



CDFI

Abstract

딥러닝을 기반으로 한 frame interpolation은 연속적인 두 장의 사진을 이용하여 보간을 한다. 이는 전형적으로 무거운 모델을 기반으로 한다. 우리는 가치치기를 통한 최적화와 성능 향상을 가져오는 CDFI model을 제시한다. 우리는 먼저 Adacof model을 10배로 압축한 모델을 보여 준다. 그리고 warping module을 제시하여 시각적 일관성을 향상시킨다. 결과적으로 4분의 1크기에 해당하며 더 뛰어난 성능을 지니는 Adacof가 된다. 또한 우리의 방법은 다른 모델에도 쉽게 적용이 가능하다.



Figure 1. A challenging example consists of large motion, severe occlusion and non-stationary finer details. From top to bottom: the overlaid two inputs, the ground-truth middle frame, the frame generated by AdaCoF [32], the frame generated by the $10\times$ compressed AdaCoF, and the frame generated by our method. The compressed AdaCoF even outperforms the full one in this case.

Introduction

VFI는 실제 존재하는 이미지간 사이의 이미지를 생성하는 작업이다. 이는 슬로우 모션 생성, 프레임을 증가 및 여러 작업에서 사용되어진다. VFI는 여러 문제에 당면해있다. 예를 들어, 복잡한 움직임, 흐릿함, 그리고 실제 세계의 광범위한 이미지가 있다.

많은 연구자는 이 분야에 대해 많은 탐구를 진행했고, 딥러닝 방식이 수행되어졌다. 특히, flow기반의 연구가 많이 이루어졌다. 그러나 flow기반은 pre-trained된 flow 모델이 필요하며 이는 end-to-end를 이룰 수 없다. 반면 kernel-based 방법은 kernel-size와 높은 컴퓨팅 자원으로 인해 어려움을 겪는다. 그래서 두 가지를 적절히 융합한 방식이 많이 사용된다.

연구가 활발해질수록 많은 모델은 점점 더 무거워지게 된다. 특히 MEMC-Net의 경우는 70m의 파라미터로 이루어져 있다. 그래서 우리는 CDFI model을 제시하며 이는 최근에 제시된 AdaCoF의 파라미터를 감소시키며 더 높은 성능을 보여준다.

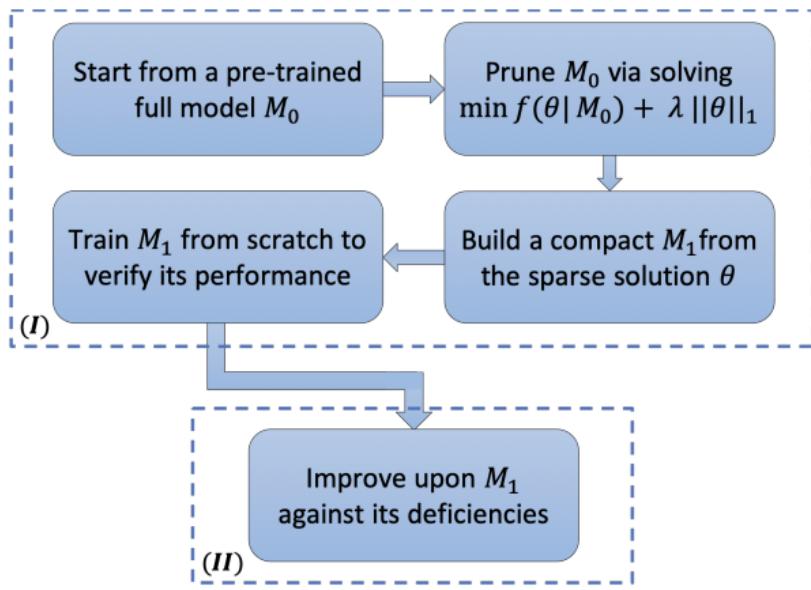


Figure 2. Pipeline of CDFI. Stage (I): compression of the baseline; Stage (II): improvements upon the compression.

Related work

CNN기법은 최근에 시간적인 motion예측에서 성공하였습니다. flow based는 pixel마다의 예측을 진행하는 반면 2d기반인 CNN은 근처 픽셀 간의 상호작용을 통해 예측을 진행하여 더욱 좋은 성능을 보였습니다. 그러나 이러한 방식은 정해진 영역만에서 작동하였고 large motion을 다룰 수 없었습니다. 이에 따라, 발전한 방식이 Deformable CNN 방식으로 정해진 규격의 CNN이 아닌 좀 더 자유로운 영역에 기반한 CNN방식을 사용하는 것이었고 이는 AdaCoF에 기반합니다. 우리는 이러한 AdaCoF를 파라미터 감소화시키며 성능 향상을 하고자 합니다.

