2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) | 978-1-6654-9810-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/INISTA55318.2022.9894190

A Fully Convolutional Encoder-Decoder Network for Moving Object Segmentation

Anil Turker
Graduate School, Istanbul Technical University, Istanbul, Turkey
ASELSAN Inc., Ankara, Turkey
anilturker17@gmail.com
Ender M. Eksioglu
Electronics and Communication Engineering Department, Istanbul Technical University, Istanbul, Turkey
eksioglue@itu.edu.tr

***Abstract*—Moving object segmentation (MOS) is one of the important and well-studied computer vision problems. It is used in applications such as video surveillance systems, human tracking, self-driving cars, and video compression. Traditional approaches solve this problem by using hand-crafted features and then modeling the background by using these features. Convolutional Neural Networks (CNNs), on the other hand, have proven to be more powerful than traditional methods in extracting features. In this work, a hybrid system is presented that contains flux tensors together with 3D CNN, enhancing the performance of the algorithm on the unseen videos. 3D CNN can extract spatial and temporal features, thus exploiting motion information between adjacent frames. Motion entropy feature maps extracted by 3D CNN and the output of the flux tensor are jointly fed into an encoder-decoder network. ChangeDetection 2014 dataset is used for both training and test stages. Training and test videos are selected separately, and the networks are tested on unseen videos. Our proposed network gives promising segmentation results, which are competitive with existing methods.**

***Keywords*—Moving object segmentation, flux tensor, deep learning, spatiotemporal, change detection, foreground segmentation, background subtraction**

I. INTRODUCTION

Moving object detection or segmentation is one of the crucial tasks in many systems such as video surveillance systems, human tracking, action recognition, self-driving cars, and video compression. It is the main component in video surveillance systems, and also it is used as a subsystem in applications such as video compression and self-driving cars. Moving object segmentation also known as background subtraction for foreground segmentation in the literature. In traditional approaches, the background is modeled by using historical frames. Then the current frame is compared with the modeled background. As a result of comparing modeled background and the current frame, the algorithm classifies the pixels as background or foreground. Consequently, the pixels of a moving object are defined as foreground, and others are classified as background. The background may contain stationary objects such as walls, buildings, furniture, and roads, or moving objects such as the shaking of moving tree leaves, rain, snow, and waves.

In the traditional approaches, the modeling background is the most crucial part affecting the performance of the algorithm. The biggest shortcoming is the inability to extract sufficient features to model the background. Recently, CNNs are popular for extracting features and are used in many areas such as classification, object detection, and segmentation. A robust background model can be obtained by extracting spatial and temporal features via convolutional filters.

Moving Object Segmentation (MOS) can be considered a binary segmentation problem. U-net [1] network, which is one of the successful approaches for segmentation problems and is used frequently in the literature. In this paper, we preferred to use a hybrid algorithm that contains a flux tensor and 3D CNN. Flux tensor method [2] is an efficient way to extract motion information without using eigenvalue decomposition. Also, the network contains 3D CNN for extracting the spatiotemporal features by using temporal depth in the kernel. Combining the output of flux tensor and motion entropy maps extracted by convolutional filters makes the network more robust for unseen videos. The flux tensor output, the feature map extracted by the 3D CNN network, and the current frame are the inputs of our U-net model, and networks produce a foreground probability map as a result of the sigmoid activation in the last layer of the network. After applying the thresholding to the foreground probability map, the pixels can be classified as foreground or background.

II. RELATED WORKS

There are many efficient and influential studies on moving object detection/segmentation in the literature. Gaussian Mixture Model (GMM) [3] is one of the parametric approaches proposed by C. Stauffer. It is robust to model complex backgrounds such as moving backgrounds and instant illumination changes. Also, it is suitable for real-time applications due to its low computational cost. In this method, each pixel has multiple Gaussian distributions, and the mean and standard deviation of each Gaussian distribution is optimized by using the expectation-maximization algorithm. GMM models the background, then the current frame and modeled background

are compared to classify each pixel as foreground or background. Codebook [4] is another method that is non-parametric one proposed by Kim et al. The pixels are clustered as codewords in the compressed form of the background. Vibe [5] is another the popular non-parametric method. It has a pixel library that contains past pixels in the same location or neighbor ones. The updating mechanism of the Vibe [5] algorithm is different from the other traditional approaches. It updates the pixel library randomly rather than the oldest pixel being replaced first. LBP [6] is another method that exploits neighborhood relations between pixels. It classifies each pixel by using the texture descriptor. Then, in the SubSense algorithm [7], a spatiotemporal descriptor that combines LBSP [8] and a feedback mechanism is used to change the algorithm parameters. There are different variations of the SubSense algorithm [9], [10], and they have provided competitive results in the ChangeDetection 2014(CDNet-2014) [11] dataset. IUTIS-5 [12], which was developed by combining different background subtraction algorithms utilizing generic programming, has reached state-of-the-art performance in the CD2014.

