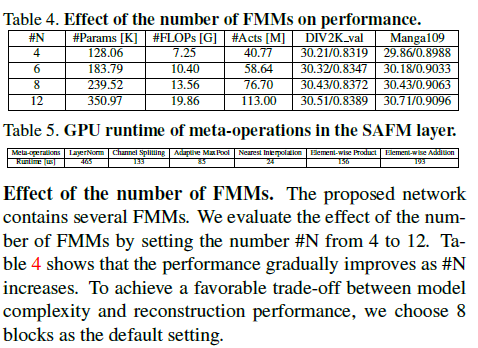
LMLT: Low-to-high Multi-Level Vision Transformer for Image Super-Resolution

향후 실험 (**베이스는 dim=36, block=8개 : Efficient 모델**)

Dim 60, block 4 / 12의 선정 기준 : Dim60은 SwinIR, SRFormer, NGram 등 다양한 모델에서 사용하는 모델 사이즈. 4, 12는 SAFMN이 자신의 모델의 블록 수에 따른 성능을 평가하기 위해 사용.



1. *현 진행 중*

*Dim=48, block=8 (Banana\_1) / Dim=36, block=12 (Banana\_2) -> 진행 중*

*No CCM(A6000\_1)*

*Dim = 86 -> A6000\_2*

*3090\_1 : 오리지날 재실험(재검증)*

*3090\_2 : Head 1개로 (x2 x3 x4) -> 진행중. 다 끝나면 CCM -> IVF로*

1. *향후 진행* ***2 :*** *No Aggr, No Act (3090\_1) / No Aggr (3090\_2)*

*일반 모델처럼 일반적인 [64 64]로 8개의 블록을 4번 통과 (x2 x3 x4), head는 4개 << 필요한가??*

*Dpr 넣어서 해보기 -< A6000\_1*

*No Sum 재검증(x3 x4)*

*1) 기존 ViT는 같은 stage 내의 차원의 size는 전부 같으며, head별로 size를 다르게 한 경우는 없음.*

*2) 사이즈를 다르게 하여 각자 더해주는 방식은 window communication 문제를 해결(no window sum과 DownSampling을 하지 않는 실험의 결과가 이를 증명). Training size는 (64, 64)이므로, win=(8,8), downsample /8은 fully-attn과 유사 효과.*

*3) 여기에 더불어, CNN이 가진 근본적 한계인 limited receptive field를 ViT를 통해 해결하고자 함. 과도한 연산횟수를 downsampling + channel split을 통해 해결.*

*4) 한 번에 QKV를 진행하지 않고, split으로 쪼갠 다음 qkv attn 연산을 하면 #param과 #MACs가 4배로 감소((36\*108)에서 4\*(9\*36)으로 감소).*

*5) 일반적인 ViT의 depth 개념을 (split + downsample + 각자의 SA)로 대체*

*6) SAFMN에도 상위 단계에 더하기를 추가하여 성능 추이를 봄. 전체적으로 오를 시, motivation을 뒷받침하는 강력한 증거가 될 것.*

*7) 앞선 모델인 SAFMN과 유사 내지는 살짝 증가한 #param 및 inference time 및 메모리 점유율을 가지면서도 성능을 높일 수 있으리라 예상.*

1. Introduction (3400자)

**단일 이미지 초해상도(Single Image Super Resolution, SR)는 저해상도 이미지를 고해상도 이미지로 변환하는 작업**으로, 컴퓨터 비전 분야에서 활발히 연구되고 있는 분야이다. 과거에는 nearest neighbor interpolation, bilinear interpolation과 같은 방법들이 사용되었지만, 최근에는 많은 Super-resolution 연구들이 딥러닝, 특히 CNN 기반 방법[]과 Vision Transformer 기반 방법[]을 이용하여 획기적인 성능의 향상을 이루었다.

**Super-Resolution을 에지 장치에서 실제 적용을 하기 위해, 계산 비용은 줄이면서도 성능이 보장되는 효율적인 모델이 요구된다.** CNN을 사용하는 이미지 초해상도 아키텍처는 [10]가 등장한 이래로 dense connection[1]이나 residual learning[]과 같은 방식을 사용하여 발전시켜왔다. 한편, ViT을 사용하는 모델은 CNN 기반 모델에 비해 월등한 성능을 보였다. SwinIR[2]의 경우 Swin Transformer[17]을 Image Super-Resolution에 적용하여 self-attention이 SISR에 효과적일 수 있음을 보였다. 이를 따라 [20, 37]와 같은 다양한 모델이 등장하였으며, self-attention과 CNN을 혼합한 모델[22] 역시 등장하여 큰 성과를 이루었다.

**CNN과 ViT를 사용하는 Super-resolution의 경우, 성능과 효율성 측면에서 다음과 같은 문제가 있다.** 우선 CNN의 경우, 성능을 높이기 위해 모델이 과도하게 깊어지는 경우가 있다[]. 이는 모델의 사이즈를 키우거나 메모리를 증가시키는 문제점이 있으므로, 이를 보상하기 위해 모델의 모듈들이 파라미터를 공유하는 방식을 취한다[]. 그러나 이는 여전히 추론 시의 효율성을 보장하지 않는다는 문제가 있다[]. 한편 ViT 계열의 경우, 인풋 사이즈가 증가할 때마다 연산량이 제곱으로 증가하는 문제점이 존재[]하며, 이를 완화하기 위해 이미지를 윈도우로 잘라 Self Attention을 진행하는 방식이 대두되었지만[], 이는 윈도우와 윈도우 간의 통신이 되지 않아 윈도우의 경계선 상에서의 품질을 떨어트린다[].

SAFMN[16]은 이미지의 feature을 Multi-Head[11] 와 같은 방식으로 나눈 다음 각각의 스케일을 다르게 하여 비국소적 feature 관계를 구현하고, FMBConv[18]를 기반으로 하여 각 컨볼루션 채널을 믹싱 함으로써 모델의 정확도와 복잡성 사이에서 유리한 절충점을 달성한다. **우리는 여기에서 영감을 받아, 추론 시의 효율성은 높이면서도 성능을 최대한 유지하는 LMLT(Low-to-high Multi-Level Transformer)을 제안한다.** 우리의 방식은 SAFMN처럼 이미지의 feature을 Multi-Head와 같은 방식으로 나눈 다음 각각을 지정된 사이즈로 다운샘플링하는 것까지는 같다. 이후에는 [16]의 방식을 따라, 나눠진 feature에 대해 고정된 사이즈로 self-attention을 진행한다. 이는 추론 과정에서 이미지의 스케일을 줄이므로 다른 ViT 기반 모델에 비해 추론 시간이 감소한다.

**스케일 감소에 따른 공간적 정보 손실과 window communication 부재의 문제를 해결하기 위해, 하나의 헤드에서 진행한 Self Attentinon의 결과를 상위 헤드에 더해주는 스케일 간 residual network[]를 통해 보완한다.** 윈도우 사이즈는 고정되어 있기에, 다운샘플링을 진행하지 않은 상위 헤드의 윈도우는 한정된 부분만을 보게 된다는 제약에서 자유롭지 않다. 반면, 다운샘플링을 할수록 상대적으로 윈도우가 이미지 전체 대비 보게 되는 범위는 증가하며, 따라서 하위 헤드의 결과를 업샘플링하여 상위 헤드에 전달하는 것만으로 상위 헤드의 윈도우가 보는 영역을 증가시킬 수 있다. 이를 통해, 우리는 ViT 기반 모델도 충분히 모델의 복잡성과 정확도 사이에서 절충점을 달성할 수 있다는 점을 보인다. 같은 차원 수로 비교할 때, SAFMN[16]에 비해 추론 시간은 n%, 점유 메모리는 n% 증가하지만 대부분의 스케일에서 벤치마크 데이터셋에서 우위를 보이며, 다른 State-of-the-Art의 결과와 비교해볼 때 점유 메모리와 추론 시간을 상당히 줄이면서 성능을 증가시킨다. 특히, ViT 기반 Super-Resolution에 비해 추론시간과 메모리 점유율을 크게 감소시킨다.

**우리의 기여는 다음과 같다 :**

1. 쿵따리샤바라
2. 빠빠빠빠
3. 너는 매일
4. 같이 학교가는 버스 안에서
5. Related works (5000자) 각각 2500자로…

2.1. Vision Transformer

**Transformer[11]은 NLP의 다양한 분야에서 큰 성과를 거뒀으며, 최근 vision task 역시 Transformer 아키텍처[12]가 소개된 뒤로 이를 이용하여 classification[], object detection[] 그리고 semantic segmentation[26] 등 다양한 분야에서 큰 성공을 이루었다.** Transformer은 기존 CNN이 가지고 있던 한계인, 제한된 커널 사이즈로 인해 오직 이미지의 local region에 대해서만 정보의 처리가 가능하기에, long-range dependency modeling에 대한 효율성을 달성하기 어렵다는 문제점[]을 context간의 global interaction을 캡쳐함으로써 해결한다[].

