Learning Temporal Regularity in Video Sequences

arXiv 2016
Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, Larry S. Davis

SeulGi Park

March 5, 2020

Contents

1. Introduction

- 비정상 이벤트(드물게 불규칙하거나 특정한 순간)를 supervised로 모델링하기 보다는 정상 이벤트의 시간적 규칙을 이용하여 limited supervision으로 모델링
- 정상 이벤트의 모션 특징을 추출하기 위해 여러 가지 데이터셋을 사용
  - CUHK Avenue, Subway (Enter and Exit), UCSD Pedestrian datasets (Ped1 and Ped2)

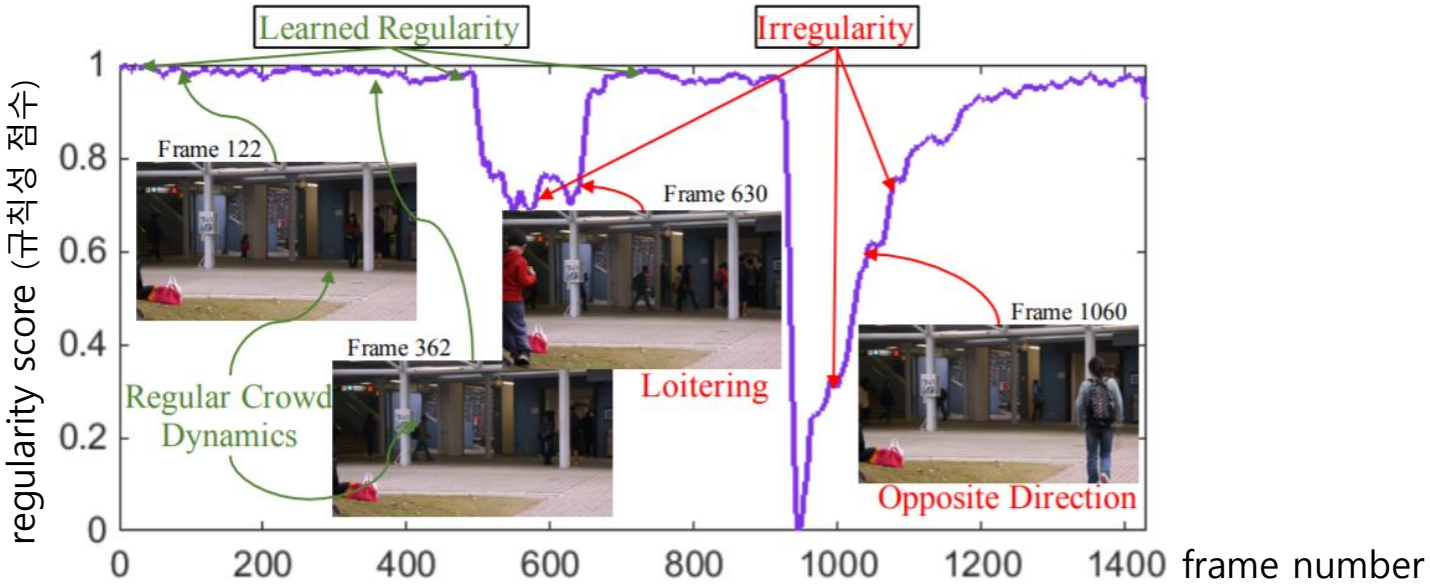


Figure 1. Learned regularity of a video sequence. Y-axis refers to regularity score and X-axis refers to frame number. When there are irregular motions, the regularity score drops significantly (from CUHK-Avenue dataset [8]).

2. Approach

1. Handcrafted Features: Improved Trajectory(HOG, HOF + Trajectory)로 Motion Features 추출
2. Fully Connected Autoencoder(layer): Handcrafted Features로 Frame에서의 Motion Pattern 추출
3. Fully Convolutional Autoencoder(layer): Handcrafted Features로 Videos에서의 Regular Motion Signatures를 추출