CNNs have been used by researchers to detect or segment moving objects like the other fields in computer vision. Braham and Van Droogen Broeck proposed the method [13] that uses convolutional filters for the background subtraction. In their work, the background and current images are obtained by using the temporal median filter, and then they fed into LeNet-5 [14] network as a form of *N*N* patches. This method is significant for inspiring other researchers about the deep learning methods that can be used in this area. Another popular method is DeepBs [15] which combines traditional and deep learning-based approaches and makes them more robust on unseen videos. It uses the Subsense algorithm [7] as a traditional approach. Then, the current frame and modeled background are the input to the network as a form of *N*N* patches. Also, it has a pixel library that contains the historical values of each pixel. The length of the library changes depends on the motion information generated by the flux tensor algorithm [2]. In contradistinction to DeepBs algorithm [15], Yongqiang Gao and Huayue Cai proposed an end-to-end network [16] that uses 3D convolution to extract both spatial and temporal features. FgSegNet [18] network has state-of-the-art result on the CDNet-2014 dataset. It is a segmentation network consisting of an encoder and decoder network that extracts multi-scale features. It uses 50 or 200 frames for each video for training the network. This algorithm provides a video-specific result for foreground segmentation. Another recent work is the BSUV-Net network [19], which consists of a fully convolutional network and is tested on unseen videos. BSUV-Net [19] is a network that has similar network structure to U-net [1]. The network uses the current frame and two temporal median filters with different numbers for the historical frames. In addition to the three channels, they exploit the segmentation network in their proposed method.

In most of the approaches, the training and test strategy are not announced, which makes it difficult to compare the algorithm results. In addition, the proposed method can be evaluated differently as video-optimized or video-agnostic. In video-optimized approaches, the training and test set is obtained from randomly separated video frames. Even if the datasets are selected from different video frames, they may contain similar images. Therefore, the network's performance on unseen videos is not sufficient. The training and test videos are entirely different in the video-agnostic methods, so the network test on unseen videos. Consequently, video-optimized approaches have an unfair advantage over video-agnostic techniques.

III. THE PROPOSED METHOD

This section gives details about our proposed network MOS-Net consisting of 3D CNN, the flux tensor algorithm, and the encoder-decoder network. The proposed network is shown in Figure 1. In addition, we give the details of our training strategy and loss function.

A. 3D Convolutional Neural Network

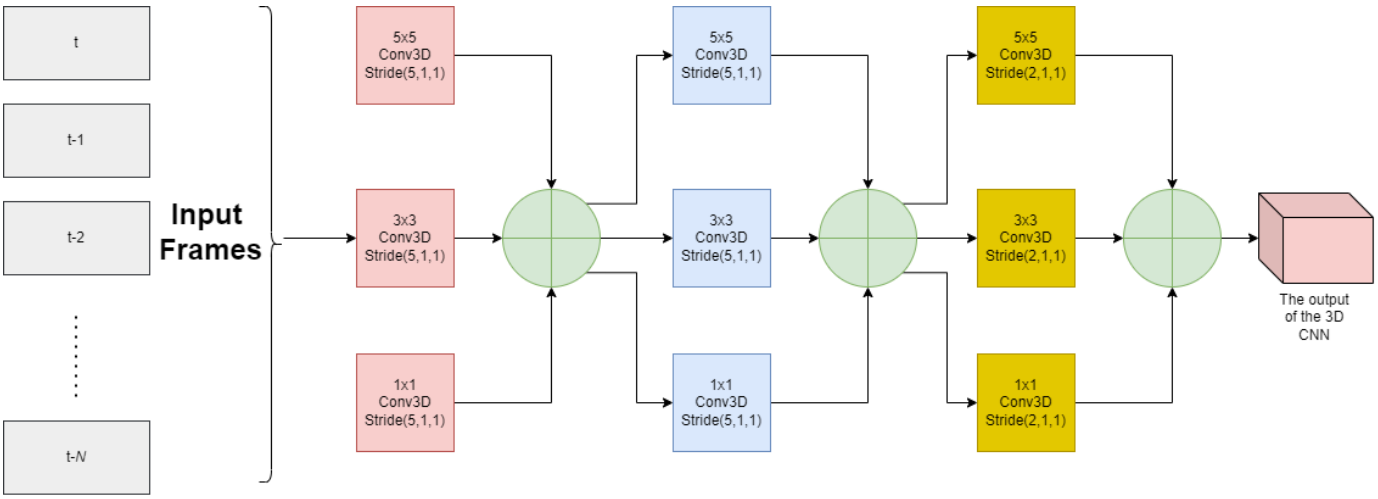
As is known, the 2D convolution operation is popularly used in the extraction of spatial features. 2D convolution layer is computed as:

