

CGAN에 기반한 비디오 월 컨트롤러의 레이아웃 생성

김성진
(주)리드텍
sjkim@leadtech21.com

Layout Generation for Video Wall Controller Using CGAN

Sung-jin Kim
LEADTECH Co., Ltd

요 약

비디오 월 컨트롤러의 멀티스크린에 여러 영상을 동시에 표출하고자 하는 경우, 운영자는 표출할 영상의 레이아웃을 미리 저장해두고 필요할 때 이를 불러와서 표출한다. 비디오 월 컨트롤러의 운용 중에 발생하는 상황에 따라 레이아웃이 달라지므로 모든 상황을 고려하여 레이아웃을 저장하는 것이 일반적인 운용 방식이다. 멀티스크린의 모니터 개수가 많지 않은 소규모의 시스템에서는 서너 개의 레이아웃을 저장하는 것만으로 충분하지만, 모니터의 개수가 늘어날수록 레이아웃 디자인의 경우 수가 늘어나므로 레이아웃의 디자인과 관리에 많은 시간과 비용이 소요된다. 따라서 본 논문에서는 CGAN을 이용하여 비디오 월 컨트롤러의 레이아웃을 자동으로 생성하는 모델을 제안한다.

1. 서론

비디오 월 컨트롤러는 여러 개의 모니터를 연속적으로 배치하여 하나의 큰 스크린으로 표출하는 특수한 형태의 멀티 모니터와 입력 소스의 영상을 멀티 모니터에 표출하기 위한 제어 시스템으로 구성된다. 운영자는 제어 시스템을 사용하여 멀티스크린에 영상을 표출한다. 비디오 월 컨트롤러로 모니터링 작업을 수행할 때 상황에 따라 필요한 영상을 그때 그때 표출하기도 하지만, 영상을 개별적으로 표출하는 것보다는 여러 개의 영상을 동시에 표출하는 것이 더 효율적이다. 따라서 운영자는 표출할 영상과 표출할 위치 및 크기를 미리 저장해두고 필요한 경우에 이를 불러와서 여러 영상을 동시에 표출한다. 그림 1과 같이 영상을 표출할 위치와 크기의 정보를 레이아웃이라고 한다.



(그림 1) 레이아웃과 표출된 영상

운영자는 비디오 월 컨트롤러의 제어 시스템을 사

용하여 운용 중에 발생할 수 있는 모든 상황에 대해 레이아웃을 디자인하고 저장한다. 멀티스크린의 모니터 개수가 많지 않은 소규모의 시스템에서는 서너 개의 레이아웃을 저장하는 것만으로 충분하지만, 모니터의 개수가 늘어날수록 레이아웃 디자인의 수가 늘어나므로 레이아웃의 디자인과 관리에 많은 시간과 비용이 소요되고 운영자의 입장에서는 번거로운 작업일 수밖에 없다. 이에 본 논문에서는 레이아웃을 미리 디자인하거나 저장할 필요 없이 운용 중의 상황에 따라 레이아웃을 자동으로 생성하는 모델을 제안한다.

2. 관련 연구

CGAN(Conditional Generative Adversarial Network)[1]은 생성 모델의 훈련에 사용하는 GAN[2]을 확장한 모델로서, 2014년 Mehdi Mirza와 Simon Osindero가 제안하였다. CGAN의 생성자(generator)와 판별자(discriminator)는 추가적으로 제공되는 정보에 의해 조건적인 관계를 학습할 수 있는데, 추가적인 정보로는 클래스의 레이블, 태그셋 그리고 텍스트에 의한 설명 등이 사용된다. 생성자는 훈련 데이터셋의 레이블과 매칭되는 샘플을 생성

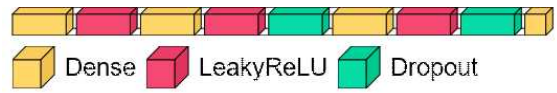
하도록 학습하는 것과 동시에 진짜와 구분하기 힘들 정도의 합성 샘플을 생성하도록 학습한다. 한편, 판별자는 가짜와 진짜 샘플을 구분할 수 있도록 학습하는 것과 병행하여 샘플이 레이블과 매칭되는 지를 구분할 수 있도록 학습한다. 결과적으로 생성자는 판별자를 속이기 위해 진짜 같은 샘플을 생성할 뿐 아니라 레이블과 매칭되는 샘플을 생성하도록 훈련되므로, 훈련이 완료되면 생성자에게 원하는 레이블을 제공하여 레이블과 매칭되는 샘플을 얻을 수 있게 된다. 레이블 정보를 생성자와 판별자의 모델에 인코딩 하기 위한 방법에는 여러 가지가 있지만 임베딩 레이어(Embedding Layer)를 사용하는 것이 일반적이다.