Pruning-based model compression

모델 압축은 딥러닝에서 매우 중요합니다. 모델 경량화는 가지치기, 양자화, 층화추출, AutoML방식 등이 있습니다. 경량화를 위한 위와 같은 많은 방식이 존재하며 우리 모델에도 적용할 수 있으나, 압축 이후 우리의 모델은 더 이상 우리의 목적과는 달라집니다.

따라서, 우리는 최적화 기반의 l_0 , l_1 규제화를 통한 가지치기 방식을 사용합니다. 우리는 다음과 같은 방식을 따릅니다.

- 1) l_1 norm을 통해 학습을 진행합니다.
- 2) 각각의 레이어에서 알려진 드문 구조에 따라서 작은 네트워크로 재구성합니다.
- 3) 새로 만든 네트워크를 학습합니다.

The proposed approach

비디오에서 연속적인 두개의 이미지는 가운데 시점의 이미지를 합성합니다.

Motivation

AdaCoF의 핵심은 하나의 이미지를 합성하기 위해 DConv방식을 사용하는 것입니다.

$$\sum_{k=0}^{F-1} \sum_{l=0}^{F-1} W_{i,j}^{(k,l)} I_{\text{in}}(i + dk + \alpha_{i,j}^{(k,l)}, j + dl + \beta_{i,j}^{(k,l)}), \quad (1)$$

각 두개의 이미지로 부터 DCN방식을 사용하여 참조할 영역의 크기(kernel size F)를 정하며 d는 이동할 수 있는 거리를 뜻합니다. F, D는 하이퍼 파라미터로 작동합니다. 일반적인 DCN과 다른점은 W가중치로 인해 각각 독립성이 부과됩니다.

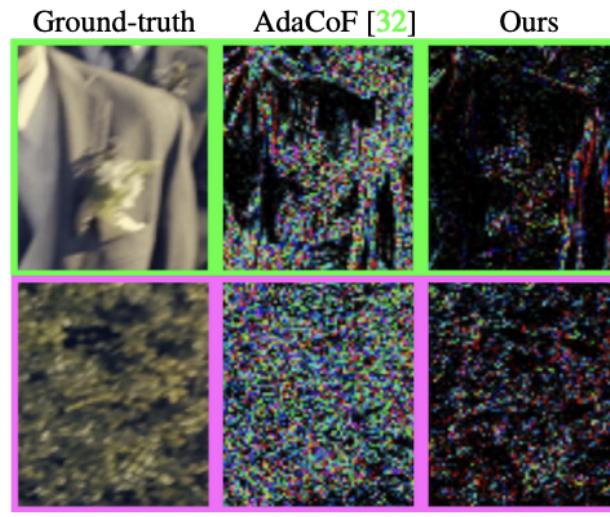


Figure 3. Visualization of the difference between the interpolation and the ground-truth image.

AdaCoF는 복잡한 알고리즘으로 인해 크고 복잡한 모션을 다루지만, 흐릿함과 디테일한 부분을 잡아내는데는 어려움을 겪습니다. (Figure.3) 이는 두개의 이미지를 합성하는데에 있어서 간단한 sigmoid mask V1에 따라 정보를 보존하지 못합니다.

근본적인 질문은 우리가 이러한 점을 직접적으로 개선할 수 있는가입니다. 그러나 우리는 AdaCoF 모델은 불충분하다는 것을 밝혀냈습니다. 예를 들어, 가운데에 있는 여섯개의 $512 \times 512 \times 3 \times 3$ conv layer는 불필요한것으로 나타났습니다. 기전의 AdaCoF model은 21.8m 파라미터를 가집니다. 이러한 무거운 모델은 학습하는데 오래 걸려 개선시키기에 어렵습니다.