$$a_{ikp}^{l+1} = RELU(b_i^l + \sum_{k=1}^{x}\sum_{p}^{y} w_{ikp}^l a_{ikp}^l) \tag{1}$$

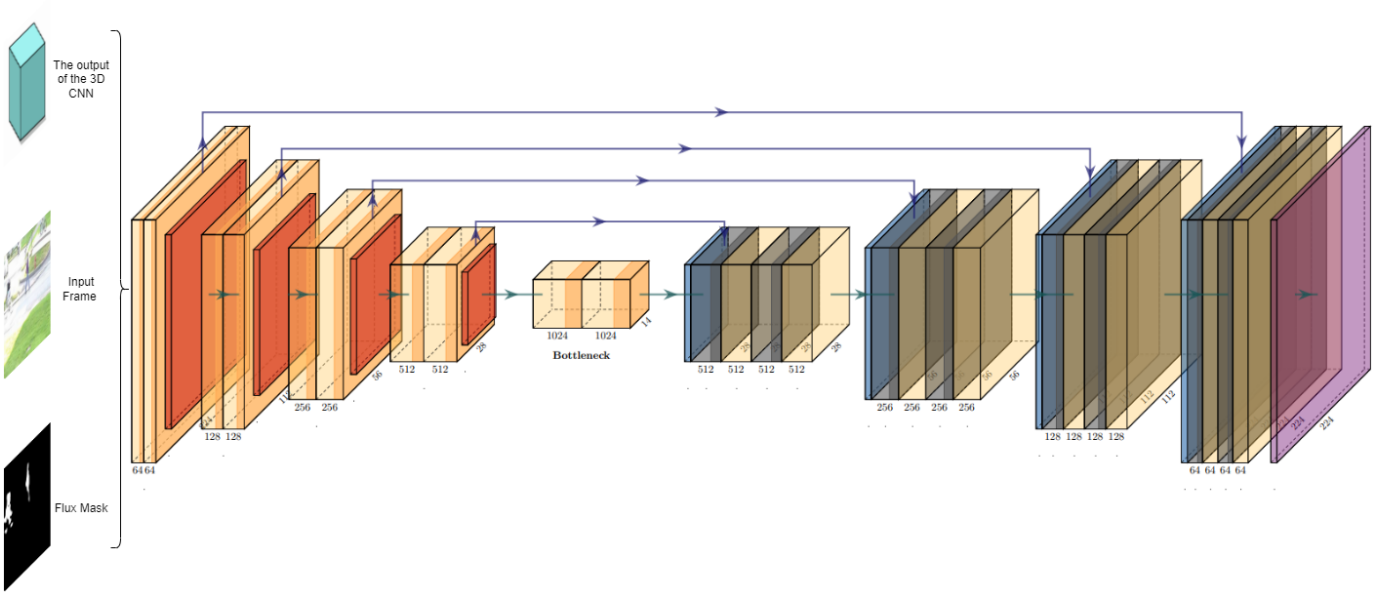
Relu is a frequently used activation function in the convolutional layer. a_{ikp}^{l+1} is the output of i-th input matrix k-th columns of the (l+1)-th layer. w_{ikp} is the weight parameter, and b_i^l bias parameter of the l-th layer of the neural network. As shown in the Eq. (1), the weight parameter only uses spatial information. They don't use the other input matrix. In 3D convolution, the convolution operation is 3-dimensional, and the third dimension is the temporal dimension. 3D convolution layer is computed as:

$$a_{ijkp}^{l+1} = RELU(b_{ij}^l + \sum_{j=1}^{n}\sum_{k=1}^{x}\sum_{p}^{y} w_{ijkp}^l a_{ijkp}^l) \tag{2}$$

