F3DsCNN: A Fast Two-Branch 3D Separable CNN for Moving Object Detection

Bingxin Hou*, Ying Liu*, Nam Ling*, Lingzhi Liu†, Yongxiong Ren†, and Ming Kai Hsu†
*Department of Computer Science and Engineering, Santa Clara University, Santa Clara, USA
†Kwai, Inc., Palo Alto, USA
Emails: {bhou, yliu15, nling}@scu.edu, {l.liu, yongxiongren, mingkaihsu}@kwai.com

2021 International Conference on Visual Communications and Image Processing (VCIP) | 978-1-7281-8551-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/VCIP53242.2021.9675395

***Abstract*—Deep learning methods have been actively applied in moving object detection and achieved great performance. However, many existing models render superior detection accuracy at the cost of high computational complexity and slow inference speed, which hindered the application on mobile and embedded devices with limited computing resources. In this paper, we propose a two-branch 3D separable convolutional neural network named "F3DsCNN" for moving object detection. The network extracts both high-level global features and low-level detailed features. It achieves a fast inference speed of 120 frames per second, suitable for tasks that need to be carried out in a timely manner on a computationally limited platform with high accuracy.**

***Index Terms*—computer vision, convolutional neural network, depthwise convolution, inverted residual, linear bottleneck, moving object detection, pointwise convolution, 3D convolution, 3D separable convolution, two-branch network, unseen videos, video analytics, video surveillance.**

I. Introduction

Moving object detection (MOD) is an essential step of a video processing pipeline which extracts dynamic foreground content from the video frames, while discarding the non-moving background. It plays an important role in many real-world applications [1]–[7]. However, processing a large amount of video data at a fast speed on a resource-limited platform is quite challenging and crucial. In this paper, we propose a fast MOD algorithm called "F3DsCNN" based on a two-branch network architecture and the 3D separable convolution. It is a real-time algorithm tailored for computation-resource-limited and delay-sensitive applications.

The rest of the paper is organized as follows. In Section II, we introduce existing algorithms for moving object detection. In Section III, we elaborate on our proposed network in detail. Section IV describes our experimental results compared to the state-of-the-art models. Section V concludes the paper.

II. Related Works

The methods for MOD problems can be broadly categorized into traditional methods and deep learning methods.

Traditional methods [8]–[14] basically consist of two components: (1) background modeling which initializes the background scene and updates it over time, and (2) classification which classifies each pixel to be foreground or background. There are many background modeling schemes adopted in traditional methods such as temporal adaptive filter [8] and temporal median filter [9], etc. However, it is quite difficult for traditional methods to perform well in complex scenarios.

Deep learning-based methods have been recently proposed for MOD problems. The first convolutional neural netwrok (CNN)-based approaches ConvNet-GT [15] and DeepBS [16] estimate background in traditional ways with temporal filters, and then utilize CNN for classification. Other 2D convolution-based models [17]–[19] use end-to-end neural networks such as FgSegNet_M [18] and FgSegNet_v2 [19], which take each video frame at three different resolution scales in parallel as the input of the encoding network. Besides 2D convolution, 3D convolution is applied to MOD problems to utilize spatial-temporal information in visual data. In [20], 3D CNN and a fully connected layer are adopted in a patch-wise method. 3D-CNN-BGS [21] performs 3D convolution in a multi-scale manner to enhance segmentation accuracy. Recently, generative adversarial networks (GAN) are adopted in MOD problems. BScGAN [22] is based on conditional generative adversarial network (cGAN). BSGAN [23] and BSPVGAN [24] are based on Bayesian GANs.

However, the performance of all the aforementioned deep learning-based methods comes at a slow inference speed due to complex network structures and intense convolution operations. In this paper, we devise a fast two-branch 3D separable CNN that extracts both high-level global features and low-level detailed features for moving object detection in computation-resource-limited and delay-sensitive scenarios with high accuracy.

III. Proposed F3DsCNN Network

The proposed deep moving object detection network shown in Fig. 1 is based on a two-branch structure that captures global context and detailed information. While existing two-branch models [1], [25]–[27] adopt 2D convolutions, we adopt 3D convolution in the proposed two-branch network to explore spatial-temporal information. Further, to reduce complexity and to increase the inference speed, we replace the standard 3D convolution by the 3D separable convolution.