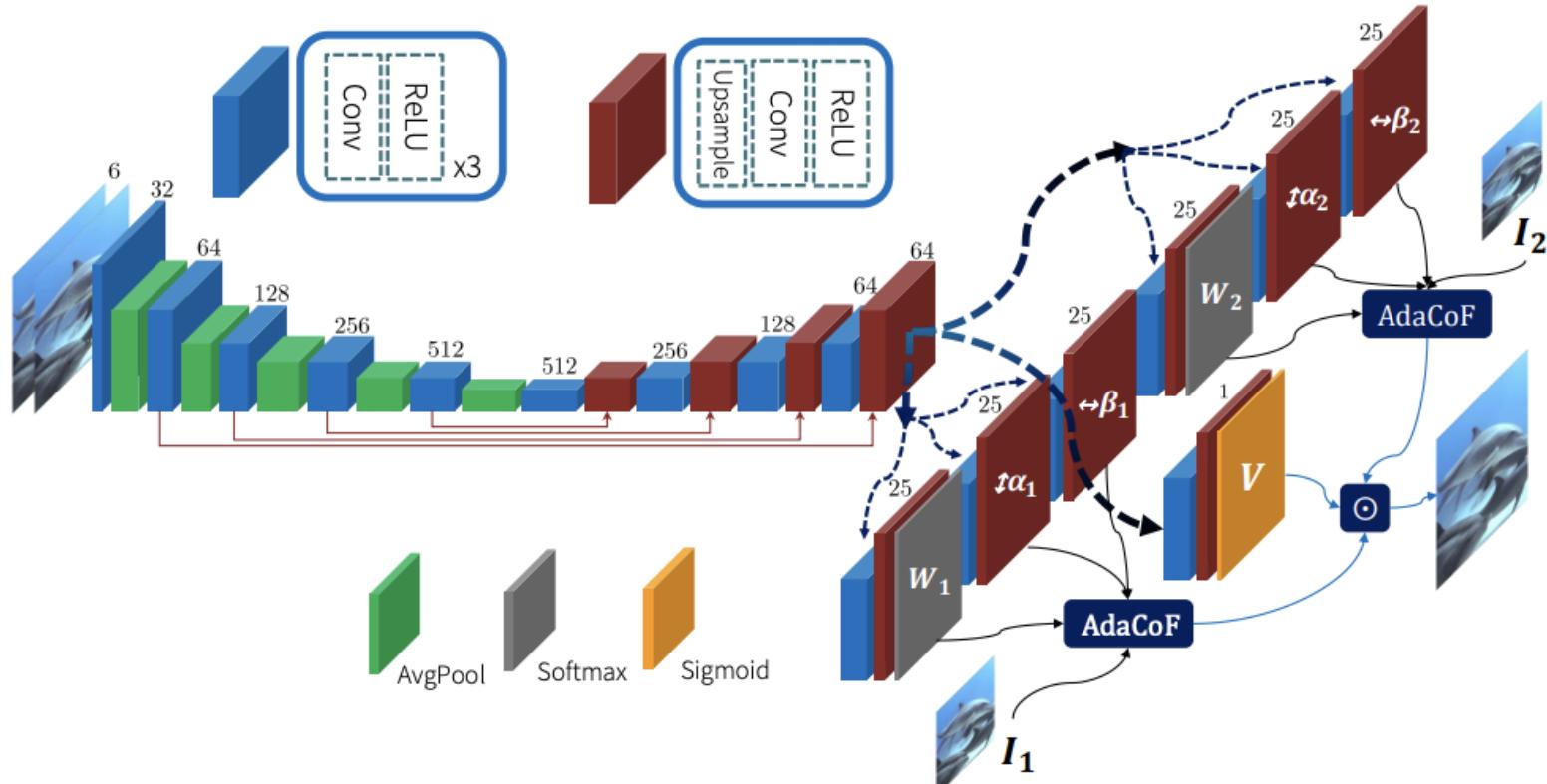


Figure 3: The neural network architecture. The model consists of three main parts: the U-Net, sub-networks, and Adaptive Collaboration of Flows (AdaCoF). The U-Net architecture extracts features from the input image. Then the sub-networks estimates the parameters needed for AdaCoF from the extracted features. The output's height and width of each sub-network are the same as that of the input. Each parameter group for an output pixel is obtained as a 1D vector along the channel axis. The AdaCoF part synthesizes the intermediate frame using the input frames and parameters.

AdaCoF Architecture.