Unlike the 2D convolution operation, the j part, which shows the frame number as the third dimension, has been added to the mathematical expression in the Eq. (2). *N* parameter specifies the size of the kernel in the third dimension. Spatial and temporal features can be extracted from the image by using the relationship in neighboring frames by using a 3D convolutional operation. Stride is a parameter in the neural network, and it indicates the movement of the kernel over the image. In the 3D convolution operation, the stride can be used by the kernel both in the spatial and temporal domain. If we define the stride rate as (s, t), the neighboring (s-1) pixels will be assumed as zeros between each convolution operation in the spatial domain. And, (t-1) neighboring input frames will consist of zeros. Figure 2 shows the convolution where the stride rate is (2,2).



(a)



(b)

Fig. 1. (a) 3D CNN that extracts the motion entropy maps from the N historical frames, (b) The proposed network:MOS-Net.

Our proposed 3D CNN is demonstrated in Figure 1a. It extracts the spatiotemporal feature and creates motion-entropy maps which are one of the inputs of encoder-decoder network.

B. The flux tensor algorithm

There are different algorithms used to obtain motion information in the literature. One of the most well-known of these is the Lucas Kanade method [20] for estimation of optical flow. This algorithm makes the optical flow estimation by assuming that the motion in neighboring pixels is similar. The least-square fit method is used for estimating the motion vectors.

In the flux tensor algorithm [2], motion information can be extracted with a lower computational cost compared to other algorithms without using the eigenvalue decomposition.

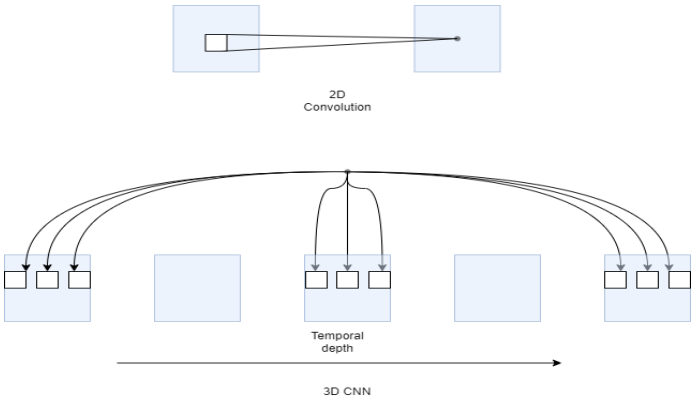


Fig. 2. The Comparison of 2D and 3D convolution operation, stride rate (2,2)

The elements of the flux tensor contain motion information depending on time. The matrix of flux tensor is shown in the Eq. (3). It is possible to distinguish between static and moving objects by using the flux tensor matrix. The trace of the matrix can be used directly to find the moving object in the current frame. The trace of the matrix is shown in the Eq. (4)

$$\mathbf{J} = \begin{bmatrix} \int_\Omega (\frac{d^2I}{dxdt})^2 dy & \int_\Omega \frac{d^2I}{dxdt}\frac{d^2I}{dydt} dy & \int_\Omega \frac{d^2I}{dxdt}\frac{d^2I}{dt^2} dy \\ \int_\Omega \frac{d^2I}{dydt}\frac{d^2I}{dxdt} dy & \int_\Omega (\frac{d^2I}{dydt})^2 dy & \int_\Omega \frac{d^2I}{dydt}\frac{d^2I}{dt^2} dy \\ \int_\Omega \frac{d^2I}{dt^2}\frac{d^2I}{dxdt} dy & \int_\Omega \frac{d^2I}{dt^2}\frac{d^2I}{dydt} dy & \int_\Omega (\frac{d^2I}{dt^2})^2 dy \end{bmatrix} \quad (3)$$

$$\mathbf{Trace}_J = \int_{\Omega)} \left\| \frac{d}{dt}(\nabla I) \right\|^2 dy \quad (4)$$

C. Encoder-Decoder Network

Our proposed encoder-decoder network is shown in Figure 1. The encoder module reduces the spatial resolution by using convolution and max pooling operations. The decoder module upsamples the reduced form to the original resolution. The decoder part maps the low-resolution image containing the features to the original resolution and utilizes the skip connection in the network. The upsampling operation can apply by using transposed convolution in the decoder part.

