일반논문 (Regular Paper)
방송공학회논문지 제30권 제1호, 2025년 1월 (JBE Vol.30, No.1, January 2025)
https://doi.org/10.5909/JBE.2025.30.1.61
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

객체 추적을 활용한 비디오 인페인팅용 사용자 기반 마스크 생성 기법

김 은 지[a)], 김 동 휘[a)], 강 다 빈[a)], 송 호 준[a)], 박 상 효[a)‡]

User-based Mask Generation Method using Object Tracking for Video Inpainting

Eunji Kim[a)], Dong-hwi Kim[a)], Dabin Kang[a)], Hojun Song[a)], and Sang-hyo Park[a)‡]

요 약

미디어 촬영 과정에서 의도치 않은 행인 등의 객체가 포함되는 경우, 사용자는 그러한 객체를 가리는 방식을 사용할 수 있지만, 동시에 미디어의 부자연스러움이 증가할 수 있다. 이때 객체를 가리는 대신 자연스레 삭제할 수 있는 비디오 인페인팅 기술을 적용할 수 있으나, 이는 모든 프레임마다 마스크가 필요하여 객체의 위치를 일일이 레이블링하는 비용이 필연적으로 발생한다. 이를 해결하기 위해 본 논문에서는 사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 인페인팅 프레임워크를 제안한다. 본 프레임워크에서는 객체 제거를 위한 비디오 인페인팅을 진행하기 위해 객체 추적 기술과 분할 기술을 결합하여 자동으로 레이블링한다. 이때 사용자가 지정한 객체를 추적하기 때문에 다양한 객체가 등장하더라도 원하는 객체만 제거한 뒤 인페인팅 할 수 있다. 제안된 기법은 비디오 인페인팅에서 비디오 분할 기술만 활용해 마스크를 생성했을 때보다 더 높은 결과를 기록했으며 프레임 당 처리 시간도 절반 수준으로 감소시켰다.

Abstract

When unintended objects such as passersby appear in media, traditional methods like covering the object are used to remove them, which can lead to unnatural results. Instead of covering objects, video inpainting can offer a better alternative. However, video inpainting needs to generate masks for every frame, which can be quite costly. In this paper, we propose user-based object tracking method for video inpainting to address this problem. Our framework combines object tracking and segmentation to automatically generate masks for object removal in video inpainting. Since it tracks user-specified objects, it enables the selective removal of desired objects, even when multiple objects are present. The proposed method achieves better results compared to using only video segmentation for mask generation and reduces per-frame execution time by half.

Keyword : Visual object tracking, Video inpainting, Image segmentation, Video generation

a) School of Computer Science and Engineering, Kyungpook National University
‡ Corresponding Author : Sang-hyo Park
E-mail: s.park@knu.ac.kr
Tel: +82-53-950-6373
ORCID: https://orcid.org/0000-0002-7282-7686

※ 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2023-00227431, 3차원 공간 디지털미디어 규격화 기술 개발)
· Manuscript December 23, 2024; Revised December 24, 2024; Accepted January 10, 2024.

Ⅰ. 서 론

미디어 콘텐츠 산업이 급격히 발전하면서 이에 따른 미디어 후처리 기술의 중요성 또한 함께 부각되고 있다. 미디어 촬영 과정에서 때때로 행인 등 의도치 않은 객체가 포함되는 경우가 발생하고, 이를 수정하기 위해 미디어를 다시 생성하는 것은 시간과 비용 면에서 큰 부담이 된다. 기존에는 해당 객체를 모자이크로 가리는 방법을 사용할 수 있으나 이는 미디어의 부자연스러움을 증가시킨다. 이때 객체를 가리는 대신 제거하여 해결할 수 있다. 따라서 이미지나 비디오의 불필요한 객체를 제거하고 이를 자연스럽게 채워나가는 인페인팅 기술의 중요성은 날로 커지고 있다.

초기의 인페인팅 기술은 손상되거나 가려진 부분을 복원하는 데 중점을 두었다[1,2]. 딥러닝 기술의 발전으로 인해 넓은 영역의 손상 복원이 가능해졌고, 최근에는 미디어 복원의 영역을 넘어 객체를 삭제하거나 새로운 객체로 대체하는 기술로 확장되고 있다[3,4]. 그러나 이미지나 비디오에서 원하는 객체를 정확히 삭제하기 위해서는 우선으로 해당 객체의 정확한 영역을 표현하는 마스크가 필수적이다. 예컨대 만약 마스크가 객체의 영역을 정확하게 표현하지 못하면 삭제하고자 하는 객체의 일부분이 남아 부자연스러운 결과를 초래할 수 있다. 또한 이미지의 경우 사용자가 객체의 영역을 직접 지정하여 마스크를 빠르게 생성할 수 있지만, 비디오에서는 프레임 전체에 대한 마스크가 필요하기 때문에 레이블링 비용이 필연적으로 발생한다. 따라서 비디오 인페인팅 과정에서 비용을 효율적으로 줄이거나 최소화하기 위해서는 자동으로 마스크를 생성하는 기술이 필요하다.

본 논문에서는 비디오 인페인팅 과정에서 발생하는 레이블링 비용을 감소시키기 위해 사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 인페인팅 프레임워크를 제안한다. 제안한 프레임워크는 사용자가 첫 번째 프레임에서 삭제할 객체의 위치를 지정하면 객체 추적을 통해 전체 비디오에서 객체의 위치를 파악한다. 그 후, 객체의 위치 정보를 기반으로 해당 객체를 분할하여 마스크를 생성한다. 생성된 마스크는 비디오 인페인팅 모델의 입력으로 들어가 마스크 영역에 해당하는 객체를 제거한 뒤 자연스럽게 채워 넣는다. 위 과정을 통해 본 프레임워크는 비디오에서 지정한 객체를 제거한 비디오를 생성한다. 그리고 생성된 비디오를 정량적, 정성적 평가를 통해 비디오 분할로 만들어진 마스크를 사용한 인페인팅 결과와 비교하여 효율성을 입증했다. 본 논문에서 제안한 객체 추적과 이미지 분할을 결합한 기법은 비디오 분할만 적용한 결과보다 정량적 평가에서 좋은 결과를 보였으며 프레임 당 실행 시간을 약 1/2 수준으로 감소시켜 효과적임을 입증하였다.