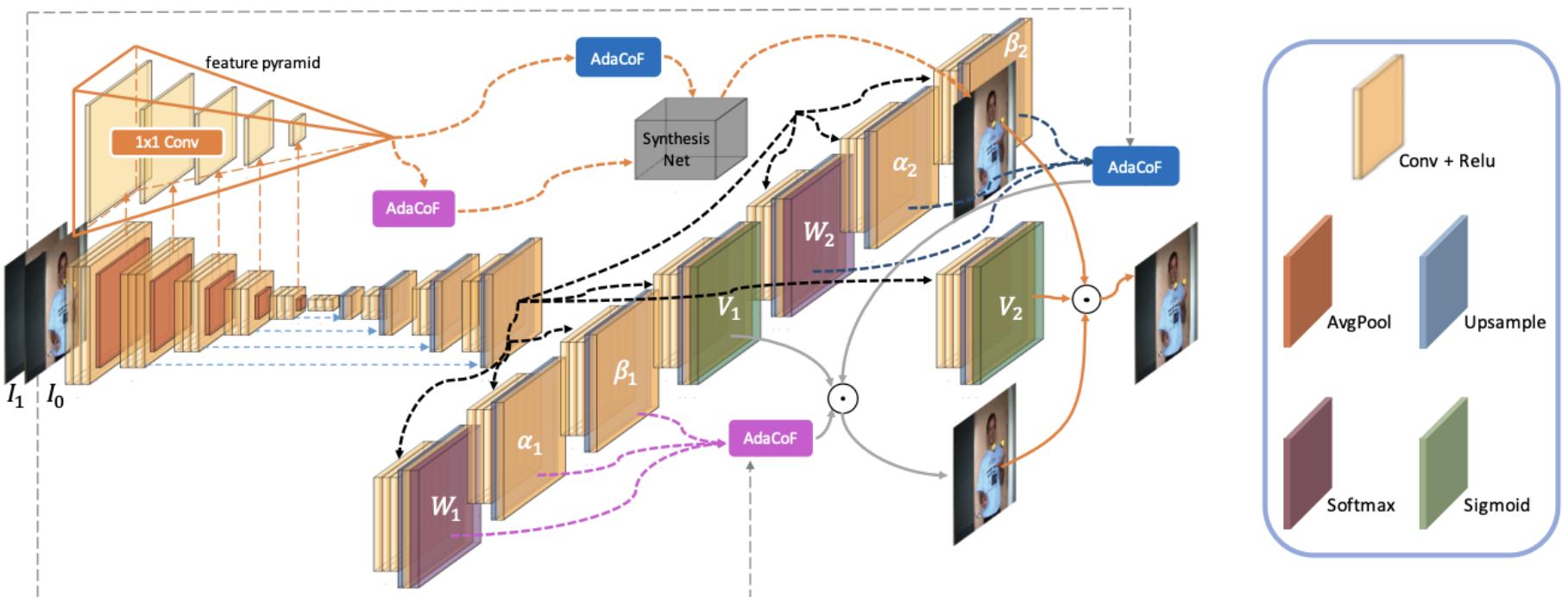


Figure 4. Illustration of our architecture design based on the compressed AdaCoF [32]. The lower part (AdaCoF) consists of a U-Net, a group of sub-networks for estimating two sets of $\{W_i, \alpha_i, \beta_i\}$ in AdaCoF operation (1) correspond to backward/forward warping, and an occlusion mask V_1 for synthesizing one candidate intermediate frame $I_{0.5}^{(1)}$. The upper part (our design) extracts a feature pyramid representation of the input frames through 1-by-1 convolutions from the encoder of the U-Net, then the multi-scale features are warped by AdaCoF operation of learned backward/forward parameters, which are fed into a synthesis network to generate another candidate intermediate frame $I_{0.5}^{(2)}$. Note that the pink and blue AdaCoF modules are associated with $\{W_1, \alpha_1, \beta_1\}$ and $\{W_2, \alpha_2, \beta_2\}$, respectively. Finally, the network outputs the interpolation frame by blending $I_{0.5}^{(1)}$ and $I_{0.5}^{(2)}$ via an extra occlusion mask V_2 .

CDFI Architecture.

Compression of the baseline

먼저 우리는 모델을 압축시키고자 합니다. AdaCoF는 여기서 가지치기 모델에 의해 활용됩니다. 학습되어진 full model M0는 I1 규제에 의한 가중치에 의해 재학습되어집니다.

$$\min_{\theta} f(\theta|M_0) + \lambda \|\theta\|_1,$$

이는 중요한 뉴런에 대한 가중치를 부과합니다. 우리는 새로 제시된 orthant-based stochastic method를 사용하여 효율적인 메커니즘을 추구합니다. 이를 통해 sparse solution $\hat{\theta}$ 를 얻어 small network M1을 제작합니다.

최종적으로 우리는 M1을 초기상태부터 재학습을 진행합니다.

