딥 러닝 기반 Super Slow 비디오 서비스

이동연, 박진수, 남진우, 최해철
한밭대학교 정보통신공학과
ycbkr123@naver.com, jsp0314@naver.com, doert03@naver.com,
choihc@hanbat.ac.kr

Deep Learning-Based Super Slow Video Service

DongYeon Lee, JinSu Park, JinWoo Nam, Haechul Choi
Dept. of Information and Communication Engineering, HANBAT NATIONAL UNIVERSITY

요약

최근 스포츠 경기나 차량 블랙박스 등에서 비디오를 이용한 판정이 점차 확대되고 있지만, 일반 카메라로 촬영된 비디오에서 정확한 판정을 하기 어려울 때가 빈번히 발생한다. 초고속 카메라로 촬영한 슬로우 모션 비디오를 이용할 수 있다면 판정의 정확성을 향상시킬 수 있을 것이다. 본 논문에서는 일반 카메라로 촬영한 비디오로부터 마치 초고속 카메라로 촬영한 것과 같은 슬로우 모션 비디오를 생성하여 제공하는 서비스를 제안한다. 제안 방법은 NVIDIA에서 개발한 Super Slomo 기술을 기반으로, 초당 30장의 표준 비디오를 초당 60장에서 240장까지의 고품질 슬로우 모션 비디오로 변환한다. 이 기술은 시간적으로 이웃한 두 영상을 입력하여 딥 러닝 기반으로 중간 프레임을 보간함으로써 프레임율을 향상시킨다. 또한 본 논문에서는 Super Slomo 기술에 FP16을 적용하여 처리속도를 향상 시켰으며, 웹 서버를 이용하여 비디오를 업로드하고 슬로우 모션으로 변환된 비디오를 다운로드 할 수 있는 사이트를 구현했다.

1. 제작 동기

축구를 시청하다 보면 골 장면이나 VAR 장면을 슬로우 모션으로 본적이 있을 것이다. 하지만 이런 슬로우 모션 비디오는 콘텐츠 제공자 혹은 심판에 의해서만 볼 수 있다. 본 논문은 사용자가 원하면 언제나 어떤 콘텐츠도 슬로우 모션으로 제공하여, 사용자 주도적 몰입 형 콘텐츠 서비스를 할 수 있는 핵심 기술을 제안한다.

2. 설계 및 구현

영상의 프레임을 늘리기 위한 연구는 국내외에서도 꾸준히 진행되고 있다. 최근에 딥 러닝 분야에 대한 활용이 활발해지며 다양한 알고리즘들 및 컨볼루션을 활용하는데 본 논문은 그 중 Super Slomo를 활용하고 있다.

2.1 딥 러닝 기반 연구

SuperSlomo의 알고리즘을 알기에 앞서 컨볼루션 네트워크인 U-Net구조[4]가 무엇인지 알아야 한다.

Fig.1 U-Net

U-Net은 이미지 분할 (Image Segmentation)을 목적으로 제안된 End-to-End 방식의 컨볼루션 네트워크기반 모델이다. 이 모델은 Contract(수축)와 upsampling(맵 증가시키기)을 이용한 방식이고 Contract 과정에서 image의 크기를 줄 여나가면서 많은 수의 feature들을 가지게 된다. 그리고 다시 segmentation(분할)된 image를 얻기 위해서 upsampling과정을 거치게 된다. Localize를 위해서 contract부분에서의고해상도 feature들을 upsampling과정에서의 대응되는 layer에 copy-and-crop(복사 및 다듬기)을 통해서 concatenation(합)해준다. 이러한 방법으로 인해 더 정확한 출력을 낼 수 있다. SuperSlomo는 컨볼루션 네트워크인 U-Net을 2번 이용한다.

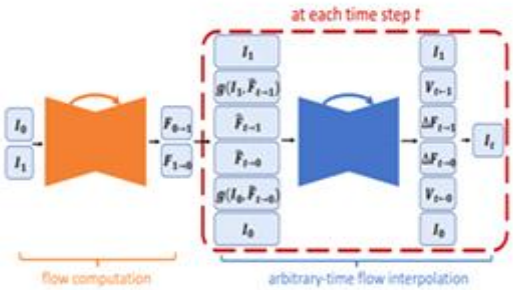

Fig.2 first U-Net

먼저 대략적인 Flow를 계산하기 위해 첫 번째 네트워크에서 U-Net을 사용하여 Optical Flow를 예측한다. 여기서 Optical Flow란, 대상을 기준으로 카메라가 움직이는 방향을 측정하여 흐름을 파악하는 것이다.

