Autoencoder-based background reconstruction and foreground segmentation with background noise estimation

Supplementary material

1. Autoencoder architecture

The autoencoder is deterministic and takes as input a RGB image of size $h \times w$, and produces a RGB image (3 channels) and an error estimation map of the same size (1 channel).

The encoder and decoder structures in the proposed model are computed dynamically using as input the size (height h and width w) of the input frames of the dataset. The number of latent variables produced by the encoder is fixed to 16.

We use a fully convolutional autoencoder architecture, which appears to be more robust to overfitting than architectures including fully connected layers or locally connected layers. We add two fixed positional encoding channels as inputs to all layers of the encoder and the decoder, one channel coding for the horizontal coordinates, the other one for the vertical coordinates .

The encoder is a sequence of blocks composed of a convolution layer with kernel size 5, stride 3 and padding equal to 2, followed by a group normalization layer and a CELU nonlinearity layer. The generator is a symmetric sequence of blocks composed of transpose convolution layers with kernel size 5 and stride 3 and padding equal to 2 followed by group normalization and a CELU nonlinearity, except for the last layer where the transpose convolution layer is followed by a sigmoid to generate the final image. The number of layers of the encoder and the decoder is then equal to 5 or 6 depending on the image size (assuming that the maximum of the image height and image width is in the range $200 - 1000$). The number of channels per convolutional layer is fixed according to Table 1, depending on the image size and the background complexity.

These channel distributions are motivated by the fact that a larger number of parameters is required in the generator in order to handle complex backgrounds, but that we have experimentally observed that a large number of channels in the last layer of the encoder and the first layer of the decoder increases the risk of overfitting on foreground objects, so that reducing this number for long training schedule is necessary to improve the robustness of the auto-encoder with respect to the risk of overfitting. For example, we have measured that increasing the numbers of channels in the last hidden layer of the encoder and first hidden layer of the decoder to 160 and 256 leads to de 2,3 % degradation of the average F-Measure on the CDnet dataset.

For non-video dataset experiments, which handle small images, we use a smaller stride, set to 2 instead of 3. The autoencoder architectures for 64×64 images (ShapeStacks and ObjectRooms datasets) and 128×128 images (Clevrtex dataset) are described in Table 2 and 3:

2. Additional implementation details

The datasets and preprocessing codes for CLEVRTEX, Shapestacks and ObjectsRoom were downloaded from the following public repositories:

- https://www.robots.ox.ac.uk/~vgg/data/clevrtex/
- https://ogroth.github.io/shapestacks/
- https://github.com/deepmind/multi_object_datasets

3. Additional image samples

We provide in figures $1 - 7$ additional samples of background reconstruction and foreground segmentation obtained using the proposed model.

Table 1. Number of channels for each layer of the encoder and decoder (excluding positional encoding input channels)

background complexity	image size max(h,w)	Encoder	Decoder
simple	200-405	(3,64,160,160,32,16)	(16,32,256,256,144,4)
simple	406-1000	(3,64,160,160,160,32,16)	(16,32,256,512,256,144,4)
complex	200-405	(3,64,160,160,16,16)	(16,16,640,640,144,4)
complex	406-1000	(3,64,160,160,160,16,16)	(16,16,640,1280,640,144,4)

Table 2. autoencoder architecture for 64 $\times$ 64 images

Encoder

Layer	Size	Ch	Stride	Norm./Act.
Input	64	3		
Conv 5 $\times$ 5	32	64	2	GroupNorm/CELU
Conv 5 $\times$ 5	16	160	2	GroupNorm/CELU
Conv 5 $\times$ 5	8	320	2	GroupNorm/CELU
Conv 5 $\times$ 5	4	160	2	GroupNorm/CELU
Conv 4 $\times$ 4	2	16	2	GroupNorm/CELU
Conv 2 $\times$ 2	1	16	1	GroupNorm/CELU

Decoder

Layer	Size	Ch	Stride	Norm./Act.
Input	1	16		
Conv Transp 2 $\times$ 2	2	16	1	GroupNorm/CELU
Conv Transp 4 $\times$ 4	4	640	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	8	1280	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	16	640	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	32	144	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	64	4	2	
Sigmoid	64	4		