Ⅱ. 관련 연구

1. 객체 추적

객체 추적(Visual Object Tracking)은 3D 프레임 시퀀스와 초기 객체 상태를 입력받아 전체 프레임에서 객체의 위치를 추적하는 기술이다. 딥러닝 이전에는 칼만 필터와 같은 기법을 사용하여 과거 상태로부터 객체를 추적했다[5]. 딥러닝이 도입되면서 빠르고 효율적인 추적 모델이 등장하였으나[6], 외형이 변화하면 성능 저하가 발생하는 단점이 있었다. 이를 해결하기 위해 [7]은 시각적 템플릿 매칭과 모션 모델링을 통합한 프레임워크를 제안했다. 이러한 외형 변화에 강인한 객체 추적 기법은 비디오에서 연속적인 객체의 위치 정보를 빠르게 파악할 수 있어서 본 논문에서는 [7]을 활용하여 자동으로 마스크를 생성하는 비디오 인페인팅 프레임워크를 제안한다.

2. 객체 제거를 위한 비디오 인페인팅

비디오 인페인팅은 비디오에서 객체를 삭제하거나 결손난 부분을 복원하는 기술이다. 이미지와 다르게 비디오는 시간적 정보를 가지고 있으며 초기의 비디오 인페인팅 기법은 프레임 간 객체의 움직임을 추적하는 흐름 기반 기법들을 사용했다[8,9]. 그러나 초기의 흐름 기반 기법들은 수작업 연산이 많아 속도 저하 문제가 있었다. 이를 해결하기 위해 Flow Completion을 다운샘플된 영역에서 단일 단계로 처리하여 속도를 개선한 프레임워크가 고안되었고[10], 컨볼루션과 어텐션을 활용한 비디오 인페인팅 기법도 등장

했다[11,12]. 특히, [12]는 프레임 간 정보가 연속적으로 연결되지 않을 때 발생하는 성능 저하 문제를 해결하기 위해 이미지 인페인팅과 비디오 인페인팅 모델을 결합하여 SOTA(State-of-the-Art) 성능을 달성하였다. 이러한 비디오 인페인팅 기술들은 삭제할 부분의 영역을 나타낸 마스크가 필요하며, 객체를 제거하기 위해서는 프레임마다 객체의 위치를 레이블링하는 비용이 필수적으로 발생한다. 비디오 인페인팅에서는 비디오의 길이가 길어질수록 프레임 수가 많아져 레이블링 비용이 많이 발생하지만, 현재의 연구들은 인페인팅 과정의 속도를 향상하는 것에 집중하여 발전하였다. 따라서 본 논문에서는 객체 추적 기술을 도입하여 객체 제거를 위한 비디오 인페인팅 기술에서 객체의 위치를 레이블링하는 비용을 절감하여 기존의 문제를 해결하였다.

3. 분할 기술과 마스크 생성

분할(Segmentation)은 이미지나 비디오 내에서 특정 객체나 영역을 구분하는 컴퓨터비전 기술 중 하나이다. 이미지 분할은 컨볼루션 기반 모델의 등장으로 성능과 속도 부분에서 큰 발전을 이루었으며 최근에는 사전 학습된 거대한 데이터셋을 기반으로 다양한 객체를 분할하는 SAM (Segment Anything) 모델이 등장하였다[13]. 비디오도 연속된 이미지로 해석하여 이미지 분할 기술을 적용할 수 있으나, 객체를 일관성 있게 추적하지 못하는 단점을 해결하기 위해 광학 흐름을 적용하는 연구가 많이 이루어졌다[14,15]. SAM 모델은 후에 주의 기법을 적용하여 광역적인 프레임 정보를 통해 비디오 분할에도 사용할 수 있게 발전하였다[16]. 이처럼 이미지와 비디오 분할 기술의 정확도가 높아지면서 농업, 의료 영상 분석 등 다양한 분야에서 사용되고 있다[17,18]. 인페인팅 분야에서도 SAM 모델을 결합해 마스크 생성 과정을 자동화하려는 시도가 있었으나[19], 비디오 인페인팅에서 마스크를 자동으로 생성하는 연구는 아직 충분히 이루어지지 않았다. 이에 본 논문에서는 비디오 인페인팅에서 객체를 표현하는 마스크를 자동으로 생성하기 위해 분할 모델을 활용하는 방법을 제안한다. 비디오 마스크 생성을 위해 비디오 분할 모델을 사용할 수도 있지만, 시간적 처리 문제로 인해 속도가 저하되는 단점이 발생한다. 따라서 본 연구는 객체 추적 모델과 이미지 분할 모델을 결합하여 성능을 유지하면서 처리 속도를 향상했다.

Ⅲ. 제안 기법

본 논문에서는 사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 인페인팅 프레임워크를 제안한다. 제안한 프레임워크는 비디오와 첫 프레임에서의 객체의 위치를 토대로 최종적으로 객체가 제거된 비디오를 출력한다. 그림 1은 제안한 프레임워크의 작동 과정을 시각적으로 나타낸 것이며, 전체 프로세스는 크게 세 단계로 이루어진다. 첫 번째 단계에서는 사용자의 입력을 기반으로 객체의 위치를 바운딩박스로 변환한다. 입력 데이터는 객체의 위치를 마스크나 바운딩박스 좌표 형태로 제공할 수 있다. 두 번째 단계에서는 객체 기반 마스크 생성자를 통해 모든 프레임에서의 객체 위치를 파악하고 분할을 통해 자동으로 마스크를 생성한다. 마지막 단계에서는 생성된 마스크를 활용해 비디오 인페인팅을 수행하여 최종적으로 객체가 제거된 비디오를 생성한다. 이 프레임워크의 각 단계에 대한 구체적인 설명과 작동 방식은 뒤에서 자세히 다룰 예정이다. 이를 통해 본 프레임워크는 첫 프레임에서의 객체 위치 정보만을 사용해 비디오 전체에서 객체를 제거할 수 있어 모든 프레임에 대해 마스크가 필요했던 기존 비디오 인페인팅 기술의 한계를 보완할 수 있다.

