A 3D CNN-LSTM-Based Image-to-Image Foreground Segmentation

Thangarajah Akilan, *Student Member, IEEE*, Qingming Jonathan Wu, *Senior Member, IEEE*, Amin Safaei, *Senior Member, IEEE*, Jie Huo, *Student Member, IEEE*, and Yimin Yang, *Member, IEEE*

***Abstract*— The video-based separation of foreground (FG) and background (BG) has been widely studied due to its vital role in many applications, including intelligent transportation and video surveillance. Most of the existing algorithms are based on traditional computer vision techniques that perform pixel-level processing assuming that FG and BG possess distinct visual characteristics. Recently, state-of-the-art solutions exploit deep learning models targeted originally for image classification. Major drawbacks of such a strategy are the lacking delineation of FG regions due to missing temporal information as they segment the FG based on a single frame object detection strategy. To grapple with this issue, we excogitate a 3D convolutional neural network (3D CNN) with long short-term memory (LSTM) pipelines that harness seminal ideas, viz., fully convolutional networking, 3D transpose convolution, and residual feature flows. Thence, an FG-BG segmenter is implemented in an encoder-decoder fashion and trained on representative FG-BG segments. The model devises a strategy called double encoding and slow decoding, which fuses the learned spatio-temporal cues with appropriate feature maps both in the down-sampling and up-sampling paths for achieving well generalized FG object representation. Finally, from the Sigmoid confidence map generated by the 3D CNN-LSTM model, the FG is identified automatically by using Nobuyuki Otsu's method and an empirical global threshold. The analysis of experimental results via standard quantitative metrics on 16 benchmark datasets including both indoor and outdoor scenes validates that the proposed 3D CNN-LSTM achieves competitive performance in terms of figure of merit evaluated against prior and state-of-the-art methods. Besides, a failure analysis is conducted on 20 video sequences from the DAVIS 2016 dataset.**

***Index Terms*— Deep learning, foreground-background segmentation, intelligent systems, LSTM, spatiotemporal cues.**

I. Introduction

IN THE field of Intelligent Transportation Systems (ITS), including autonomous driving, driver assistance, Simultaneous Localization and Mapping (SLAM), pedestrian and vehicle detection, the FG-BG segmentation using visual cues has been an integral subsystem. Hence, the video-based intelligent systems have become ubiquitous due to a myriad of easily accessible low-priced camera modules. Such applications face a crucial challenge of processing massive volume of data from multiple feeds at the same time. It is also required for them to tackle with varying environmental factors, like illumination changes, dynamic backgrounds, and so forth [1], [2]. These demands perplex the real-time operation of the systems. In the analysis of traffic flow or human activity, the performance of an ITS substantially depends on the robustness of FG-BG segmentation.

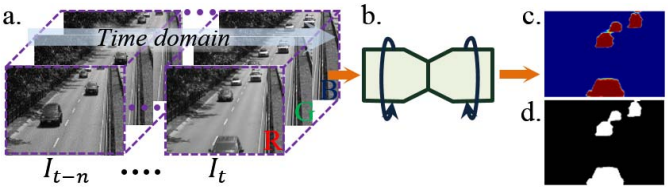


Fig. 1. Traffic flow identification: (a) Multi-channel [R,G,B] spatio-temporal input, (b) 3D CNN-LSTM, (c) FG score map, and (d) identified traffic flow (white - FG and dark - BG).

Besides being a core unit of video analytic intelligent framework, the FG-BG segmentation is also an inherent part of various machine-/computer- vision problems, for instance, attention-aware video analysis [1], [3], video saliency-based object segmentation and retrieval [4]–[6], image quality assessment [7], visual tracking [8], and human-robot or machine interaction [9]. The primary objective of FG-BG segmentation is to place a tight mask, where the appearance of an object, a vehicle or human is monitored. Such mask is very informative than using bounding box as it allows a close localization of the FG objects. An example of FG detection and BG suppression is shown in Fig. 1, where the FG-BG separation is employed to identify the traffic flow on a busy highway. It can be achieved by employing several algorithms categorized into five groups: i) Sample-based [10]–[15], ii). Probabilistic-based [16]–[21], iii). Subspace-based [22]–[24], iv). Codebook-based [25]–[27], and v). Neural network (NN)-based [28]–[33].

The sample-based algorithms create a generalized BG model based on the evidence collected in local-level, global-level, or a hybrid-level of the two from the past set of N frames, i.e., for each pixel/super-pixel location or region

Manuscript received January 26, 2018; revised July 20, 2018, October 9, 2018 and December 6, 2018; accepted February 16, 2019. This work was supported in part by the Canada Research Chair Program and in part by the NSERC Discovery Grant. The Associate Editor for this paper was C. Guo. *(Corresponding author: Qingming Jonathan Wu.)*

T. Akilan, Q. J. Wu, and J. Huo are with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: (thangara@uwindsor.ca; jwu@uwindsor.ca; huo11@uwindsor.ca).

A. Safaei is with Toronto Micro Electronics Inc., Mississauga, ON L5T 2H7, Canada (e-mail: safaei.a.s@ieee.org).

Y. Yang is with the Computer Science Department, Lakehead University, Thunder Bay, ON P7B 5E1, Canada (e-mail: yyang48@lakeheadu.ca).

Digital Object Identifier 10.1109/TITS.2019.2900426

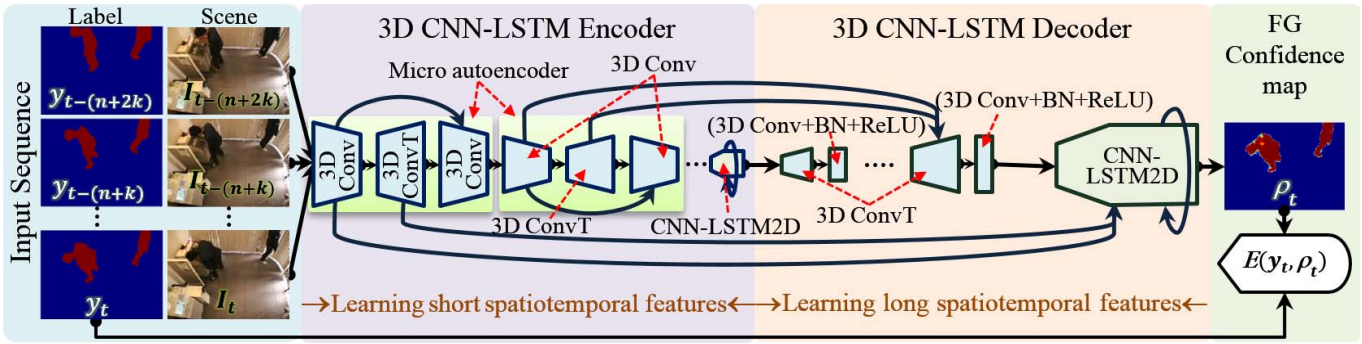


Fig. 2. An overview of the proposed 3D CNN-LSTM image-to-image network: 3D Conv - 3D Convolution w/t down sampling, 3D ConvT - 3D transpose conv (up-sampling), BN - Batch normalization, $E(\cdot)$ - binary cross-entropy error.

there are N samples stored. If there are k number of samples in the BG that have a distance smaller than a set threshold τ to the incoming pixel/super-pixel or the region in the current frame, then it is classified as BG, otherwise FG. The probabilistic models work on the principle of stochastic process, like Gaussian Mixture Models (GMM) [17], [34] and Conditional Random Field (CRF)-based algorithms [35]. The subspace-based approaches perform a transformation of data to a subspace, such as Eigenspace or Principal Component Analysis (PCA)-based subspace. Then, they form a BG model using the subspace and estimate the FG. The Codebook generates a dictionary that consists of color, intensity, temporal features, or similar representations. Same properties of a new pixel are compared with the dictionary values to determine its status. The NN-based models formulate the FG-BG segmentation as a structured input-output matching problem. Such models have gained their reputation after series of breakthrough performances in the ImageNet-Large-Scale Visual Recognition Challenge (ILSVRC) since the year of 2012. The NN-based techniques have been exploited for visual semantics/labeling [36], [37], medical image partitioning [38], [39], and recently for video FG-BG segmentation [1] as well. The main challenges in CNN-based FG detection and BG suppression is that dealing with time-dependent motion and the dithering effect at bordering pixels of the FG objects. We address these issues, by excogitating a 3D EnDec CNN that utilizes a strategy called double encoding with micro-autoencoders and slow decoding using residual connections like in ResNet [40] for lost feature recovery and 3D Conv and LSTM units to handle local to global long-short term spatio-temporal motion of the FG objects. To facilitate the training process, we take advantage of intra-domain transfer learning.