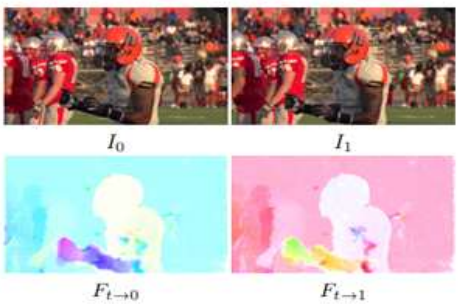

Fig.3 Optical Flow(예시)

계산하여 나온 Optical Flow를 보안하기 위해 다시 U-Net을 사용하여 VisibilityMap을 생성한다.

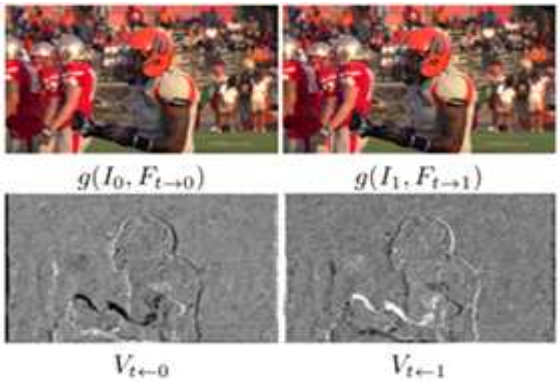

Fig.4 Visibility Map(예시)

가상 이미지 I를 생성하여 픽셀에 대한 이미지(t<-0 or t<-1)를 비교하여 기여도가 높을수록 해당 픽셀의 값이 흰색에 가깝게 한다.
마지막으로 Occlusion을 고려하여 최종 이미지 It를 도출한다[1].

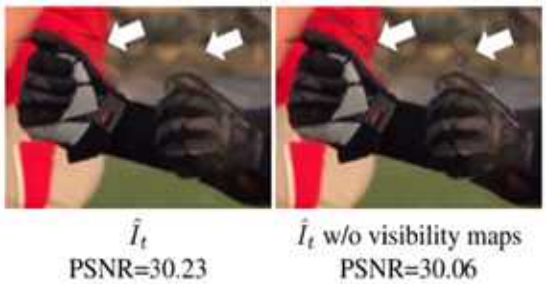

Fig.5 (좌)Artifact제거 후 / (우)Artifact제거 전

2.2 Hyper parameter(하이퍼 파라미터)
Hyper Parameter는 모델링을 할 때 사용자가 직접 세팅해주는 값을 뜻한다. 이것을 사용하면 더욱 정밀화하고 최적화 시킬 수 있다. 본 논문에 쓰일 Hyper parameter는 Learning rate, Epochs, Milestone, Train_batch_size, Loss등이 있다 [3]에서 제공되는 파라미터를 사용했다.

Hyper parameter		Loss	
LEARNING_RATE	0.0001	Loss 1	204
Epochs	200	Loss 2	0.005
MILESTONES	[100 , 150]	Loss 3	102
TRAIN_BATCH_SIZE	6	Loss 4	default

Fig.6 Test과정에서 사용된 Hyper parameter
먼저 Learning rate는 네트워크가 매개 변수를 얼마나 빨리 업데이트 하는지 정의한다. 쉽게 말해 학습속도가 낮으면 학습 과정이 느려지지만 원활하게 수렴된다. 학습 속도가 클수록 학습 속

도가 빨라지지만 수렴되지 않을 수 있다.
Epochs는 전체 데이터를 자신이 설정한 횟수만큼 거치는 것으로 200이라면 전체 데이터를 200번 사용하는 parameter이다.
Milestone은 자신이 지정한 n번째의 Epoches부터 Learning rate에 0.1곱 해서 Training을 하겠다는 의미의 parameter이다.
Train_batch_size는 한번의 Batch마다 주는 데이터 샘플의 size(프레임의 개수를 의미하며, 2장의 프레임이 필요하므로 최소 2개 이상의 size가 필요)를 나타내는 parameter 이다.
마지막으로 Loss는 학습을 통해 얻은 데이터의 추정치가 실제 데이터의 얼마나 차이 나는지 평가하는 것이다. 본 문에 사용된 Loss는 총 4가지로 Loss1: 보간 픽셀 자체의 구현 도를 비교하여 손실률을 측정한다. Loss2: VGG네트워크의 출력을 통한 불완전한 영상의 구조의 정도를 측정한다. Loss3: 흐름 맵의 정확도를 측정한다. Loss4: 흐름 보완하는 맵의 정확도를 측정한다.