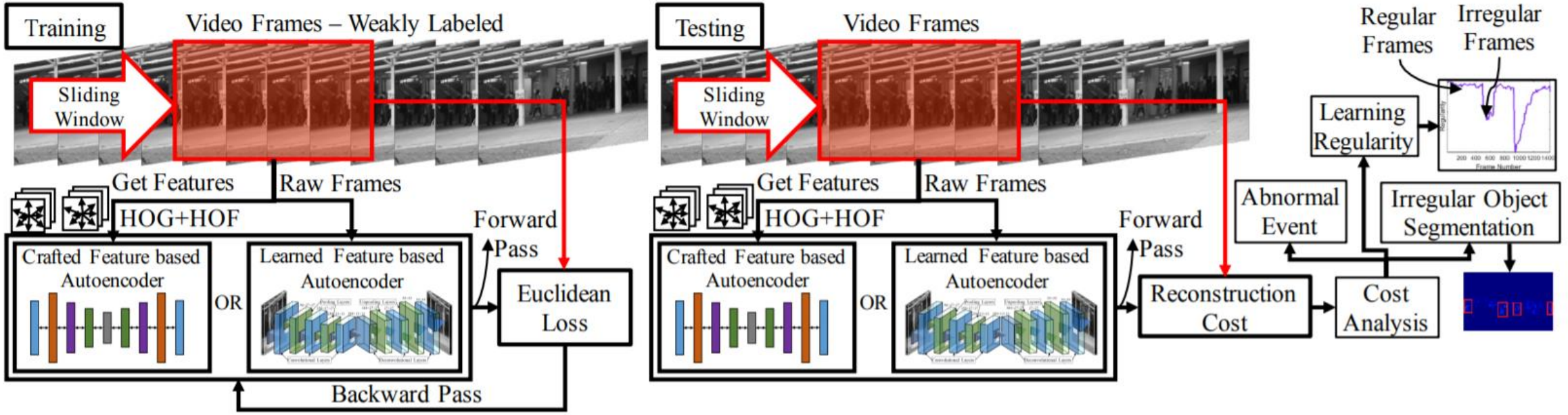


Figure 2. Overview of our approach. It utilizes either state-of-the-art motion features or learned features combined with autoencoder to reconstruct the scene. The reconstruction error is used to measure the regularity score that can be further analyzed for different applications.

2. Approach

2.1 Learning Motions on Handcrafted Features => limited supervision

- **다양한 데이터셋(비디오)에서 정상 이벤트(프레임) Features를 추출**하기 위해
- 입력 값: Trajectory information + HOG, HOF Handcrafted Features
- 출력 값: Motion Features(row-level Spatio-temporal)

1) 프레임에서 Handcrafted Appearance features와 Motion features 추출
2) 추출된 기능을 Fully Connected layer의 입력으로 사용하여 프레임의 시간 규칙성을 학습

- **Low-Level Motion Information in a Small Temporal Cuboid**
  - Appearance features와 Motion features 효율적으로 인코딩하기 위해 HOG, HOF를 Spatio-temporal cuboid
- **Trajectory Encoding**
  - Improved Trajectory[Wanget al.]: HOG, HOF Features + Trajectory information
  - 프레임에서 interest points 기준으로 HOG와 HOF를 사용하여 다음 프레임에서의 interest points를 Tracking
- **Final Motion Feature**
  - Local Appearance features와 Motion features 주변의 Trajectories HOG와 HOF로 인코딩 됨

2. Approach

2.1.1 Model Architecture

- **정상 이벤트(프레임) Motion Features의 Pattern을 추출**하기 위해
- 입력 값: Motion Features(row-level Spatio-temporal) = Handcrafted Features output
- 출력 값: Regular Motion Patterns

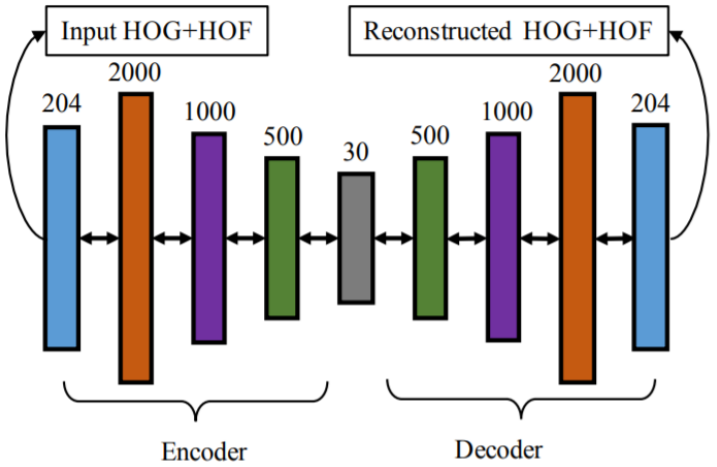


Figure 3. Structure of our autoencoder taking the HOG+HOF feature as input.

