Deep Background Subtraction with Scene-Specific Convolutional Neural Networks

M. Braham and M. Van Droogenbroeck

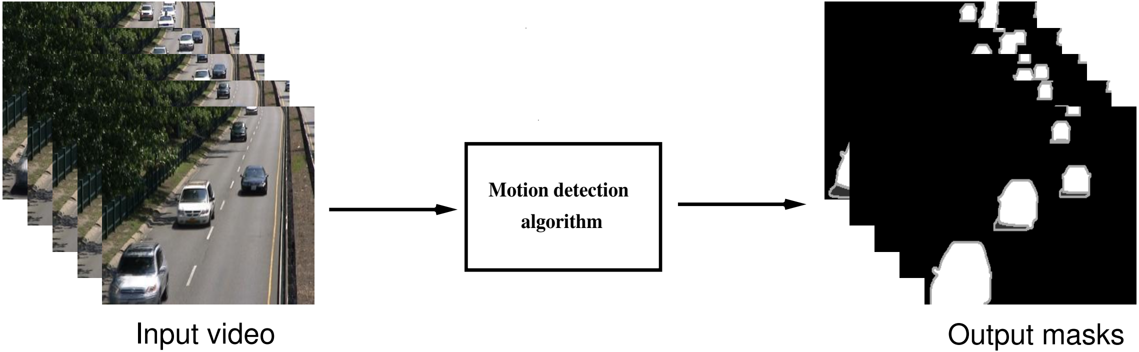
INTELSIG, Department of Electrical Engineering and Computer Science, University of Liège, Belgium

IWSSIP 2016 - 23rd International Conference on Systems, Signals and Image Processing - Bratislava, Slovakia - May 2016

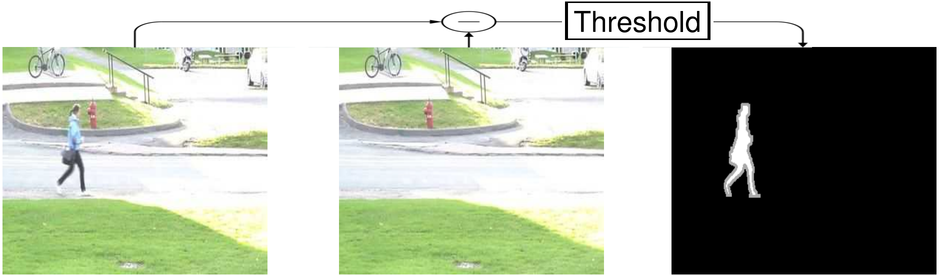
Outline

1. Introduction to background subtraction
   - Motion detection in video sequences
   - Principle of background subtraction
   - Common problems and traditional solutions
2. Proposed method
   - Deep background subtraction with scene-specific ConvNets
   - Pipeline of our algorithm
   - Network architecture and training
3. Experimental results
   - Methodology
   - Quantitative results
   - Qualitative results
4. Conclusion

Motion detection in video sequences



Principle of background subtraction



Current frame   Background model   Output binary mask

Main questions

How to **model** the background ? How to **initialize** the model? How to **update** the model? How to **subtract** the background model?

Common problems and traditional solutions

Common problems:

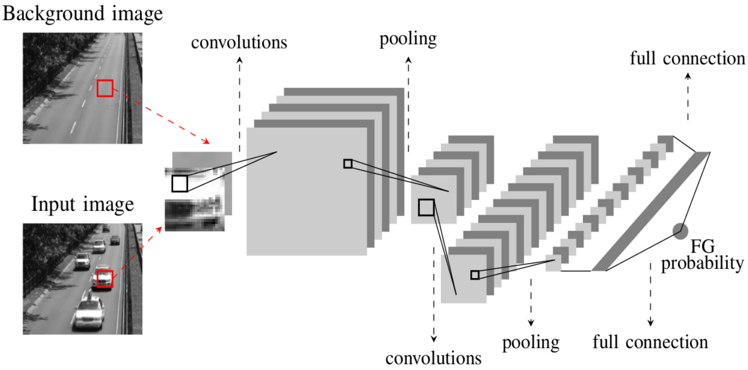
- Camouflage
- Noise
- Light changes
- Dynamic background
- Shadows
- ...

Traditional solutions:

- Complex background modeling strategies (GMM, KDE, Codebook, ViBe, ...)
- Hand-crafted features (Gradient, LBSP, HRI, ...)
- Post-processing (median filtering, area filtering, morphological filtering, ...)
- More recently : feedback loops

Deep background subtraction with scene-specific ConvNets

Our main idea is to face the complexity of the task in the subtraction operation itself, not in the background modeling strategy.



- Simple background model: a single grayscale image
- Deep subtraction operation
- Learned spatial features
- No post-processing or feedback loop
- Scene-specific ConvNet

Pipeline of our algorithm

Background image extraction → Dataset Generation → Network training → Background subtraction

Background image extraction

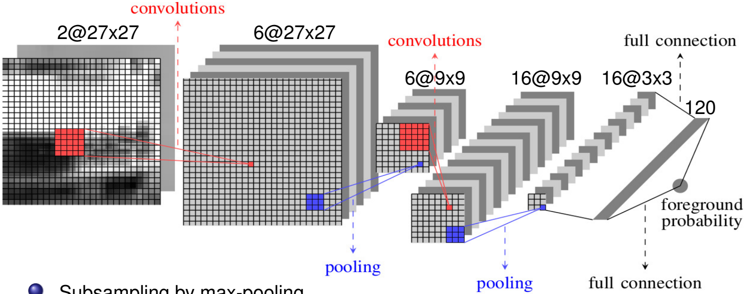
Pixel-based temporal median filter (150 frames)

