

Motivation behind the proposed algorithm

In the last few years, convolutional neural networks have been a well-established technique for background subtraction and used in many real-time applications. These networks can extract and learn features at different levels rather than hand-crafted features. In particular, the convolutional networks that are utilized transfer learning have shown substantial improvement over conventional techniques by large margins. Thus, we explore a VGG-19 deep network with a transfer learning strategy as an encoder that deeply learns and extracts useful features at low, mid, and high levels. In the proposed encoder, to keep the contextual details of the challenging scene, we have utilized the initial four blocks of the VGG-19 network and removed the max-pooling layer between the third and fourth blocks. Also, to reduce overfitting, the fourth block of the VGG-19 network is fine-tuned by inserting dropout regularisation after every convolutional layer with a rate of 0.5. The proposed Multi-scale Feature Pulling (MFP) block extracts features at multi-scales by operating a max-pooling layer and various atrous convolutional layers with a sampling rate of 4, 8, and 16. In this work, we have designed a decoder network consisting of stacked transposed convolutional layers which efficiently predict that each pixel of the target frame belongs to the background or foreground.

Results analysis

To check the effectiveness of the proposed scheme, we have compared the results obtained by it with seventeen existing background subtraction techniques: six non-deep-learning and eleven deep learning techniques. Due to page constraints, the visual assessment of the proposed model is compared with four competitive state-of-the-art techniques: DeepBS, BSPVGAN, WisenetMD, and BSUVNet2.0. All the original frames and the corresponding ground-truth images of the five challenging sequences are shown in Fig. 1 (a) and (b). The results obtained by the DeepBS technique, as shown in Fig. 1 (c), depict that many of the edge pixels are missing due to imbalanced pixel values in various video frames. Thus, the DeepBS technique provides many missed alarms in the change detection results. Fig. 1 (d) illustrates the segmented foreground results obtained by the BSPVGAN technique where many false alarms appear in the scene. The object detection results attained by the WisenetMD technique are given in Fig. 1 (e), where it can be perceived that a few details of the moving objects are missing. Fig. 1 (f) represents the BSUVNet2.0 results where the background is identified as the foreground. From Fig. 1 (g), it is observed that the results obtained by the proposed scheme can handle the challenging scene effectively and produce results with lesser isolated points against the existing techniques.

Also, for the objective assessment of the proposed scheme, we have utilized three evaluation measures: average precision, average recall, and average F-measure, and results obtained by the proposed scheme compared with seventeen state-of-the-art techniques. From Table 1, it is found the proposed scheme attained higher accuracy in all considered measures on the CDNet-2014 database than existing techniques.