Table 3. autoencoder architecture for 128 $\times$ 128 images

Encoder

Layer	Size	Ch	Stride	Norm./Act.
Input	128	3		
Conv 5 $\times$ 5	64	64	2	GroupNorm/CELU
Conv 5 $\times$ 5	32	320	2	GroupNorm/CELU
Conv 5 $\times$ 5	16	640	2	GroupNorm/CELU
Conv 5 $\times$ 5	8	640	2	GroupNorm/CELU
Conv 5 $\times$ 5	4	320	2	GroupNorm/CELU
Conv 4 $\times$ 4	2	16	2	GroupNorm/CELU
Conv 2 $\times$ 2	1	16	1	GroupNorm/CELU

Decoder

Layer	Size	Ch	Stride	Norm./Act.
Input	1	16		
Conv Transp 2 $\times$ 2	2	16	1	GroupNorm/CELU
Conv Transp 4 $\times$ 4	4	320	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	8	640	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	16	1280	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	32	640	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	64	144	2	GroupNorm/CELU
Conv Transp 5 $\times$ 5	128	4	2	
Sigmoid	128	4		

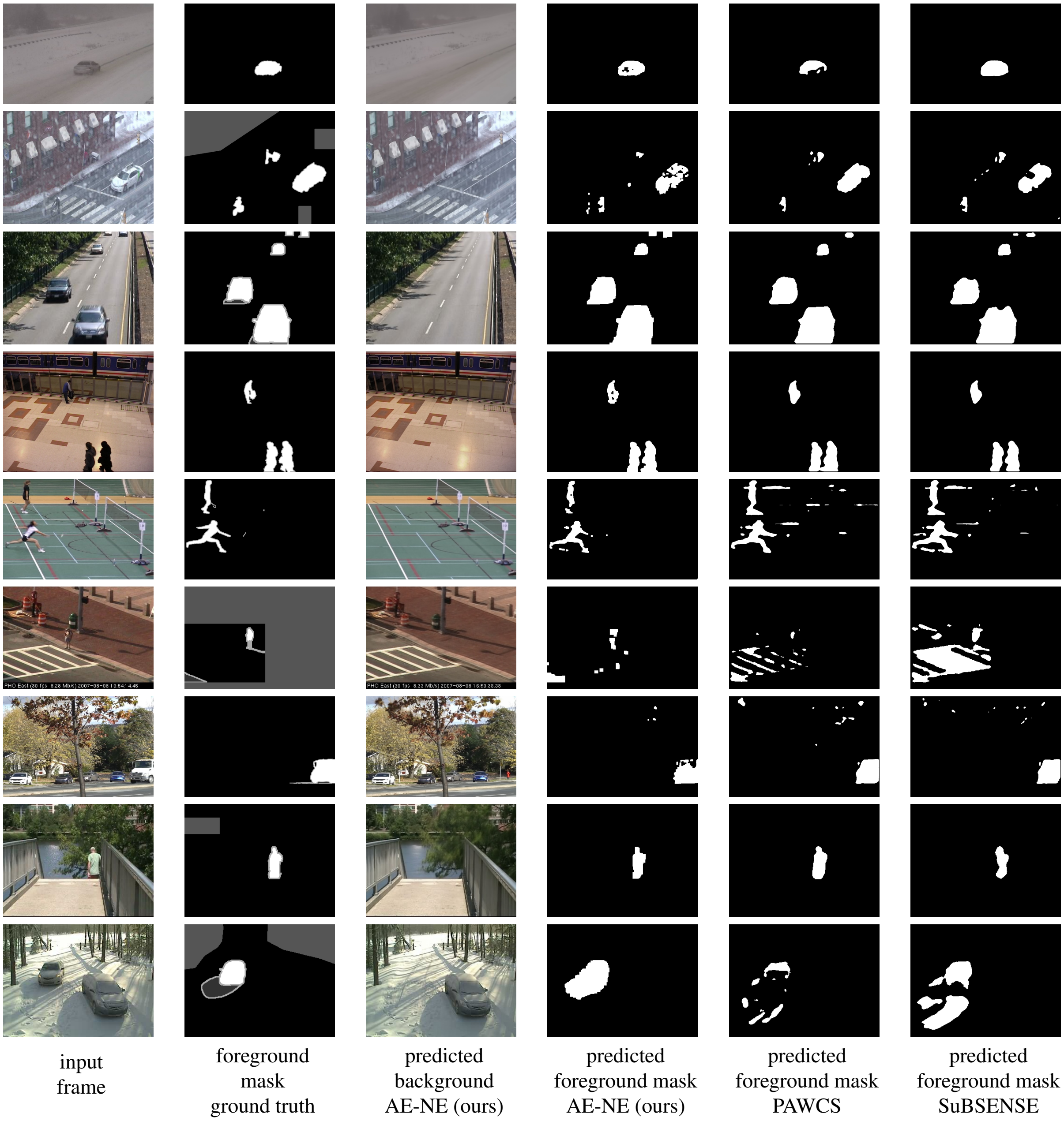
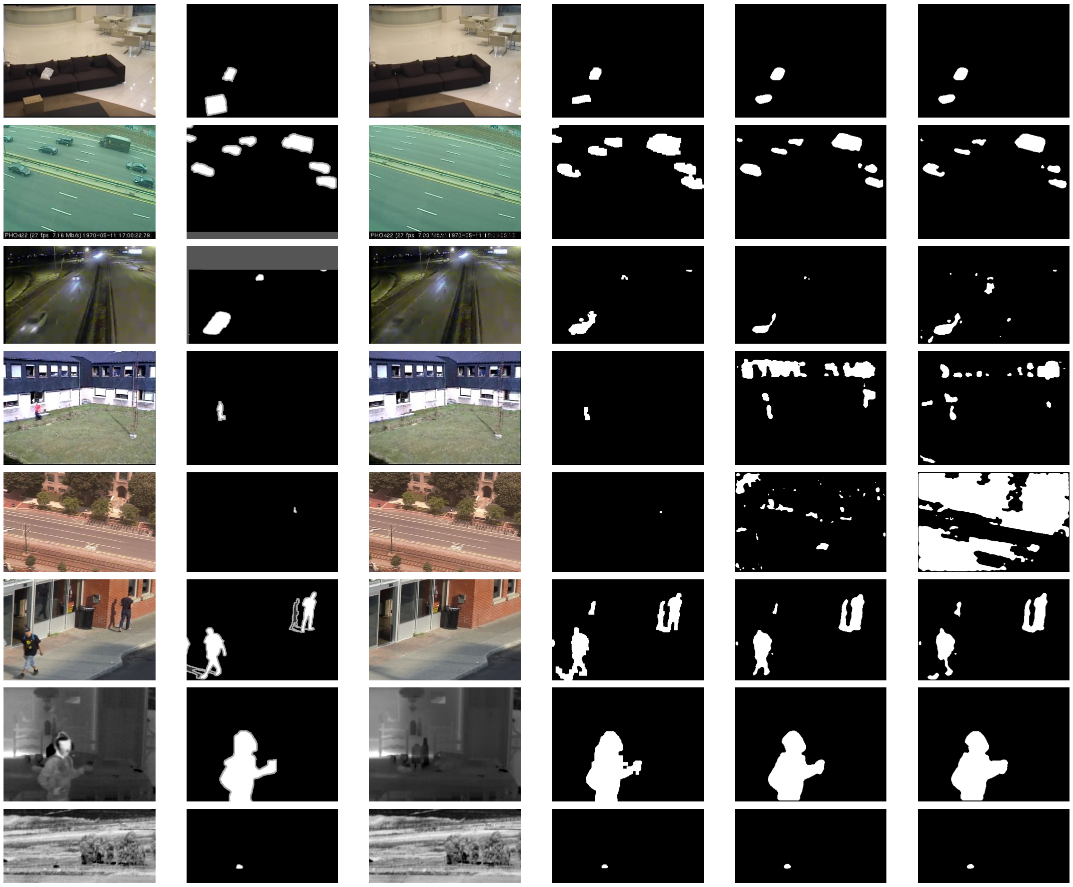