This work was supported in part by Kwai, Inc. under the Kwai, Inc. grant.

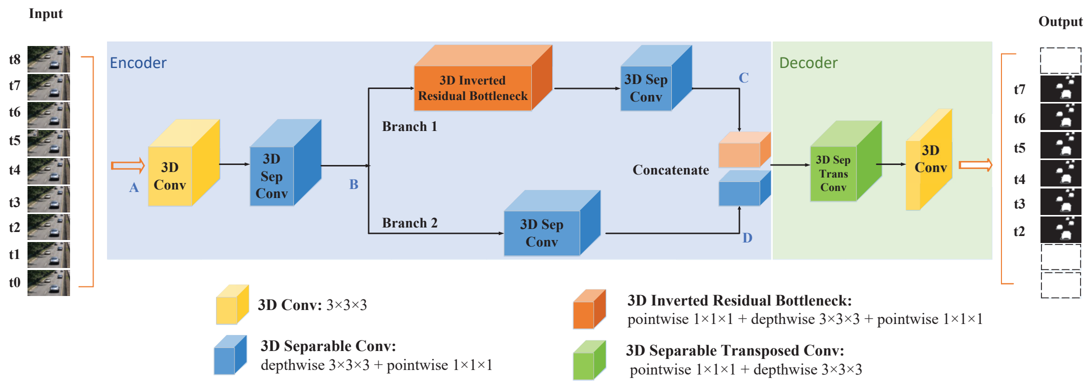

Input
t8
t7
t6
t5
t4
t3
t2
t1
t0
Encoder
A
3D Conv
3D Sep Conv
B
Branch 1
3D Inverted Residual Bottleneck
3D Sep Conv
C
Branch 2
3D Sep Conv
Concatenate
D
Decoder
3D Sep Trans Conv
3D Conv
Output
t7
t6
t5
t4
t3
t2
3D Conv: 3×3×3
3D Separable Conv: depthwise 3×3×3 + pointwise 1×1×1
3D Inverted Residual Bottleneck: pointwise 1×1×1 + depthwise 3×3×3 + pointwise 1×1×1
3D Separable Transposed Conv: pointwise 1×1×1 + depthwise 3×3×3

Fig. 1. The architecture of the proposed network F3DsCNN.

TABLE I
THE CONFIGURATION OF PROPOSED NETWORK F3DSCNN.

Block		Input	Output	Layers
Encoder	3D Conv	3×9×240×320	32×9×240×320	1
	3D SepConv	32×9×240×320	64×5×30×40	9
Branch1	3D InvertedBottleNeck	64×5×30×40	256×2×15×20	24
Branch2	3D SepConv	64×5×30×40	256×2×15×20	8
Decoder	3D SepTransposedConv	512×2×15×20	1×6×120×160	4
	3D Conv	1×6×120×160	1×6×240×320	1

Input/Output data format (CLHW): C: Channel, L: Temporal length, H: Image height, W: Image width.

A. *Design of Proposed Network*

1) Two-branch Network: In order to increase detection accuracy, we design a two-branch network for feature extraction. Traditional two-branch methods [25]–[27] extract global features from low-resolution images with deeper neural network, and extract spatial details from full-resolution images with shallow neural network structures. To reduce computational complexity, the two branches can share the first few layers [1]. Our proposed method is shown in Fig.1. Branch 1 adopts deep inverted residual bottleneck layers to extract global features, and branch 2 adopts a shallower network to extract lower-level features. The two branches share layers from point A to B. The advantage of such layer sharing is that it learns a low-resolution representation from a full-resolution image, which is then used as the input of global feature extraction in branch 1, and simultaneously this low-resolution representation encodes the full-resolution image for detailed spatial feature extraction in branch 2. Finally, these two types of features are concatenated and fed into the decoder.