**그러나 Vision Transformer 역시 한계점을 지니고 있다.** 대표적으로, input의 사이즈가 증가할수록 복잡도가 제곱(quadratic)으로 증가한다. 이를 해결하기 위해 SASA[31]에서 local window 내에서 self-attention을 계산하는 방식을 제안한 이래로, HaloNet[32]은 쿼리 주위를 감싸는 밴드를 결합하여 공유 이웃 블록을 획득하는 방식을 통해 ResNet[33]에 비해 높은 성능을 달성했다. Swin Transformer[17]는 이미지를 고정된 윈도우로 자른 다음, 윈도우 내에서 self-attention 매커니즘을 진행한 다음 점차 병합해가는 방식을 사용해 계산량을 감소시킴과 동시에 놀라운 성능을 이끌어내었다. PvT[14, 15]와 MViT[25]는 멀티스케일을 통해 Self-Attention을 적용하는 효과적임을 보였다. 한편 [27, 28]의 경우, CNN과 ViT를 함께 쓰는 hybrid 아키텍처를 통해 성능과 이미지 throughput 간의 trade-off를 이끌어낸다.

**하나의 윈도우와 다른 윈도우 간의 문맥 통신이 없고 global context의 획득이 까다로운 것은 Swin Transformer가 극복해야 할 과제이다.** CSwin Transformer[13]은 윈도우를 stripe 형식으로 만들어 글로벌 정보를 포착한다. NAT[29]나 CvT[30]는 Self-attention 매커니즘에 CNN을 적용시켜, kernel과 stride를 조정해 cross-window communication 문제를 해결한다.

2.2. Image Super-Resolution

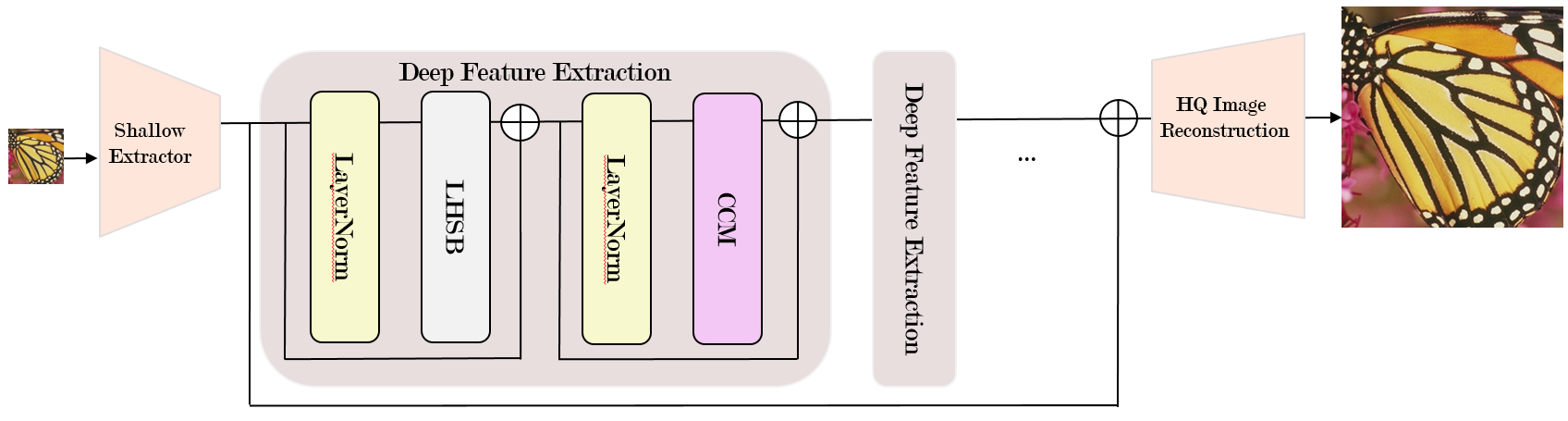
SRCNN[10]이 3개의 end-to-end 레이어를 이용해 HR image를 restore하는 방식을 제안한 이래로, **다양한 CNN 기반 모델이 Single Image Super-Resolution의 성능과 효율성을 높이기 위해 제안되었다.** VDSR[34]과 DRCN[35]은 더 깊은 신경망 구조를 사용하고, 네트워크 양 끝단 매핑에 재귀적 신경망 구조를 제안하여 더 높은 품질의 결과를 산출했다. ESPCN[36]은 네트워크의 끝에 있던 업샘플링을 bicubic-filter에서 sub-pixel convolution으로 대체함으로써 SR의 속도를 크게 높였고, 이후 여러 작업에서 사용되게 되었다[]. ~~LapSRN[]은 LR 이미지를 입력으로 받아 다양한 해상도에서 feature을 추출한 다음 residual block을 사용하여 LR과 HR 사이의 correlation을 학습하여 성능을 높였다.~~

**ViT를 포함한 Attention 매커니즘을 Image Super Resolution과 같은 low-level vision task에 적용하는 것 역시 최근에 활발하게 연구되고 있는 분야**이다. IPT[]는 image denoising, deraining, super resloution과 같은 이미지 프로세싱 문제를 large pre-trained transformer model을 통해 해결한다. Liang et al.[2]은 Swin Transformer에 기반한 Image Super Resolution모델인 SwinIR을 제안하였고, 뒤를 이어 ELAN[37]은 다양한 윈도우 사이즈를 self-attention에 사용하여 공간적 적응성을 높이고자 했다.

**~~Self-attention은 이미지의 공간적 특징을 잘 포착하지만, channel-wise로 self-attention을 계산하는 방향으로도 low-level vision task의 해결이 가능하다.~~** ~~SE-Net[]이 CNN의 feature을 채널 별로 중요도를 다시 매겨 성능을 올린 이후, RCAN[]은 이를 super-resolution 모델에 적용하여 residual block과 channel attention 매커니즘을 이용하여 더욱 깊은 네트워크를 구성하였다. 이후 Restormer[]은 channel-wise self attention만을 이용해 deraining 등 여러 이미지 프로세싱에 탁월한 성능을 보였고, DAT[]는 Spatial-wise self attention과 Channel-wise self attention을 병용하여, 두 방식을 함께 씀으로 인해 global dependency를 model할 수 있음을 보였다.~~

**Super Resolution 네트워크의 효율성을 증가시키기 위해 CNN, ViT를 이용한 다양한 모델이 제안되었다.** CARN[1]은 다단계 방식으로 연결되는 캐스캐이딩 잔차블록을 통해 다양한 단계에서 feature을 재사용했다. IMDN[19]은 네트워크를 통과하는 feature을 점진적으로 정제했다. SwinIR-light[2]는 Swin Transformer[17]을 이용한 SR모델인 SwinIR을 가벼운 형태로 만든 것이다. HNCT[22]는 CNN과 Swin Transformer[17]을 결합하여 성능의 향상을 꾀했다. NGSwin[20]은 기존 Swin Transformer[17]가 가지고 있던 문제점인 cross-window communication 문제를 ngram[21]을 적용하여 해결했다. 이외에도 지식 증류[39], 모델 양자화[40] 또는 프루닝[41]을 적용하는 등[43, 44] 다양한 방식을 통해 네트워크를 가볍게 만들려는 시도가 있으며, 같은 아키텍처를 사용하되 네트워크의 블록 수나 feature 차원의 수 등 하이퍼파라미터를 변경하여 classical image super resolution 모델과 lightweight image super resolution 모델을 분리하는 경우도 존재한다[38, 49, 50].

1. Proposed Method (7000자)

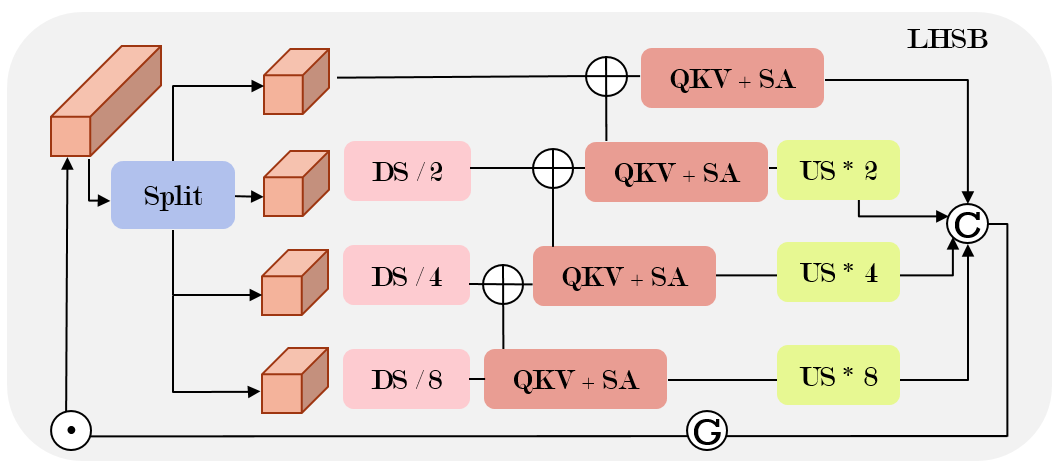


3.1. Method Overview

**이 섹션에서 우리는 우리의 모델에 대한 구조를 설명한다.** 우리의 모델 구조는 그림 #과 같이 이루어져 있다. 우선 이미지로부터 3x3 컨볼루션을 통해 얕은 수준의 feature을 추출한다.

3.2. Multi-Scale Head Attention

ViT 모델은 놀라운 성능을 보여주지만, 문제점이 있다. (1. 모든 head의 shape가 같아서 연산량이 많고, 이는 ViT의 효율을 높이는 데 장애가 된다. 2. 윈도우로 나누는 경우 cross-window communication 문제)



우리는 SAFMN에서 아이디어를 차용해, feature을 head로 split한 다음 각 헤드의 feature에 다운샘플링을 적용한다. (qkv를 head에서 적용한다는 애기는 여기에서 하라)

SA를 진행한 다음, 각각을 upscaling하여 상위 레이어의 feature에 더한 다음 더해준다 (업스케일링 식이랑 더하는 식, 인코더 식1로 짜기).