3. 접근 방식 및 모델 아키텍처

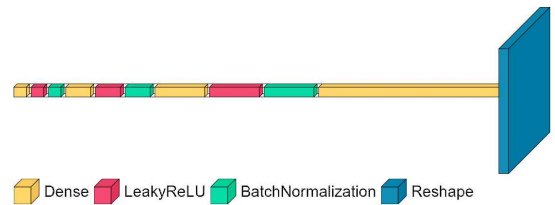
본 논문에서는 CGAN을 이용하여 비디오 월 컨트롤러의 레이아웃을 자동으로 생성한다. 비디오 월 컨트롤러의 운용 중에 발생할 수 있는 상황의 유형과 상황 유형별 레이아웃의 이미지 데이터로 CGAN을 훈련하고, 운용 중의 상황에 따라 상황 유형을 CGAN에 제공하여 레이아웃을 자동으로 생성한다. 운용 중에 발생할 수 있는 상황은 모니터링 하는 대상과 설치 장소 그리고 설치 규모 등에 따라 다양하지만 본 논문에서는 실험상의 편의를 위해 3가지의 유형을 사용하였는데, 바로 평상시, 이벤트 발생, 행사이다. 평상시는 일반적인 모니터링 업무를 수행하는 상황으로 가능한 한 많은 영상을 표출하여 모니터링의 효율성을 높이는 유형이다. 이벤트 발생은 거동수상자가 나타난다거나 화재 발생 등과 같은 모니터링 하는 대상에게 이상이 발생한 상황으로 이벤트가 발생한 영상을 확대하여 표출하므로 평시에 비해 표출 크기가 큰 유형이다. 행사는 실내 행사나 교육 그리고 프레젠테이션 등을 실시하는 상황으로 전체 스크린으로 확대하여 표출하는 경향이 있으므로 상황 유형 중에서 표출 크기를 가장 크게 디자인한다. 상황 유형 별 레이아웃의 이미지 데이터는 비디오 월 컨트롤러의 제어 시스템을 사용하여 레이아웃을 저장한 후 수집한다.

CGAN의 판별자는 입력으로 주어지는 레이아웃 이미지가 실제로 수집한 데이터인지 생성자가 만들어낸 합성 이미지인지를 구분하는 것과 동시에 상황 유형과 매칭되지 않는 레이아웃을 구분하도록 훈련되며 아키텍처는 그림 2와 같다. CGAN의 생성자는 판별자가 구분하기 힘든 수준의 레이아웃 이미지를

생성하는 것과 병행하여 상황 유형과 매칭되는 레이아웃을 생성하도록 훈련되며 아키텍처는 그림 3과 같다.



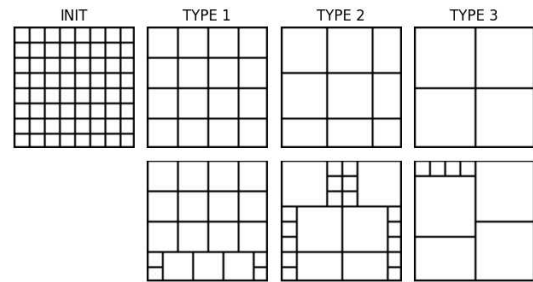
(그림 2) 판별자의 아키텍처



(그림 3) 생성자의 아키텍처

4. 실험 및 결과

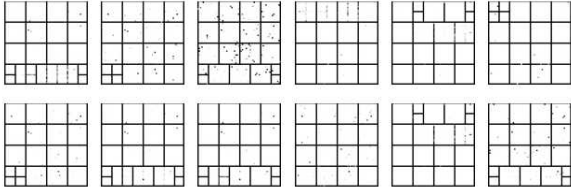
실험에 사용한 비디오 월 컨트롤러는 8단 8열의 FHD(1920x1080) 모니터 64대로 구성된 시스템이다. 비디오 월 컨트롤러의 제어 시스템을 사용하여 상황 유형별로 레이아웃을 디자인하고 그림 4와 같은 이미지 데이터를 수집하였다.