input frame
foreground mask ground truth
predicted background AE-NE (ours)
predicted foreground mask AE-NE (ours)
predicted foreground mask PAWCS
predicted foreground mask SuBSENSE

Figure 1. Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE



input frame

foreground mask ground truth

predicted background AE-NE (ours)

predicted foreground mask AE-NE (ours)

predicted foreground mask PAWCS

predicted foreground mask SuBSENSE

Figure 2. Examples of background reconstruction and foreground segmentation on the CDnet 2014 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE

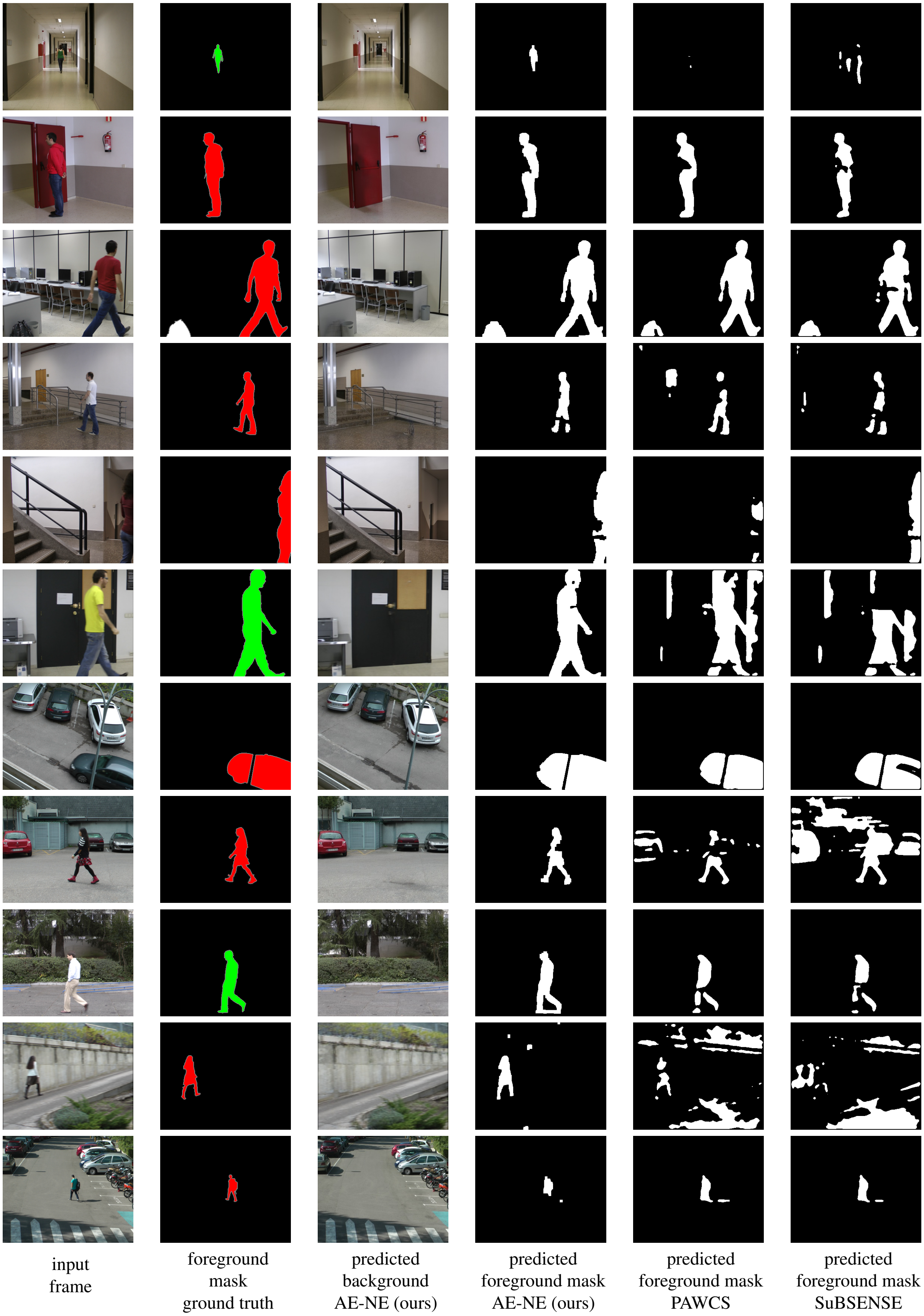

input frame
foreground mask ground truth
predicted background AE-NE (ours)
predicted foreground mask AE-NE (ours)
predicted foreground mask PAWCS
predicted foreground mask SuBSENSE

Figure 3. Examples of background reconstruction and foreground segmentation on the LASIESTA dataset produced using the proposed model and comparison with PAWCS and SuBSENSE