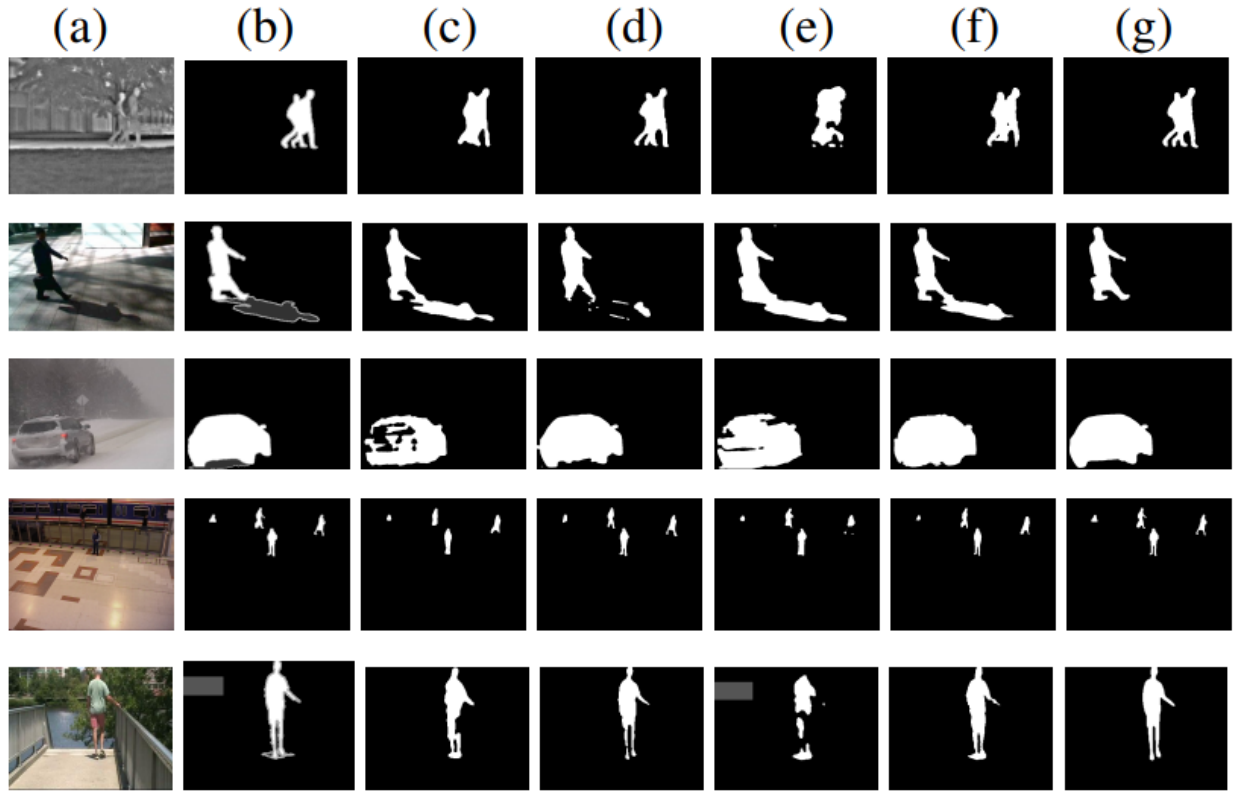


Fig. 1: Visual Results: (a) original frame (b) ground-truth, results obtained by (c) DeepBS, (d) BSPVGAN , (e) WisenetMD, (f) BSUVNet2.0, and (g) MFP (Proposed).

Table 1: Quantitative comparison with all the sequences of CDNet-2014 database

	Approaches	Avg.Precision	Avg.Recall	Avg.F-measure
Non-deep learning	KDE	0.5811	0.7375	0.5688
	GMM	0.6025	0.6846	0.5707
	PAWCS	0.7857	0.7718	0.7403
	SuBSENSE	0.7509	0.8124	0.7408
	Kernelized Fuzzy	0.8912	0.8672	0.8792
	Possibilistic Fuzzy	0.9322	0.8929	0. 9121
Deep learning	DeepBS	0.8332	0.7545	0.7458
	BSPVGAN	0.9472	0.9544	0.9501
	WisenetMD	0.7668	0.8179	0.7535

	Cascade CNN	0.8997	0.9506	0.9209
	IUTIS-5	0.8087	0.7849	0.7717
	BSUV-Net	0.8113	0.8203	0.7868
	SemanticBGS	0.8305	0.7890	0.7892
	BSUV-Net 2.0	0.9011	0.8136	0.8387
	BMN-BSN	0.7032	0.8250	0.7188
	MU-Net2	0.9407	0.9454	0.9369
	DeepSphere	0.9512	0.7795	0.9158
	MFP (Proposed)	0.9696	0.9612	0.9643

Ablation study

In this work, we have conducted an ablation study to choose the most appropriate deep learning network for the proposed background subtraction algorithm. From Table 2, it may be observed that the use of existing deep neural networks such as VGG-16, GoogLeNet, and ResNet-50 for local change detection cannot accurately preserve the fine and coarse-scale features. Thus, these techniques produce a lesser value of the average F-measure for the challenging CDNet-2014 dataset. So, in the proposed scheme, a VGG-19 network is adhered to as an encoder that provides better accuracy than the existing deep neural networks.

Table2: Ablation study of average F-measure comparison on CDNet-2014 with different Encoder

Category	VGG-16	GoogLeNet	ResNet-50	Proposed
BadWeather	0.9594	0.8557	0.9294	0.9905
Baseline	0.8949	0.7961	0.9461	0.9981
Camera Jitter	0.9422	0.8864	0.9518	0.9970
Dynamic Background	0.7356	0.6588	0.8220	0.9962
Intermittent Object Motion	0.7538	0.6488	0.8453	0.9926
Low Framerate	0.6175	0.5947	0.8080	0.7480

Night Videos	0.7526	0.6003	0.8585	0.9852
PTZ	0.7816	0.7136	0.7776	0.9924
Shadow	0.9084	0.8049	0.9647	0.9971
Thermal	0.8546	0.7725	0.9444	0.9954
Turbulence	0.9207	0.7637	0.8011	0.9148
Overall	0.8292	0.7360	0.8772	0.9643

Also, we have conducted an ablation study to check the effectiveness of the proposed algorithm with the decoder network consisting of stacked transposed convolutional layers and without transposed convolutional layers. From Table 3, it may be observed that the proposed model with a decoder network composed of stacked transposed convolutional layers attained better accuracy and effectively projected the feature level into pixel level than without transposed convolutional layers.

Table3: Ablation study of average F-measure comparison on CDNet-2014 with different Decoder

Category	Proposed model with decoder network without transposed convolutional layers	Proposed model with decoder network consisting of transposed convolutional layers
BadWeather	0.9781	0.9905
Baseline	0.9927	0.9981
Camera Jitter	0.9785	0.9970
Dynamic Background	0.9796	0.9962
Intermittent Object Motion	0.9727	0.9926
Low Framerate	0.7698	0.7480
Night Videos	0.9340	0.9852
PTZ	0.9495	0.9924
Shadow	0.9866	0.9971
Thermal	0.9689	0.9954
Turbulence	0.9597	0.9148
Overall	0.9518	0.9643

Training and Testing time for the proposed algorithm

In Table 4, we have reported the training and testing time of the proposed algorithm on the CDNet-2014 database for the image size of 240 x 320, 480 x 720, and 576 x 720.

Table4: Computational time for the proposed scheme on CDNet-2014 database

Database	Framesize	Training time (per epoch) in seconds	Testing time in seconds
CDNet-2014	240 x 320	22	0.04
	480 x 720	91	0.17
	576 x 720	109	0.21