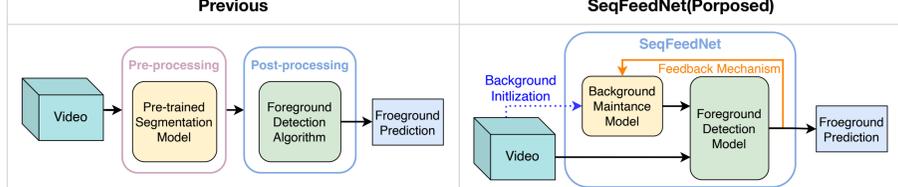


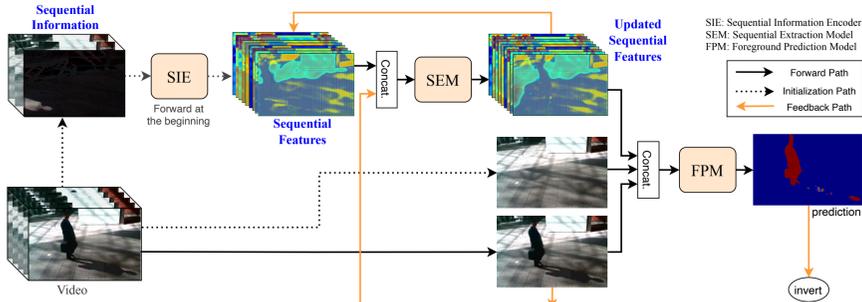
SeqFeedNet: Sequential Feature Feedback Network for Background Subtraction

Abstract

Background subtraction (BGS) is a fundamental task in computer vision with applications in video surveillance, object tracking, and people recognition. Despite recent advancements, many deep learning-based BGS algorithms rely on large models to extract high-level representations, demanding significant computational resources and leading to inefficiencies in processing video streams. To address these limitations, we introduce the Sequential Feature Feedback Network (SeqFeedNet), a novel supervised algorithm for BGS in unseen videos that operates without additional pre-processing models. SeqFeedNet innovatively incorporates time-scale diverse sequential features and employs a feedback mechanism for each iteration. Moreover, we propose the Sequential Fit Training (SeqFIT) technique, enhancing model convergence during training. Evaluated on the CDNet 2014 dataset, SeqFeedNet not only achieves ~5 times increase in processing speed but also having better F-Measure scores than the leading supervised algorithms, making it highly suitable for real-world applications. Our experiment demonstrates that SeqFeedNet surpasses state-of-the-art network without pre-trained segmentation model by **3.83%** F-Measure on the CDnet 2014 dataset. Leading the way to establish a new benchmark for efficient and effective BGS on unseen videos.



- The framework of SeqFeedNet.



Results

- Comparison of methods according to the per-category F-Measure for unseen videos from CDnet 2014.

Method	baseline	camjitt	dynbg	intmot	shadow	thermal	badwea	lowfr	night	PTZ	turbul	Overall
With pre-trained segmentation model												
<i>Unsupervised algorithms</i>												
SWCD	0.9214	0.7411	0.8645	0.7092	0.8779	0.8581	0.8233	0.7374	0.5807	0.4545	0.7735	0.7583
SemanticBGS	0.9604	0.8388	0.9489	0.7878	0.9478	0.8219	0.8260	0.7888	0.5014	0.5673	0.6921	0.7892
RTSS	0.9597	0.8396	0.9325	0.7864	0.9551	0.8510	0.8662	0.6771	0.5295	0.5489	0.7630	0.7917
RT-SBS-v2	0.9535	0.8233	0.9217	0.8946	0.9497	0.8697	0.8279	0.7341	0.5629	0.5808	0.7315	0.8045
ZBS	0.9653	0.9545	0.9290	0.8758	0.9765	0.8698	0.9229	0.7433	0.6800	0.8133	0.6358	0.8515
<i>Supervised algorithms</i>												
BSUV-Net	0.9693	0.7743	0.7967	0.7499	0.9233	0.8581	0.8713	0.6797	0.6987	0.6282	0.7051	0.7868
BSUV-Net 2.0	0.9620	0.9004	0.9057	0.8263	0.9562	0.8932	0.8844	0.7902	0.5857	0.7037	0.8174	0.8387
Without pre-trained segmentation model												
<i>Unsupervised algorithms</i>												
PAWCS	0.9397	0.8137	0.8938	0.7764	0.8913	0.8324	0.8152	0.6588	0.4152	0.4615	0.6450	0.7403
SuBSENSE	0.9503	0.8152	0.8177	0.6569	0.8986	0.8171	0.8619	0.6445	0.5599	0.3476	0.7792	0.7408
WisenetMD	0.9487	0.8228	0.8376	0.7264	0.8984	0.8152	0.8616	0.6404	0.5701	0.3367	0.8304	0.7535
<i>Supervised algorithms</i>												
FgSegNetv2(unseen video)	0.6926	0.4266	0.3634	0.2002	0.5295	0.6038	0.3277	0.2482	0.2800	0.3503	0.0643	0.3715
STPNet	0.9587	0.7721	0.8058	0.8267	0.9114	0.8688	0.8898	0.7297	0.6961	0.6076	0.7248	0.7992
Fast BSUV-Net 2.0	0.9694	0.8828	0.7320	0.9016	0.8890	0.8379	0.8909	0.7824	0.6551	0.5014	0.8379	0.8039
SeqFeedNet(Ours)	0.9747	0.9080	0.7743	0.8455	0.9380	0.8977	0.8437	0.7615	0.7487	0.6989	0.8737	0.8422

Without the pre-trained segmentation model, SeqFeedNet achieves the best BGS on unseen videos.