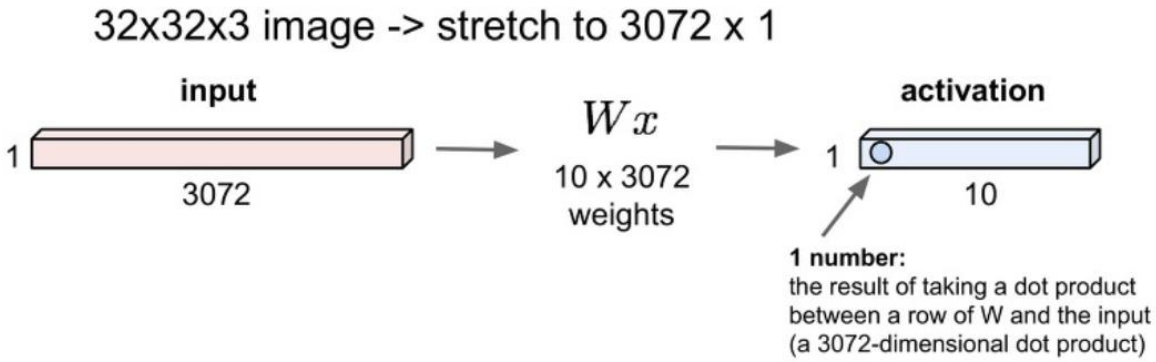
- Objective Function : Input feature = $\mathbf{x}_i$, Output reconstructed feature = $f_W(\mathbf{x}_i)$, Euclidean loss, L2 regularization, N = size of mini batch, γ = hyper-parameter(loss, regularization balance), W = weights

$$\hat{f}_W = \arg\min_W \frac{1}{2N} \sum_i \|\mathbf{x}_i - f_W(\mathbf{x}_i)\|_2^2 + \gamma \|W\|_2^2, \quad (1)$$

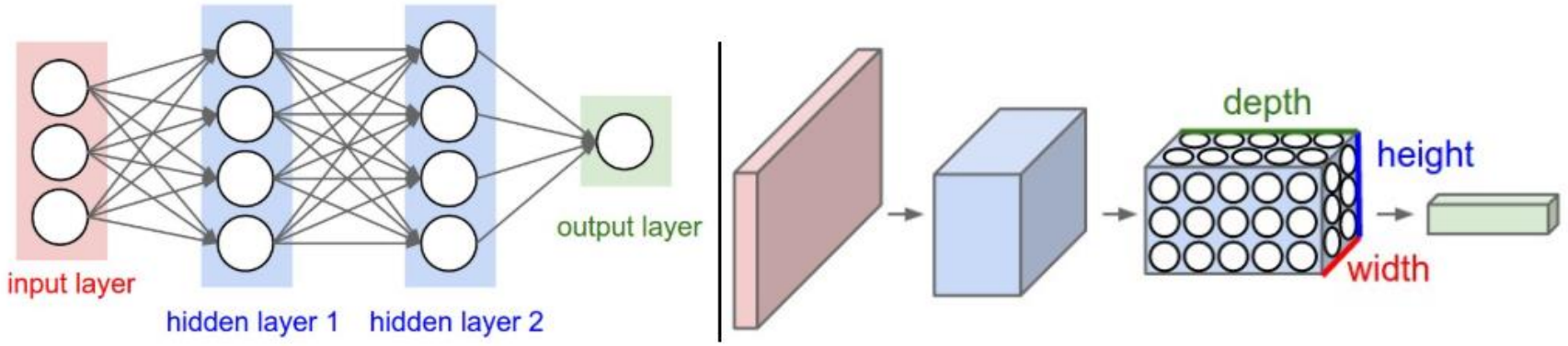
2. Approach

2.2 Learning Features and Motions

– Fully Connected layer는 1차원 데이터만 입력 받을 수 있어, 3차원 데이터를 입력하고자 하는 경우 평탄화 하여 입력하여야 함



– Fully Connected layer는 비디오에서의 시공간 features를 학습하기에 부족, Fully Convolution layer 사용



2. Approach

2.2.1 Model Architecture

- **정상 이벤트(비디오)의 시공간의 정보를 유지하면서 Regular Motion Signatures를 추출**하기 위해
- 입력 값: Videos (Motion Features(row-level Spatio-temporal))
- 출력 값: Regular Motion Signatures (비디오의 시공간 정보 유지)