In summary, this paper focuses on improving a Vanilla image-to-image Conv-LSTM model for enhanced FG object localization. To this end, the key contributions of this paper are as follows:

i. It introduces a novel technique named double-encoding using autoencoder-like micro modules and slow-decoding using feature passing residual connections. Here, an input feature at a stage during down-sampling process is encoded twice before it reaches completely to the next level of dimension reduced feature map. While, the up-sampling process decodes the feature maps with two sets of residual feature flows from down-sampling stages for every new spatial dimension of the feature space.
ii. The time-dependent video cues are handled by 3D convolutions to capture the short temporal motions while the long-short term temporal motions are captured by LSTM modules in the down-sampling and up-sampling stages, respectively.
iii. It provides empirical manifest to show the effectiveness of the proposed model compared to a Vanilla Conv-LSTM network.
iv. It carries out testing in an exhaustive manner on various video datasets from the benchmark database called change detection 2014 (CDnet) [20] and failure analysis on DAVIS-2016 dynamic camera video sequences [41].

The rest of this paper is organized as follows: Section II reviews related literature. Section III elaborates the architectural information. Section IV describes the experimental set-up, analyses the performance, and highlights some key characteristics of the compared existing methods. Finally, Section VI concludes the paper with future directions.

II. Review: CNN for Segmentation

Deep CNNs have shown state-of-the-art performance in object segmentation/detection/localization over traditional methods, like GMM [5], [17], Graph-cut, Nonparametric models [15], Visual background extractor (ViBe) [11], and Pixel-Based Adaptive Segmenter (PBAS) [42]. Here, the FCN [36] is a pioneer of CNN architecture that reinterprets the standard visual classification network as layers of full 2D convolutional computations without flattened fully connected layers. This model introduces feature-level augmentations through skip connections that combine deep, coarse, semantic detail and shallow, fine, appearance cues from chosen mid-layers. In contrast, our model performs 3D convolutions with LSTM modules and does the coarse-level feature fusion in a structured manner, as shown in Fig. 2. The introduction of micro autoencoder blocks are based on the philosophy of increasing the network depth instead of widening for a better feature generalization. Hence, the residual feature flows negate the vanishing gradient of deep networks by carrying important information from earlier to later layers. Although such shortcuts seem, like

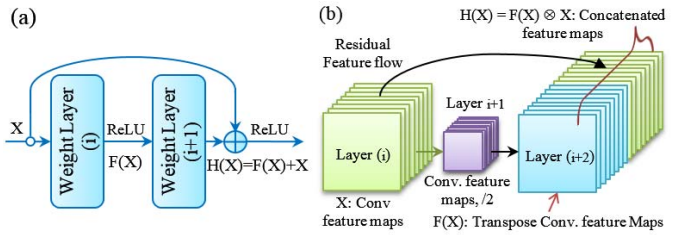


Fig. 3. CNN feature flows: (a) ResNet flow, and (b) the residual feature mapping of our 3D CNN-LSTM FG segmenter.

an addition to the conventional CNN connections, it alleviates training and reduces the number of parameters [40].

An illustration for the ResNet connection is depicted in Fig. 3 (a), where X is an input feature, $H(X)$ is a desired transformation, and $F(X)$ is a residual mapping. In [40], the feature fusion operation $H(X) = F(X) + X$ is performed by a shortcut connection and element-wise addition. Contrastingly, our model stacks the features depth-wise as $H(X) = F(X) \bigotimes X$, like in Fig. 3 (b), where $\bigotimes$ denotes coarse-level feature concatenation. This favors to have less number of filters in conv layers resulting less computation.

Ronneberger *et al.* [39] restructures the FCN as EnDec CNN, referred U-net for biomedical cell segmentation. In that, the activation maps after each convolution (conv) in the encoding stage are concatenated with the spatially matching activation maps in the decoding stage. It allows the network to exploit the original contextual information to supplement the features after upsampling in the higher layers. In other words, it is a remedy for the lost spatial resolution due to pooling operations or consecutive convolutional kernel striding. The proposed 3D CNN-LSTM model abstracted by Fig. 2 has the following variations from the U-net:

i. The max-pooling operations achieve invariant features but has a toll on object localization accuracy [43]. To circumvent this, we perform subsampling process through 3D strided conv (kernel size of 3 and stride of 2).
ii. Our model entirely uses 3D conv layers instead of 2D.
iii. Our model uses 3D convolutions in the encoding stage, so it employs 3D transpose conv (3D convT) in contrast to the 2D standard upsampling operations.
iv. Our model requires a 5D input data (for time-domain: $[b, n, H, W, D]$), while the U-net takes 4D data ($[b, H, W, D]$) without consideration of temporal information. The b, n, H, W, D stand for the input's batch size, look back time steps, height, width, and number of channels.

The LSTM is an advanced version of Recurrent Neural Networks (RNNs) [44]. The Constant Error Carousel (CEC) cells in LSTM use an identity activation function and have self-routed connections to themselves with a constant weight of 1.0. So, the errors backpropagate through the LSTMs cannot explode or vanish [45]. It is considered to be biologically plausible structure, to a certain extent and has been proved to solve previously unlearnable DL tasks involving temporal data. There are many variations of LSTMs, such as decoupled extended Kalman filter LSTM-RNN [46], bi-directional [47], and Connectionist temporal classification (CTC) [48]. The LSTM unit is applicable to several real-world tasks, like handwriting recognition [49], speech/language identification [47], robot control/localization [50], and driver distraction detection [51]. Thus, our model also harnesses the LSTM to capture long-short term temporal connections of FG and BG in the consecutive frames.

III. Proposed Model: The 3D CNN-LSTM Foreground Segmenter

We take advantage of a bottom-up implementation strategy. As we want to handle the sequence learning using LSTM modules, we start from a scratch model as shown in Fig. 4. It is similar to the U-net [39], but instead of standard Conv layers, it employs Conv-LSTM2D layers. Then, we improve the scratch to the proposed 3D CNN-LSTM model. We test both the models on selected datasets as a sanity check and Proof of concept (PoC), then based on the empirical prove we finalize the model and carry out extensive experiments on sixteen benchmark CD-net video sequences and limitation analysis on twenty dynamic camera videos from the DAVIS 2016 dataset.

A. A Start From Scratch

Figure 4 overviews the Vanilla Conv-LSTM2D network that is the stepping stone of our proposed model. The size of the kernel k, stride rate s, and dimension of the output are denoted in the following order and enclosing braces $(k, s)[b, n, H, W, D]$ on each layer. Where, b, n, H, W, and D represents the batch size, number of samples taken by the Conv-LSTM modules to capture the temporal information, height and width of the frame, and number of output feature maps, respectively. The network has 24 layers with 298,529 trainable parameters that integrate three major components: encoder, decoder, and classifier. It maintains a constant number of filters (16) at each layer (except the penultimate layer, that produces 20 feature maps) and the kernel size, $k = 3$.

Hence, the spatial dimension of feature maps is linearly reduced by half through striding the kernel at a rate of 2 in the encoding phase. Thus, the last layer of the encoder generates feature maps that have spatial dimension of 15×20 as the network's input layer accepts frames with spatial dimension of 240×360. To achieve precisely decoded feature maps there are four mini-decoder blocks sequentially networked. Where, each block subsumes a 3D transpose conv, a Conv-LSTM2D, a concatenation, and again a Conv-LSTM2D layer. Hence, the final layer of the decoder produces feature maps with same spatial dimension as the network's input. Every stage in the decoder receives residual cues from the encoding stage via shortcuts as in the U-net. The final classifier module consists of a Batch Normalization (BN) and 3D conv layers with $Sigmoid$ function as classifier. Thus, the output of the Vanilla model is the FG-BG probability map of the current frame estimated based on the observed n frames.

B. The Proposed 3D CNN-LSTM Model

The proposed model overcomes the issue of lacking FG delineation by capturing short and long-short

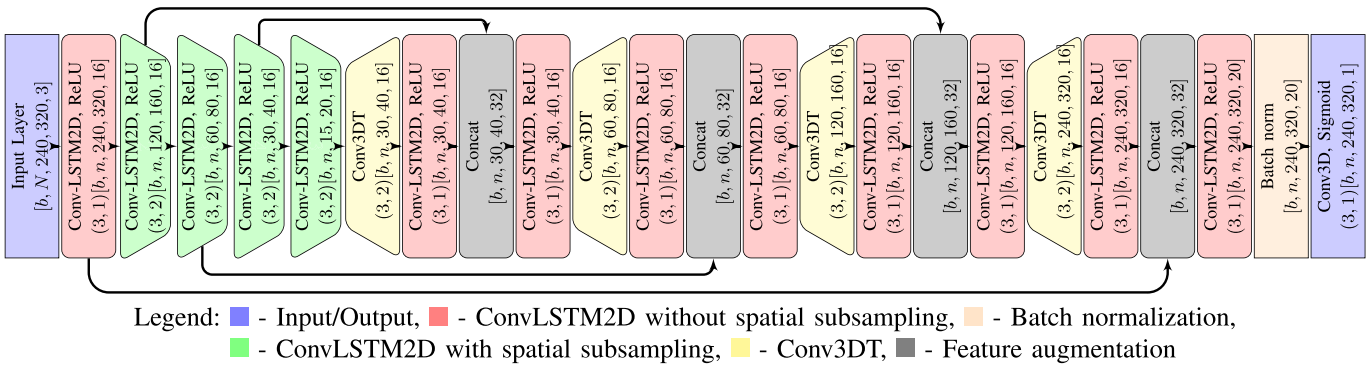


Fig. 4. A layer-wise schematic of the Vanilla model inspired by the U-net [39]. It exploits Conv-LSTM2D modules in an encoder-decoder fashion with residual feature concatenations.

spatiotemporal cues in local and global levels through 3D convolution and LSTM modules. The proposed model is an improved version of the Vanilla network in Fig. 4. Figure 5 elucidates the 3D CNN-LSTM model with type of operation carried out in each layer, the associated activation function, and the output dimension. It is optimized for number of parameters and performance. It gets rid of continuous heavy computing ConvLSTM2D layers in the Vanilla model and replaces them with 3D Conv layers. It has three micro-autoencoder blocks in the down sampling path while up-sampling process is slowed down by frequent feature concatenation blocks. End of down- and up- sampling stages a Conv-LSTM module is used to create long-short-term temporal features. Thus, the network becomes deeper with 42 layers and robust than the Vanilla model. Hence, as in the Vanilla model the FG-BG segmentation is handled by the Sigmoid classifier, which generates higher scores for all possible FG objects and lower scores (close to 0) for BG pixels.