다음으로 모든 헤드의 feature을 원래의 사이즈로 복원한 다음 concat하고, aggr과 activation을 통과시켜 원래의 피처와 더해준다.

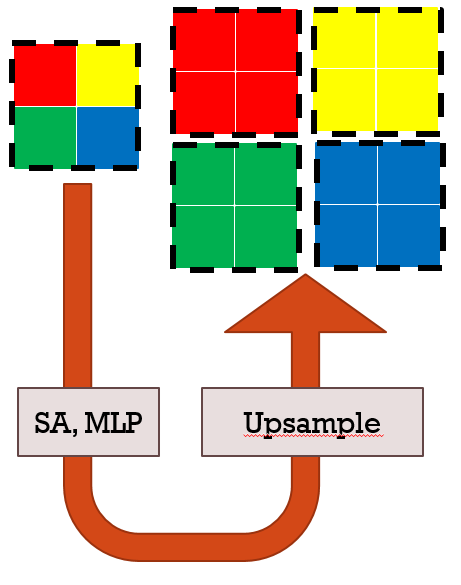
3.3. Feature Mixing Module????

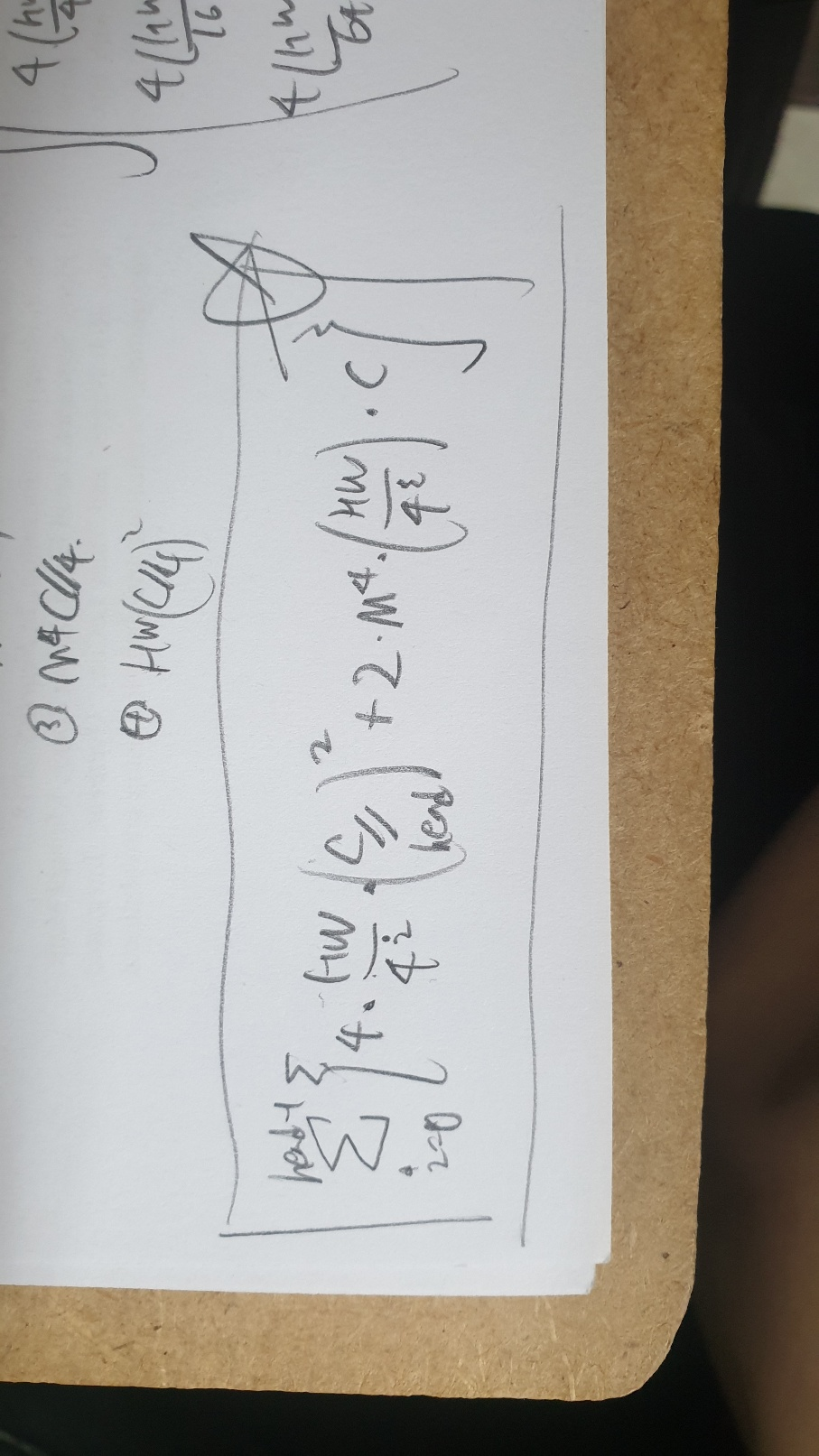
SAFMN에 있던 CCM과 이런거 다 갖다 쓴건데 뭘 쓰지? 더 이상 3번 아키텍처에 쓸 말이 없다… 3.2에 있는 게 내가 바꾼 전부고 나머진 SAFMN의 그것들인데 그거라도 설명해야 하나??

3.4. Downsampling and Cross-window Communication (이 섹션은 지울 수 있음)

*왜 다운샘플링하여 self-attn하고 더해주는 게 long-term 정보 획득에 도움이 되는지 설명. 그림이랑 함께. 자세한 건 섹션 4의 테이블 n 참조하라고 하기. Sum을 안 했을 때의 feature과 했을 때의 feature을 시각화해서 보여주는 그림 + cross widnow communication 그림 총 2장.*

윈도우 사이즈가 고정되어 있고 feature size를 줄이면 1/8로 줄인 것 기준으로 64x64에 SA하는 것과 유사한 효과임. 또한 감소되는 연산량도 식으로 표현. << 이 섹션은 지울 수 있다.

**





1. Experiments
   1. Experimental Setup

**Experimental Setup**

SAFMN에서 arch말고 아무거도 안바꿨으니 이거 걍 비슷하게 말해

**Dataset**

우리는 이전 연구[]들의 방법을 따라, 훈련 데이터셋으로 총 800장의 이미지를 가진 DIV2K[3]와 2650장의 이미지를 갖고 있는 Flickr2K[4]을 사용한다. 테스트 데이터셋으로는 Set5[5], Set14[6], BSD100[7], Urban100[8] 그리고 Manga109[9]를 사용한다. 해상도가 복구된 이미지의 질을 평가하기 위해 PSNR(peak signal to noise ratio)과 SSIM(structural similarity index)[45]을 사용하며, values of these metrics are calculated on the Y channel of the YCbCr space.