1. 사용자 기반 객체 선택

비디오 인페인팅을 위한 마스크 생성을 위해 본 프레임워크는 객체 추적 기술과 이미지 분할 기술을 결합하여 사용한다. 먼저, 삭제할 객체의 위치 정보를 추적 모델에 전달해야 하며 본 프레임워크는 객체 위치를 나타내는 바운딩박스 또는 마스크 형태의 입력을 허용한다. 만약 첫 프레임에서 객체 위치를 나타내는 마스크가 입력된 경우에는 경계 부분을 판단하여 객체의 위치를 바운딩박스로 변환한다. 만약 마스크가 없는 경우에는 사용자가 직접 객체의 영역을 $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ 형태로 제공해야 한다.

2. 객체 기반 마스크 생성자

객체 기반 마스크 생성자는 객체 추적 기술과 이미지 분할 기술을 결합하여 마스크를 생성한다. 이때, 사용자는 추적될 객체의 구체적인 정보를 선택적으로 입력할 수 있으며 이 정보는 객체 추적 과정에서 보조적인 역할을 한다. 추적된 객체의 위치는 각 프레임마다 바운딩박스 $(B_1, B_2, \ldots B_t, \ldots B_n)$로 반환되며 이는 객체 기반 이미지 분할 모델의 입력으로 사용된다. 객체 기반 이미지 분할 모델은 추적을 통해 파악된 객체가 있는 부분만 집중하여 분할하기 때문에, 이미지 전체 영역을 처리하는 기존의 방식보다 효율적인 분할이 가능하다. 또한, 분할은 프레임을 각각의 이미지로 보고 독립적으로 진행되어 병렬적인 처리가 가능하고, 이를 통해 마스크 $M_G(M_1, M_2, \ldots M_t, \ldots M_n)$가 생성된다.

3. 비디오 인페인팅

비디오 인페인팅 과정에서는 객체 기반 마스크 생성자를 활용해 생성된 마스크 $M_G(M_1, M_2, \ldots M_t, \ldots M_n)$를 사용하여 비디오 내 객체의 영역을 명확히 지정한 후, 해당 영역을 배경으로 자연스럽게 채워 객체를 삭제한다. 이를 통해 비디오에서 객체 제거를 위해 인페인팅을 수행할 때 모든 프레임에 대한 마스크를 사용자가 직접 제공해야 했던 기존의 문제를 해결하고, 첫 번째 프레임에서 제공된 객체 위치 정보만으로 전체 비디오에서 객체를 제거한 영상을 생성할 수 있다.

4. 사용 모델

객체 제거를 위한 자동 마스크 생성 인페인팅 프레임워크에서는 시각적 템플릿 매칭으로 객체의 외형 변화에 강인한 ARTrack[7]을 객체 추적 모델로 사용하였다. 추적된 객체의 위치 정보를 바탕으로 Segment Anything 2(SAM 2)[16]를 활용하여 이미지 분할을 수행하였다. SAM 2는 비디오 분할 모델이지만 각 프레임을 각각 독립적인 단일 프레임 비디오로 간주하여 이미지 분할처럼 활용하였다. 이후, 생성된 마스크는 E2FGVI[10] 모델을 통해 인페인팅을 진행하였다. 마스크 생성 효율성을 검증하기 위해 SAM 2로 비디오 분할을 진행해 마스크를 생성한 후 이를 인페인

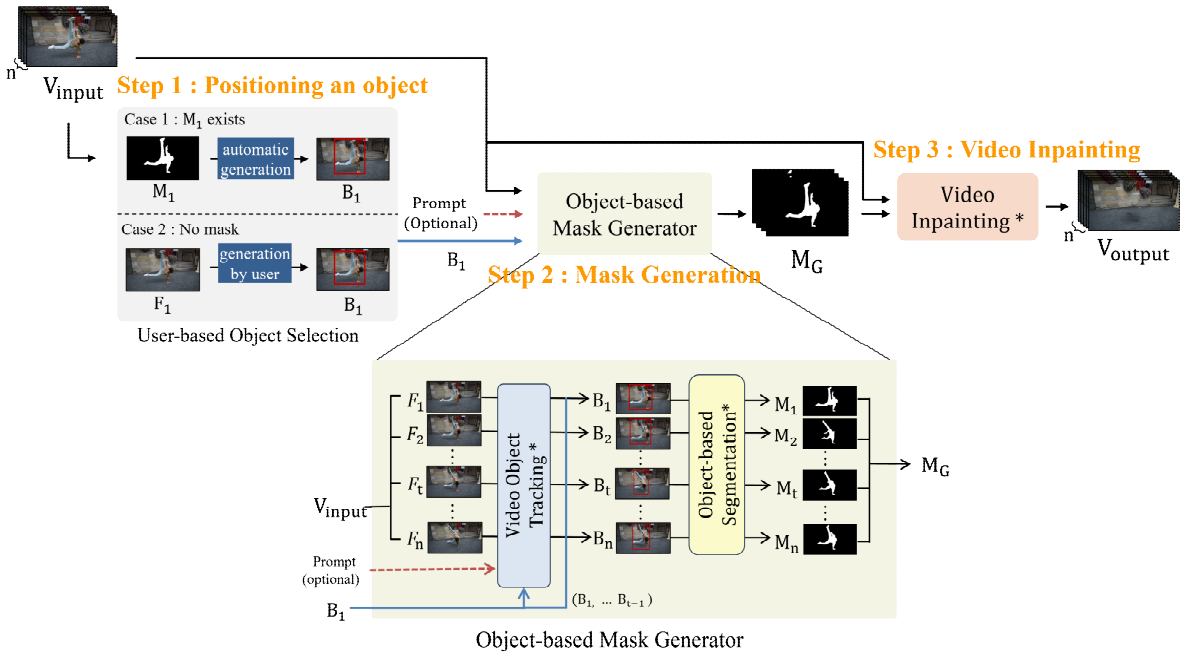


그림 1. 사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 비디오 인페인팅 프레임워크
Fig. 1. An automatic mask generation video inpainting framework based on object visual tracking for object removal

팅에 적용하여 성능을 비교하였다.