It is noticed that although the 3D CNN-LSTM model is deeper than the Vanilla model, it consumes only 221,367 trainable parameters. It is understandable, since a ConvLSTM2D layer has more nodes than a Conv3D layer. For instance, a ConvLSTM2D with a kernel size of 3 and 16 output feature maps for a four-frame look back input sequence with a dimension of $240 \times 320 \times 3$ subsumes 11,008 trainable parameters while a Conv3D has only 1,312 trainable parameters for the same setting as shown in Table I. So by replacing a ConvLSTM2D with a Conv3D, we can achieve $\approx$ 84% reduction in the trainable parameters. That in return allows us to design a deeper architecture. Thus, our 3D CNN-LSTM neural net becomes 1.75 times deeper and $\approx$ 25% lesser parameters than the Vanilla model.

C. 3D Conv-LSTM

The Conv3D is pertinent to spatiotemporal representation learning [52]. It performs convolutional operations spatiotemporally unlike 2D Conv layer that does only spatially. Thus, a Conv3D extracts short-term or local temporal features resulting in an output volume. Later, these short-term temporal features are fed into Conv-LSTM units to retain long-term or global temporal connectivity of FG cues between

TABLE I
TRAINABLE PARAMETER COMPARISON: CONVLSTM2D VS. CONV3D BASED ON KERAS 2.1.5-TF API

Case	Layer type	Output shape	# Param
A	InputLayer	$b \times 240 \times 320 \times 3$	0
	ConvLSTM2D	$b \times 240 \times 320 \times 16$	11,008
B	InputLayer	$b \times 240 \times 320 \times 3$	0
	Conv3D	$b \times 240 \times 320 \times 16$	1,312

consecutive frames. The Conv operation is determined by its filter weights that are updated through training. An output feature map of a standard 2D Conv C w.r.t. kernel ω, and an input image/patch $\mathbf{x}$ is computed as

$$C(m,n) = \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} \omega(k,l) * \mathbf{x}(m+k, n+l), \tag{1}$$

where $*$, K, $\{m,n\}$, and $\{k,l\}$ represent the Conv operation, size of the kernel, first coordinate or origin of the image/patch, and element index of the kernel respectively. Hence, feature map dimension of the conv layer is given by $(I_s - K_s + 2 \times P)/S + 1$, where I_s, K_s, P, and S denotes size of input image/path, filter size, number of zero-padded pixels, and stride rate respectively.

The 2D Conv can be extended for a Conv3D as follows. Let the input path $\mathbf{x}$ as a volume of data. Then, the 3D Conv C_{3D} w.r.t. kernel ω, is computed as

$$C_{3D}(q,m,n) = \sum_{t=0}^{T-1} \sum_{k=0}^{K-1} \sum_{l=0}^{K-1} \omega(t,k,l) * \mathbf{x}(q+t, m+k, n+l), \tag{2}$$

where $*$, T, K, $\{q,m,n\}$, and $\{t,k,l\}$ represent the Conv operation, temporal length of the data, size of the kernel, first coordinate or origin of the input patch, and element index of the kernel respectively.

Hence, the conventional 1D LSTMs take temporal dependency into consideration, but not the spatial dependency. However, in this work, the 2D LSTMs cover both the spatial and temporal relationships as they are integrated with 3D Conv. Figure 6 describes a standard LSTM unit, where $X_1, \ldots, X_t$ are the inputs, C_t is the cell state, H_t is the hidden state,

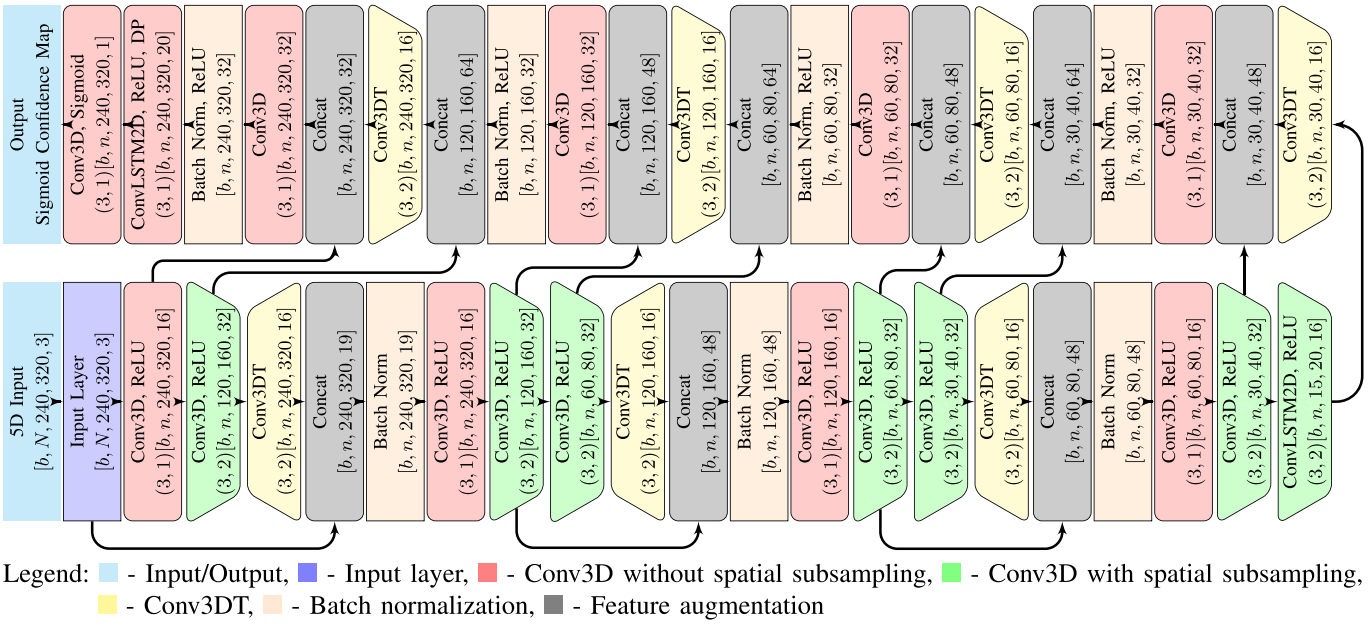


Fig. 5. A layer-wise schematic of the 3D CNN-LSTM model. It exploits autoencoder-like micro modules and slow-decoding strategy.

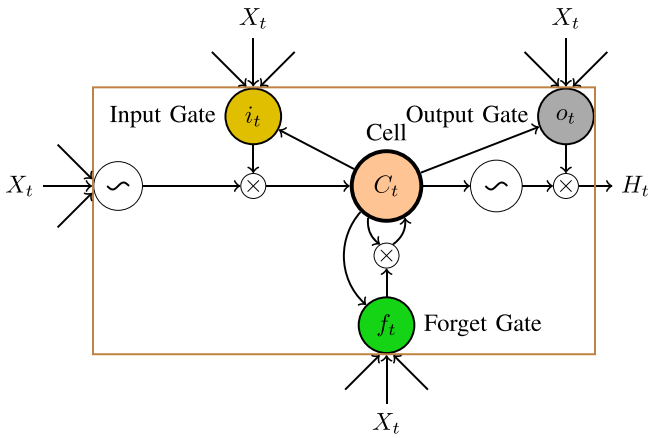


Fig. 6. A standard LSTM module with three gates.

and i_t, f_t, and o_t are the gates of a ConvLSTM block. If '$*$' and '$\circ$' denote the conv operator and Hadamard product, then computation of the ConvLSTM block can be derived as:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i), \quad (3)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f), \quad (4)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o), \quad (5)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \quad (6)$$

$$H_t = o_t \circ \tanh(C_t), \quad (7)$$

where σ is the recurrent activator, $W_{x\_}$ and $W_{h\_}$ are the spatial dimension of conv kernels. In this case, σ is a $hard\ sigmoid$ function.