* 1. Comparisons of State-of-the-arts models
* *일단 SAFMN과 똑같은 구조 취해봐.*

**Quantitative and Qualitative Comparisons**

제안한 모델의 성능을 평가하기 위해, 우리는 다른 lightweight SR모델을 각기 다른 scaling factor에서 다른 모델들과 비교한다. ~~~을 포함해서. (Table 1 – stage \* 4, dim = 36)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| IMDN | X2 | 691 | 159 | 423 | 38.00 | 0.9605 | 33.63 | 0.9177 | 32.19 | 0.8996 | 32.17 | 0.9283 | 38.88 | 0.9774 |
| PAN | X2 | 261 | 71 | 677 | 38.00 | 0.9605 | 33.59 | 0.9181 | 32.18 | 0.8997 | 32.01 | 0.9273 | 38.70 | 0.9773 |
| LAPAR-A | X2 | 548 | 171 | 656 | 38.01 | 0.9605 | 33.62 | 0.9183 | 32.19 | 0.8999 | 32.10 | 0.9283 | 38.67 | 0.9772 |
| ECBSR | X2 | 596 | 137 | 252 | 37.90 | 0.9615 | 33.34 | 0.9178 | 32.10 | 0.9018 | 31.71 | 0.9250 |  |  |
| SMSR | X2 | 985 | 132 | - | 38.00 | 0.9601 | 33.64 | 0.9179 | 32.17 | 0.8990 | 32.19 | 0.9284 | 38.76 | 0.9771 |
| Shuffle-Mixer | X2 | 394 | 91 | 832 | 38.01 | 0.9606 | 33.63 | 0.9180 | 32.17 | 0.8995 | 31.89 | 0.9257 | 38.83 | 0.9774 |
| SAFMN | X2 | 228 | 52 | 299 | 38.00 | 0.9605 | 33.54 | 0.9177 | 32.16 | 0.8995 | 31.84 | 0.9256 | 38.71 | 0.9771 |
| **Ours** | X2 | **239** | **59** | 603 | 38.01 | 0.9606 | 33.60 | 0.9185 | 32.18 | 0.8999 | 32.04 | 0.9272 | 38.92 | 0.9776 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| IMDN | X3 | 703 | 72 | 190 | 34.36 | 0.9270 | 30.32 | 0.8417 | 29.09 | 0.8046 | 28.17 | 0.8519 | 33.61 | 0.9445 |
| PAN | X3 | 261 | 39 | 340 | 34.40 | 0.9271 | 30.36 | 0.8423 | 29.11 | 0.8050 | 28.11 | 0.8511 | 33.61 | 0.9448 |
| LAPAR-A | X3 | 594 | 114 | 505 | 34.36 | 0.9267 | 30.34 | 0.8421 | 29.11 | 0.8054 | 28.15 | 0.8523 | 33.51 | 0.9441 |
| SMSR | X3 | 993 | 68 | - | 34.40 | 0.9270 | 30.34 | 0.8412 | 29.10 | 0.8050 | 28.25 | 0.8536 | 33.68 | 0.9445 |
| Shuffle-Mixer | X3 | 415 | 43 | 404 | 34.40 | 0.9272 | 33.37 | 0.8423 | 29.12 | 0.8051 | 28.08 | 0.8498 | 33.69 | 0.9448 |
| SAFMN | X3 | 233 | 23 | 134 | 34.34 | 0.9269 | 30.33 | 0.8418 | 29.08 | 0.8048 | 27.95 | 0.8474 | 33.52 | 0.9437 |
| **Ours** | X3 | **244** | **28** | 283 | 34.36 | 0.9270 | 30.38 | 0.8428 | 29.11 | 0.8056 | 28.10 | 0.8503 | 33.72 | 0.9448 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| IMDN [19] | X4 | 715 | 41 | 108 | 32.21 | 0.8948 | 28.58 | 0.7811 | 27.56 | 0.7353 | 26.04 | 0.7838 | 30.45 | 0.9075 |
| PAN [46] | X4 | 272 | 28 | 238 | 32.13 | 0.8948 | 28.61 | 0.7822 | 27.59 | 0.7363 | 26.11 | 0.7854 | 30.51 | 0.9095 |
| LAPAR-A [24] | X4 | 659 | 94 | 452 | 32.15 | 0.8944 | 28.61 | 0.7818 | 27.61 | 0.7336 | 26.14 | 0.7871 | 30.42 | 0.9074 |
| ECBSR [47] | X4 | 603 | 35 | 64 | 31.92 | 0.8946 | 28.34 | 0.7817 | 27.48 | 0.7393 | 25.81 | 0.7773 |  |  |
| SMSR [48] | X4 | 1006 | 42 | - | 32.12 | 0.8932 | 28.55 | 0.7808 | 27.55 | 0.7351 | 26.11 | 0.7868 | 30.54 | 0.9085 |
| Shuffle-Mixer [23] | X4 | 411 | 28 | 269 | 32.21 | 0.8953 | 28.66 | 0.7827 | 27.61 | 0.7366 | 26.08 | 0.7835 | 30.65 | 0.9093 |
| SAFMN [16] | X4 | 240 | 14 | 77 | 32.18 | 0.8948 | 28.60 | 0.7813 | 27.58 | 0.7359 | 25.97 | 0.7809 | 30.43 | 0.9063 |
| **Ours** | X4 | **251** | **15** | 152 | 32.21 | 0.8949 | 28.63 | 0.7822 | 27.60 | 0.7367 | 26.08 | 0.7837 | 30.59 | 0.9081 |

Quantitative comparison에 이어, 우리는 제안한 모델의 성능을 qualitative comparison을 통해 비교한다.

(그림 많이 넣기)

**Memory and Running time Comaprisons**

우리는 SAFMN[16]의 방식을 따라, 제안한 모델의 효율성을 평가하기 위해 여러 모델에 대해 총 50장의 이미지를 스케일링 할 때 사용되는 #GPU MEM과 #AVG TIME을 측정한다. #GPU MEM은 추론 도중에 가장 최대로 메모리를 소비할 때의 결과이며, #AVG TIME은 총 50장의 이미지를 추론할 때의 장당 평균 소요 시간이다. X2 Scale과 x3 scale, 그리고 x4 scale은 각각 640x320, 427x240 그리고 320x180 사이즈의 랜덤 이미지를 2배, 3배, 그리고 4배 업스케일링했을 때의 결과이다. 우리는 CARN-M[1], CARN[1], IMDN[19], SAFMN[16] 그리고 우리의 36차원의 모델을 여러 스케일에서 비교하며, 추가로 60차원에서 진행한 실험의 결과를 SwinIR-light[2], NGSwin[20], 그리고 HNCT[22]와 비교한 것을 Table #에서 보여준다.

Table #

테이블 #는 우리의 제안한 36차원의 모델이 CNN 기반인 SAFMN[16]에 비해 최대 메모리 점유율이 약 25%에서 32%정도 증가하고 추론 시간 역시 약 70% 증가하지만 여전히 다른 State-of-the-art 모델에 비해 현저히 comparable함을 보여준다. 특히 추론시간의 경우 IMDN과 SAFMN을 제외하고는 30초대 후반으로 전부

* 1. Ablation Studies

이 섹션에서 우리는 우리가 제안한 모델의 효율성을 실험하기 위해 진행한 여러 Ablation study에 대해 demonstrate한다. 우선 우리는 스케일을 서로 다르게 한 다음 더하기를 넣은 것의 효과를 진행하고, 다음으로 각 스케일을 다르게 하는 것의 효과를 통해 스케일을 다르게 한 다음 더하기를 넣은 것이 좋다는 것 말한 다음, 각 모듈에 대해 평가함.

**Impact of Each Module**

이 섹션에서, 우리는 우리가 제안한 모델의 여러 모듈에 대해 실험한다. 테이블 n는 여러 스케일로 나눈 다음 상위 스케일로 sum을 하는 방식에 대해 실험함. SAFMN에 plus를 넣는 경우(x2, x3, x4)s와 내 거에서 sum 없는 경우(x2, x3, x4) 🡪 총 기존 SAFMN 3개 + plus 3개 + 내거 3개 + 내거 sum 없는거 3개 = 12개

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| SAFMN | X2 | 228 |  |  | **38.00** | **0.9605** | **33.54** | **0.9177** | 32.16 | 0.8995 | 31.84 | 0.9256 | 38.71 | 0.9771 |
| SAFMN **w/ sum** | X2 | 228 |  |  | 37.99 | 0.9604 | 33.52 | 0.9174 | **32.17** | **0.8996** | **31.86** | **0.9257** | **38.77** | **0.9773** |
| SAFMN | X3 | 233 |  |  | 34.34 | 0.9267 | **30.33** | 0.8418 | 29.08 | 0.8048 | 27.95 | 0.8474 | 33.52 | 0.9437 |
| SAFMN **w/ sum** | X3 | 233 |  |  | 34.34 | **0.9269** | 30.32 | 0.8418 | **29.09** | **0.8049** | **27.96** | **0.8476** | **33.55** | **0.9438** |
| SAFMN | X4 | 240 |  |  | **32.18** | **0.8948** | **28.60** | **0.7813** | 27.58 | **0.7359** | **25.97** | **0.7809** | 30.43 | 0.9063 |
| SAFMN **w/ sum** | X4 | 240 |  |  | 32.10 | 0.8937 | 28.59 | 0.7812 | 27.58 | 0.7358 | 25.96 | 0.7799 | **30.44** | 0.9063 |