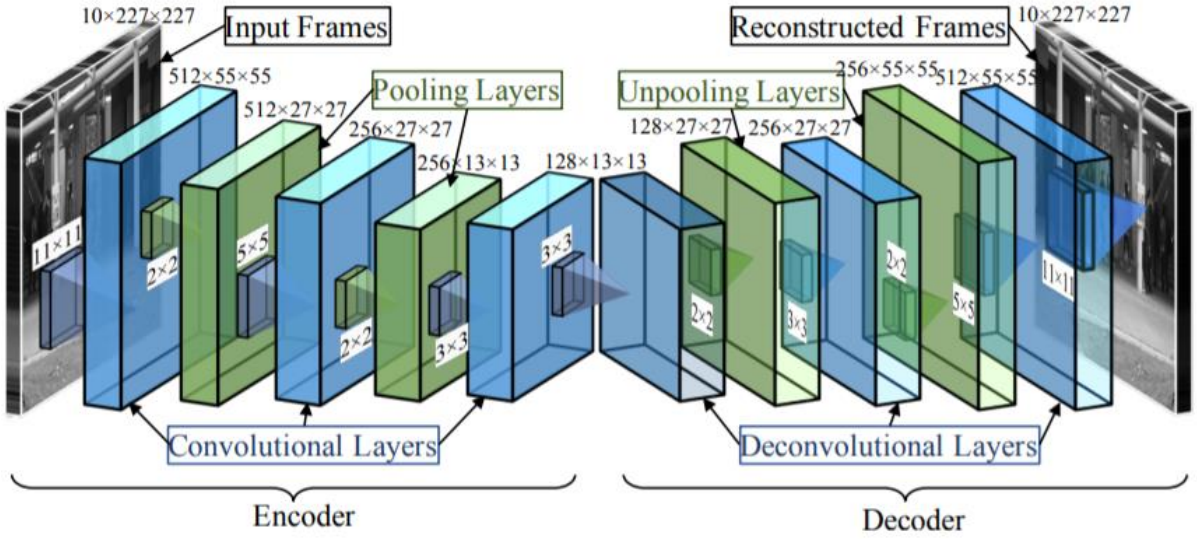


Figure 4. Structure of our fully convolutional autoencoder.

The first convolutional layer has 512 filters with a stride of 4. It produces 512 feature maps with a resolution of 55×55 pixels. Both of the pooling layers have kernel of size 2×2 pixels and perform max poling. The first pooling layer produces 512 feature maps of size 27×27 pixels. The second and third convolutional layers have 256 and 128 filters respectively. Finally, the encoder produces 128 feature maps of size 13×13 pixels. Then, the decoder reconstructs the input by deconvolving and unpooling the input in reverse order of size. The output of final deconvolutional layer is the reconstructed version of the input.

2. Approach

Input Data Layer

- 임의의 수의 채널로 구성된 비디오
- T 프레임을 함께 쌓아서 자동 인코더에 대한 입력으로 사용 (T: 슬라이딩 윈도우의 길이)
- T가 증가하면 더 긴 동작이나 시간 정보를 통합함

Data Augmentation In the Temporal Dimension

- stride-1: 모든 T 프레임이 연속적
- stride-2: 각각 하나의 프레임을 건너 뜀
- stride-3: 각각 두개의 프레임을 건너 뜀

Optimization Objective

- Xi는 i 번째 시공간 정보(Spatio-temporal cuboid)

$$\hat{f}_W = \arg\min_W \frac{1}{2N} \sum_i \|\mathbf{X}_i - f_W(\mathbf{X}_i)\|_2^2 + \gamma \|W\|_2^2, \quad (2)$$

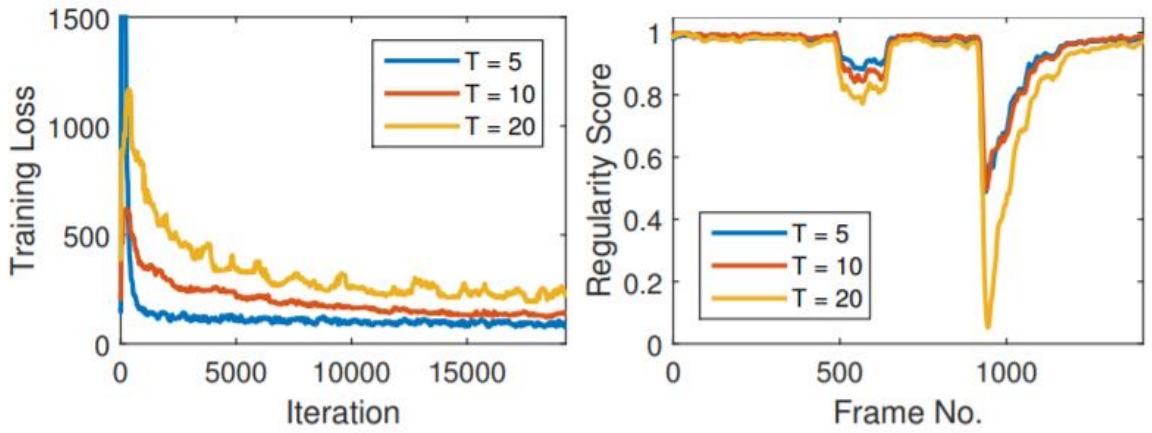


Figure 5. Effect of temporal length (T) of input video cuboid. (Left) X-axis is the increasing number of iterations, Y-axis is the training loss, and three plots correspond to three different values of T. (Right) X-axis is the increasing number of video frames and Y-axis is the regularity score. As T increases, the training loss takes more iterations to converge as it is more likely that the inputs with more channels have more irregularity to hamper learning regularity. On the other hand, once the model is learned, the regularity score is more distinguishable for higher values of T between regular and irregular regions (note that there are irregular motions in the frame from 480 to 680, and 950 to 1250).