The first four convolutional layers of the encoder module are the same as the VGG-16 [22] network. All convolution filters have a 3x3 kernel size and max-pooling has a 2x2 kernel size as a stride rate of 2. Batch normalization [21] is used at the end of each convolution layer and standardizes the input for each mini-batch. It stabilizes the network, speeds up the training process, and has a regularization effect.

Motion information obtained by using flux tensor and 3D CNN is input to the encoder-decoder network together with the current frame. The network gives an output of a binary image with the same spatial resolution with the current frame. The binary image that contains the foreground probability map is obtained by using the sigmoid activation function. The foreground probability map consists of pixels value between 0 and 1, and pixels greater than 0.5 are classified as the foreground.

D. Loss Function

Moving object segmentation is fundamentally the classification of each pixel as foreground or background. The number of background pixels on the dataset is much more than the number of foreground pixels. This imbalance problem degrades the performance of the proposed fully convolutional network. The cross-entropy loss function is prevalent in semantic segmentation tasks. When this loss function is used in an imbalanced dataset, the classifier favors the majority classes, and the network will be a biased model. The weighted cross-entropy loss is proposed as a solution to this imbalance problem. The effect of the samples with a minority in the dataset on the loss function has been increased in the weighted cross-entropy loss. The Jaccard index or IoU(intersection over union) is used widely for imbalanced dataset problems in object detection tasks. As used in object detection, this loss function can also be used for semantic segmentation tasks. The Jaccard index is computed as:

$$\mathbf{Jaccard} = \frac{TP}{TP + FP + FN} \quad (5)$$

The Jaccard index measures how sensitive the network finds the foreground pixels. As the network performance increases, this value approaches one, while as the performance decreases, this value approaches 0. It can be defined as a loss function with the following equation:

$$\mathbf{Jaccard} = (1 - JaccardIndex) * \alpha \quad (6)$$

α is a smoothing parameter.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Dataset

The proposed method is trained and tested using the CD2014 [11] dataset, which is frequently used in MOS. In the CD2014 dataset, there are 53 videos, 11 categories in total. This dataset contains challenging videos such as dynamic background, bad weather, and illumination changes. The data division strategy [23] for training and testing videos on the dataset is given in Table I. The frames for training and testing are selected from different videos, one video is in each category is chosen as a testing set, and others are used for training.

B. Evaluation Metric

The proposed method and several existing works in the literature are evaluated by using the training and test dataset given in Table I. We evaluated the performance of our proposed MOS-Net on unseen videos since it is a video-agnostic algorithm. Then, we compared the other methods using the same data division strategy for a fair comparison. The proposed method and other methods are compared with respect to the F1-score. Precision, recall, and F1-score are defined in the following equations:

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP}$$

$$\textit{F1-score} = \frac{2(Recall \times Precision)}{Recall + Precision}$$

C. Training Configuration

We implemented the proposed MOS-Net by using the Pytorch framework and by using Tesla P100-PCIE-16GB GPU with a batch size of 8. Adam was used as the optimization algorithm during the training, and the learning rate started at 0.0001. And then, it has been gradually reduced for every 20 epochs. The network is trained by using randomly selected 200 video frames for each video in Table I.

Fully convolutional networks are independent of the particular spatial resolution of the input. The spatial resolutions of the videos in the CD2014 dataset differ from each other. We used