Ⅳ. 실험

1. 데이터셋

본 논문에서는 Youtube-VOS[20], DAVIS[21] 데이터셋을 사용하였다. Youtube-VOS는 4,450개의 비디오와 94개의 카테고리를 가지고 있는 대규모 비디오 객체 분할 데이터셋이다. 본 논문의 프레임워크는 단일 객체를 삭제한 뒤 인페인팅 하는 것에 초점을 두고 있으므로, 단일 객체에 대한 마스크를 제공하는 1,476개 중 500개를 임의로 선정하여 실험에 사용하였다. Youtube-VOS는 한 비디오당 평균 150개의 프레임을 제공하고 있으며, 마스크는 5프레임마다 제공된다. DAVIS 데이터셋은 비디오 분할을 위한 데이터셋으로 약 150개의 비디오를 포함하고 있다. 이 중 단일 객체에 대한 마스크를 제공하는 31개의 비디오를 사용하여 실험하였다. DAVIS는 Youtube-VOS와는 다르게 전체 프레임에 대한 마스크를 제공하고 있다. 객체 추적 과정에서 선택적으로 입력받는 객체 정보의 경우 DAVIS는 'bear', 'person' 등 객체의 정보를 직접 판단해 입력으로 넣었으나 Youtube-VOS는 데이터의 수가 많아 객체의 정보를 입력하지 않고 실험을 진행하였다.

2. 실험 환경

본 논문의 실험은 NVIDIA RTX 3060 (VRAM 12GB), RAM 32GB 환경에서 수행되었다. 이미지 분할 모델을 이용한 인페인팅은 추적, 분할, 인페인팅이 통합된 프레임워크를 사용하여 한 번에 처리되었다. 반면, 비디오 분할 인페인팅은 메모리 부족 문제로 인해 분할, 인페인팅을 각각 분리하여 진행하였다.

3. 평가 지표

본 논문에서 제안한 기법의 성능을 측정하기 위해 PSNR, SSIM[22], VFID[23], E_{warp} [24]를 사용하였다. PSNR과 SSIM은 이미지, 비디오 인페인팅 분야에서 널리 사용되는 평가 지표이다. PSNR은 최대 신호 대 잡음비로 가질 수 있는 최대 전력에 대한 잡음 면적의 비를 나타내며, SSIM은 왜곡에 대한 원본 영상의 유사도를 측정하는 기법이다. 두 평가 지표 모두 높은 값을 나타낼수록 좋은 성능을 의미한다. video-based Frechet inception distance(VFID)는 사전 학습된 비디오 인지 네트워크를 기반으로 시공간적인 특징 맵을 추출한다. 그 후, FID와 동일한 과정으로 계산을 진행한다. 본 논문에서는 특징 맵을 추출하기 위해 I3D backbone[25]을 사용한다. E_{warp}는 전체 프레임에 대한 평균적인 warping error를 계산한다.

4. 실험 결과

4.1. 정량적 평가

사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 비디오 인페인팅 프레임워크의 정량적 평가는 Youtube-VOS와 DAVIS 데이터셋을 사용하여 진행되었으며 그 결

표 1. DAVIS, Youtube-VOS 데이터셋에 대한 정량적 평가. Original Mask는 데이터셋에서 제공한 마스크로 인페인팅 한 결과에 대한 성능이다. E_{warp}*는 $E_{warp} \times 10^{-2}$를 나타낸다. ↑는 더 높은 값이 더 좋은 성능임을 의미하고 ↓는 더 낮은 값이 좋은 성능임을 의미한다.
Table 1. Quantitative comparisons with our framework on DAVIS and Youtube-VOS datasets. Original Mask is performance of inpainting with masks provided in the datasets. E_{warp}* denotes $E_{warp} \times 10^{-2}$. ↑ indicates higher is better and ↓ indicates lower is better.

Mask generation Method	DAVIS				Youtube-VOS			
	PSNR↑	SSIM↑	VFID↓	E_{warp}*↓	PSNR↑	SSIM↑	VFID↓	E_{warp}*↓
Original Mask	32.19	0.9662	0.2858	0.1422	34.14	0.9759	0.2503	0.2964
Video Segmentation	31.69	0.9641	0.3249	0.1421	33.53	0.9671	0.3495	0.3053
Ours	32.27	0.9667	0.2859	0.1415	33.57	0.9696	0.3410	0.3035

과는 표 1과 같다. 원본 마스크는 각 데이터셋에서 기본적으로 제공된 마스크를 활용해 인페인팅을 수행한 결과를 나타낸다. 비디오 분할은 비디오 분할 모델을 통해 생성된 마스크로 인페인팅 결과로 SAM 2 모델을 사용하였다. DAVIS 데이터셋에서는 PSNR 32.27, SSIM 0.9667, E_{warp}* 0.1415로 본 논문에서 제안한 기법이 원본 마스크보다 높은 성능이 나왔다. 이러한 결과가 가능한 이유는 원본 마스크는 객체의 경계를 정확히 구분하지만, 때로는 주변 그림자 등으로 인해 객체 경계보다 더 넓은 영역을 제거해야 자연스러운 결과를 얻을 수 있기 때문으로 분석된다. Youtube-VOS 데이터셋에서는 본 논문의 기법이 원본 마스크를 사용한 방식보다는 성능이 낮았으나 비디오 분할 방식보다는 우수한 결과를 보였다. 특히, DAVIS와 Youtube-VOS 모두 본 논문의 방식이 비디오 분할 방식보다 더 우수한 성능을 보여, 제안된 기법이 인페인팅을 위한 마스크를 효과적으로 생성하고 있음을 확인할 수 있다.