2) 3D Separable Convolution: While existing two-branch models [1], [25]–[27] adopt 2D convolution, our proposed F3DsCNN adopts 3D convolution in both branch 1 and branch 2 to explore temporal information and to improve detection accuracy. In a 3D convolution layer, the input is 4D and is of size $C \times L \times H \times W$, where C is the number of input channels, L is the temporal length, H and W are the height and width of feature maps. Filters of size $C \times K \times K \times K$ (channel × time × height × width) move in three directions aligning with the temporal length, height, and width axes of the 4D input to output a 4D tensor.

Further, to reduce model size and computational complexity, we propose to separate the aforementioned standard 3D convolution into a 3D depthwise convolution and a 1D pointwise convolution. The 3D depthwise convolution adopts independent filters of size $K \times K \times K$ (time × height × width) to perform a 3D convolution on each of the C input channels. Then, the pointwise convolution adopts filters of size $C \times 1 \times 1 \times 1$ (channel × time × height × width), performs a linear projection along the channel axis, and generates a new representation. Such a factorization can reduce computational complexity by roughly $\frac{1}{K^3}$ times where K is the filter size.

3) Inverted Residual BottleNeck with 3D Separable Convolution: In branch 1, we adopt the inverted residual bottleneck module that was originally proposed in MobileNet-V2 [2]. In an inverted residual bottleneck module [2], the input features with C_l channels are first expanded to a high-dimensional space with $C_h > C_l$ channels using a pointwise convolution. Subsequently, a 2D depthwise convolution with nonlinear activations is performed on each of these C_h channels. Afterwards, another pointwise convolution with linear activatons projects the features back onto a low-dimensional space with C_l channels. The reason for such operations is that it is better to apply nonlinear activations in a high-dimensional space than in a low-dimensional space to prevent information loss. To utilize spatio-temporal information in video data and to increase detection accuracy, we propose to replace such 2D separable convolutions in the inverted residual bottleneck [2] by 3D separable convolutions. The redesigned 3D inverted residual bottleneck first expands the 4D input of size $C_l \times L \times H \times W$ to a high-dimensional space by a pointwise convolution with C_h filters of size $C_l \times 1 \times 1 \times 1$ (C_l is the low-dimensional input channel, C_h is the high-dimensional output channel, $C_h > C_l$). Subsequently, a 3D depthwise convolution with a filter of $3 \times 3 \times 3$ (time × height × width) is performed on each of the C_h channels, and the output is then projected back to the low-dimensional space using another pointwise convolution with C_l filters of size $C_h \times 1 \times 1 \times 1$ with linear activations.

B. Configuration of Proposed Network

We use the format of “CLHW” to represent data, which denotes the number of channels C, the temporal length L, the height of the image H, and the width of the image W.

In Table I, for each training sample, the input to the encoder network is a set of consecutive video frames in a 4D shape of $3 \times 9 \times 240 \times 320$, where 3 is the RGB color channels, 9 is the number of video frames, and 240 and 320 are the height and width of the video frames. In Fig. 1, t_0, t_1, t_2, t_3, t_4... represent different time slots. In the first step, standard 3D convolution is adopted with 32 filters of size $3 \times 3 \times 3 \times 3$ to calculate the convolution on nine input frames. The input video frames are transformed to 32 feature maps in a shape of $32 \times 9 \times 240 \times 320$ at the output. In the following blocks, the feature maps are down-sampled by 9 layers of 3D separable CNN and then separately go through the 24 layers of 8 consecutive 3D inverted residual bottleneck modules in branch 1 for deep global feature extraction, and through 8 layers of 3D separable CNN in branch 2 for shallower feature extraction, and then the outputs of the two branches are concatenated together to be fed into the decoder. In the decoder, we employ 4 layers of 3D separable transposed convolution and 1 layer of standard 3D convolution. A sigmoid activation function is appended at the end to generate the probability masks for 6 successive frames in a shape of $1 \times 6 \times 240 \times 320$.

IV. Experimental Results and Analysis

To analyze how the proposed model performs, we conducted two experiments: (A) Training and evaluation on seen videos of CDnet2014 dataset [28], and (B) Training and evaluation on unseen videos of DAVIS2016 dataset [29]. In Experiment (A), frames in training and test sets were non-overlapped, but from the same video, whereas, in Experiment (B), videos completely unseen in the training set were used for testing.

