Frame Interpolation Using Generative Adversarial Networks(FINNiGAN)

http://cs231n.stanford.edu/reports/2017/pdfs/317.pdf

### Abstract

FINNiGAN:

- Frame Interpolation Using Generative Adversarial Networks

- CNN + GAN -> take a pair of sequential video frames & generate frame in-between them. -> increase the frame rate of videos.

- training on a combination of l\_1, MS-SSIM, GAN losses.

### Intro

Frame interpolation: generating a frame given immediate frame(바로 앞, 바로 뒤)

input: a pair of sequential frames from a video

### Related work

1. Linear Frame Interpolation(LFI)

- pixel to pixel -> "ghosting" effect

2. Motion-Compensated Frame Interpolation(MCFI)

https://blog.naver.com/hskkhr/222140161149

- ME(motion estimation): compute 'velocity' of each pixel

- MC(motion compensation): move each pixel halfway in the same direction with that estimates(ME)

-> "soap-opera" effect. 부자연스러운 이미지 생성

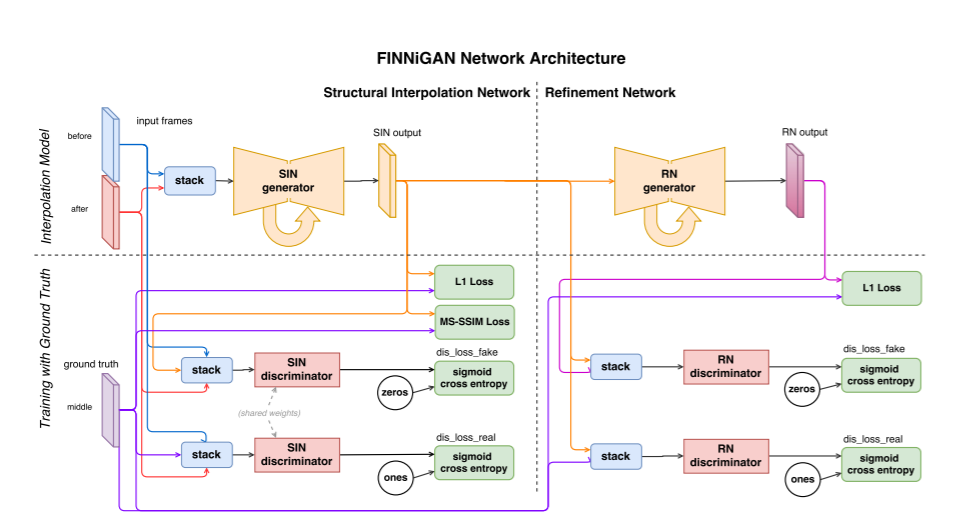
3. I-MCFI / AVMF / MVS.

"frame interpolation: 2 image -> single image translation"

- 연속된 2 image에서 info를 얻어 (CNN: encoder-decoder setup to learn "implicit features". -> blur / noisy image)

- single image를 generate(GAN: signle to single -> 이미지를 좀 더 "realistic"하게)

### Methods: FINNiGAN



1. Structural Interpolation Network(SIN)

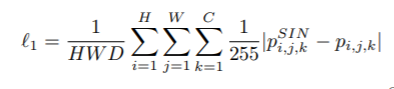
- takes two adjacent frames as an input -> generate the structure of the middle

- weights the MS-SSIM loss heavily.

\* MS-SSIM loss takes a gray-scale image -> color x

- loss function:

1. L1 loss:



- capture the general colors & intensity.

- image의 high-frequency domain을 capture x -> blurry하게.

- But 다른 loss랑 같이 사용하면 useful.

2. MS-SSIM loss: Multi Scale-Structural Similarity Index

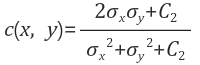
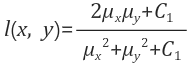
https://blueskyvision.tistory.com/396

- SSIM + “ 스케일 스페이스”: MS-SSIM

- 여러 스케일에서 SSIM 점수 산출 -> 가중곱해서 최종 score get.

- kill gradients & lose ability to learn color info -> 다른 loss 필요!

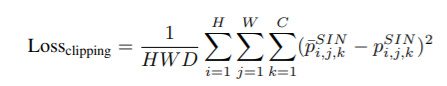
\* SSIM(Structural Similarity): 사람 시각 시스템은 이미지에서 구조 정보를 도출하는 것에 특화. -> 구조 정보의 왜곡 정도가 지각 정도에 큰 영향을 끼침. 원본 이미지 x와 왜곡 이미지 y의 밝기, 콘트라스트(이미지의 표준편차값), 구조(이미지에서 평균 밝기 빼고, 표준편차로 나눠준 것, ) 비교.





\* 스케일 스페이스: 영상처리를 할 때 단 하나의 스케일의 이미지가 아니라 여러 스케일로 본 이미지들을 가지고 필요한 작업을 함. 여러 스케일의 이미지들을 모아놓은 것: scale space.

3. Clipping loss:

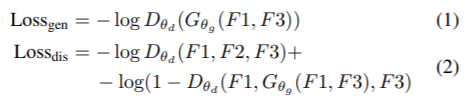


- MS-SSIM이 clipping to [0, 1]된 후 계산됨.

-> prevent gradient on MS-SSIM loss to flow through pixels clipped

-> penalize G against outputting values outside the allowed range.

4. Discriminator Loss:



2. Refinement Network(RN)

- corrects color & clean up some structural error

- uses only L1 loss

- outputs of SIN: structure 😊, ghosting(LFI) x. but color ☹, testure ☹

-> train “pix2pix model” to re-color and re-texture.

- G takes single image. D scores pairs of images.

- augments the GAN loss with an l\_1 loss.

3. Evaluation Metrics

1. Qualitative: compare to LFI / DFI

- avoid ghosting, generate jarring artifacts.

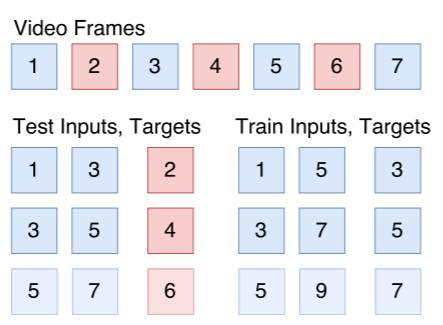
- preserve details

2. Quantitative: compare to MCFI(Motion Compensated Frame Interpolation)

- with single-scale SSIM, PSNR(Peak-Signal-To-Noise-Ratio)

### Dataset

Goal: up-sample a video from original to twice frame-rate.



* down-sample to half -> test-set
* left-over -> training-set

### Experimental Results

1. Qualitative
   1. Comparison with baseline
      * SIN: ghosting x. color x.
      * FINNiGAN(SIN + RN): color o. structure o.
      * LFI: ghosting o.
   2. DFI(Deep Frame Interpolation):
      * trained on l\_1 loss -> blur & color 날라감.
2. Quantitative
   1. SSIM: DFI랑 비슷. (DFI: over-estimates. Down-sampled된 애로 학습)
   2. PSNR(l\_2 loss): I-MCFI가 압도. (FINNiGAN은 l\_2 loss 반영 x)

### Conclusion \*\*\* 우리가 해볼 수 있을 듯!

1. Hyperparameters 조정
2. Network size
3. Extrapolate the next frame.