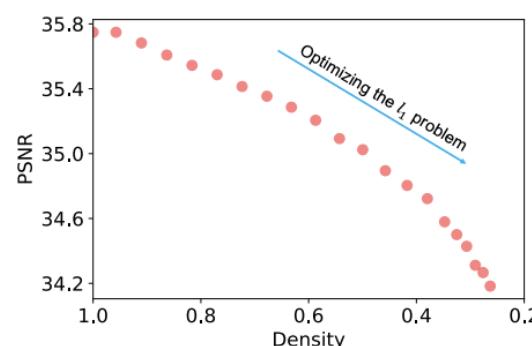


Figure 5. Plot of PSNR against the density of AdaCoF, trained on Middlebury, within 20 epochs of optimizing equation (2).

and density ratio d_l of this layer are respectively defined as

$$s_l := (\# \text{ of zeros in } \hat{\theta}_l) / K_l \quad \text{and} \quad d_l := 1 - s_l. \quad (3)$$

orthant-based stochastic solve를 20epoch동안 1000개의 triplet셋을 이용해 학습을 진행하였고, 매 epoch마다 작은 Density를 기록하였습니다. 마지막 epoch에서는 기존보다 26%의 density를 갖는 네트워크를 구성합니다. 매 레이어의 density를 살펴보면 우리는 가운데 여섯개의 $512 \times 512 \times 3 \times 3$ conv layer는 중복되는 부분으로 나왔으며 다음에 나오는 업샘플링 부분은 density가 7%에 불과합니다. 이는 이전의 원본 모델은 상당수의 파라미터가 불필요한 것을 확인할 수 있었습니다. 그래서 우리는 계산된 density에 따라서 간결한 네트워크를 재구성합니다. 그리고 재학습을 진행합니다.

	Original AdaCoF ($F = 5, d = 1$)	After Compression
PSNR	35.72	35.43
SSIM	0.96	0.96
Size (MB)	83.4	9.4
Time (ms)	82.6	60.4
FLOPS (G)	359.2	185.9
Parameters (M)	21.8	2.45

Table 1. The statistics of AdaCoF and the compressed version.

위에서 density를 줄여가며 성능을 보았을 때는 34.2까지 감소했으나, 다시 전체 데이터셋으로 재학습을 진행한 결과 35.46을 기록했습니다. 그리고 우리는 AdaCoF의 파라미터 수를 10분의 1로 감소시켰습니다.

Improve upon the compression

이 부분은 파라미터 감소와는 관련이 없지만 간결함으로 인해 우리는 많은 공간 확보를 이루었기에 추가적인 모듈을 사용하여 성능을 올릴 수 있었습니다. AdaCoF는 흐릿함과 디테일이 떨어졌기에 우리는 이에 대해 세가지 feature pyramid, image synthesis network, selection mechanism을 구성합니다.

Feature pyramid

마지막 두개의 frame을 결합함에 있어 간단한 single sigmoid mask V1은 정보 손실이 불가피합니다. 그래서 우리는 feature pyramid에서 1*1 conv layer를 통해 다양한 scale로 추출하였고 이는 AdaCoF operation에 들어갑니다.

Image synthesis network

multi-scale에서 더 좋은 피쳐를 추출하기 위해 GridNet을 사용합니다. 이는 다양한 scale의 결합시 유용합니다.

Path selection

기존의 AdaCoF에서는 완성된 featuremap을 통해 V1을 사용하여 output을 제작했습니다. 이와 병렬적으로 다른 다양한 scale에서 결합된 featuremap을 또 다른 occlusion인 V2를 이용해 결합을 진행합니다. 이는 이전의 부족한 정보를 더해줄 수 있을거라 기대되어집니다.

이러한 세가지 요소는 파라미터 감소로 이루어지지는 않지만 성능을 향상시키는데 도움을 줍니다.

Training

AdaMax를 이용하여 $\beta_1 = 0.9, \beta_2 = 0.999$, initial lr = 0.001을 사용합니다. 그리고 매 20epoch마다 lr을 절반으로 감소시킵니다.

우리가 사용하는 로스는 다음과 같습니다.

$$\mathcal{L}_{\text{Charbon}} = \rho(I_{\text{out}} - I_{\text{gt}})$$

$$\mathcal{L}_{\text{vgg}} = \|\phi(I_{\text{out}}) - \phi(I_{\text{gt}})\|_2.$$

$$\mathcal{L}_{\text{tv}} = \tau(\alpha_1) + \tau(\alpha_2) + \tau(\beta_1) + \tau(\beta_2)$$

$$\mathcal{L} = \mathcal{L}_{\text{Charbon}} + \lambda_{\text{vgg}} \mathcal{L}_{\text{vgg}} + \lambda_{\text{tv}} \mathcal{L}_{\text{tv}}$$