To evaluate the performance of our proposed model, the inference speed is measured in frames per second (fps), and the detection accuracy is measured by F-measure defined as:

$$F\text{-}measure = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \tag{1}$$

where $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$, given the true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

We used the RMSprop optimizer with binary cross-entropy loss function and trained the model for 30 epochs with batch size 5. The learning rate was initialized at 1×10^{-3} and was reduced by a factor of 10 if the validation loss did not decrease for five successive epochs. Both experiments were carried out on an Intel Xeon with an 8-core 3GHz CPU and an Nvidia Titan RTX 24GB GPU. The number of trainable parameters for the proposed model is 4.34 millions.

A. Training and Evaluation on Seen Videos (CDnet2014)

The CDnet2014 dataset has 11 video categories which include a total of 53 video sequences. In Experiment (A), we excluded the PTZ (pan-tilt-zoom) category since the camera has excessive motion. The proposed model was trained on the first 50% frames in each of the 49 videos, and test on the last 50% frames from the same videos.

All the other nine compared deep learning-based methods such as VGG-PSL-CRF [30], DeepBS [16], BSPVGAN [24], MsEDNet [31], MSCNN+Cascade [32], MSFgNet [33], as well as FgSegNet_S [18], FgSegNet_M [18], and FgSegNet_v2 [19] were trained and tested in the same setup as our proposed model F3DsCNN.

Table II shows the objective performance. Each column lists the inference speed in fps and detection accuracy in F-measure values averaged on test frames from a certain video category, while the last column shows the average F-measure values across all the 10 video categories. We found that our proposed model outperforms the other nine deep-learning methods by 8.3% on average in F-measure and achieves the highest inference speed at 120 fps. Fig. 2 (a)shows the visual subjective performance of our proposed model in Experiment (A) on CDnet2014 dataset. We randomly picked a sample test frame from categories BSL-baseline, LFR-lowFramerate, and NVD-nightVideos. We observe that the proposed F3DsCNN provides more details and clearer edges in the detected foreground objects, such as the car mirrors in “BSL” and the truck in ”LFR”. Moreover, the proposed method detects more accurate and contiguous objects such as the bus in “NVD”. In contrast, the detected binary masks of other methods in comparison have either blurry edges or missing parts.

B. Training and Evaluation on Unseen Videos (DAVIS2016)

To evaluate the generalization ability of the proposed model, we also conducted Experiment (B) on unseen videos of DAVIS2016 dataset. In this experiment, 30 videos in DAVIS2016 dataset were used in training, and 10 completely unseen videos were used for testing. Table III shows the comparison between the proposed model and publicly published deep learning-based methods such as MSK [34], CTN [35], SIAMMASK [36], and PLM [37] from the benchmark DAVIS2016 challenge [38], as well as FgSegNet_S [18], FgSegNet_M [18], and FgSegNet_v2 [19] which were trained and tested in the same setup as our proposed model F3DsCNN.

Table III shows the objective performance. We found that our proposed model offers overwhelmingly faster inference speed at 120 fps. Although the proposed method only offers slightly higher F-measure of 0.8006 than that of MSK [34], its advantage in inference speed at 120 fps is more suitable for mobile and embedded devices. Compared to the remaining existing methods, the proposed method F3DsCNN enhanced the F-measure by 12.6% on average.