2. Approach

2.3. Optimization and Initialization

- Optimization: gradient method AdaGrad 사용(Adam, RMSProp 보다 결과가 좋았음)
- Initialization: Xavier algorithm를 사용(입,출력 뉴런 수를 기반으로 초기화 규모를 자동으로 결정하여 신호를 여러 계층을 통해 합리적인 범위의 값으로 유지, Gaussian 보다 결과가 좋았음)

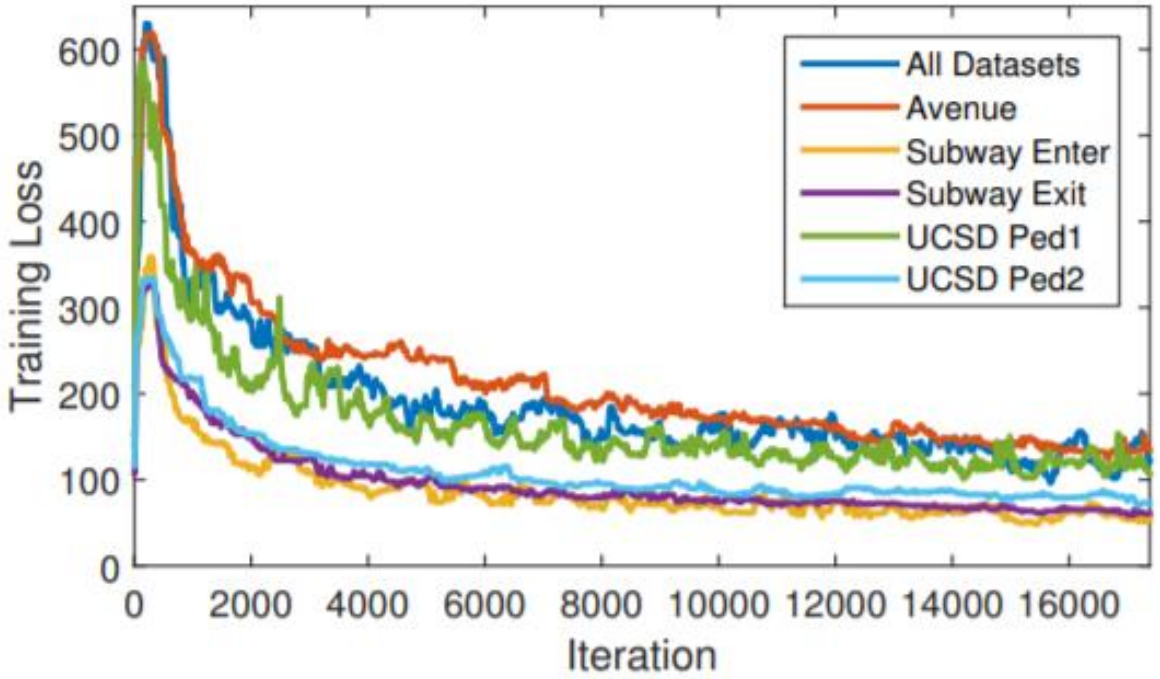


Figure 6. Loss value of models trained on each dataset and all datasets as a function of optimization iterations.

2. Approach

2.4. Regularity Score(규칙 점수)

Reconstruction Error = e(x, y, t)

- 학습한 모델(f_W)에 대한 비디오 시퀀스 프레임 t에서 위치 (x, y)에서 픽셀의 intensity 값 I
- f_W : fully convolutional autoencoder가 학습한 모델

$$e(x, y, t) = \|I(x, y, t) - f_W(I(x, y, t))\|_2, \qquad (3)$$

Regularity Score = s(t)

- 프레임 t의 x,y 픽셀의 reconstruction error 계산 후
- 모든 픽셀의 error를 합산하여 프레임의 Reconstruction Error를 계산 $e(t) = \sum_{(x,y)} e(x, y, t)$