4.2. 정성적 평가

본 논문의 제안 기법과 비디오 분할을 통해 나온 결과를 시각적으로 비교한 결과, 모두 객체의 움직임이나 외형 변화가 크지 않으면 안정적인 성능을 보여주었다. 하지만, 객체가 빠르게 움직일 때는 일부 프레임에서 부자연스러운 결과를 보였다. 그림 2는 Youtube-VOS와 DAVIS 데이터셋에서 객체의 움직임이 많은 비디오에 대한 결과를 시각적으로 나타낸다. 첫 번째와 두 번째 비디오에서는 모든 모델이 객체를 잘 분할한 뒤 자연스럽게 제거하였다. 반면, 세 번째 비디오처럼 새의 날갯짓 등 움직임이 큰 객체의 경우 비디오 분할 방식이 제안 기법보다 더 자연스러운 결과를 보였다. 하지만 비디오 분할 방식은 초기 단계에서 객체를 제대로 분할하지 못하면 이후 프레임에서도 올바르게 분할되지 않는 한계가 있었다. 그림 2의 네 번째 비디오에서 비디오 분할 방식은 첫 프레임부터 물고기를 분할하지 못해, 결과적으로 모든 프레임에서 객체를 제거하지 못하는 문제가 발생하였다. 따라서 비디오 분할 모델은 첫 프레임에서 객체를 분할하지 못하면 이에 대한 손실 비용이 크게 나타남을 알 수 있다. 반면, 제안 기법은 분할이 프레임별로 별도로 진행되기 때문에 이러한 위험을 줄일 수 있다. 이를 통해 제안된 기법이 객체의 움직임이 심한 경우에는 제한적인 성능을 보이지만 누적된 오류로 전체 성능이 저하되는 문제를 효과적으로 방지할 수 있음을 확인할 수 있다.

4.3. 효율성 비교

표 2는 프레임 당 전체 실행 시간을 나타낸다. 본 논문의

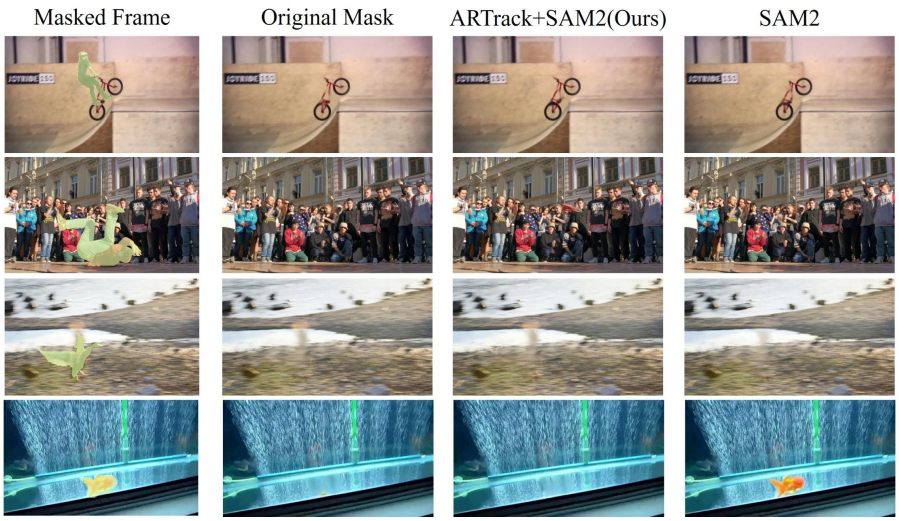


그림 2. DAVIS, Youtube-VOS 데이터셋 중 객체의 움직임이 빠른 비디오에 대한 결과
Fig. 2. Qualitative results for videos with fast moving objects among DAVIS and Youtube-VOS datasets

표 2. DAVIS, Youtube-VOS 데이터셋에 분할 모델에 따른 실행 시간
Table 2. Execution time according to segmentation model on DAVIS and Youtube-VOS datasets

Methods	Execution time (s/frame)	
	DAVIS	Youtube-VOS
Ours	0.607963	1.201448
Video Segmentation	0.749175	1.340546

제안 기법은 추적, 이미지 분할, 인페인팅 총 3개의 과정으로 이루어졌으며, 비디오 분할 모델로 마스크를 생성한 방식은 비디오 분할, 인페인팅, 총 2개의 과정으로 진행되었다. 실행 시간을 비교한 결과 본 논문의 방식이 DAVIS에서 0.6초, Youtube-VOS에서 1.2초로 비디오 분할 방식보다 약 0.14초씩 빨랐다. 또한 본 논문의 실험 환경인 RTX 3060을 사용한 경우, 비디오 분할 모델은 인페인팅 모델과 결합하면 메모리 초과가 발생해 따로 실행한 뒤 각각의 실행 시간을 합산했다. 이에 반해 본 논문의 기법은 객체 선택부터 인페인팅까지 전체 과정을 통합하여 실행해도 자원 문제가 발생하지 않았다. 이로써 본 논문에서 제안한 프레임워크를 따르는 과정이 실행 시간이 짧고 효율적임을 알 수 있다.

Ⅴ. 절제 연구(Ablation Study)

1. 다양한 이미지 분할 모델에 대한 실험

본 논문의 제안 기법은 이미지 분할 모델의 성능에 크게 영향을 받는다. 따라서 분할 모델을 선정하기 위해 추적 모델(ARTrack)과 인페인팅 모델(E2FGVI)은 동일하게 유지한 채 이미지 분할 모델만 바꾸어 비교 실험을 수행하였다. 실험에서는 SAM 1과 비디오 분할 모델인 SAM 2를 비교하였다. SAM 2는 한 장짜리 비디오로 간주에 이미지 분할처럼 활용하였다. 표 3에서 확인할 수 있듯이 SSIM가 약 0.01, VFID가 약 0.001 차이의 차이로 두 모델의 성능은 유사했으나 전반적으로 SAM 2가 SAM 1보다 더 나은 결과를 보였다. 특히, SAM 2는 프레임당 실행 시간이 SAM 1 대비 절반 수준으로 감소하여 효율성 면에서도 우수한 성능을 보였다. 이러한 결과를 바탕으로 따라서 본 논문에서는 SAM 2를 이미지 분할 모델로 채택하였다.