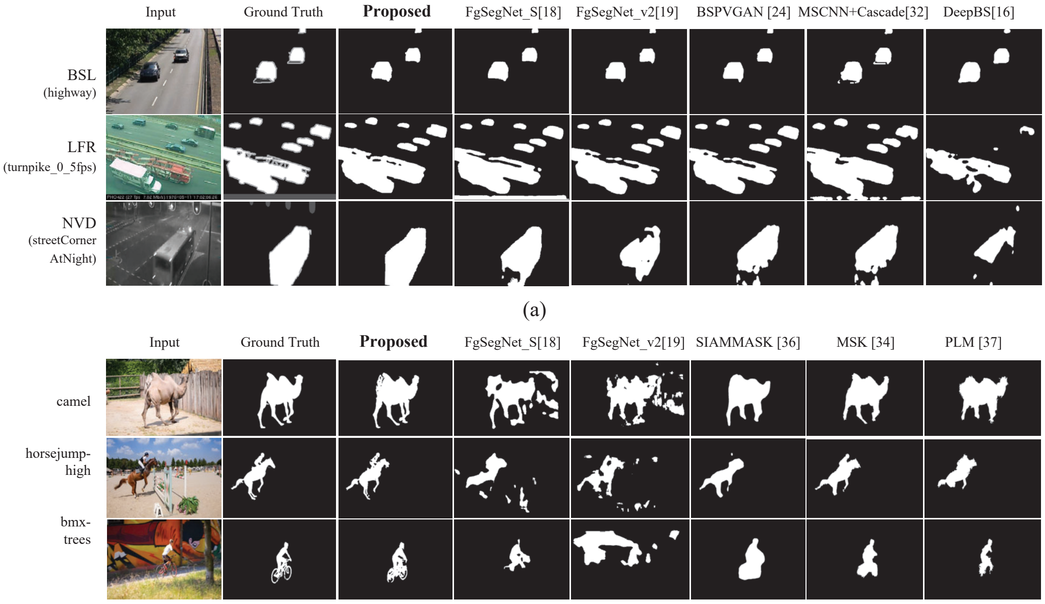
In Fig. 2 (b), we randomly select three videos (camel, horsejump-high, and bmx-trees) for comparison illustration. Our proposed model accurately and clearly detects the shapes and details of objects such as the bike and person in ”bmx-trees”, the camel and the horse in ”camel” and ”horsejump-high”, while the other models hardly detect correct shapes of objects or can only detect blurry, noisy or incomplete objects.

TABLE II
COMPARATIVE F-MEASURE PERFORMANCE FOR **SEEN** VIDEOS ON CDNET2014 DATASET.

Method	**Inference Speed (fps)**	**F-measure**										
		BDW	**BSL**	**CJT**	**DBG**	**IOM**	**NVD**	**LFR**	**SHD**	**THM**	**TBL**	**Avg**
VGG-PSL-CRF [30]	4.9	0.8869	0.9474	0.9276	0.7190	0.7405	0.7398	0.6105	0.8890	0.8352	0.9137	0.8210
DeepBS [16]	10	0.8221	0.9460	0.8844	0.8593	0.5962	0.5777	0.5932	0.9116	0.7389	0.8385	0.7768
BSPVGAN[24]	10	0.9564	0.9717	0.9747	0.9683	0.9230	0.8873	0.8448	**0.9732**	0.9570	0.9240	0.9380
MsEDNet [31]	13.6	0.8975	0.9248	0.9027	0.8902	0.8051	-	-	0.9002	0.8621	-	0.8832
MSCNN+Cascade [32]	50	0.9351	0.9666	0.9612	0.9492	0.8358	0.8837	0.8312	0.9227	0.8764	0.9038	0.9066
FgSegNet_M [18]	69	0.9307	0.9528	0.9403	0.9136	0.8943	0.8830	0.8897	0.9153	0.9160	0.7964	0.9032
FgSegNet_S [18]	82	0.9331	0.9608	0.9407	0.9233	0.9045	0.8871	0.9123	0.9197	0.9152	0.7980	0.9095
MSFgNet [33]	83.8	0.8424	0.9091	0.8167	0.8348	0.7669	0.7973	0.8352	0.9151	0.7822	0.8572	0.8357
FgSegNet_v2 [19]	89	0.9396	0.9680	0.9475	0.9143	0.8985	0.8736	0.9247	0.9152	0.9196	0.8179	0.9119
Proposed F3DsCNN	**120**	**0.9712**	**0.9784**	**0.9755**	**0.9721**	**0.9737**	**0.8878**	**0.9718**	0.9432	**0.9576**	**0.9581**	**0.9589**

TABLE III
COMPARATIVE F-MEASURE PERFORMANCE FOR **UNSEEN** VIDEOS ON DAVIS2016 DATASET.