$$s(t) = 1 - \frac{e(t) - \min_t e(t)}{\max_t e(t)}. \qquad (4)$$

3. Experiments

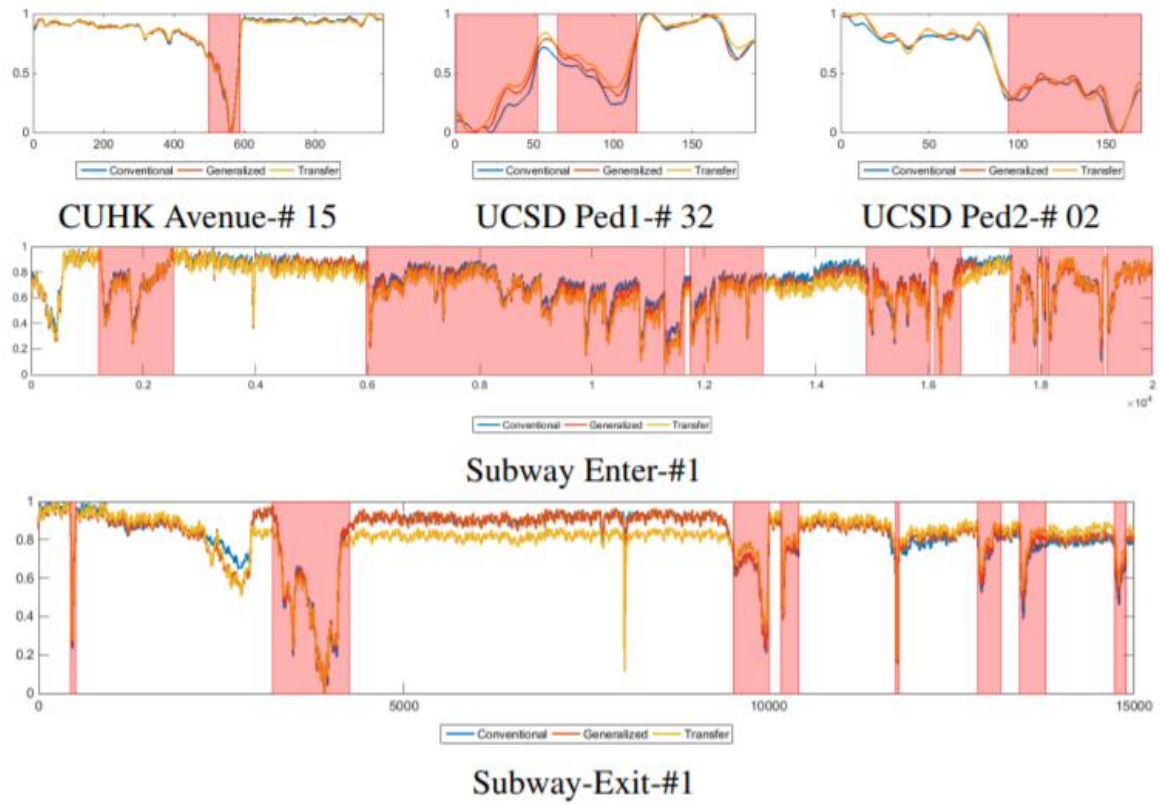


Figure 7. Generalizability of Models by Obtained Regularity Scores. 'Conventional' is by a model trained on the *specific target* dataset. 'Generalized' is by a model trained on *all* datasets. 'Transfer' is by a model trained on *all datasets except that specific target* datasets. Best viewed in zoom.