2. 마스크 없이 수행되는 비디오 인페인팅 모델과의 성능 비교

표 3. 이미지 분할 모델의 변화에 따른 실험 결과
Table 3. Experimental results based on the change of image segmentation model

Method	DAVIS				Execution time (frame/s)
	PSNR	SSIM	VFID	E_{warp}*	
SAM 1	32.18	0.9657	0.2781	0.1428	1.417961
SAM 2	32.27	0.9667	0.2859	0.1415	0.607963

표 4. 마스크 없이 수행되는 비디오 인페인팅 모델의 성능 비교
Table 4. Performance comparison of video inpainting models performed without Masks

Method	DAVIS			
	PSNR	SSIM	VFID	E_{warp}*
Inpaint Anything	27.40	0.9131	1.8295	0.1504
LGVI	24.29	0.8162	1.7506	0.2306
Ours	32.27	0.9667	0.2859	0.1415

비디오 인페인팅 분야에서는 레이블링 비용을 줄이기 위한 여러 연구가 진행되어왔다. Language Driven Video Inpainting (LGVI)[26]는 입력된 프롬프트를 활용해 객체를 자동으로 탐지하고 마스크를 생성하여 비디오 인페인팅을 수행한다. Inpainting Anything[19]은 본 논문과 유사하게 추적, 분할, 인페인팅을 결합한 구조로 각각 OSTrack[27], SAM, STTN[28]을 활용한다. Inpainting Anything은 객체의 위치를 한 점의 좌표로 입력받기에 바운딩박스를 입력받아 처리하는 본 논문과 차이점이 있다. 이들의 효율성을 검증하기 위해 앞서 소개한 두 모델을 DAVIS 데이터셋으로 테스트하였다. LGVI의 프롬프트는 ‘remove the [object]’로 통일하였으며, 여기서 [object]는 제거 대상 객체의 이름을 의미한다. Inpainting Anything의 경우 삭제하려는 객체의 정중앙 좌표를 수작업으로 찾아 입력하였다. 정량적 비교 결과는 표 4에 제시되어 있다. 표 4의 결과를 통해 본 논문의 제안 기법이 LGVI와 Inpainting Anything에 비해 PSNR, SSIM, VFID, E_{warp}*가 모두 높았음을 확인할 수 있다. 따라서 이는 본 논문의 방법이 인페인팅을 위한 마스크 생성 과정에서 효과적임을 입증한다.

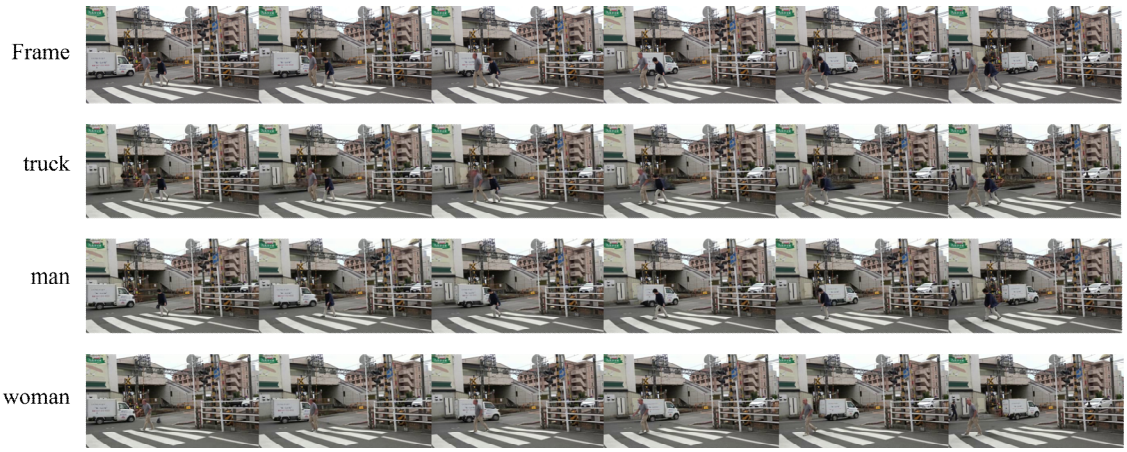


그림 3. 여러 객체가 등장하는 비디오에서 서로 다른 객체를 지정해 인페인팅 한 결과
Fig 3. Result of inpainting by specifying different objects in a video containing multiple objects

3. 단일 비디오 내 여러 객체를 대상으로 한 실험

본 논문에서 제안한 프레임워크는 삭제하고자 하는 객체를 직접 지정할 수 있다. 한 영상에 다양한 객체가 나오는 경우나 여러 객체가 겹치는 경우 서로 다른 객체를 지정해 인페인팅을 진행할 수 있다. 그림 3은 여러 객체가 등장하는 비디오에서 서로 다른 객체를 지정해 인페인팅 한 결과를 나타낸다. 영상에는 남자, 여자, 트럭 총 3개의 주요 객체가 등장한다. 중간에 남자와 여자가 각각 트럭과 겹친다. 그럼에도 불구하고 다른 객체와 혼동하지 않고 객체가 적절하게 삭제되었다.

4. 생성 비디오에 대한 비디오 인페인팅 실험

딥러닝 기술의 발전으로 AI를 활용해 비디오를 생성하는 사례가 점차 증가하고 있다[29]. 그러나 생성된 비디오에 원하지 않는 객체가 포함된다면 비디오를 재생성해야 한다. 하지만, 재생성은 비용 측면에서 손실을 초래할 수 있으며 원하는 결과를 보장할 수 없다. 재생성 대신 비디오 인페인팅 기술 활용을 고려할 수 있으나 생성된 비디오에는 마스크가 없기에 비디오 인페인팅을 수행할 수 없다. 본 논문에서 제안한 프레임워크는 자동으로 마스크를 생성하기 때문에 이러한 문제점을 해결할 수 있다. 다만, 자연 비디오의 분포와 생성 비디오의 분포가 다르므로 제안된 프레임워크가 생성 비디오에 적합하게 적용되는지 검증하는 과정이 필요하다.