Method	**Inference Speed (fps)**	**F-measure**										
		camel	**car-roundabout**	**car-shadow**	**cows**	**goat**	**horsejump-high**	**kite-surf**	**bmx-trees**	**parkour**	**soapbox**	**Avg**
MSK [34]	0.5	0.7350	**0.9260**	0.9480	0.8120	0.8140	0.8510	0.4380	0.7360	0.8740	**0.8420**	0.7976
CTN [35]	4.5	0.7250	0.7750	0.8670	0.7750	0.7460	0.8660	0.4600	0.4800	**0.8820**	0.7440	0.7320
PLM [37]	9.5	0.6130	0.7140	0.7310	0.7410	0.6940	0.7860	0.4560	0.6840	0.8120	0.6300	0.6861
SIAMMASK [36]	78	0.7480	0.8720	**0.9780**	0.7720	0.7210	0.6880	0.3260	0.6590	0.8290	0.5470	0.7140
FgSegNet_M [18]	69	0.6047	0.4892	0.8704	0.5620	0.4009	0.6199	0.6308	0.5895	0.5190	0.5835	0.5870
FgSegNet_S [18]	82	0.6163	0.5194	0.8940	0.5356	0.4063	0.6273	0.6904	0.6948	0.5345	0.5902	0.6109
FgSegNet_v2 [19]	89	0.6201	0.5120	0.8744	0.5309	0.4509	0.5940	0.6820	0.5498	0.5029	0.6194	0.5936
Proposed F3DsCNN	**120**	**0.8144**	0.8155	0.8456	**0.8162**	**0.8213**	**0.8721**	**0.7020**	**0.8860**	0.7060	0.7271	**0.8006**



(a)

(b)

Fig. 2. Visual comparison results (a) Experiment (A) on seen sample of CDnet2014 dataset, (b) Experiment (B) on unseen samples of DAVIS2016 dataset.

V. CONCLUSION

In this paper, we propose the F3DsCNN model for moving object detection. Our model is designed specifically for environments with limited computing resources and for delay-sensitive tasks. Our model increases detection accuracy by utilizing the spatial-temporal information in the video data via 3D convolution, and also by feature fusion via the two-branch structure. Our model improves the efficiency of the model via 3D separable convolution and 3D inverted residual bottleneck module. Moreover, the two experiments conducted on seen videos and unseen videos demonstrate that our proposed model achieves superior detection accuracy among all compared models with high inference speeds suitable for low-latency vision applications. In terms of future study, we plan to use data-augmentation techniques to improve detection accuracy, and we plan to extend the work to object-aware moving object detection tasks to discriminate different moving objects.

References

[1] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," in *30th British Machine Vision Conference (BMVC)*, 2019, p. 289.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 4510–4520.

[3] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: arXiv:1704.04861

[4] A. Ignatov, R. Timofte, S. Ko, S. Kim, K. Uhm, S. Ji, S. Cho, J. Hong, K. Mei, J. Li, J. Zhang, H. Wu, J. Li, R. Huang, M. Haris, G. Shakhnarovich, N. Ukita, Y. Zhao, L. Po, T. Zhang, Z. Liao, X. Shi, Y. Zhang, W. Ou, P. Xian, J. Xiong, C. Zhou, W. Y. Yu, Y. Yubin, B. Hou, B. Park, S. Yu, S. Kim, and J. Jeong, "AIM 2019 challenge on raw to rgb mapping: Methods and results," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3584–3590.

[5] A. Ignatov, R. Timofte, Z. Zhang, M. Liu, H. Wang, W. Zuo, J. Zhang, R. Zhang, Z. Peng, S. Ren, L. Dai, X. Liu, C. Li, J. Chen, Y. Ito, B. Vasudeva, P. Deora, U. Pal, Z. Guo, Y. Zhu, T. Liang, C. Li, C. Leng, Z. Pan, B. Li, B.-H. Kim, J. Song, J. C. Ye, J. Baek, M. Zhussip, Y. Koishekenov, H. C. Ye, X. Liu, X. Hu, J. Jiang, J. Gu, K. Li, P. Tan, and B. Hou, "AIM 2020 challenge on learned image signal processing pipeline," 2020. [Online]. Available: arXiv:2011.04994

[6] R. Berns and B. Hou, "RIT-DuPont supra-threshold color-tolerance individual color-difference pair dataset," *Color Research and Application*, vol. 35, pp. 274–283, 2009.