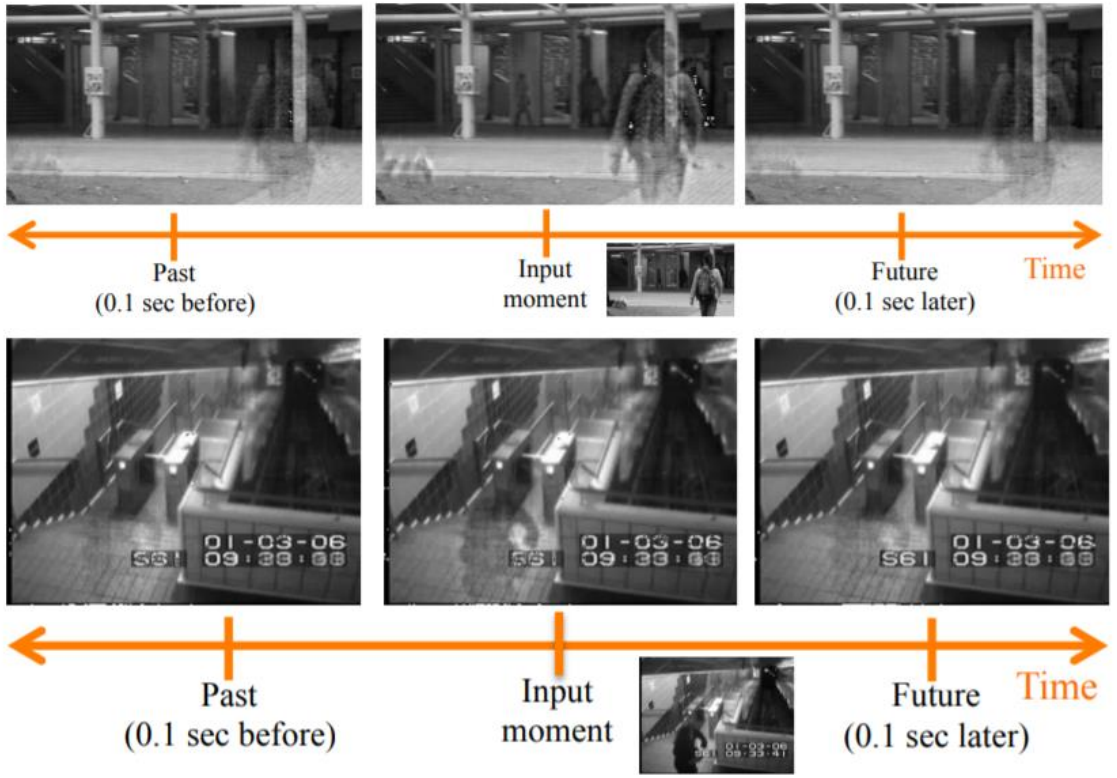


Figure 10. Synthesizing a video of regular motion from a single seed image (at the center). Upper: CUHK-Avenue. Bottom: Subway-Exit.

3. Experiments

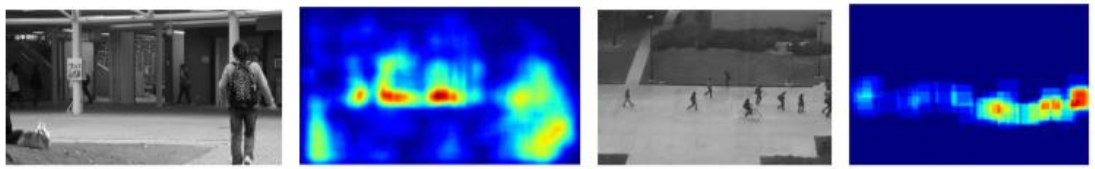
Avenue Dataset

UCSD Ped1 Dataset

UCSD Ped2 Dataset

Subway Exit Dataset

Figure 8. (Left) A sample irregular frame. (Middle) Synthesized regular frame. (Right) Regularity Scores of the frame. Blue represents regular pixel. Red represents irregular pixel.



Avenue Dataset UCSD Ped2 Dataset

Figure 9. Learned regularity by improved trajectory features. (Left) Frames with irregular motion. (Right) Learned regularity on the entire video sequence. Blue represents regular region. Red represents irregular region.

3. Experiments

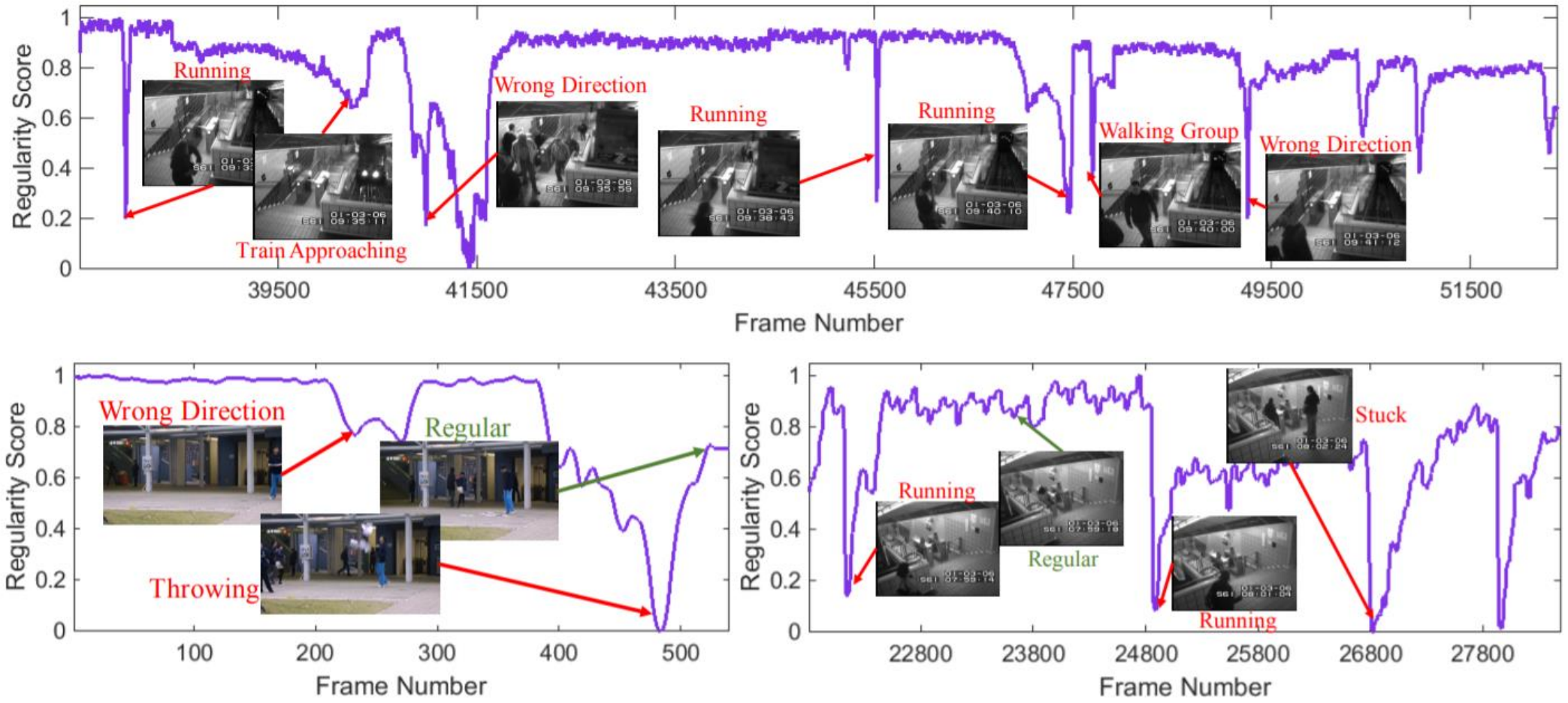
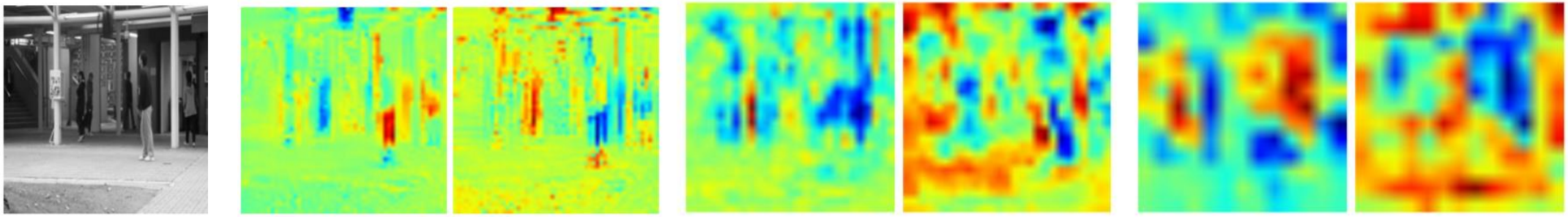