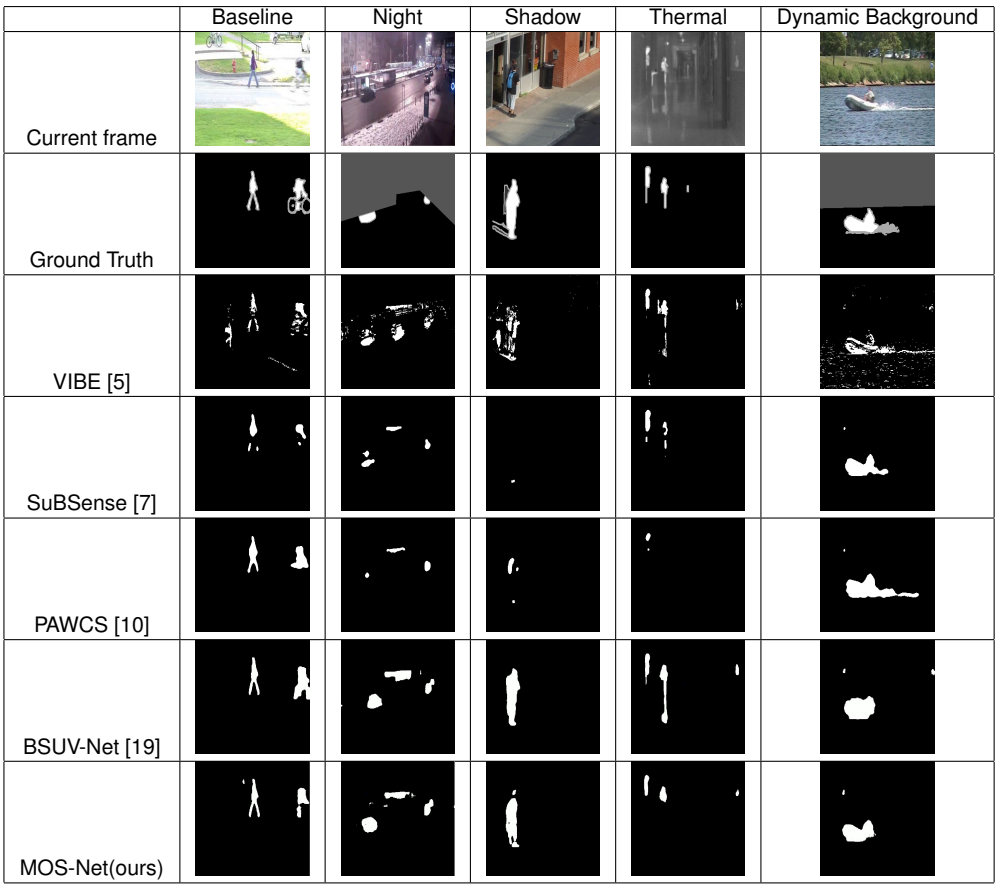
TABLE I
CDNET-2014 DATASET DATA DIVISION

Category	Train Data	Test Data
Baseline	highway, office, PETS2006	pedestrians
badWeather	skating, snowFall, wetSnow	blizzard
cameraJitter	badminton, boulevard, sidewalk	traffic
dynamicBackground	canoe, fall, fountain01, fountain02, overpass	boats
intermittentObjectMotion	abandonedBox, sofa, streetLight, tramstop, winterDriveway	parking
lowFramerate	port_0_17fps, tramCrossroad_1fps, tunnelExit_0_35fps	turnpike_00_05fps
nightVideos	bridgeEntry, busyBoulvard, fluidHighway, streetCornerAtNight, winterStreet	tramStation
shadow	backdoor, bungalows, cubicle, peopleInShade	busStation
thermal	diningRoom, lakeSide, library, park	corridor
turbulence	turbulence1, turbulence2, turbulence3	turbulence0

TABLE II
COMPARISON OF METHOD IN TERMS OF F-MEASURE ON CDNET-2014 DATASET

Method	Baseline	Bad weather	Camera Jitter	Dynamic Background	Int.Obj. motion	Low Framerate	Night	Shadow	Thermal	Turbulence	Overall
ViBe [5]	0.90	0.53	0.66	0.22	0.26	0.60	0.62	0.67	0.75	0.58	0.58
SubSense [7]	0.92	0.81	0.80	0.69	0.48	,0.81	0.76	0.82	0.86	0.79	0.77
PAWCS [10]	0.93	0.66	0.83	**0.88**	0.21	0.91	0.63	0.86	0.65	0.68	0.73
IUTIS-5 [12]	0.96	0.80	0.83	0.75	0.65	0.85	0.75	0.82	0.88	0.63	0.79
DeepBS [15]	0.95	0.61	**0.88**	0.81	0.60	0.49	0.16	**0.94**	**0.89**	0.77	0.71
FgSegNet [18]	0.06	0.12	0.36	0.12	0.46	0.60	0.27	0.27	0.18	0.02	0.22
BSUV-Net [19]	**0.97**	0.88	0.76	0.77	0.59	0.96	**0.82**	0.92	0.87	0.41	0.80
MOS-Net(ours)	0.96	**0.91**	0.65	0.66	**0.75**	**0.97**	0.78	0.84	**0.89**	**0.89**	**0.83**

TABLE III
QUALITATIVE RESULTS OF SEVERAL EXISTING METHODS AND THE PROPOSED ALGORITHM ON UNSEEN VIDEOS



the fixed size input for speeding up the training process. The input can make fixed by resizing operations or cropping fixed-sized patches from the images. Randomly cropped images gives the network an extra augmentation and regularization effect. At the same time, random noise has been added to the input image as for another augmentation technique that makes the network robust for sudden changes. In the inference process, we used the original spatial resolution of the input frame without any scaling or cropping operation.