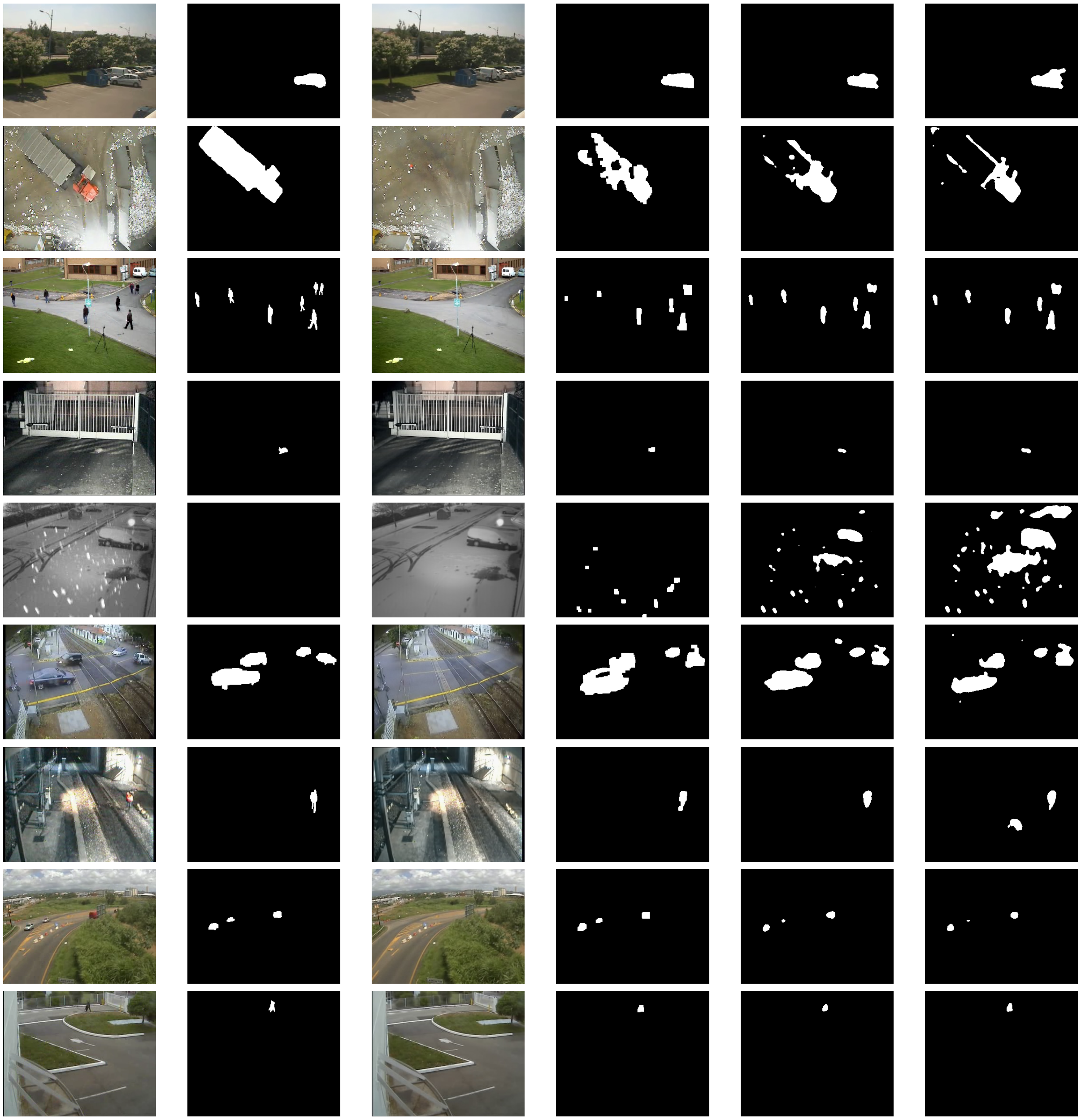

input frame
foreground mask ground truth
predicted background AE-NE (ours)
predicted foreground mask AE-NE (ours)
predicted foreground mask PAWCS
predicted foreground mask SuBSENSE

Figure 4. Examples of background reconstruction and foreground segmentation on the BMC 2012 dataset produced using the proposed model and comparison with PAWCS and SuBSENSE