본 논문에서는 SORA[30], Emu Video[31], Runway Gen-2[32]로 생성한 비디오에 대해 인페인팅 성능을 검증하였다. SORA와 Emu Video는 해당 모델로 생성한 뒤 배포된 비디오를 사용하였다. Runway Gen-2는 DAVIS 데이터셋의 첫

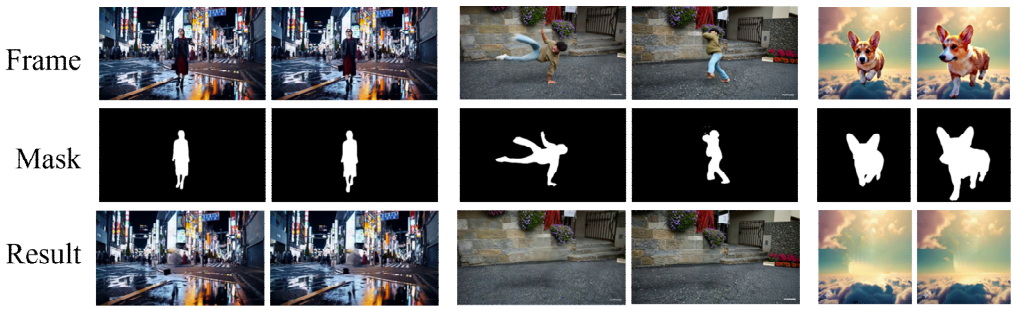


그림 4. SORA, Runway, Emu Video로 생성한 비디오에 대한 프레임워크 적용 실험 결과
Fig. 4. Experiment results of the videos generated by SORA, Runway, and Meta Emu Video

프레임을 입력하여 생성 이미지를 생성하였다. 실험 시간 단축을 위해 SORA와 Runway Gen-2의 비디오는 모델은 해상도를 432×240으로, Emu Video는 512×512로 변환하였다. 그림 4는 각 모델로 인페인팅한 결과를 보여준다. 세 개의 생성 모델 모두 원하는 객체를 성공적으로 제거한 결과를 확인할 수 있었다. Runway로 생성한 비디오에서는 사람의 다리가 비정상적으로 휘어지는 등 의미론적으로 부자연스러운 프레임이 생성되었으나 정상적으로 객체를 추적해 삭제하는 모습을 보였다.

Ⅵ. 결 론

본 논문에서는 객체 제거를 위한 비디오 인페인팅 과정에서 모든 프레임에서의 마스크를 생성해야 하는 문제를 해결하기 위해 사용자 기반 객체 추적 기법을 도입한 자동 마스크 생성 인페인팅 프레임워크를 제안하였다. 사용자가 삭제하고자 하는 객체를 지정하면, 객체 추적을 통해 각 프레임에서 객체의 위치를 파악하고, 해당 위치를 기반으로 이미지 분할을 수행한다. 이렇게 생성된 마스크는 비디오 인페인팅에 사용되며, 최종적으로 객체가 삭제된 비디오를 출력한다. 프레임워크의 성능과 효율성을 검증하기 위해 비디오 분할로 생성한 마스크와 비교한 결과 추적과 분할을 결합한 본 논문의 방식이 가장 높은 성능을 기록했다. 특히, 이 방식은 비디오 분할에 비해 프레임 당 실행 시간을 절반 수준으로 낮추었다. 이를 통해 마스크가 없는 비디오에서도 제한 없이 인페인팅 기술을 적용할 수 있을 것으로 기대한다.

참 고 문 헌 (References)

[1] Ballester, Coloma, et al. “Filling-in by joint interpolation of vector fields and gray levels.” IEEE transactions on image processing Vol.10 No.8 pp.1200-1211. August 2001.
doi: https://doi.org/10.1109/83.935036

[2] Bertalmio, M. “Image Inpainting.” 2000.

[3] Li, Wenbo, et al. “Mat: Mask-aware transformer for large hole image inpainting.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.10758-10768. 2022.
doi: https://doi.org/10.48550/arXiv.2203.15270

[4] Lugmayr, Andreas, et al. “Repaint: Inpainting using denoising diffusion probabilistic models.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.11461-11471. June 2022.
doi: https://doi.org/10.48550/arXiv.2201.09865

[5] R. Dugad and N. Ahuja, “Video denoising by combining Kalman and Wiener estimates,” Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), pp.152-156. 1999.
doi: https://doi.org/10.1109/ICIP.1999.819568

[6] Bertinetto, Luca, et al. “Fully-convolutional siamese networks for object tracking.” Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14.Springer International Publishing, pp.850-865. 2016.
doi: https://doi.org/10.1007/978-3-319-48881-3_56

[7] Wei, Xing, et al. “Autoregressive visual tracking.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp.9697-9706. June 2023.
doi: https://doi.org/10.1109/CVPR52729.2023.00935

[8] Kim, Dahun, et al. “Deep video inpainting.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.5792-5801. June 2019.
doi: https://doi.org/10.1109/CVPR.2019.00594

[9] Gao, Chen, et al. “Flow-edge guided video completion.” Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16. Springer International Publishing, pp.713-729. 2020.
doi: https://doi.org/10.1007/978-3-030-58610-2_42

[10] Li, Zhen, et al. “Towards an end-to-end framework for flow-guided video inpainting.” Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.17562-17571. June 2022.
doi: https://doi.org/10.48550/arXiv.2204.02663

[11] Chang, Ya-Liang, et al. “Free-form video inpainting with 3d gated convolution and temporal patchgan.” Proceedings of the IEEE/CVF International Conference on Computer Vision. pp.9066-9075. October 2019.
doi: https://doi.org/10.1109/ICCV.2019.00916

[12] Yu, Yongsheng, Heng Fan, and Libo Zhang. “Deficiency-aware masked transformer for video inpainting.” arXiv preprint arXiv:2307.08629 2023.
doi: https://doi.org/10.48550/arXiv.2307.08629

[13] Kirillov, Alexander, et al. “Segment anything.” Proceedings of the IEEE/CVF International Conference on Computer Vision. pp.4015-4026. October 2023.
doi: https://doi.org/10.48550/arXiv.2304.02643

[14] Dosovitskiy, Alexey, et al. “Flownet: Learning optical flow with convolutional networks.” Proceedings of the IEEE international conference on computer vision. pp.2758-2766. 2015.
doi: https://doi.org/10.1109/ICCV.2015.31

[15] Zhao, Shengyu, et al. “Maskflownet: Asymmetric feature matching with learnable occlusion mask.” Proceedings of the IEEE/CVF

conference on computer vision and pattern recognition. pp.6278-6287. 2020.
doi: https://doi.org/10.48550/arXiv.2003.10955

[16] Ravi, Nikhila, et al. “Sam 2: Segment anything in images and videos.” arXiv preprint arXiv:2408.00714 2024.
doi: https://doi.org/10.48550/arXiv.2408.00714