2.3 FP16(Floating Point 16-bit)
본 연구에선 RTX NVIDIA 그래픽카드에서 지원하는 FP16기술[2]를 사용한다. FP16은 딥러닝 연산속도를 가속화 하는데 도움이 되는 Tensor코어로써 RTX 버전의 NVIDIA 그래픽카드에 장착 되어있다. Tensor코어를 이용해 PyTorch에서 FP16 dmf 사용하여 훈련하면 FP32 대비 x8의 연산처리량, x2의 메모리 처리량, 1/2의 메모리 사용량의 효과를 볼 수 있어 메모리프로세싱 속도가 증가가 된다.

```
        # Iterate over data.
for batch_size,(inputs, labels) in enumerate(data_loader):
    inputs = inputs.to(device).half()
    labels = labels.to(device)
```

Fig.7 FP16변환을 위한 .half()함수 호출 코드

2.4 웹 서버-클라이언트
본 연구에 사용되는 웹 서버는 Ubuntu 운영체제를 기반으로 Apache2.2.29를 사용했다. 또한 동적 컨텐츠(데이터 요청, 처리된 데이터 리턴) 제공을 위해 Python기반 웹 어플리케이션 서버 Flask를 .wsgi프레임워크를 이용하여 연동시켜 웹 페이지에 File Upload/Download가능하도록 설계했다.

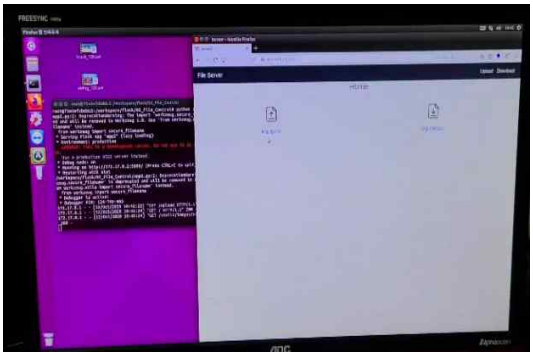

Fig.8 웹 서버, 클라이언트 구현

3. 구현 결과

본 연구는 FP16을 사용하여 테스트의 결과를 UCF101 dataset using the evaluation scrip로 PSNR과 SSIM으로 판별하기로 했다. PSNR은 최대 신호대 잡음비로써 신호가 가질 수 있는 최대의 신호에 대한 잡음의 비를 나타낸다. 이는 동영상 손실 압축에서 화질 손실 정보를 평가할 때 사용되는데 손실이 적을수록 높은 값을 가진다.

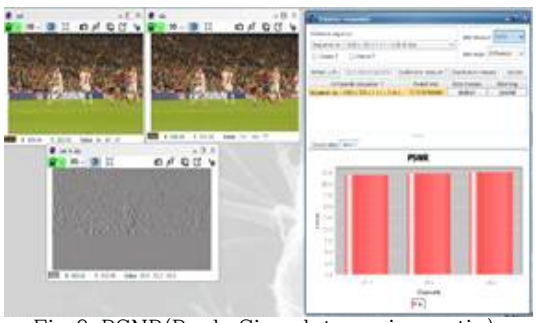

Fig.9 PSNR(Peak Signal-to-noise ratio)

실험결과 22.333의 수치가 나왔다. 이는 손실률이 기대 이상으로 적다는 알려준다.

SSIM은 구조적 유사 지수로써 압축 및 변환에 의해 발생하는 왜곡에 대하여 원본 영상에 대한 유사도를 측정하는 방법이다.

Fig.10 SSIM(Structural Similarity)

실험결과 0.999993 값을 얻을 수 있었다. 이는 이전 영상과 새로운 영상의 유사도가 높다는 것을 알 수 있다.

4. 기대효과

현재 스포츠 경기나 차량 블랙박스 등에서 비디오를 이용한 판정으로 인해 교통사고 분쟁이 감소되고 있다. 하지만 일반 카메라로 촬영된 비디오에서 정확한 판정을 하기 어려운 경우가 빈번하게 발생되어 문제가 된다.
본 작품을 토대로 스포츠 비디오판독으로만 할 수 있는 기능을 웹사이트, 어플리케이션에 접목시킴으로써 하나의 서비스를 만들 수 있다. 또한 교통사고 등의 사건 사고에 활용하여 일반 카메라에 비해 더 개선되고 정확한 판결을 낼 수 있다.

5. 참고문헌

[1] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller and Jan Kautz, “Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation,” In CVPR, 2018, June.
[2] NVIDIA AI Conference 정리, Tensorcore를 이용한 딥러닝 학습 가속을 쉽게 하는 방법, https://brstar96.github.io/dl%20training%20tip/event&seminar/NVIDIAAIConf_session1, 2020년 10월
[3] GitHub-avinashpalival/Super-SloMo, https://gi,thub.com/avinashpaliwal/Super-SloMo, 2020년 3월.
[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” In CVPR, 2015 Feb.