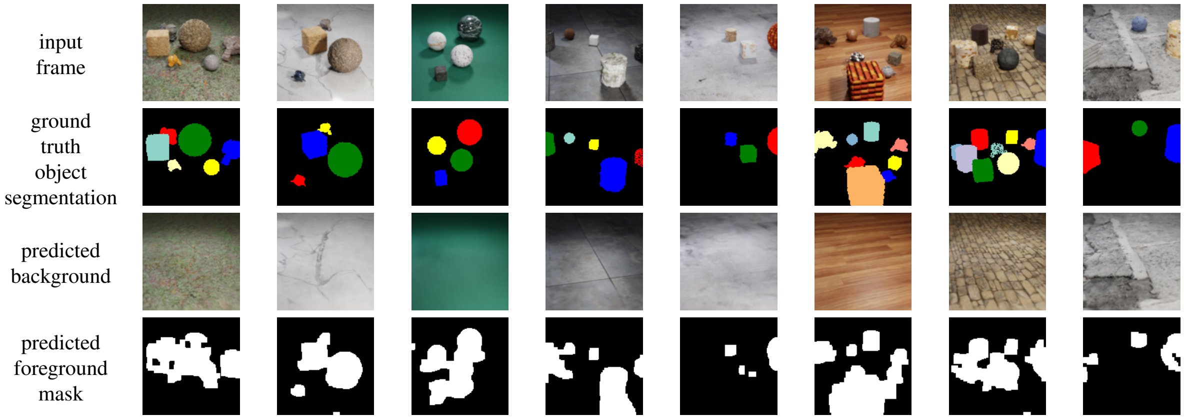


Figure 5. Examples of background reconstruction and foreground segmentation on Clevrtex dataset

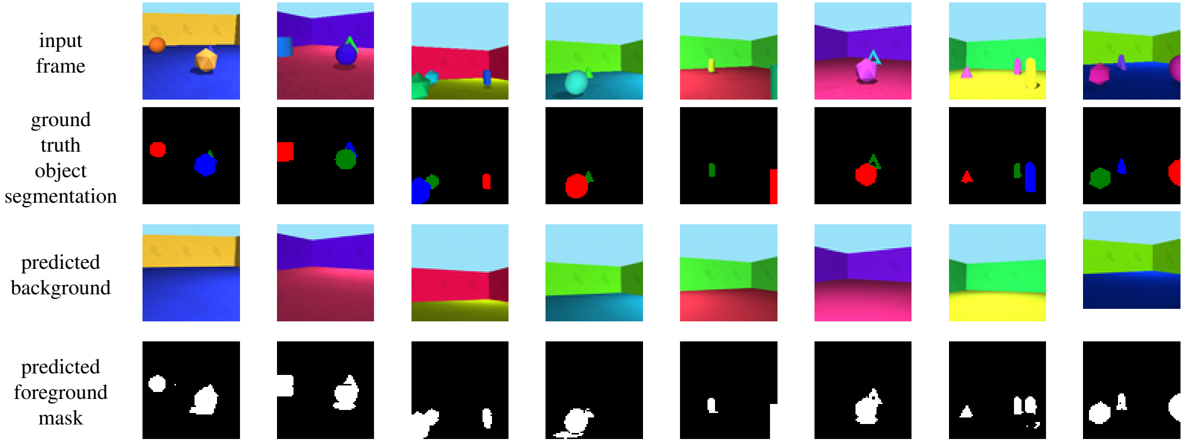


Figure 6. Examples of background reconstruction and foreground segmentation on ObjectsRoom dataset

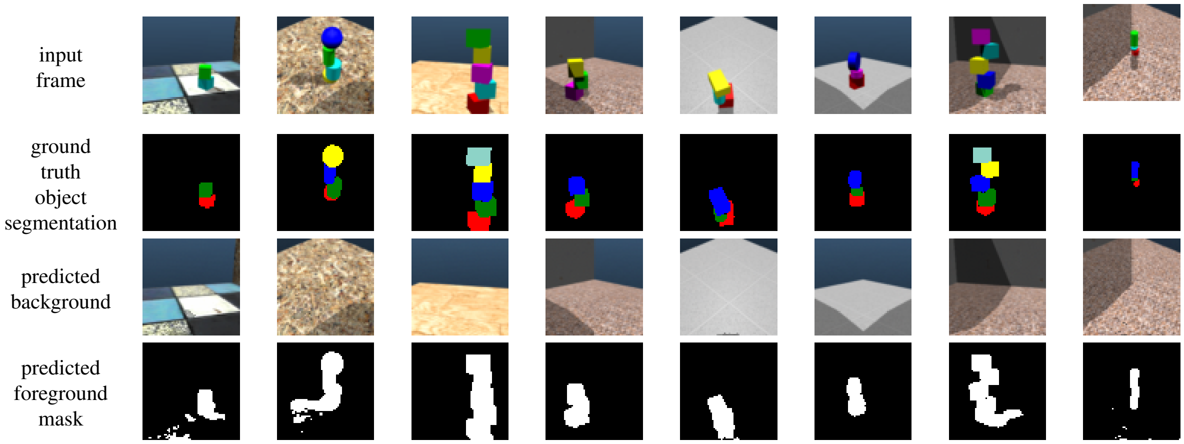


Figure 7. Examples of background reconstruction and foreground segmentation on ShapeStacks dataset