- Qualitative practical demand comparison for BGS.

Method	F_1	Arbitrary Size Image	Parameter Size (MParams/iter.)	Inference Speed	Effectiveness on Unseen Videos ($F_1 > 0.8$)
FgSegNetv2	0.9770	Yes	-	fast	No
FgSegNetv2 (unseen video)	0.3715	Yes	-	fast	No
ZBS	0.8515	No	175.5	slow	Yes
BSUV-Net 2.0	0.8387	Yes	230.0	slow	Yes
Fast BSUV-Net 2.0	0.8039	Yes	30.4	fast	Yes
SeqFeedNet(Ours)	0.8422	Yes	60.8	fast	Yes

SeqFeedNet aims to achieve all the key factors for the practical application and deployment.

- FPS is calculated using PyTorch 2.1 implementation on a single Nvidia GeForce RTX 3090 GPU.

Method	FPS			
	Re	Pr	F_1	320x240 640x480
BSUV-Net 2.0 (Median on GPU)	0.8136	0.9011	0.8387	~11.8 ~3.2
Fast BSUV-Net 2.0 (Median on GPU)	0.8181	0.8425	0.8039	~56.8 ~15.4
SeqFeedNet	0.8324	0.8782	0.8422	~57.8 ~16.0

Note: ZBS was not included in the speed tests due to its higher setup costs, although it reportedly uses C++ language and reaches about 20 FPS on an A100 GPU according to its publication.

SeqFeedNet is ~5 times faster than the best supervised BGS algorithm.

Overall, SeqFeedNet become a **leading BGS algorithm for real-world application scenarios.**

Performance

Follow the link to download the evaluation models mention on paper:

https://drive.google.com/drive/folders/1GjFqnQp7vxh_96-moHWdUFEdTrITvh2?usp=sharing

- Cross-validation of overall and per-category results of SeqFeedNet on the CDnet 2014 dataset.

Model	baseline	cameraJitter	badWeather	dynamicBackground	intermittentObjectMotion	lov
cv1	0.9719	0.8808	0.8846	0.9353	0.9111	0.1
cv2	0.9685	0.8874	0.9432	0.3776	0.8228	0.9
cv3	0.9881	0.9129	0.7645	0.9674	0.9004	0.9
cv4	0.9701	0.9509	0.7823	0.8166	0.7478	0.9
Overall	0.9747	0.9080	0.8437	0.7743	0.8455	0.7

- Ablation

Frame Group G	Seq. Loop	Re	Pr	F_1
1	0	0.0734	0.0067	0.0078
2	1	0.6850	0.8507	0.7338
5	4	0.8182	0.8791	0.8351
7	6	0.8324	0.8782	0.8422

The findings reveal that increasing the number of sequential loops by including additional frame groups significantly boosts the performance of SeqFeedNet.

Installation

Test on python 3.10, cuda 12.1, and torch 2.1.1

```
pip3 install -r requirements.txt
pip3 install torch torchvision
```

Dataset

- Download the CDnet 2014 Dataset: <http://jacarini.dinf.usherbrooke.ca/static/dataset/dataset2014.zip>
- Download the model trained using both an empty background and a recent background as provided by M. Ozan Tezcan et al. at the following link: https://drive.google.com/drive/folders/1fskxV1paCsoZvqTVLjnAdPOCHk1_XmF
- Prepare the folder structure as follow:

```
Data
├── currentFr -> cdnet2014/dataset/
├── emptyBg -> BSUV-Net-2.0_Training_Data/emptyBg
└── recentBg -> BSUV-Net-2.0_Training_Data/recentBg
```

Training

```
cross_validation_set=<number_of_set>
python3 training.py --device 0 --epochs 200 --batch_size 9 --workers 9 --cv $cross_validation_set --imghw
```

The model's weight will be saved to the `out/` directory.

Testing

```
cross_validation_set=<number_of_set>

# generate measurement csv
python3 testing.py -cv $cross_validation_set --device 0 --weight <model_weight>
# generate measurement csv & video result
python3 testing.py -cv $cross_validation_set --device 0 --weight <model_weight> --save
```

Demo

The order of the first row, from left to right, is as follows: input, label, prediction mask(threshold=0.5), and prediction probability.

The second row displays the sequential features' visual results.

- PTZ/intermittentPan

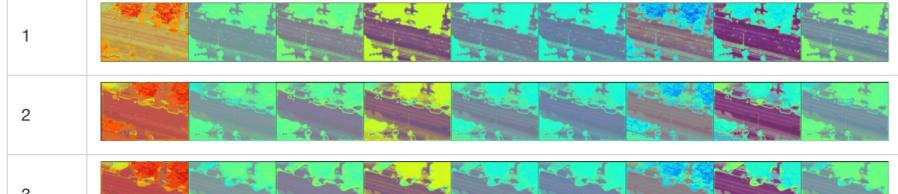
link: <https://drive.google.com/file/d/12-dvboDZkGfXo4dM-YxM1wBMWJJCwCb1>



From Frame 1 (initialize) to Frame 3, it is obvious that the sequential features can be adapted quickly.

- shadow/peopleInShade

link: https://drive.google.com/file/d/1WbR_CjdeGmQn2NpPwPVL57v_eL2RmEK1/view?usp=drive_link



The result shows that the sequential features can divide the shadow shape to prevent misjudgment of the shadow as foreground.