D. Transpose Convolution

The Conv3DT layers perform upsampling of 3D Conv such that spatial dimension of the output feature maps become as twice as the input without losing the connectivity pattern. In contrast to spatial resizing (extrapolation), the transpose layer has trainable parameters. It is done by inserting zeros between consecutive neurons in the input receptive field, then sliding the conv kernel with unit strides [53].

E. Activation Functions

The activation functions improve NN's representation ability by introducing non-linear factors, since the linear representation of Conv operation faces its limits when it comes to deep architectures. The **ReLU** can be formally defined as (8) when taken a case with K number of anchor vectors, denoted by $\mathbf{w}_k \in \mathbb{R}^N, k = 1, 2, \ldots, K$. For a given input $\mathbf{x}$, the correlations with $\mathbf{a}_k$ and $k = 1, 2, \ldots, K$, defines a nonlinear rectification to an output $\mathbf{y} = (y_1, \ldots, y_K)^T$, where

$$y_k(\mathbf{x}, \mathbf{a}_k) = \max(0, \mathbf{a}_k^T \mathbf{x}) \equiv ReLU(\mathbf{a}_k^T \mathbf{x}), \quad (8)$$

i.e., it clips negative values to zero while keeping positive quantities intact. The benefit of ReLU is sparsity, overcoming vanishing gradient, and efficient computation than other activations. **Sigmoid**, on the other hand, has output in the range [0, 1] for an input $\mathbf{x}$ and it is defined by

$$f(\mathbf{x}) = \frac{1}{1 + exp(-\mathbf{x})}. \quad (9)$$

It befits a binary classifier, as used in this work and linear regression problems. **Hard-Sigmoid** is a linear piece-wise function that approximates the outputs as a linear interpolation between pair of cut-points. It is computationally very fast [54].

F. Batch Normalization

The BN operation can be mathematically formulated as follows. Let the output of a layer $\mathbf{X} \in \mathbb{R}^{N,D}$, where N is the number of samples in the mini-batch and D is the number

TABLE II
DATASET SUMMARY

Dataset name	Frame size ($W \times H$)	Nature	N frames
Highway	320 × 240	Baseline	1229
Office	360 × 240		1447
Pedestrians	360 × 240		753
PETS2006	720 × 576		900
Canoe	320 × 240	Dynamic background	342
Boats	320 × 240		6026
Overpass	320 × 240		440
Fall	720 × 480		1400
Boulevard	320 × 240	Camera jitter	1004
CopyMachine	720 × 480	Shadow	1401
PeopleInShade	380 × 244		829
BusStation	360 × 240		832
TwoPositionPTZCam	570 × 340	PTZ camera	449
Turnpike_0_5fps	320 × 240	Low framerate	350
Sofa	320 × 240	Intermittent object motion	2243
TramStation	480 × 295	Nighttime	1250

of hidden neurons, then normalized matrix $\hat{\mathbf{X}}$ is given as

$$\hat{\mathbf{X}} = \frac{\mathbf{X} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \tag{10}$$

where μ_B, σ_B^2, and ϵ refer to the mean and variance of the mini-batch, and a small value of 0.001 to prevent division by zero, respectively. Then, the layer maintains its representational strength by testing the identity transform as

$$y = \gamma \hat{X} + \beta, \tag{11}$$

where, β and γ are trainable parameters that are initialized with $\beta = 0$ and $\gamma = 1$, in this work. Note that, when $\beta = \mu_B$ and $\gamma = \sqrt{\sigma_B^2 + \epsilon}$ it returns the previous layer's activation map. Employing BN has multifaceted benefits: i) reducing internal Covariate shift by keeping μ_B and σ_B close to 0 and 1. ii) Since the batch of examples given in the training are normalized, it increases the generalization of the model. iii) When the BN is located prior to non-linearity, it avoids an undesirable situation, where the training saturates areas of non-linearities, solving the issues of vanishing exploding gradients, and iv) It allows the training process with much higher learning rates without much attention to initialization [55].

G. Training Strategy

1) Exclusive Sets: Experiments are carried on widely accepted video sequences from change detection 2014 [20] benchmark database, a.k.a. CDnet. Table II briefs the properties of the datasets. To form exclusive sets of training and test, the available samples with ground truths are divided such a way the training set takes first 70% of frames and the test set takes the rest. This approach is more appropriate than a random selection of frames used in [31] for video FG segmentation. Because, an arbitrary choice of samples may pick a frame, I_t for training set while picking a temporally closest frame, like I_{t+1} or I_{t-1} for test set. There can be many such instances in random selection resulting in mere exclusiveness of training and test sets. Note that, in [1], 90% of the samples are selected for training and only 20 samples from the rest are considered for testing from each dataset. To meet the input layer requirement of the proposed model, the divided training and test sets, have to be rearranged to form a 5D data sequence as shown in Fig. 7, where t, k refer the number of frames taken to represent the temporal domain and total number of sequential samples in the particular dataset. Accordingly, the same arrangement is done for the corresponding ground truths as well. Here, the batch size b, and the number of look back frames t are set to 16 and 4 respectively for all the video sequences. The selection of these values depends on the available GPU memory. In our case, the GPU is an NVIDIA GeForce GTX 1080 Ti with 11GB memory.

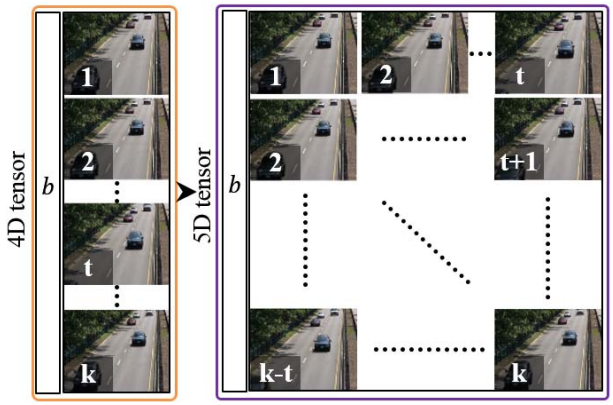


Fig. 7. Sequence generation for the 3D CNN-LSTM model: $4D \in \mathbb{R}^{b \times H \times W \times D}$, $5D \in \mathbb{R}^{b \times t \times H \times W \times D}$, where b, H, W, D, and t stand for input batch size, height, width, number of channels and number of look back frames, respectively.

2) Training: The 3D CNN-LSTM is trained individually on each dataset with *Adadelta* optimizer that minimizes binary cross-entropy loss defined by Eqn. (12), where the base learning is set to 0.0002 with a scheduler that reduces the learning rate by factor of 0.8.

$$E = \frac{-1}{n} \sum_{n=1}^{N} \left[p_n \log \hat{p}_n + (1 - p_n) \log(1 - \hat{p}_n) \right], \tag{12}$$

where it takes two inputs; first one is the output from the final layer of the network with dimension of $N \times t \times C \times H \times W$, which maps the FG pixel probabilities $\hat{p}_n = \sigma(x_n) \in [0, 1]$ using Sigmoid classifier, $\sigma(.)$ defined earlier in Eqn. 9. And the second one is target $p_n \in [0, 1]$ with the same dimension as the first one, where N, t, C, H, and W represent the batch size, the number of frames in temporal axis and channels, height, and width respectively. In this case, p_n is the normalized segmentation ground truth images.

3) Transfer Learning: To improve the network's trainability in short-span of epochs, it is necessary to have proper weight initialization. It can be achieved by transfer learning, where the model learns new task efficiently by using already learned parameters or knowledge [56]. To this end, we incorporate intraclass domain transfer and fine-tuning following the dataset pairs given in Table III. For instance, the model is trained with

TABLE III
THE DATASET PAIRS USED FOR MODEL FINE-TUNING

Fine-tuned to	Transferred from	Fine-tuned to	Transferred from
Highway	TramStation	Boulevard	Highway
Office	PETS2006	CopyMachine	PETS2006
Pedestrians	Overpass	PeopleInShade	BusStation
PETS2006	Pedestrians	BusStation	Sofa
Canoe	Boats	Sofa	BusStaion
Boats	Canoe	TramStation	Boulevard
Overpass	Pedestrians	Turnpike_0_5fps	TwoPositionPTZCam
Fall	Boulevard	TwoPositionPTZCam	Turnpike_0_5fps

random initialization on *TramStation* then fine-tuned for *Highway*. Note that, since the network has few trainable parameters we fine-tune the entire layers with a smaller learning rate. We expect the pre-trained weights to be quite good already when compared to random initialization, so we do not like to distort them too quickly and too much. As a rule of thumb, we set the initial learning rate ten times smaller than the one used for training from scratch.

4) Environment: Python with Keras (Tensorflow backend) is used as a software paradigm. The network is trained on a GTX 1080 Ti 11 GiB with Intel(R) Core(TM) i7-6850K CPU @ 3.60 GHz, 64 GiB memory, and Ubuntu 64-bit OS. In average, the training takes about 1.5 to 2 hours depends on the properties of the video sequence.