Dataset

- Collection of TxT 2-channel patches with central pixel class as target value
- Scene-specific training data
- Automatic labeling with an existing BGS method or human expert labeling

Network architecture and training

Architecture



- Subsampling by max-pooling
- Rectified linear units
- 20243 trainable weights

Training

- Cross-entropy error function
- RMSProp optimization strategy
- Mini-batch size = 100
- Learning rate = 0.001
- Training stopped after 10000 iterations

Methodology

- Benchmarking on the 2014 ChangeDetection.net dataset (CDnet 2014)[1]
- The first half of each video is used to generate the training data while the second one is used as a test set
- Experiments restricted to sequences with different foreground objects between the training set and the test set (21 videos considered from 9 categories)
- Results compared to those of traditional and state-of-the-art methods on the test set in terms of F performance metric:

$$F = \frac{2PrRe}{Pr + Re}$$

- 2 variants of our method evaluated: ConvNet-GT (dataset labeling by human expert) and ConvNet-IUTIS (dataset labeling by IUTIS-5 BGS algorithm[2])

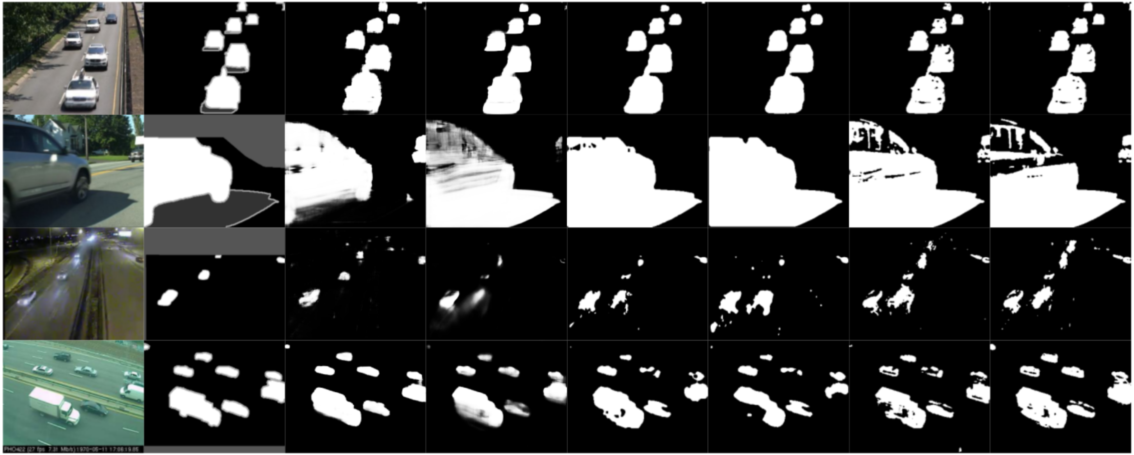
[1] Goyette *et al.*, "A novel video dataset for change detection benchmarking", *IEEE Trans. Image Process.*, 2014

[2] Bianco *et al.*, "How far can you get by combining change detection algorithms", *arXiv.org*, 2015

Quantitative results

Method	$F_{overall}$	$F_{Baseline}$	F_{Jitter}	$F_{DynamicBG}$	$F_{Shadows}$	$F_{Thermal}$	$F_{BadWeather}$	$F_{LowFramerate}$	F_{Night}	$F_{turbulence}$
ConvNet-GT	**0.9046**	**0.9813**	**0.9020**	0.8845	**0.9454**	**0.8543**	**0.9264**	**0.9612**	**0.7565**	**0.9297**
IUTIS-5	0.8093	0.9683	0.8022	0.8389	0.8807	0.7074	0.9043	0.8515	0.5384	0.7924
SuBSENSE	0.8018	0.9603	0.7675	0.7634	0.8732	0.6991	0.9195	0.8441	0.5123	0.8764
PAWCS	0.7984	0.9500	0.8473	**0.8965**	0.8750	0.7064	0.8587	0.8988	0.4194	0.7335
PSP-MRF	0.7927	0.9566	0.7690	0.7982	0.8735	0.6598	0.9135	0.8109	0.5156	0.8368
ConvNet-IUTIS	0.7897	0.9647	0.8013	0.7923	0.8590	0.7559	0.8849	0.8273	0.4715	0.7506
EFIC	0.7883	0.9231	0.8050	0.5247	0.8270	0.8246	0.8871	0.9336	0.6266	0.7429
Spectral-360	0.7867	0.9477	0.7511	0.7775	0.7156	0.7576	0.8830	0.8797	0.4729	0.8956
SC_SOBS	0.7450	0.9491	0.7073	0.6199	0.8602	0.7874	0.7750	0.7985	0.4031	0.8043
GMM	0.7444	0.9478	0.6103	0.7085	0.8396	0.7397	0.8472	0.8182	0.4004	0.7883
GraphCut	0.7394	0.9304	0.5183	0.7372	0.7543	0.7149	0.9166	0.8208	0.4751	0.7867
KDE	0.7298	0.9623	0.5462	0.5511	0.8357	0.7626	0.8691	0.8580	0.4057	0.7776

Qualitative results



Input image Ground truth ConvNet-GT ConvNet-IUTIS IUTIS-5 SuBSENSE GMM KDE

Conclusion

The proposed background subtraction algorithm:

- models the background with a single grayscale image
- faces the complexity of the task in the subtraction operation itself
- performs a deep subtraction using a trained convolutional neural network
- requires scene-specific labeled data
- outperforms state-of-the-art methods significantly when prior knowledge about the scene is considered