	Gr, BGS, Flux	ER	One	RN-18	RN	TWE + BCE	0.9332	0.7594	–
DeepFTSG-1	Gr, BGS, Flux	ER	One	SE-RN-50	SE-RN	DICE + BCE	0.9652	0.8605	0.8465
DeepFTSG-1	Gr(T-1), Gr(T), Gr(T+1)	ER	One	SE-RN-50	SE-RN	DICE + BCE	0.9567	0.8412	–
DeepFTSG-1	Gr, SuB, Flux	ER	One	SE-RN-50	SE-RN	DICE + BCE	0.9675	0.8413	0.8547
DeepFTSG-1	Gr, BGS, Flux	ER	One	SE-RN-50	SE-RN	FL + BCE	0.9593	0.8595	
DeepFTSG-1	Gr, BGS, Flux	ER	One	SE-RN-50	SE-RN, FPM	DICE + BCE	0.9651	0.8515	0.8499
DeepFTSG-2	1st-st: RGB 2nd-st: BGS, Flux	ER, MD	Two	1st-st: RN-18 2nd-st: RN-18	RN, ICL	TWE + BCE	0.9414	0.8187	–
DeepFTSG-2	1st-st: RGB 2nd-st: BGS, Flux	ER, MD	Two	1st-st: SE-RN-50 2nd-st: SE-RN-50	SE-RN	DICE + BCE	0.9514	0.8394	0.8503
DeepFTSG-2	1st-st: RGB 2nd-st: BGS, Flux	ER, MD	Two	1st-st: SE-RN-50 2nd-st: RN-18	1st-st: SE-RN 2nd-st: RN	DICE + BCE	0.9700	0.8715	0.8510
DeepFTSG-2	1st-st: RGB 2nd-st: SuB, Flux	ER, MD	Two	1st-st:SE-RN-50 2nd-st: RN-18	1st-st: SE-RN 2nd-st: RN	DICE + BCE	0.9731	0.8605	0.7645
DeepFTSG-2	1st-st: RGB 2nd-st: BGS, Flux	ER, MD	Two	1st-st: SE-RN-50 2nd-st: RN-18	1st-st: SE-RN 2nd-st: RN	FL + BCE	0.9691	0.8591	0.8282

Table 18 continued

Methods	Input	Fusion	Stream	Encoder	Operators	Loss	CD (F)	SBI (F)	LAS (F)
DeepFTSG-2	1st-st: RGB 2nd-st: BGS, Flux	ER, MD	Two	1st-st: SE-RN-50 2nd-st: RN-18	1st-st: SE-RN 2nd-st: RN, FPM	DICE + BCE	0.9694	0.8619	0.8480
DeepFTSG-3	1st-st: Gr(T-1). Gr(T), Gr(T+1) 2nd-st: BGS(T-1), BGS(T), BGS(T+1) 3rd-st: Flux(T-1), Flux(T), Flux(T+1)	ER, MD	Three	1st-st: RN-18 2nd-st: RN-18 3rd-st: RN-18	RN, ICL	TWE + BCE	0.9461	–	–
DeepFTSG-3	1st-st: Gr(T-1), BGS(T-1), Flux(T-1) 2nd-st: Gr(T), BGS(T), Flux(T) 3rd-st: Gr(T+1), BGS(T+1), Flux(T+1)	ER, MD	Three	1st-st: RN-18 2nd-st: RN-18 3rd-st: RN-18	RN, ICL	TWE + BCE	0.9443	0.8043	–

References

Akilan, T., Wu, Q. J., Safaei, A., Huo, J., & Yang, Y. (2020). A 3D CNN-LSTM-based image-to-image foreground segmentation. *IEEE Transactions on Intelligent Transportation Systems, 21*(3), 959–971.

Andrews, S., & Antoine, V. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding, 122*, 4–21.

Babaee, M., Dinh, D. T., & Rigoll, G. (2018). A deep convolutional neural network for video sequence background subtraction. *Pattern Recognition, 76*, 635–649.