H. Binary Foreground Mask

It is also crucial to create a binary mask that segments FG region from BG. We apply an empirically determined dataset-specific global threshold value ($[0.05, 0.75]$) to transform FG confidence maps generated during inferencing. Then to clean noisy artifacts, a neighborhood connectivity-based post-processing is carried out removing regions of 50 pixels or less. As the FG confidence map represents a bi-modal grayscale image, we also employ the Nobuyuki Otsu's clustering algorithm to choose an appropriate threshold adaptively. Otsu iteratively computes a threshold value, τ that lies in-between two peaks of the intensity histogram of a bi-model image, whereby intraclass variances are minimum [57]. The weighted sum of within-class variance is defined as

$$\sigma_{\rho}^2(\tau) = \rho_0(\tau)\sigma_0^2(\tau) + \rho_1(\tau)\sigma_1^2(\tau), \tag{13}$$

where the weights ρ_0 and ρ_1 are the probabilities of BG and FG clustered by a threshold τ, and the variances of these two classes are σ_0^2 and σ_1^2 respectively. This binarization process is part of testing stage only as the numerical analysis is made on the binarized FG.

IV. EXPERIMENTAL SETUP, RESULTS, AND DISCUSSION

This section provides an empirical evidence for the proposed 3D CNN-LSTM model as a performance comparison to the Vanilla model. Then, it extends the examination of the proposed model through comparisons to existing methods, including classical approaches and recent NN-based models. Some highlights of the compared methods are also provided on-the-fly.

The model is evaluated on sixteen video sequences from the benchmark change detection database [20] that consists of both indoor and outdoor scenes. A succinct description of the datasets is given in Table II. General nature of the datasets as follows: the **baseline** represents a mixture of mild challenges, like subtle background motion, isolated shadows, swaying tree branches, and natural illumination changes; the **dynamic background** includes scenes with strong (parasitic) BG motion, and shimmering water; the **camera jitter** contains outdoor videos captured by vibrating cameras due to high wind; and the **shadow** category comprises indoor video exhibiting strong as well as faint shadows, where the shadows are even cast by the moving FG objects on the scene; the **low frame rate** contains sequences recoded with low frame rate; The PTZ camera and nighttime categories contain surveillance videos shot with PTZ cameras and shot at night, respectively; and **the intermittent object motion** set contains videos containing BG objects moving away, abandoned objects, and objects stopping for a short while and then moving away.

A. Evaluation Matrix

The standard performance measure that evaluates the similarity between predicted FG and the ground truth segmentation. It is a weighted harmonic mean measure of recall and precision, i.e., a region of intersection taken as ratio of union of predicted and actual FG segments as:

$$FoM = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \times 100\%, \tag{14}$$

where $[0 \leq FoM \leq 100]$, recall is the detection rate defined by $TP/(TP+FN)$ and precision is the percentage of correct prediction compared to the total number of detections as positives, given by $TP/(TP+FP)$, where TP, FN, and FP refer true positive, false negative, and false positive respectively.

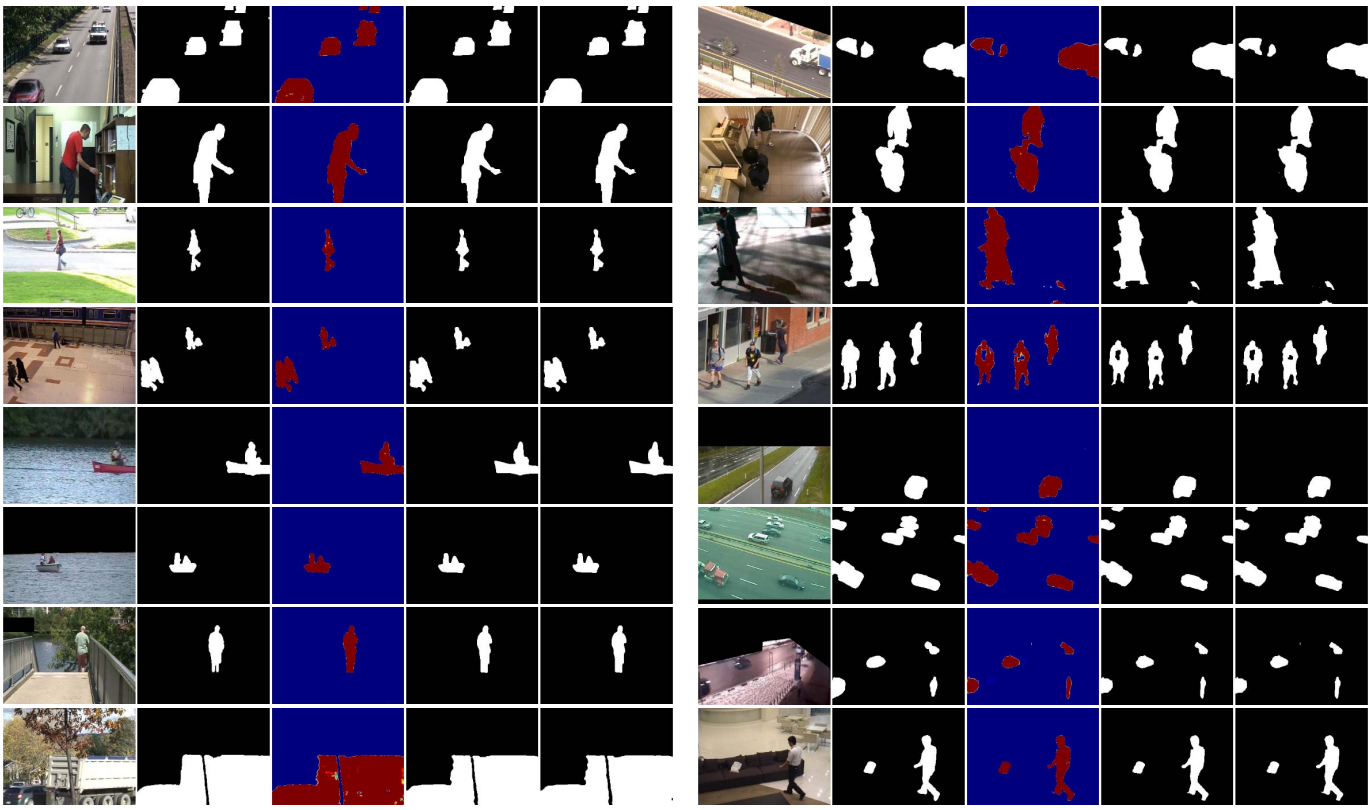
B. Sanity Test: Vanilla Model vs 3D CNN-LSTM

To validate the performance gain of the proposed 3D CNN-LSTM model that exploits the micro-autoencoder and slow decoding blocks, firstly we carry out a comparative analysis between the Vanilla model (Fig. 4) and the improved architecture (Fig. 5). The sanity test is performed on a subset of video sequences from Table II, including three categories: the baseline, dynamic background, and cast shadow. The sanity check results are tabularized in Table IV.

The results show that the 3D CNN-LSTM records the best performance on all the datasets while there is an $\approx$ 4% improvement overall. Hence, the proposed 3D CNN-LSTM model has inferencing speed of $\approx$ 24 frames-per-second (FPS) which is 9 frames higher than the Vanilla model that only produces $\approx$ 15 FPS. Considering these results as an empirical foundation, extensive investigation is followed through on the proposed model with all sixteen video sequences described in Section III-G.

C. Qualitative Analysis

A visual inspection is carried out by comparing the predicted FG regions with the ground truth segmentations. We limit the qualitative presentation with one sample per data



(a) (b)

Fig. 8. Qualitative results of the proposed 3D CNN-LSTM image to image encoder-decoder FG-BG segmenter: bright and dark pixels represent the FG and BG respectively. (a) Samples for Highway, Office, Pedestrians, PETS2006, Canoe, Boats, OverPass, and Fall datasets (from row 1 to 8). Where, from column 1 to 5: input frames, ground truths, Sigmoid confidence maps, binary FG-BG segmentations generated through G-th and O-th respectively. (b) Samples for Boulevard, CopyMachine, PeopleInShade, BusStation, TwoPositionPTZCam, Turnpike_5_fps, TramStation, and Sofa data sequences (from row 1 to 8). Where, column 1 to 5: input frames, ground truths, Sigmoid confidence maps, binary FG-BG segmentations generated through G-th and O-th respectively.

TABLE IV

SANITY CHECK RESULTS OF THE PROPOSED 3D CNN-LSTM COMPARED TO THE VANILLA MODEL IN TERMS OF FOM

Dataset	Model	G-th	O-th	Average
Highway	Vanilla	95.20	95.58	95.39
	3D CNN-LSTM	97.71	97.69	97.70
Office	Vanilla	95.19	95.14	95.17
	3D CNN-LSTM	96.86	96.74	96.80
Canoe	Vanilla	93.83	88.23	91.03
	3D CNN-LSTM	95.39	95.34	95.37
Boats	Vanilla	90.88	91.14	91.01
	3D CNN-LSTM	92.60	92.47	92.54
Overpass	Vanilla	90.21	88.02	89.12
	3D CNN-LSTM	95.78	95.46	95.62
Boulevard	Vanilla	87.07	81.35	84.35
	3D CNN-LSTM	95.70	95.16	95.43
CopyMachine	Vanilla	95.53	94.58	95.06
	3D CNN-LSTM	96.83	96.67	96.75
Overall average	Vanilla	92.56	90.58	91.57
	3D CNN-LSTM	95.84	95.65	95.74

sequence as shown in Fig. 8 due to space constraints. The qualitative results show that the proposed model has tightly detected the FG and BG when compared to the ground truth segmentations. However, it is important to evaluate its performance across all the test frames of all the video sequences numerically.