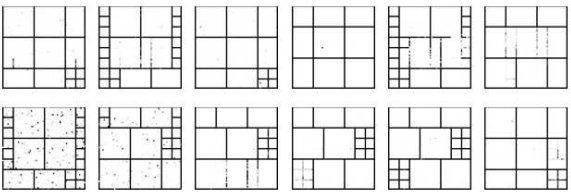
(그림 4) 레이아웃 이미지 데이터셋의 예

가장 왼쪽의 이미지는 8단 8열의 모니터 구성을 나타내고 있으며 하나의 작은 사각형이 한 대의 FHD 모니터를 나타낸다. 유형 1은 평상시로 2단 2열의 크기를 중심으로 레이아웃을 디자인하였다. 유형 2는 이벤트 발생으로 3단 3열, 유형 3은 행사로 4단 4열의 크기를 중심으로 디자인하였다. 상황 유형별로 수집한 이미지를 [-1, 1]의 범위로 스케일링하였고 그 외의 이미지 전처리는 수행하지 않았다. CGAN은 배치의 크기를 16으로 설정하고 미니 배치 SGD(Mini-batch Stochastic Gradient Descent)로 훈련 하였다. 모든 파라미터는 랜덤으로 초기화하였다. 활성화 함수는 LeakyReLU를 사용하였고 파라미터는 0.2를 설정하였다. 최적화 알고리즘은 Adam

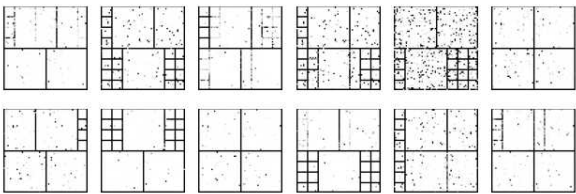
optimizer(Kingma & Ba[3], 2014)를 사용하였고 학습률(Learning Rate)은 0.0002, β_1 은 0.5를 설정하였다. 25000 epoch 후에 CGAN의 생성자는 상황 유형별로 그림 5, 그림 6 그리고 그림 7과 같은 샘플 이미지를 생성하였다.



(그림 5) 유형 1의 샘플 이미지



(그림 6) 유형 2의 샘플 이미지



(그림 7) 유형 3의 샘플 이미지

노이즈가 발생한 이미지도 있지만 CGAN은 상황 유형의 대표적인 레이아웃 크기를 학습하여 잘 생성하고 있는 것을 알 수 있다.

5. 결론

본 논문에서는 비디오 월 컨트롤러의 멀티스크린에 여러 영상을 동시에 표출하고자 할 때 레이아웃을 미리 디자인하거나 저장할 필요 없이 상황에 따라 레이아웃을 자동으로 생성하는 모델을 제안하였다. 비디오 월 컨트롤러의 운용 중에 발생할 수 있는 상황의 유형과 상황 유형별 레이아웃의 이미지 데이터로 CGAN을 훈련하고 상황 유형을 CGAN에 제공하여 레이아웃을 자동으로 생성하였다. 레이아웃을 자동으로 생성함으로써 레이아웃의 디자인과 관리에 소요되는 시간과 비용을 절감할 수 있을 것으로 예상된다. 한편, 본 논문에서는 실험상의 편의

를 위해 8단 8열의 모니터로 구성된 시스템을 사용하였지만 일반적으로는 4단 2열이나 6단 4열 등 가로 방향의 모니터 수가 많은 구성을 사용하고 있으므로 직사각형 모양의 모니터 구성에서도 활용할 수 있도록 모델을 개선해나갈 계획이다.

참고문헌

- [1] Mehdi Mirza, Simon Osindero, “Conditional Generative Adversarial Nets”, 2014, <https://arxiv.org/abs/1411.1784>.
- [2] Ian J. Goodfellow et al., “Generative Adversarial Networks,” 2014, <http://arxiv.org/abs/1406.2661>.
- [3] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, 2014, <http://arxiv.org/abs/1412.6980>.