도표 해석 : SAFMN의 경우, 상위 레이어에 sum을 해줘도 별로 영향 없음.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| Ours | X2 | 239 |  |  | **38.01** | **0.9606** | **33.60** | **0.9185** | **32.18** | **0.8999** | **32.04** | **0.9272** | **38.92** | **0.9776** |
| Ours **w/o sum** | X2 | 239 |  |  | 37.91 | 0.9602 | 33.47 | 0.9170 | 32.10 | 0.8987 | 31.75 | 0.9242 | 38.46 | 0.9766 |
| Ours | X3 | 244 |  |  | 34.36 | 0.9270 | **30.38** | **0.8428** | 29.11 | 0.8056 | **28.10** | **0.8503** | **33.72** | 0.9448 |
| Ours **w/o sum** | X3 | 244 |  |  | **34.40** | **0.9272** | 30.35 | 0.8425 | 29.11 | 0.8056 | 28.05 | 0.8496 | 33.71 | 0.9448 |
| Ours | X4 | 251 |  |  | 32.21 | 0.8949 | 28.63 | 0.7822 | 27.60 | 0.7367 | 26.08 | 0.7837 | 30.59 | 0.9081 |
| Ours **w/o sum** | X4 | 251 |  |  |  |  |  |  |  |  |  |  |  |  |

도표 해석 : ours의 경우, 상위 레이어에 이전 레이어의 값을 더해주면 성능이 향상(일단 x2에선 증가) 🡪 총 6개

보면 알 수 있듯, SAFMN에서는 상위 레벨로 더해주는 것이 큰 의미 없는데, 내건 의미 있음. 오히려 더하지 않는 경우 그냥 SAFMN에 비해 성능 부족함. 그림(n)의 feature을 보면, scale을 더해주는 것이 더 좋음을 알 수 있음(각 그림 별 스케일 당 feature 보여줌).

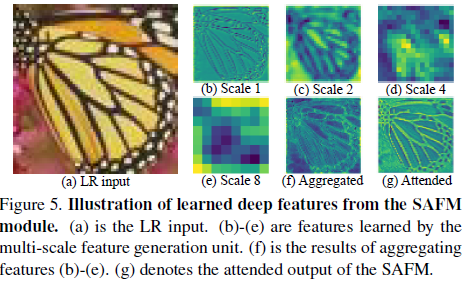
예. 더하기 안 넣은거랑 넣은 걸 비교하는 그림 넣기.

Table n은 스케일을 두지 않고 모두 기본 학습 사이즈 [64 64]로 두는 경우에 대해 실험함으로써, 여러 스케일을 두고 합치는 방식이 유효함을 설명. X2에 대해서만?? 각각 원래, with merge, without merge임. 오히려 모든 HEAD를 같은 spatial size로 두는 경우, downsampling하고 더하는 것에 비해 성능이 떨어짐.(이거도 그림 필요한가? 넣으면 좋기야 할텐데 분량 부족하면 넣고 안부족하면 위의 예시 더하기 안넣기/넣기 비교한 그림 같은 걸 넣어?) 만약 x2만 하면 3개, 아니면 + 6개해서 9개 🡪 6개

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| **Ours** | X2 | 239 | 59 | 603 | **38.01** | **0.9606** | **33.60** | **0.9185** | **32.18** | **0.8999** | **32.04** | **0.9272** | **38.92** | **0.9776** |
| w/o DS, w/M | X2 | 239 | 67 | 1223 | 37.96 | 0.9605 | 33.54 | 0.9177 | 32.16 | 0.8996 | 31.88 | 0.9257 | 38.78 | 0.9773 |
| w/o DS, w/o M | X2 | 236 | 66 | 1152 | 37.96 | 0.9604 | 3351 | 0.9175 | 32.15 | 0.8994 | 31.83 | 0.9250 | 38.71 | 0.9772 |
| **Ours** | X3 | 244 | 28 | 283 | 34.36 | 0.9270 | 30.38 | 0.8428 | 29.11 | 0.8056 | 28.10 | 0.8503 | 33.72 | 0.9448 |
| w/o DS, w/M | X3 |  | 32 | 572 |  |  |  |  |  |  |  |  |  |  |
| w/o DS, w/o M | X3 |  | 31 | 539 |  |  |  |  |  |  |  |  |  |  |
| **Ours** | X4 | 251 | 15 | 152 | 32.21 | 0.8949 | 28.63 | 0.7822 | 27.60 | 0.7367 | 26.08 | 0.7837 | 30.59 | 0.9081 |
| w/o DS, w/M | X4 |  | 17 | 308 |  |  |  |  |  |  |  |  |  |  |
| w/o DS, w/o M | X4 |  | 17 | 290 |  |  |  |  |  |  |  |  |  |  |

위 테이블의 x3 x4는 안 채울 수도 있나?? 우선순위 잘 모르겠다.

Table n는 스케일 또는 더하기를 제외한, 내 모델의 PE와 MLP, 그리고 aggr, window sum 방식에 대해 실험함. SwinT의 PE와 CSwinT의 LePE에 대해 실험 + ViT의 일반적인 MLP와 SAFMN에서 제안한 CCM에 대해 성능 실험. 또는 aggr이 있을 때와 없을 때 실험. (x2에 대해서만) 이건 딱히 뭔가 그림으로 보여줄 수 있는게 없을 듯.

*RPE로 위치 임베딩 한게 더 성능 좋긴 한데, #param이 더 증가함. 따라서 걍 LePE로 유지.*

(Tab N)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | Acts | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| **Ours** | **X2** | **239** | **59** | **603** | **38.01** | **0.9606** | **33.60** | **0.9185** | **32.18** | **0.8999** | **32.04** | **0.9272** | **38.92** | **0.9776** |
| PE -> RPE | X2 | 244 |  |  | 38.02 | 0.9606 | 33.62 | 0.9182 | 32.20 | 0.9000 | 32.05 | 0.9275 | 38.90 | 0.9775 |
| 보간 -> WWS | X2 | 236 |  |  | 37.99 | 0.9606 | 33.57 | 0.9178 | 32.18 | 0.8999 | 32.01 | 0.9273 | 38.85 | 0.9774 |
| No ACT | X2 |  |  |  | 37.99 | 0.9605 | 33.55 | 0.9180 | 32.16 | 0.8997 | 31.92 | 0.9260 | 38.83 | 0.9774 |
| No Aggr | X2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| No Act, Aggr | X2 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CCM -> MLP | X2 | 73 |  |  | 37.71 | 0.9593 | 33.25 | 0.9150 | 31.96 | 0.8968 | 31.19 | 0.9187 | 38.03 | 0.9754 |
| Each CCM | X2 | 83 |  |  | 34.87 | 0.9426 | 31.30 | 0.8952 | 30.41 | 0.8749 | 27.76 | 0.8678 | 32.50 | 0.9507 |
| Only LMLT | X2 |  |  |  |  |  |  |  |  |  |  |  |  |  |

테이블 해석 : RPE 쓰는게 더 성능 좋은데 파라미터 증가함. 그래서 베이스 삼음. Act 없음 성능 줄고, 다른 것도 없거나 바꾸면 성능 줄음

**Impact of Number of Heads**

Head가 2개 / 3개일 때의 성능 평가.

**Impact of Number of Blocks and Channels**

*Dim의 개수(60[x2 x3 x4]과 48, 24의 x2 [x3 x4??])와 head 수(2개 3개의 x2), Block 개수(36의 x2의 4 6 12)의 성능 평가*

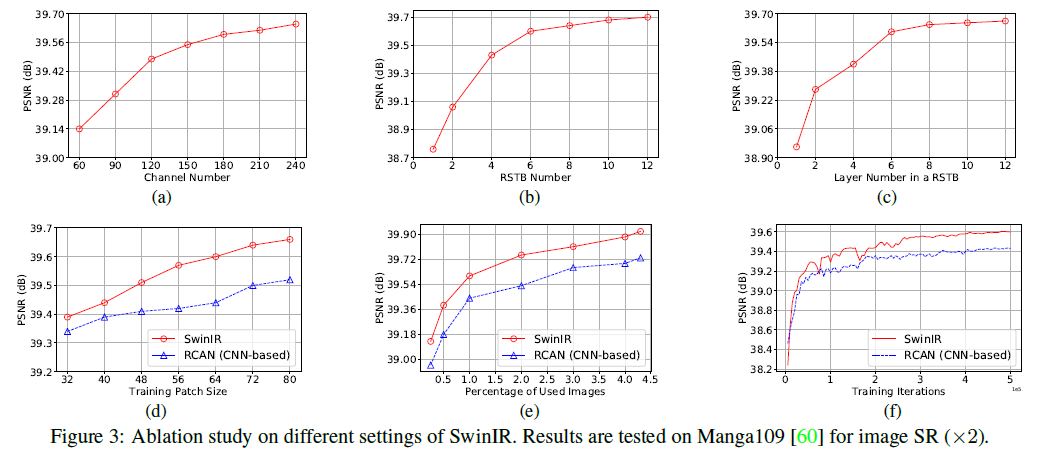
이 섹션에서, 우리는 dim과 block을 여러 개로 나누어 성능을 평가한다. 우선 table n을 보듯, block 개수가 8일 때 가장 효율적임.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # of Block | #Param[k] | #FLOPs[G] | #Acts[M] | Set14 | | Manga109 | |
| PSNR | SSIM | PSNR | SSIM |
| 4 | 122 | 30 | 307 | 33.40 | 0.9166 | 38.47 | 0.9765 |
| 6 | 181 | 44 | 455 | 33.52 | 0.9175 | 38.74 | 0.9771 |
| 8 | 239 | 59 | 603 | 33.60 | 0.9185 | 38.92 | 0.9776 |
| 12 | 357 | 88 | 898 | 33.65 | 0.9187 | 39.10 | 0.9780 |

다음으로, head의 개수에 따른 성능 평가. Head가 4개, 즉 각 head 당 dim이 9일 때 성능이 제일 효율적.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # of head | #Param[k] | Each head dim | #FLOPs[G] | #Acts[M] | Set14 | | Manga109 | |
| PSNR | SSIM | PSNR | SSIM |
| 4 | 239 | 9 |  | 603 | 33.60 | 0.9185 | 38.92 | 0.9776 |
| 3 |  | 12 |  |  |  |  |  |  |
| 2 |  | 18 |  |  |  |  |  |  |

Dim에 따른 성능평가는 모든 결과는 appendix로 빼든가 하고, 아래처럼 간단하게 보여줄 예정. 단 dim이 60인 경우는 따로 빼서 보여주기.