D. Quantitative Analysis: FoM

The quantitative analysis is provided in Table V as a comparison between the proposed model and existing methods ranging from traditional approaches to deep learning techniques. It shows that none of the algorithms output performs others on all the video sequences. However, the prosed 3D CNN-LSTM model exhibits a robustness with average FoM of $\approx 94\%$. The overall improvements of the proposed model are 14%, 6%, 7%, 7%, 8%, and 6% compared to PBAS [42], PAWCS [13], IUTIS-5 [58], MBS [59], DeepBS [31], and DBFCN [1], respectively.

Hence, It is noticed that there is not much significance gain between Global threshold and Otsu's algorithm to transform the Sigmoid confidence map generated by the 3D CNN-LSTM. It proves that the network generates very strong probability scores that have ignorable noise between FG and BG.

The key aspects of the compared existing methods are as follows. Hofmann *et al.* [42] come up a Pixel-Based Adaptive Segmenter (PBAS) using local features, that generates a generic BG from a dictionary of recently observed pixel values in a non-parametric manner. Then, the FG region is separated based on a set threshold. Researchers, like Charles *et al.* [12], [13] also exploit local features to model the BG. They adapt and integrate Local Binary Similarity Patterns (LBSP) as additional cues to pixel intensities in a

TABLE V

PERFORMANCE COMPARISON IN TERMS OF FOM: GLOBAL-TH AND OTSU-TH STAND FOR THE TWO METHODS APPLIED TO TRANSFORM SIGMOID SCORES TO BINARY MASK. VALUES IN ARE THE BEST WHILE THE ONES IN ARE THE SECOND BEST FOM (NA - NOT AVAILABLE)

Dataset / Method	PBAS [42] (2012)	PAWCS [13] (2015)	IUTIS-5 [58] (2017)	MBS [59] (2017)	DeepBS [31] (2018)	DBFCN [1] (2018)	Proposed 3D CNN-LSTM Global-th	Proposed 3D CNN-LSTM Otsu-th
Highway	94.51	94.36	95.35	92.17	96.55	94.12	97.71	97.69
Office	94.20	93.75	96.86	97.19	97.80	92.36	96.86	96.74
Pedestrians	93.63	94.61	96.69	95.66	94.59	83.94	95.53	95.01
PETS2006	87.36	93.15	93.54	86.48	94.25	90.59	93.79	93.63
Canoe	71.96	93.79	94.62	93.45	97.94	na	95.39	95.34
Boats	36.11	84.16	73.32	90.41	81.21	na	92.60	92.47
OverPass	79.25	95.90	92.72	89.90	94.16	na	95.78	95.46
Fall	87.14	90.52	93.61	56.68	82.94	82.03	95.73	95.66
Boulevard	66.02	84.44	76.80	86.72	86.23	na	95.70	95.16
CopyMachine	87.27	91.43	92.60	87.11	95.34	na	96.83	96.67
PeopleInShade	89.19	89.86	91.03	90.16	91.97	na	95.89	95.72
BusStation	86.09	87.29	88.26	86.95	93.74	na	87.85	87.34
TwoPosPTZCam	na	81.68	79.09	79.59	87.04	na	90.08	90.01
Turnpike	na	91.46	88.02	89.01	49.17	na	95.58	95.30
Sofa	73.81	72.47	79.14	84.55	81.34	86.45	92.85	92.86
TramStation	82.43	74.28	60.80	88.56	47.54	na	86.11	85.00
Overall	80.64	88.32	87.03	87.16	85.74	88.25	94.02	93.75

non-parametric consensus-based BG model that is then automatically tuned using pixel-level feedback loops. Meanwhile, Babaee *et al.* [31] employ a conventional CNN, train the network with randomly selected video frames with ground-truth segmentations patch-wise, like in [28], and carry out spatial-median filtering as the post-processing of the network outputs. Yang *et al.* [1] follow the structure of the FCN described in [36] with replacing the few standard Conv layers to atrous convolution branches that use different dilate to extract spatial information from extended neighborhoods of pixels. They also include a CRF-based refinement step. Although over the past two decades many algorithms have been proposed, none of them can be the ultimate model for video FG inferencing. Therefore, Bianco *et al.* [58] explore a way of harnessing multiple state-of-the-art moving object detection algorithms to improve the FG segmentation. They obtain a solution tree through Genetic Programming (GP); however, this approach also cannot be acclaimed as a universal solution. Similarly, Sajid and Cheung [59] introduce a multi-modality framework that estimates multiple BG models and use them as Background Model Bank (BMB). Then, to segment the FG from the dynamic BG, they apply a spatial de-noising approach based on Mega-Pixel (MP) to pixel-level probability estimation using various color spaces and get multiple FG regions. Later, a fusion technique is employed to define a final FG.

In summary, most of the state-of-the-art methodologies use patch-wise processing and multi-modality-based algorithms for BG establishment and a feedback-based approach as post processing to refine the primarily detected FG regions. Such setup ensues complex computations and higher processing time due to the time-consuming iterative pursuit of low-rank or sparse matrix. On the contrary, the proposed model processes the whole input image as a single entity during inferencing. Then, it refines the output by a non-iterative process, resulting $\approx$ 24 FPS for FG-BG inferencing.

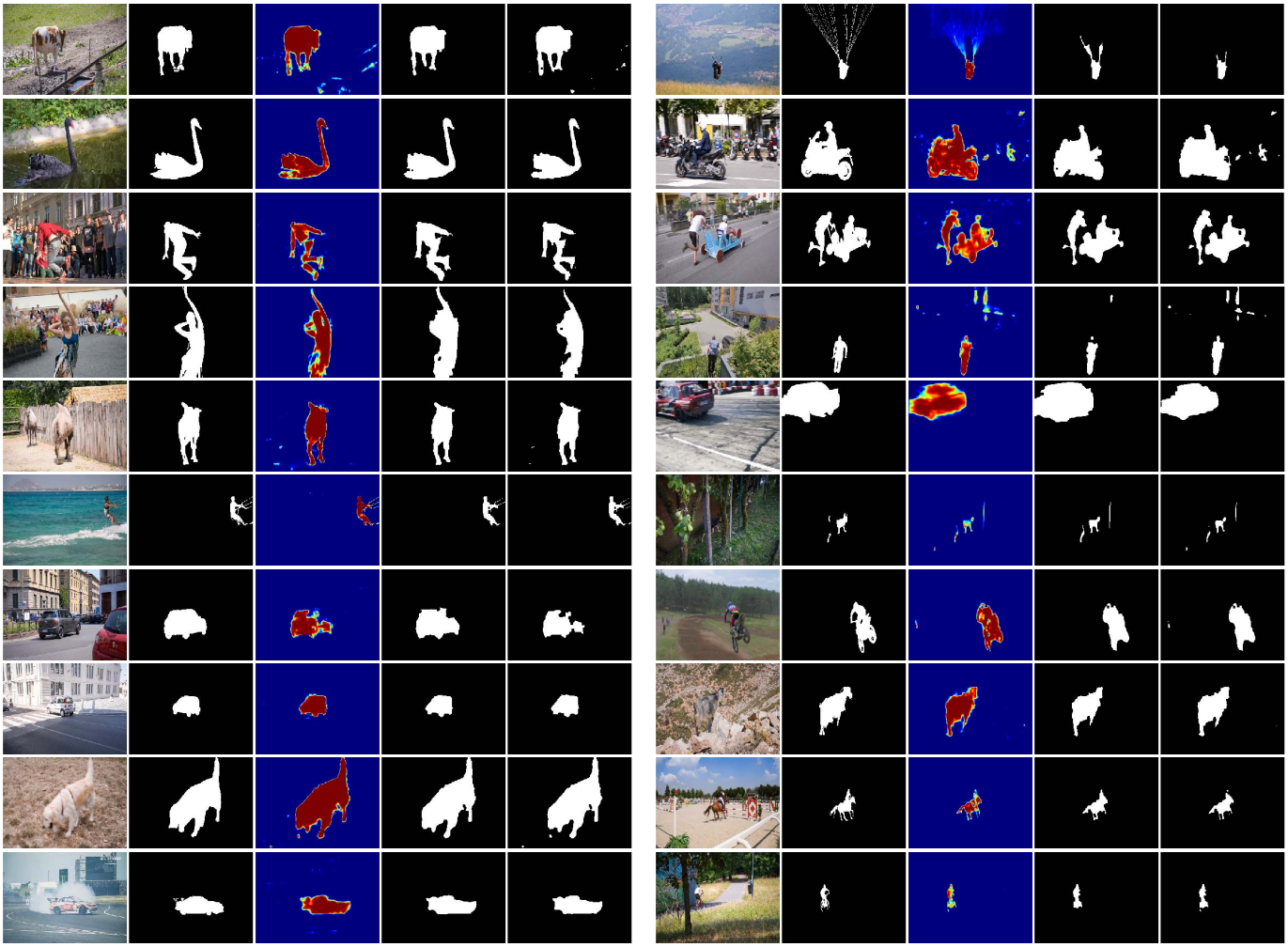
TABLE VI

THE DAVIS-2016 DATASET FINE TUNING PAIRS

Fine-tuned to	Transferred from	Fine-tuned to	Transferred from
Cows	Bear	Paragliding-launch	Paragliding
Blackswan	Flamingo	Scooter-black	Scooter-gray
Breakdance	Breakdance-flare	Soapbox	Parkour
Dance-twirl	Breakdance-flare	Parkour	Rollerblade
Camel	Cows	Drift-straight	Drift-chicane
Kite-surf	Kite-walk	Libby	Dog
Car-roundabout	Car-turn	Goat	Sheep
Car-shadow	Car-roundabout	Horsejump-high	Horsejump-low
Dog	Dog-agility	Horsejump-high	Horsejump-low
Drift-chicane	Drift-turn	Bmx-trees	Bmx-jump