[7] B. Hou, "Extending the RIT-DuPont suprathreshold data set: Weighted individual discrimination pair data and new chroma dependency visual data," Master's thesis, Rochester Institute of Technology, 2010.

[8] Y. Zheng and L. Fan, "Moving object detection based on running average background and temporal difference," in *IEEE International Conference on Intelligent Systems and Knowledge Engineering*, 2010, pp. 270–272.

[9] Q. Zhou and J. Aggarwal, "Tracking and classifying moving objects using single or multiple cameras," in *Handbook of Pattern Recognition and Computer Vision*, Jan 2005, pp. 499–524.

[10] 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.

[11] P. St-Charles, G. 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, 2015.

[12] Y. Liu and D. A. Pados, "Compressed-sensed-domain L1-PCA video surveillance," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 351–363, 2016.

[13] Y. Liu, Z. Bellay, P. Bradsky, G. Chandler, and B. Craig, "Edge-to-fog computing for color-assisted moving object detection," in *Big Data: Learning, Analytics, and Applications*, F. Ahmad, Ed., vol. 10989, International Society for Optics and Photonics. SPIE, 2019, pp. 9–17.

[14] Y. Pei, Y. Liu, N. Ling, L. Liu, and Y. Ren, "Class-specific neural network for video compressed sensing," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[15] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2016, pp. 1–4.

[16] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for background subtraction," *Pattern Recognition*, Sep 2017.

[17] B. Hou, Y. Liu, and N. Ling, "A super-fast deep network for moving object detection," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[18] L. A. Lim and H. Yalim Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognition Letters*, vol. 112, pp. 256–262, 2018.

[19] L. A. Lim and H. Yalim Keles, "Learning multi-scale features for foreground segmentation," *Pattern Analysis and Applications*, vol. 23, no. 3, pp. 1369–1380, Aug 2019.

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

[21] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3D convolutional neural networks," *Multimedia Tools and Applications*, vol. 77, pp. 23 023–23 041, 2017.

[22] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "BScGAN: Deep background subtraction with conditional generative adversarial networks," in *25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 4018–4022.

[23] W. Zheng and K. Wang, "Background subtraction algorithm with bayesian generative adversarial networks," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 44, 05 2018.

[24] W. Zheng, K. Wang, and F.-Y. Wang, "A novel background subtraction algorithm based on parallel vision and Bayesian GANs," *Neurocomputing*, vol. 394, pp. 178–200, 2020.

[25] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[26] R. P. K. Poudel, U. Bonde, S. Liwicki, and C. Zach, "ContextNet: Exploring context and detail for semantic segmentation in real-time," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, 2018, p. 146.

[27] D. Mazzini, "Guided upsampling network for real-time semantic segmentation," in *British Machine Vision Conference (BMVC)*, 2018, p. 117.

[28] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 393–400.

[29] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 724–732.

[30] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise deep sequence learning for moving object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2567–2579, 2019.

[31] P. W. Patil, S. Murala, A. Dhall, and S. Chaudhary, "MsEDNet: Multiscale deep saliency learning for moving object detection," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 1670–1675.

[32] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognition Letters*, vol. 96, pp. 66–75, 2017.

[33] P. W. Patil and S. Murala, "MSFgNet: A novel compact end-to-end deep network for moving object detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4066–4077, 2019.

[34] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, "Learning video object segmentation from static images," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3491–3500.

[35] W. Jang and C. Kim, "Online video object segmentation via convolutional trident network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7474–7483.

[36] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1328–1338.

[37] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. Kweon, "Pixel-level matching for video object segmentation using convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 10 2017.

[38] Benchmark video object segmentation on DAVIS2016. [Online]. Available: https://davischallenge.org/davis2016/soa_compare.html