****

예시 그림. 전체적으로 봤을 때, dim=36, 블록=8개, head=4일 때 가장 효율적이라는 그림을 넣을 예정. 또한 dim=60의 경우 다른 efficient network와 비교.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | DTS | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| LatticeNet | X2 | 756 | 170 | D2K | 38.06 | 0.9607 | 33.70 | 0.9187 | 32.20 | 0.8997 | 32.25 | 0.9288 | 38.94 | 0.9774 |
| RFDN-L | X2 | 626 | 146 | D2K | 38.08 | 0.9606 | 33.67 | 0.9190 | 32.18 | 0.8999 | 32.24 | 0.9290 | 38.95 | 0.9773 |
| SRPN-Lite | X2 | 609 | 140 | DF2K | 38.10 | 0.9618 | 33.70 | 0.9189 | 32.25 | 0.9018 | 32.26 | 0.9294 |  |  |
| HNCT | X2 | 357 | 82 | D2K | 38.08 | 0.9608 | 33.65 | 0.9182 | 32.22 | 0.8990 | 32.22 | 0.9294 | 38.87 | 0.9774 |
| FMEN | X2 | 748 | 172 | DF2K | 38.10 | 0.9609 | 33.75 | 0.9192 | 32.26 | 0.8995 | 32.41 | 0.9311 | 38.95 | 0.9778 |
| NGSwin | X2 | 998 | 140 | D2K | 38.05 | 0.9610 | 33.79 | 0.9199 | 32.27 | 0.9008 | 32.53 | 0.9324 | 38.97 | 0.9777 |
| **Ours** | X2 | **652** | **158** | DF2K | 38.10 | 0.9610 | 33.76 | 0.9201 | 32.28 | 0.9012 | 32.52 | 0.9316 | 39.24 | 0.9783 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | DTS | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| LatticeNet | X2 | 765 | 76 | D2K | 34.40 | 0.9272 | 30.32 | 0.8416 | 29.10 | 0.8049 | 28.19 | 0.8513 | 33.63 | 0.9442 |
| RFDN-L | X2 | 633 | 66 | D2K | 34.47 | 0.9280 | 30.35 | 0.8421 | 29.11 | 0.8053 | 28.32 | 0.8547 | 33.78 | 0.9458 |
| SRPN-Lite | X2 | 615 | 63 | DF2K | 34.47 | 0.9276 | 30.38 | 0.8425 | 29.16 | 0.8061 | 28.22 | 0.8534 |  |  |
| HNCT | X2 | 363 | 38 | D2K | 34.47 | 0.9275 | 30.44 | 0.8439 | 29.15 | 0.8067 | 28.28 | 0.8557 | 33.81 | 0.9459 |
| FMEN | X2 | 757 | 77 | DF2K | 34.45 | 0.9275 | 30.40 | 0.8435 | 29.17 | 0.8063 | 28.33 | 0.8562 | 33.86 | 0.9462 |
| NGSwin | X2 | 1007 | 67 | D2K | 34.52 | 0.9282 | 30.53 | 0.8456 | 29.19 | 0.8078 | 28.52 | 0.8603 | 33.89 | 0.9470 |
| **Ours** | X2 | **660** | **75** | DF2K | 34.58 | 0.9285 | 30.53 | 0.8458 | 29.21 | 0.8084 | 28.48 | 0.8581 | 34.18 | 0.9477 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Method | Scale | Par | Flops | DTS | Set5 | | Set14 | | B100 | | U100 | | M109 | |
| LatticeNet [52] | X2 | 777 | 44 | D2K | 32.18 | 0.8943 | 28.61 | 0.7812 | 27.57 | 0.7355 | 26.14 | 0.7844 | 30.54 | 0.9075 |
| RFDN-L [53] | X2 | 643 | 37 | D2K | 32.28 | 0.8957 | 28.61 | 0.7818 | 27.58 | 0.7363 | 26.20 | 0.7883 | 30.61 | 0.9096 |
| SRPN-Lite [43] | X2 | 623 | 36 | DF2K | 32.24 | 0.8958 | 28.69 | 0.7836 | 27.63 | 0.7373 | 26.16 | 0.7875 |  |  |
| HNCT [22] | X2 | 373 | 22 | D2K | 32.31 | 0.8957 | 28.71 | 0.7834 | 27.63 | 0.7381 | 26.20 | 0.7896 | 30.70 | 0.9112 |
| FMEN [51] | X2 | 769 | 44 | DF2K | 32.24 | 0.8955 | 28.70 | 0.7839 | 27.63 | 0.7379 | 26.28 | 0.7908 | 30.70 | 0.9107 |
| NGSwin [21] | X2 | 1019 | 36 | D2K | 32.33 | 0.8963 | 28.78 | 0.7859 | 27.66 | 0.7396 | 26.45 | 0.7963 | 30.80 | 0.9128 |
| **Ours** | X2 | **672** | **41** | DF2K | 32.38 | 0.8971 | 28.79 | 0.7859 | 27.70 | 0.7403 | 26.44 | 0.7947 | 31.09 | 0.9139 |

테이블 해석 : NGSwin 이전 것들 기준으로 좋음.

1. Conclusion

인용 논문. Evaluation에 쓸 것과, 뉴립스는 따로 기술.

[1] CARN - Evaluation 용 : Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In Eur. Conf. Comput. Vis., 2018

[2] SwinIR - Evaluation 용 : Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In Int. Conf. Comput. Vis. Worksh., 2021.

[3] DIV2K : Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and KyoungMuLee. Enhanced deep residual networks for single image super-resolution. In IEEE Conf. Comput. Vis. Pattern Recog. Worksh., 2017.

[4] Flickr2K : Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In IEEE Conf. Comput. Vis. Pattern Recog. Worksh., 2017.

[5] Set5 : Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In BMVC, 2012.

[6] Set14 : Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Curves and Surfaces, 2010.

[7] BSD100 : David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Int. Conf. Comput. Vis., 2001.

[8] Urban100 : Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In IEEE Conf. Comput. Vis. Pattern Recog., 2015.

[9] Manga109 : Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications, 2017.

[10] SRCNN : Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In Eur. Conf. Comput. Vis., 2014

[11] Attn is all you need - 뉴립스 : Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in Neural Information Processing Systems, 30, 2017.

[12] ViT : Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2020.

[13] CSwinT : Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12124–12134, 2022.

[14] PVT : Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578, 2021.

[15] PVTv2 : Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media, 8(3):415–424, 2022.

[16] SAFMN - Evaluation 용 :

[17] SwinT : Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022, 2021.

[18] EfficientNetV2 : Mingxing Tan and Quoc Le. EfficientNetV2: Smaller models and faster training. In ICML, 2021.

[19] IMDN - Evaluation 용 : Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multidistillation network.

[20] NGswin - Evaluation 용

[21] N-gram : P Majumder, M Mitra, and BB Chaudhuri. N-gram: a language independent approach to ir and nlp. In International conference on universal knowledge and language,

[22] HNCT - Evaluation 용 : Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. Ahybrid network of cnn and transformer for lightweight image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1103–1112, 2022.

[23] ShuffleMixer - Evaluation 용, 뉴립스 : Long Sun, Jinshan Pan, and Jinhui Tang. ShuffleMixer: An efficient convnet for image super-resolution.

[24] LAPAR - Evaluation 용, 뉴립스 : Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. LAPAR: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond.

[25] MViT : Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6824–6835, 2021.

[26] SegFormer - 뉴립스 : Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers.

[27] FastViT : P. K.A.Vasu,J.Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, “FastViT: A fast hybrid vision transformer using structural reparameterization,” in Proc. Int. Conf. Comput. Vis., 2023.

[28] FasterViT :

[29] NAT

[30] CvT

[31] SASA

[32] HaloNet

[33] ResNet : Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[34] VDSR : Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In CVPR, 2016.