V. LIMITATIONS

Although the proposed model focuses on FG-BG segmentation for static-camera recorded videos, this section conducts a failure analysis of the model with dynamic camera condition. Thus, this analysis is carried out on DAVIS-2016 [41] validation dataset with the same domain specific supervised training method and configuration used for the experiments in Section IV. This experimental study includes the following twenty video sequences, namely Blackswan (BS), Bmx-trees (BT), Breakdance (BD), Camel (CA), Car-roundabout (CR), Car-shadow (CS), Cows (CO), Dance-twirl (DT), Dog (DO), Drift-chicane (DC), Drift-straight (DS), Goat (GO), Horsejump-high (HH), Kite-surf (KS), Libby (LI), Motocross-jump (MJ), Paragliding-launch (PL), Parkour (PA), Scooter-black (SB), and Soapbox (SO). Figure 9 summarizes few visual results and Table VII lists the average performance of the 3D CNN-LSTM on each video sequence. Table VIII compares the mean average performance of the model across all the sequences with existing methods in terms of FoM. All the compared methods also apply domain-specific semi-supervised training strategy. The results of the existing methods are adopted from [69]. Similar to the employed fine-tuning strategy for the experiments on CDnet database, the model's trainable parameters are initialized as given in Table VI.



(a) (b)

Fig. 9. Qualitative results of the proposed 3D CNN-LSTM image to image encoder-decoder FG-BG segmenter on DAVIS 2016 dataset: bright and dark pixels represent FG and BG respectively. (a) Sample results. Row 1 to 10: Cows, Blackswan, Breakdance, Dance-twirl, Camel, Kite-surf, Car-roundabout, Car-shadow, Dog, and Drift-chicane. Column 1 to 5: input frames, ground truths, Sigmoid confidence maps, binary FG-BG segmentations generated through G-th and O-th, respectively. (b) Sample results. Row 1 to 10: Paragliding-launch, Scooter-black, Soapbox, Parkour, Drift-straight, Libby, Motocross-jump, Goat, Horsejump-high, and bmx-trees sequences. Column 1 to 5: input frames, ground truths, Sigmoid confidence maps, binary FG-BG segmentations generated through G-th and O-th, respectively.

TABLE VII

F-MEASURE OF THE 3D CNN-LSTM ON DAVIS-2016 BINARY FG-BG SEGMENTATION: G-TH AND O-TH STAND FOR THE TWO THRESHOLDS (TH.) APPLIED TO TRANSFORM SIGMOID SCORES TO BINARY MASK

Th.	BS	BT	BD	CA	CR	CS	CO	DT	DO	DC	DS	GO	HH	KS	LI	MJ	PL	PA	SB	SO	Ave.
G-th	96.92	82.21	91.82	93.37	91.91	89.23	95.29	88.70	94.24	91.37	88.33	81.02	86.37	88.12	72.79	83.74	61.22	85.25	91.71	90.46	87.20
O-th	94.96	81.86	87.47	92.94	81.50	88.02	93.87	86.45	93.23	91.32	71.68	72.36	85.68	89.17	71.46	83.24	58.52	79.47	87.17	90.29	84.03

TABLE VIII

PERFORMANCE COMPARISON ON DAVIS-2016 IN TERMS OF AVERAGE FOM: G-TH AND O-TH STAND FOR THE TWO THRESHOLDS APPLIED TO TRANSFORM SIGMOID SCORES TO BINARY MASK. VALUES IN ARE THE BEST WHILE THE ONES IN ARE THE SECOND BEST FOM

HVS [60] (2010)	TSP [61] (2013)	SEA [62] (2014)	FCP [41] (2015)	JMP [63] (2015)	OFL [64] (2016)	BVS [65] (2016)	OSOVS [66] (2017)	MSK [67] (2017)	VPN [68] (2017)	MRNN [69] (2017)	Ours G-th	Ours O-th
54.60	31.90	50.40	58.40	57.00	68.00	60.00	79.80	79.70	70.20	80.40	87.20	84.03

The results are encouraging values. However, when compared to the achievements on CDnet video sequences there is a drop of $\approx 7\%$ in overall mean average FoM. On CDnet the proposed model records overall performance of 94.02%, but on DAVIS-2016 it gets 87.20%. Two factors affect the performance of the 3D CNN-LSTM: i. Dynamic camera

motion, and ii. The limited number of available frames (< 100) in each video. Note that, in CDnet each sequence has hundreds of frames.

Hence, the proposed 3D CNN-LSTM has to be retrained for the cases of domain transfer. It is because, the temporal features will be different when the nature of the domain differs, like changes in the frame rate, the motion of the FG objects, motion in the BG, and dynamic cameras. From the extensive experiments, we found that when the model is trained on all the data across all datasets (in Table II), the learned weights can be used as an optimal initial model weights for the new domain. Resulting in, a quicker fine tuning and better performance than train from scratch. The number of samples required in the fine-tuning process is subjected to the demand of the new task.

VI. Conclusion

This work excogitates a DL model for video fore-/background segmentation. Initially, a Conv-LSTM2D image to image encoder-decoder model is implemented. Then, using it as a base, a 3D CNN-LSTM model is achieved by optimizing number of trainable parameters while increasing the depth of the network via micro-autoencoders and slow decoding process with frequent residual feature forwarding. A sanity check is carried out to validate the improved model's performance over the base structure. The 3D CNN-LSTM captures short- and long-short-term spatiotemporal features through 3D convolutions and LSTM units collectively from a set of t frames before predicting the FG-BG segmentation of the current frame. In contrast to the conventional approaches, DL models do not require any feature engineering and manual parameter tuning as the network parameters are learned from exemplar FG-BG segmentations during training. Therefore, it is believed that the proposed 3D CNN-LSTM is a new addition to the state-of-the-art FG-BG segmentation algorithms.

The qualitative and quantitative analysis with sixteen benchmark video sequences demonstrates that the network is performant when dealing with FG-BG separation involving lighting variations, cast shadow, dynamic backgrounds, nighttime in indoor and outdoor environments. The results also show that our model superiorly performs most of the cases when compared with traditional and modern NN-based methods. However, this model lacks capability of handling moving camera scenarios. We leave this for the future work. The proposed 3D CNN-LSTM model is applicable to many computer vision-based intelligent systems not just limited to path segmentation for autonomous vehicles and MRI brain slice partitioning. Finally, it is understood that developing a robust FG-BG segmentation solution is still an intriguing task.

References

[1] L. Yang, J. Li, Y. Luo, Y. Zhao, H. Cheng, and J. Li, "Deep background modeling using fully convolutional network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 254–262, Jan. 2018.

[2] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, "An adaptive background modeling method for foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1109–1121, May 2017.

[3] M. Altun and M. Celenk, "Road scene content analysis for driver assistance and autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3398–3407, Dec. 2017.

[4] F. Chen, H. Yu, R. Hu, and X. Zeng, "Deep learning shape priors for object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1870–1877.

[5] A. Thangarajah, Q. M. J. Wu, and J. Huo, "A unified threshold updating strategy for multivariate Gaussian mixture based moving object detection," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2016, pp. 570–574.

[6] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3173–3181.

[7] L. S. Chow and R. Paramesran, "Review of medical image quality assessment," *Biomed. Signal Process. Control*, vol. 27, no. 1, pp. 145–154, 2016.

[8] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *Int. J. Comput. Vis.*, vol. 118, no. 3, pp. 337–363, Jul. 2016.

[9] A. I. Guha and S. Tellex, "Towards meaningful human-robot collaboration on object placement," in *Proc. RSS Workshop Planning Hum.-Robot Interact., Shared Autonomy Collaborative Robot.*, 2016, pp. 1–18.

[10] T. Huynh-The, O. Banos, S. Lee, B. H. Kang, E.-S. Kim, and T. Le-Tien, "NIC: A robust background extraction algorithm for foreground detection in dynamic scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 7, pp. 1478–1490, Jul. 2017.