[17] Zhang, Yichi, Zhenrong Shen, and Rushi Jiao. “Segment anything model for medical image segmentation: Current applications and future directions.” Computers in Biology and Medicine pp.108238. 2024.
doi: https://doi.org/10.48550/arXiv.2401.03495

[18] Li, Yaqin, et al. “Enhancing agricultural image segmentation with an agricultural segment anything model adapter.” Sensors Vol.23 No.18 pp.7884. 2023.
doi: https://doi.org/10.3390/s23187884

[19] Yu, Tao, et al. “Inpaint anything: Segment anything meets image inpainting.” arXiv preprint arXiv:2304.06790 2023.
doi: https://doi.org/10.48550/arXiv.2304.06790

[20] Xu, Ning, et al. “Youtube-vos: Sequence-to-sequence video object segmentation.” Proceedings of the European conference on computer vision (ECCV). pp.585-601. September 2018.
doi: https://doi.org/10.1007/978-3-030-01228-1_36

[21] Perazzi, Federico, et al. “A benchmark dataset and evaluation methodology for video object segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition. pp.724-732. June 2016.
doi: https://doi.org/10.1109/CVPR.2016.85

[22] Wang, Zhou, et al. “Image quality assessment: from error visibility to structural similarity.” IEEE transactions on image processing Vol.13 No.4 pp.600-612. April 2004.
doi: https://doi.org/10.1109/TIP.2003.819861

[23] Wang, Ting-Chun, et al. “Video-to-video synthesis.” arXiv preprint arXiv:1808.06601 2018.
doi: https://doi.org/10.48550/arXiv.1808.06601

[24] Lai, Wei-Sheng, et al. “Learning blind video temporal consistency.” Proceedings of the European conference on computer vision (ECCV). pp.170-185. September 2018.
doi: https://doi.org/10.1007/978-3-030-01267-0_11

[25] Carreira, Joao, and Andrew Zisserman. “Quo vadis, action recognition? a new model and the kinetics dataset.” proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp.6299-6308. July 2017.
doi: https://doi.org/10.48550/arXiv.1705.07750

[26] WU, Jianzong, et al. “Towards language-driven video inpainting via multimodal large language models.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. p. 12501-12511. 2024.
doi: https://doi.org/10.48550/arXiv.2401.10226

[27] YE, Botao, et al. “Joint feature learning and relation modeling for tracking: A one-stream framework.” In: European Conference on Computer Vision. Cham: Springer Nature Switzerland, p. 341-357. 2022.
doi: https://doi.org/10.48550/arXiv.2203.11991

[28] ZENG, Yanhong; FU, Jianlong; CHAO, Hongyang. “Learning joint spatial-temporal transformations for video inpainting.” In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16. Springer International Publishing, p. 528-543. 2020.
doi: https://doi.org/10.48550/arXiv.2007.10247

[29] Ho, Jonathan, et al. “Imagen video: High definition video generation with diffusion models.” arXiv preprint arXiv:2210.02303 2022.
doi: https://doi.org/10.48550/arXiv.2210.02303

[30] Liu, Yixin, et al. “Sora: A review on background, technology, limitations, and opportunities of large vision models.” arXiv preprint arXiv:2402.17177 2024.
doi: https://doi.org/10.48550/arXiv.2402.17177

[31] Girdhar, Rohit, et al. “Emu video: Factorizing text-to-video generation by explicit image conditioning.” arXiv preprint arXiv:2311.10709 2023.
doi: https://doi.org/10.48550/arXiv.2311.10709

[32] Esser, Patrick, et al. “Structure and content-guided video synthesis with diffusion models.” Proceedings of the IEEE/CVF International Conference on Computer Vision. pp.7346-7356. 2023.
doi: https://doi.org/10.1109/ICCV51070.2023.00675

저 자 소 개

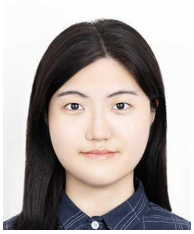
김 은 지

- 2021년 3월 ~ 현재 : 경북대학교 컴퓨터학부 학사과정
- ORCID : https://orcid.org/0009-0008-2319-0325
- 주관심분야 : Computer Vision, knowledge distillation, model compression, video generation

저 자 소 개

김 동 휘

- 2021년 2월 : 대전대학교 전자정보통신공학 학사
- 2023년 2월 : 경북대학교 컴퓨터학부 석사
- 2023년 3월 ~ 현재 : 경북대학교 컴퓨터학부 박사과정
- ORCID : https://orcid.org/0000-0002-5188-8834
- 주관심분야 : Computer vision, Deep Learning, Video compression, Generative Model



강 다 빈

- 2024년 2월 : 경북대학교 컴퓨터학부 학사
- 2024년 3월 ~ 현재 : 경북대학교 컴퓨터학부 석사과정
- ORCID : https://orcid.org/0009-0000-8242-797X
- 주관심분야 : multi-modal learning, text-to-video retrieval, video question & answering, knowledge distillation, 3D scene understanding

송 호 준

- 2024년 2월 : 경북대학교 컴퓨터학부 학사
- 2024년 3월 ~ 현재 : 경북대학교 컴퓨터학부 석사과정
- ORCID : https://orcid.org/0009-0007-4491-6458
- 주관심분야 : model compression, 3D scene understanding, point cloud data processing

박 상 효

- 2011년 2월 : 한양대학교 컴퓨터전공 학사
- 2017년 8월 : 한양대학교 컴퓨터·소프트웨어학과 박사
- 2017년 5월 ~ 2018년 2월 : 전자부품연구원 지능형영상처리센터 Post-doc
- 2018년 3월 ~ 2018년 12월 : 연세대학교 바른ICT연구소 연구원
- 2019년 2월 ~ 2020년 1월 : 이화여자대학교 전자전기공학과 박사후연구원
- 2020년 3월 ~ 현재 : 경북대학교 컴퓨터학부 부교수
- ORCID : https://orcid.org/0000-0002-7282-7686
- 주관심분야 : VVC, Encoding/Decoding Complexity, Omnidirectional Video, Model Compression, Generative Model