D. Quantitative Results

The network gives a foreground probability map resulting from the last sigmoid activation function. Then, pixel values greater than 0.5 classify as foreground and others as background. Our proposed method, MOS-Net, outperforms competing methods as in Table II by a 0.83 F1-score on the CDNet-2014 dataset. As shown in Table II, FgSegNet [18] is the state of the art method for CDNet-2014 dataset, its performance is 0.22 in terms of F1-score. It has poor performance on unseen videos. Because, FgSegnet [18] offers a video-specific solution. Another recent work is BSUV-Net [19], its performance is 0.80 in terms of F1-score on the unseen videos as listed in Table I. BSUV-Net [19], the empty and recent temporal median filters are used in the BSUV-Net [19] that suppresses the performance of the network for dynamic scenes. As can be seen from the results, our proposed network, MOS-Net, gives the best results overall among the other competitive algorithms. In Table III, there are visual results obtained in different categories in the CDNet-2014 dataset.

V. Conclusion

We have proposed a network called MOS-Net, which is based on a fully convolutional encoder-decoder network. The network provides a hybrid solution that exploits motion information extracted via the flux tensor algorithm. The flux tensor algorithm is an efficient solution for motion information extraction, and it can be used in real-time tasks. In addition to the flux tensor algorithm, the network also extracts spatiotemporal features with 3D convolutional filters. The motion information obtained from the flux tensor algorithm is preliminary to the encoder-decoder network. It has been observed that MOS-Net gives a better F1-score when compared to both traditional approaches and recent methods like FgSegnet and BSUV-Net. Our main contribution is the fusion of the spatiotemporal features with the features extracted via the flux tensor algorithm. We aim to use a recurrent neural network for extracting temporal information in future works. And we expect that recurrent neural networks like LSTM have more potent for extracting long-term temporal features than 3D CNN.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention, 2015, pp. 234–241.

[2] F. Bunyak, K. Palaniappan, S. K. Nath, and G. Seetharaman, "Flux tensor constrained geodesic active contours with sensor fusion for persistent object tracking," Journal of multimedia, vol. 2, no. 4, p. 20, 2007.

[3] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 2. IEEE, 1999, pp. 246–252.

[4] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," Real-time imaging, vol. 11, no. 3, pp. 172–185, 2005.

[5] O. Barnich and M. Van Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," IEEE Transactions on Image processing, vol. 20, no. 6, pp. 1709–1724, 2010.

[6] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 971–987, 2002.

[7] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 359–373, 2014.

[8] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, "Change detection in feature space using local binary similarity patterns," in 2013 International conference on computer and robot vision, 2013, pp. 106–112.

[9] S. Jiang and X. Lu, "Wesambe: A weight-sample-based method for background subtraction," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2105–2115, 2017.

[10] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in 2015 IEEE winter conference on applications of computer vision. IEEE, 2015, pp. 990–997.

[11] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2014, pp. 387–394.

[12] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," IEEE Transactions on Evolutionary Computation, vol. 21, no. 6, pp. 914–928, 2017.

[13] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in 2016 international conference on systems, signals and image processing (IWSSIP). IEEE, 2016, pp. 1–4.

[14] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

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

[16] Y. Gao, H. Cai, X. Zhang, L. Lan, and Z. Luo, "Background subtraction via 3D convolutional neural networks," in 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 1271–1276.

[17] K. Lim, W.-D. Jang, and C.-S. Kim, "Background subtraction using encoder-decoder structured convolutional neural network," in 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), 2017, pp. 1–6.

[18] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," Pattern Analysis and Applications, pp. 1–12, 2019.

[19] O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2774–2783.

[20] B. D. Lucas, & Kanade. "An Interative Image Registration Technique with an Application to Stereo Vision," DARPA Image Understanding Workshop, pp 121-130, 1981.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning, 2015, pp. 448–456.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", Int. Conf. on Learning Representations, 2015.

[23] M. Mandal, V. Dhar, A. Mishra, and S. K. Vipparthi, "3dfr: A swift 3d feature reductionist framework for scene independent change detection," IEEE Signal Processing Letters, vol. 26, no. 12, pp. 1882– 1886, 2019.