[11] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[12] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[13] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WCACV)*, Jan. 2015, pp. 990–997.

[14] P. Tiefenbacher, M. Hofmann, D. Merget, and G. Rigoll, "PID-based regulation of background dynamics for foreground segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3282–3286.

[15] A. Thangarajah, Q. J. Wu, A. Singh, B. Mandon, and A. Chowdhury, "Video foreground detection in non-static background using multi-dimensional color space," *Procedia Comput. Sci.*, vol. 70, pp. 55–61, Dec. 2015.

[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer, 2000, pp. 751–767.

[17] T. Akilan, Q. M. J. Wu, and Y. Yang, "Fusion-based foreground enhancement for background subtraction using multivariate multi-model Gaussian distribution," *Inf. Sci.*, vols. 430–431, pp. 414–431, Mar. 2018.

[18] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 1999, pp. 246–252.

[19] S. Varadarajan, P. Miller, and H. Zhou, "Spatial mixture of gaussians for dynamic background modelling," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2013, pp. 63–68.

[20] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2014, pp. 393–400.

[21] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung, "Or-pca with mrf for robust foreground detection in highly dynamic backgrounds," in *Proc. 12th Asian Conf. Comput. Vis., Part III*. Cham, Switzerland: Springer, 2015, pp. 284–299.

[22] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[23] C. J. Bahr and W. C. Horne, "Subspace-based background subtraction applied to aeroacoustic wind tunnel testing," *Int. J. Aeroacoustics*, vol. 16, nos. 4–5, pp. 299–325, 2017.

[24] G. Han, J. Wang, and X. Cai, "Background subtraction based on modified online robust principal component analysis," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 6, pp. 1839–1852, 2017.

[25] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Universal background subtraction using word consensus models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, Oct. 2016.

[26] L. Zhang, Y. Lu, M. Chen, and W. Zou, "A codebook based background subtraction method for image defects detection," in *Proc. Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2014, pp. 704–706.

[27] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, Aug. 2010.

[28] Y. Zhang, X. Li, Z. Zhang, F. Wu, and L. Zhao, "Deep learning driven blockwise moving object detection with binary scene modeling," *Neurocomputing*, vol. 168, pp. 454–463, Nov. 2015.

[29] Z. Zhao, X. Zhang, and Y. Fang, "Stacked multilayer self-organizing map for background modeling," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2841–2850, Sep. 2015.

[30] G. Gemignani and A. Rozza, "A novel background subtraction approach based on multi layered self-organizing maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 462–466.

[31] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognit.*, vol. 76, pp. 635–649, Apr. 2018.

[32] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3D convolutional neural networks," *Multimedia Tools Appl.*, vol. 77, pp. 23023–23041, Sep. 2018.

[33] T. Akilan, Q. J. Wu, W. Jiang, A. Safaei, and H. Jie, "Double encoding-slow decoding image to image cnn for foreground identification with application towards intelligent transportation," in *Proc. IEEE Inter. Conf. Green Comput. Communicat.*, Aug. 2018, pp. 395–403.

[34] T. M. Nguyen, Q. M. J. Wu, and H. Zhang, "Asymmetric mixture model with simultaneous feature selection and model detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 400–408, Feb. 2015.

[35] W. Zou, C. Bai, K. Kpalma, and J. Ronsin, "Online glocal transfer for automatic figure-ground segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2109–2121, May 2014.

[36] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[37] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation for autonomous driving with deep densely connected MRFS," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 669–677.

[38] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1240–1251, May 2016.

[39] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Inter. Conf. Med. Image Comput. Comput. Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[41] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 724–732.

[42] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 38–43.

[43] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[46] J. A. Pérez-Ortiz, F. A. Gers, D. Eck, and J. Schmidhuber, "Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets," *Neural Netw.*, vol. 16, no. 2, pp. 241–250, 2003.

[47] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2013, pp. 273–278.

[48] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Machi. Learn. (ICML)*, 2006, pp. 369–376.

[49] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Proc. Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Sep. 2014, pp. 285–290.

[50] A. Munawar, and R. Tachibana, "Human-like hand reaching by motion prediction using long short-term memory," in *Proc. Inter. Conf. Social Robot.* Cham, Switzerland: Springer, 2017, pp. 156–166.

[51] M. Wollmer *et al.*, "Online driver distraction detection using long short-term memory," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 574–582, Jun. 2011.

[52] T. Akilan, Q. M. J. Wu, W. Jiang, A. Safaei, and J. Huo, "New trend in video foreground detection using deep learning," in *Proc. IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2018, pp. 889–892.

[53] V. Dumoulin and F. Visin. (2016) "A guide to convolution arithmetic for deep learning." [Online]. Available: https://arxiv.org/abs/1603.07285

[54] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 406–413, Nov. 2016.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[56] T. Akilan, Q. M. J. Wu, A. Safaei, and W. Jiang, "A late fusion approach for harnessing multi-CNN model high-level features," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 566–571.

[57] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[58] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 914–928, Dec. 2017.

[59] H. Sajid and S.-C. S. Cheung, "Universal multimode background subtraction," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3249–3260, Jul. 2017.

[60] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2141–2148.

[61] J. Chang, D. Wei, and J. W. Fisher, III, "A video representation using temporal superpixels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2051–2058.

[62] S. A. Ramakanth and R. V. Babu, "SeamSeg: Video object segmentation using patch seams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 376–383.

[63] Q. Fan, F. Zhong, D. Lischinski, D. Cohen-Or, and B. Chen, "JumpCut: Non-successive mask transfer and interpolation for video cutout," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–195, 2015.

[64] Y.-H. Tsai, M.-H. Yang, and M. J. Black, "Video segmentation via object flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3899–3908.

[65] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung, "Bilateral space video segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 743–751.

[66] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixè, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2017, pp. 221–230.

[67] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2017, pp. 3491–3500.

[68] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 6, Jun. 2017, pp. 451–461.

[69] Y.-T. Hu, J.-B. Huang, and A. Schwing, "MaskRNN: Instance level video object segmentation," in *Proc. Adv. Neural In. Process. Syst.*, 2017, pp. 325–334.

Thangarajah Akilan (S'12) received the Ph.D. degree in electrical and computer engineering from the University of Windsor, Windsor, ON, Canada. He is currently a Post-Doctoral Fellow with the Computer Vision and Sensing Systems Laboratory, University of Windsor. His research interests include object and action recognition, image/video processing and segmentation, and data fusion using statistical techniques, machine learning, and deep learning. He was a Recipient of the 2015–2016 Golden Key's Premier Graduate Scholar Award and the 2013–2014 His Majesty the King's Scholarship offered by the Royal Thai Government. He serves as a Secretary of the IEEE Windsor Section, Canada, and a Reviewer for several journals, including the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.

Qingming Jonathan Wu (M'92–SM'09) received the Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990. He was affiliated with the National Research Council of Canada for ten years beginning in 1995, where he became a Senior Research Officer and a Group Leader. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He is also a Visiting Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He has published more than 300 peer-reviewed papers in computer vision, image processing, intelligent systems, robotics, and integrated microsystems. His current research interests include 3-D computer vision, active video object tracking and extraction, interactive multimedia, sensor analysis and fusion, and visual sensor networks. He holds the Tier 1 Canada Research Chair in Automotive Sensors and Information Systems. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the *Cognitive Computation*. He has served on the technical program committees and international advisory boards for many prestigious conferences.

Amin Safaei (S'08–SM'18) received the M.Sc. degree from the Sharif University of Technology, Tehran, Iran, and the Ph.D. degree in electrical and computer engineering from the University of Windsor, Windsor, ON, Canada. He is currently a Senior Design Engineer with TME, Mississauga, ON, Canada. His current research interests include hardware acceleration for machine vision applications, machine learning, deep learning, and system-on-a-chip design. He was a Recipient of the 2015 and 2016 University of Windsor Graduate Student Society Awards. He serves as a Reviewer for several journals, including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Jie Huo (S'14) received the bachelor's and master's degrees in biomedical engineering from Tianjin University, in 2011 and 2014, respectively. She is currently pursuing the Ph.D. degree in electrical and computer engineering with the Computer Vision and Sensing Systems Laboratory, University of Windsor, Canada. Her research interests include biomedical image processing and segmentation, machine learning, and deep learning. She was a Recipient of the 2016 University of Windsor Graduate Student Society Awards.

Yimin Yang (S'10–M'13) received the Ph.D. degrees in electrical engineering from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2013. From 2014 to 2018, he was a Post-Doctoral Fellow with the University of Windsor, Windsor, ON, Canada. He is currently an Assistant Professor with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada. His research interests include artificial neural networks, signal processing, and robotics. He was a Recipient of the Outstanding Ph.D. Thesis Award of Hunan Province and the Outstanding Ph.D. Thesis Award Nominations of the Chinese Association of Automation, China, in 2014 and 2015, respectively. He has been serving as a reviewer for international journals in his research field, a guest editor of multiple journals, and a program committee member of some international conferences.