Barnich, O., & Van Droogenbroeck, M. (2011). ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans Image Processing, 20*(6), 1709–1724.

Benezeth, Y., Jodoin, P.M., Emile, B., & Rosenberger, C. (2008). Review and evaluation of commonly-implemented background subtraction algorithms. In *2008 19th International Conference on Pattern Recognition*, pp. 1–4

Bianco, S., Ciocca, G., & Schettini, R. (2017). How far can you get by combining change detection algorithms? In *Image analysis and processing*Springer, pp. 96–107

Brutzer, S., Hoferlin, B., & Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. *CVPR, 2011*, 1937–1944.

Bunyak, F., Palaniappan, K., Nath, S. K., & Seetharaman, G. (2007). Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking. *Journal of Multimedia, 2*(4), 20.

Candes, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM), 58*(3), 1–37.

Carlos, C., Eva Maria, Y., & Narciso, G. (2016). Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA. *Computer Vision and Image Understanding, 152*, 103–117.

Caye Daudt, R., Le Saux, B., & Boulch, A. (2018). Fully convolutional siamese networks for change detection. In *2018 25th IEEE international conference on image processing (ICIP)*, pp. 4063–4067

Chan, A. L. (2009). *A description on the second dataset of the US army: Research laboratory force protection surveillance system*. Army Research Laboratory.

Chen, L., Papandreou, G., Kokkinos, I., et al. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans Pattern Analysis and Machine Intell, 40*(4), 834–848.

Crivelli, T., Piriou, G., Bouthemy, P., & Yao, J. F. (2011). Simultaneous motion detection and background reconstruction with a conditional mixed-State Markov random field. *International Journal of Computer Vision, 94*, 295–316.

Daniel, B., Carlos, C., Francisco, M., & Garcia, N. (2018). Real-time nonparametric background subtraction with tracking-based foreground update. *Pattern Recognition, 74*, 156–170.

Dardo, D., Palaniappan, K., & Seetharaman, G. (2016). Stream implementation of the flux tensor motion flow algorithm using GStreamer and CUDA. In*IEEE applied imagery pattern recognition (AIPR)*

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D. & Brox, T., (2015). FlowNet: Learning optical flow with convolutional networks. In*IEEE ICCV*, pp. 2758–2766

Fan, H., Lin, L., Yang. F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., & Ling, H. (2019). LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* pp. 5369–5378

Gibson, J. J. (1950). *The perception of the visual world*. Houghton-Mifflin.

Girshick, R. (2015). Fast R-CNN. In: *Proceedings of the IEEE international conference on computer vision*

Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J. & Ishwar, P. (2012). Changedetection.Net: A new change detection benchmark dataset. In: *2012 IEEE computer society conference on computer vision and pattern recognition workshops*

Haines, T. S., & Xiang, T. (2014). Background subtraction with DirichletProcess mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*, 670–683.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. In: *Proceedings of the IEEE international conference on computer vision*

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE/CVF conference on computer vision and pattern Recognition*, pp. 7132–7141

Kim, J. Y., & Ha, J. E. (2020). Foreground objects detection using a fully convolutional network with a background model image and multiple original images. *IEEE Access, 8*, 159864–159878.

Li, C., Wang, X., Zhang, L., Tang, J., Wu, H., & Lin, L. (2017). Weighted low-rank decomposition for robust grayscale-thermal foreground Detection. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(4), 725–738.

Lim, L. A., & Keles, H. Y. (2018). Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters, 112*, 256–262.

Lim, L. A., & Keles, H. Y. (2020). Learning multi-scale features for foreground segmentation. *Pattern Analysis and Applications, 23*, 1369–1380.

Liu, X., Zhao, G., Yao, J., & Qi, C. (2015). Background subtraction based on low-rank and structured sparse decomposition. *IEEE Transactions on Image Processing, 24*(8), 2502–2514.

Liu, X., Yao, J., Hong, X., Huang, X., Zhou, Z., & Qi, C. (2017). Background subtraction using spatio-temporal group sparsity recovery. *IEEE Transactions on Circuits and Systems for Video Technology, 28*(8), 1737–1751.