Figure 11. Regularity score (Eq.3) of each frame of three video sequences. (Top) Subway Exit, (Bottom-Left) Avenue, and (Bottom-Right) Subway Enter datasets. Green and red colors represent regular and irregular frames respectively.

3. Experiments

Dataset			Regularity	Anomaly Detection					
			Conv-AE	# Anomalous	Correct Detection / False Alarm			AUC/EER	
Name	# Frames	# Regular Frames	Correct Detect / FA	Event	Conv-AE	IT-AE	State of the art	Conv-AE	State of the art
CUHK Avenue	15,324	11,504	11,419/355	47	45/4	43/8	12/1 (Old Dataset) [8]	70.2/25.1	N/A
UCSD Ped1	7,200	3,195	3,135/310	40	38/6	36/11	N/A	81.0/27.9	92.7/16.0 [60]
UCSD Ped2	2,010	374	374/50	12	12/1	12/3	N/A	90.0/21.7	90.8/16.0 [30]
Subway Entrance	121,749	119,349	112,188/4,154	66	61/15	55/17	57/4 [8]	94.3/26.0	N/A
Subway Exit	64,901	64,181	62,871/1,125	19	17/5	17/9	19/2 [8]	80.7/9.9	N/A

Table 1. Comparing abnormal event detection performance. AE refers to auto-encoder. IT refers to improved trajectory.



(a) Input data frame (b) Conv1 filter responses. (c) Conv2 filter responses. (d) Conv3 filter responses.

Figure 12. Filter responses of the convolutional autoencoder trained on the Avenue dataset. Early layers (conv1) captures fine grained regular motion pattern whereas the deeper layers (conv3) captures higher level information.

4. Conclusion

- limited supervision autoencoders를 사용하여 시간적으로 규칙적인 패턴 학습
  1) Handcrafted Features: Improved Trajectory(HOG, HOF + Trajectory)로 Motion Features 추출
  2) Fully Connected Autoencoder(layer): Handcrafted Features로 Frame에서의 Motion Pattern 추출
  3) Fully Convolutional Autoencoder(layer): Handcrafted Features로 Videos에서의 Regular Motion Signatures를 추출
- 다양한 데이터셋을 이용하여 정상 이벤트를 기준으로 feature를 추출하고 학습시켰기 때문에 편향된 데이터 뿐만 아니라 일반화 가능

Thank you!