[35] DRCN : Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In IEEE Conf. Comput. Vis. Pattern Recog., 2016.

[36] ESPCN : Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR, 2016.

[37] ELAN-Light : ECCV, Efficient Long-Range Attention Network for Image Super-resolution

[38] SRFormer

[39] Fakd : Zibin He, Tao Dai, Jian Lu, Yong Jiang, and Shu-Tao Xia. Fakd: Feature-affinity based knowledge distillation for efficient image super-resolution. In ICIP, 2020.

[40] Quantization in SR : Andrey Ignatov, Radu Timofte, Maurizio Denna, and et al. Efficient and accurate quantized image super-resolution on Mobile NPUs, Mobile AI & AIM 2022 challenge: Report. In ECCV Workshops, 2022.

[41] 프루닝 : Russell Reed. Pruning algorithms-a survey. IEEE transactions on Neural Networks, 4(5):740–747, 1993.

[42] EDSR - 프루닝 : Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and KyoungMuLee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017.

[43] SRPN-Lite - 프루닝 : Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In International Conference on Learning Representations, 2021.

[44] 프루닝 : Iterative Soft Shrinkage Learning for Efficient Image Super-Resolution

[45] SSIM : Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Si moncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[46] PAN – 평가용 : Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel at tention. In ECCV Workshops, 2020.

[47] ECBSR – 평가용 : Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In ACM MM, 2021.

[48] SMSR – 평가용 : Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In CVPR, 2021.

[49] DAT – related works 용 : Dual Aggregation Transformer for Image Super-Resolution

[50] DLGSANet – RW 용 : DLGSANet: Lightweight Dynamic Local and Global Self-Attention Network for Image Super-Resolution

[51] FMEN – 평가용

[52] LatticeNet – 평가용

[53] RFDN-L – 평가용

[54]

[55]

[56]

[57]

[58]

[59]

[60]

[61]

[62]

[63]

[64]

[65]

[66]

[67]

[68]

[69]

[70]

[71]

[72]

[73]

[74]

[75]

[76]

[77]

[78]

[79]

[80]

[81]

[82]

[83]

[84]

[85]

[86]

[87]

[88]

[89]

[90]

()안의 숫자는, 파파고로 번역했을 때의 공백제외 글자수

주의할 점 –

1. N-gram SwinT는 HNCT에도 N-gram을 적용하여 실험. 나도 SRFormer에 내거 적용해서 실험?
2. Lightweight image SR + classical image SR 다 해봐야 됨??

2-1) Ngram은 각각 NGSwin / SwinIRNG로 차이를 둠

2-2) SRFormer은 depth와 embed\_dim 차이로 차이를 둠

1. HNCT(와 SRFormer)의 Ablation study 진행 시, 성능을 완벽히 맞춰야 하는가? 아니면 달라도 상관 ㄴ? (N-gram은 GPU 4개 쓰고, SRFormer 역시 그러함. HNCT는 명시 안 돼서 모름)

그림 리스트

1: 성능비교 / 2: 아키텍처 / 3: ShinePost상세 그림 / 4: 패딩 / 5: 분석 그림(between SwinIR과 SwinIRNG) / 6~ : 예시 이미지 (LAB[Interpreting Super-Resolution Networks with Local Attribution Maps, [Interpreting Super-Resolution Networks with Local Attribution Maps - Interpretable Low-Level Vision (x-lowlevel-vision.github.io)](https://x-lowlevel-vision.github.io/lam.html)]

표 리스트

1: SwinIRNG와의 param, MACs 차이(아마 param은 다르고 MAC는 다를것) / 2:내거 x2, x3, x4 / 3: 패딩 별 성능 / 4: 패딩의 Ablation study between SwinIRNG / 5: SRFormer과 내 SRFormer+SP?? / 6 : halo가 2, 3.. 일 때 성능 / 7 : 그냥 SwinIR + SP??

Title: Swin Transformer with Neighbor Cardinal Positions for Image Super-Resolution

Abstract

1. Introduction (3400자)

**단일 이미지 초해상도(Single Image Super Resolution, SR)는 저해상도 이미지를 고해상도 이미지로 변환하는 작업이다**. 최근 많은 super-resolution 연구들이 딥러닝 기반 방법, 특히 CNN과 Vision Transformer 기반 방법을 사용하여 이미지의 해상도를 높이려고 시도하고 있다.

CNN을 사용하는 이미지 초해상도 아키텍처는 dense connection이나 residual learning과 같은 방식을 사용하여, 전통적인 super-resolution 모델에 비해 획기적인 성능 향상을 달성한다. 한편, transformer 기반 모델은 self-attention 매커니즘을 디자인하여 context간의 global interaction을 캡쳐한다. SwinIR은 Swin Transformer 기술을 super-resolution에 적용함으로써 이전에 발표되었던 여러 CNN기반의 state-of-the-art 모델에 비해 높은 성능을 달성한다. 그러나 SwinIR에는 여전히 문제점이 하나 있다. **SwinIR은 이미지를 고정된 사이즈의 local window로 분할하기에, 윈도우와 윈도우 사이의 경계 간 context 교환이 이뤄지지 않는 border effect가 발생하며, 이는 곧 restored image의 visual quality를 degrade시킨다.**

**이러한 문제를 극복하기 위해 다양한 방법이 등장한다**. N-gram swin transformer은 NLP의 n-gram 아이디어를 적용해 이미지 패치를 일종의 단어로 취급하고, 인접한 윈도우 간에 self-attention을 적용하여 SwinIR의 문제점을 극복한다. 또한 SRFormer의 경우, feature의 spatial-size를 줄임으로써 연산량의 부담 없이 윈도우의 사이즈를 크게 함으로써 성능을 높인다.

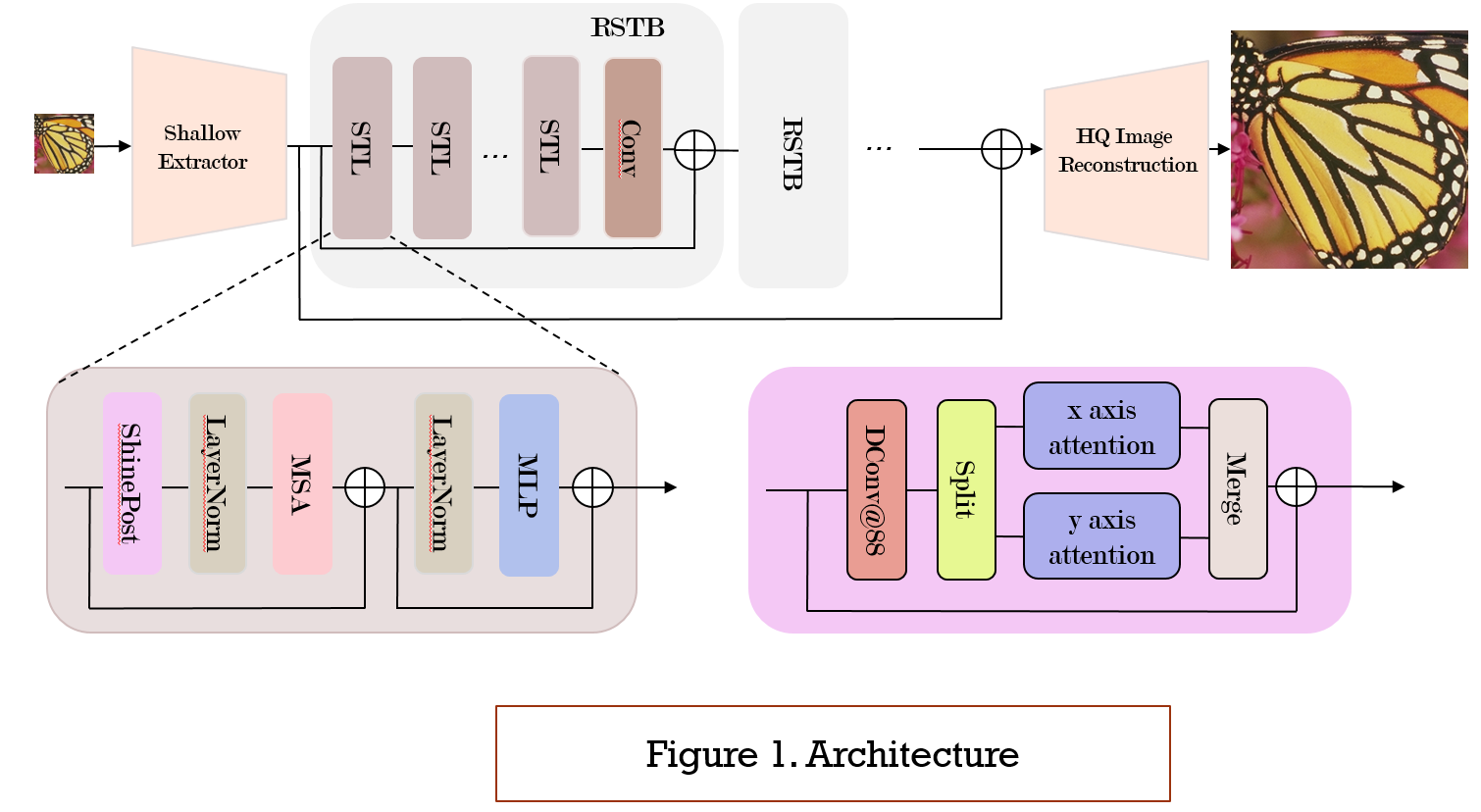
**이 논문에서는 상술한 문제를 해결하기 위해, 이 논문에서는 쿼리 윈도우에 대응되는 키와 밸류로 상하좌우의 추가 영역, 즉 헤일로를 주는 방법을 제안한다.** 기존의 HaloNet에서 제안한 헤일로를 주는 방식은 윈도우의 상하좌우와 대각선 방향까지 포함하여 추가하였기에 헤일로가 커질수록 추가되는 영역이 제곱으로 증가하고, 이는 곧 연산량에 부담이 된다. 따라서 우리는 stripe로 윈도우를 만드는 CSwin Transformer의 아이디어를 적용해, 이미지 패치에 각각 상하와 좌우를 추가 영역으로 제공하여 self-attention을 진행한 다음 둘을 merging하는 Swin Transformer with Neighbor Cardinal Position를 제안한다. 이는 오직 halo를 4방향으로만 추가하기에, 헤일로가 커져도 추가되는 영역은 linear하게 증가하여 상대적으로 연산량 부담이 덜하면서도 cross-window communication 문제를 해결한다. (1500)