Maddalena, L., & Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing, 17*, 1168–1177.

Maddalena, L., & Petrosino, A. (2012). The SOBS algorithm: What are the limits? In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pp. 21–26

Maddalena, L., & Petrosino, A. (2015). Towards benchmarking scene background initialization. *New Trends in Image Analysis and Processing, 18*, 469–476.

Mandal, M., Dhar, V., Mishra, A., et al. (2021). 3DCD: Scene independent end-to-end spatiotemporal feature learning framework for change detection in unseen videos. *IEEE Transactions on Image Processing, 30*, 546–558.

Palaniappan, K., Ersoy, I., Seetharaman, G., Davis, S., Rao, R., & Linderman, R. (2010). Multicore energy efficient flux tensor for video analysis. In *IEEE workshop on energy efficient high-performance computing*

Palaniappan, K., Ersoy, I., Seetharaman, G., Davis, S. R., Kumar, P., Rao, R. M., & Linderman, R. (2011). Parallel flux tensor analysis for efficient moving object detection. In *14th international conference on information fusion*

Radke, R., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Trans Image Proc, 14*, 294–307.

Rahmon, G., Bunyak, F., Seetharaman, G., & Palaniappan, K. (2021). Motion U-Net: Multi-cue encoder-decoder network for motion segmentation. In *2020 25th international conference on pattern recognition (ICPR)*, pp. 8125–8132

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *MICCAI, 18*, 234–241.

Schuster, R., Wasenmuller, O., Unger, C., Kuschk, G., & Stricker, D. (2020). SceneFlowFields++: Multi-frame matching, visibility prediction, and robust interpolation for scene flow estimation. *International Journal of Computer Vision, 128*, 527–546.

Shervin, M., Yuri, B., Fatih, P., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*, 3523–3542.

St-Charles, P. L., Bilodeau, G. A., & Bergevin, R. (2015). SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing, 24*, 359–373.

St-Charles, P. L., Bilodeau, G. A., & Bergevin, R. (2016). Universal background subtraction using word consensus models. *IEEE Transactions on Image Processing, 25*, 4768–4781.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. pp. 240–248

Sun, D., Yang, X., Liu, M. Y., & Kautz, J. (2018). PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE/CVF Conference on computer vision and pattern recognition*, pp 8934–8943

Tezcan, M. O., Ishwar, P., & Konrad, J. (2021). BSUV-Net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction. *IEEE Access, 9*, 53849–53860.

Theau, J. (2008). *Change detection*. Springer.

Wang, R., Bunyak, F., Seetharaman, G., & Palaniappan, K. (2014a). Static and moving object detection using flux tensor with split Gaussian models. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*

Wang, Y., Jodoin, P. M., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014b). CDnet-2014: An expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*

Wang, Y., Luo, Z., & Jodoin, P. M. (2017). Interactive deep learning method for segmenting moving objects. *Pattern Recognition Letters, 96*, 66–75.

Wenbo, Z., Kunfeng, W., & Fei-Yue, W. (2020). A novel background subtraction algorithm based on parallel vision and Bayesian GANs. *Pattern Recognition Letters, 394*, 178–200.

Xin, B., Tian, Y., Wang, Y., & Gao, W. (2015). Background subtraction via generalized fused lasso foreground modeling. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pp. 4676–4684

Yizhe, Z., & Elgammal, A. (2017). A multilayer-based framework for online background subtraction with freely moving cameras. In *IEEE international conference on computer vision (ICCV)*, pp. 5142–5151

Yuille, A., & Liu, C. (2020). Deep nets: What have they ever done for vision? *International Journal of Computer Vision, 129*, 781–802.

Zhou, X., Yang, C., & Yu, W. (2012). Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(3), 597–610.

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th international conference on pattern recognition, 2004. ICPR 2004*, pp. 28–31

Zivkovic, Z., & van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters, 27*(7), 773–780.