**그림 1**에서 보이듯, 우리가 제안한 알고리즘의 우수성은 블라블라…

추가적으로, 우리는 패딩얘기

이후 성능 얘기

우리의 기여는 아래와 같다.



1. Related works (5000자) 각각 2500자로…
   1. Vision Transformer

**Transformer은 NLP의 다양한 분야에서 큰 성과를 거뒀으며, 최근 vision task 역시 Transformer 아키텍처를 이용하여 classification, object detection 그리고 semantic segmentation 등 다양한 분야에서 큰 성공을 이루었다.** Transformer은 기존 CNN이 가지고 있던 한계인, 제한된 커널 사이즈로 인해 오직 이미지의 local region에 대해서만 정보의 처리가 가능하기에, long-range dependency modeling에 대한 효율성을 달성하기 어렵다는 문제점을 context간의 global interaction을 캡쳐함으로써 해결한다.

**그러나 Vision Transformer 역시 한계점을 지니고 있다.** 대표적으로, input의 사이즈가 증가할수록 복잡도가 제곱(quadratic)으로 증가한다. 이를 해결하기 위해 SASA에서 local window 내에서 self-attention을 계산하는 방식을 제안한 이래로, HaloNet은 쿼리 주위를 감싸는 밴드를 결합하여 공유 이웃 블록을 획득하는 방식을 통해 ResNet에 비해 높은 성능을 달성했다. Swin Transformer는 이미지를 고정된 윈도우로 자른 다음, 윈도우 내에서 self-attention 매커니즘을 진행한 다음 점차 병합해가는 방식을 사용해 계산량을 감소시킴과 동시에 놀라운 성능을 이끌어내었다. 이외에도 PvT, DAT 등 다양한 ViT 아키텍처가 제안되었다.

**하나의 윈도우와 다른 윈도우 간의 문맥 통신이 없고 global context의 획득이 까다로운 것은 Swin Transformer가 극복해야 할 과제이다.** CSwin Transformer은 윈도우를 stripe 형식으로 만들어 글로벌 정보를 포착한다. NAT나 CvT는 Self-attention 매커니즘에 CNN을 적용시켜, kernel과 stride를 조정해 cross-window communication 문제를 해결한다. 이외에도 여러 아키텍처에서 Vanilla ViT가 가지고 있는 문제점을 해결하기 위한 대안으로 등장하고 있다. (1500)

* 1. Image Super Resolution

SRCNN이 3개의 end-to-end 레이어를 이용해 HR image를 restore하는 방식을 제안한 이래로, **다양한 CNN 기반 모델이 Single Image Super-Resolution의 성능과 효율성을 높이기 위해 제안되었다.** VDSR과 DRCN은 더 깊은 신경망 구조를 사용하고, 네트워크 양 끝단 매핑에 재귀적 신경망 구조를 제안하여 더 높은 품질의 결과를 산출했다. ESPCN은 네트워크의 끝에 있던 업샘플링을 bicubic-filter에서 sub-pixel convolution으로 대체함으로써 SR의 속도를 크게 높였고, 이후 여러 작업에서 사용되게 되었다. LapSRN은 LR 이미지를 입력으로 받아 다양한 해상도에서 feature을 추출한 다음 residual block을 사용하여 LR과 HR 사이의 correlation을 학습하여 성능을 높였다.

**ViT를 포함한 Attention 매커니즘을 Image Super Resolution과 같은 low-level vision task에 적용하는 것 역시 최근에 활발하게 연구되고 있는 분야**이다. IPT는 image denoising, deraining, super resloution과 같은 이미지 프로세싱 문제를 large pre-trained transformer model을 통해 해결한다. Liang et al.은 Swin Transformer에 기반한 Image Super Resolution모델인 SwinIR을 제안하였고, 뒤를 이어 ELAN은 SwinIR과 유사한 구조를 가지되, 다양한 윈도우 사이즈를 self-attention에 사용하였다.

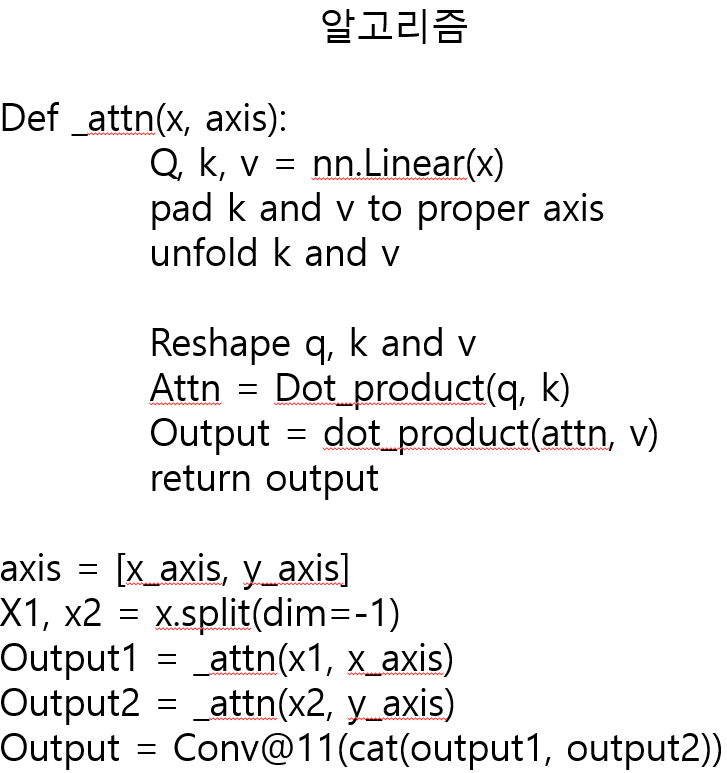
**Self-attention은 이미지의 공간적 특징을 잘 포착하지만, channel-wise로 self-attention을 계산하는 방향으로도 low-level vision task의 해결이 가능하다.** SE-Net이 CNN의 feature을 채널 별로 중요도를 다시 매겨 성능을 올린 이후, RCAN은 이를 super-resolution 모델에 적용하여 residual block과 channel attention 매커니즘을 이용하여 더욱 깊은 네트워크를 구성하였다. 이후 Restormer은 channel-wise self attention만을 이용해 deraining 등 여러 이미지 프로세싱에 탁월한 성능을 보였고, DAT는 Spatial-wise self attention과 Channel-wise self attention을 병용하여, 두 방식을 함께 씀으로 인해 global dependency를 model할 수 있음을 보였다. (1700)

(Ngram 및 SRFormer, restormer 등 여러 ViT 기반 sr 얘기도 함)

1. Proposed Method (7000자)
   1. Method Overview

**우리는 우선 우리가 제안한 모델의 전반적인 구조도에 대해 소개한다.** **그림2**에서 보여지는 것처럼, 우리의 모델은 3개의 모듈: Shallow feature extraction, deep feature extraction 그리고 high quality(HQ) image reconstruction module로 구성되어 있다. LR 이미지 ILR∈RH×W×3이 주어질 경우, 우리는 convolution layer인 SFE를 통해 shallow feature F0∈RH×W×C을 추출한다. 이 때, H와 W는 각각 input image의 높이와 너비를, 그리고 C는 feature 채널의 수를 의미한다. 다음으로 우리는 여러 개의 Swin Transformer Layer을 담고 있는 Residual Swin Transformer Block을 스택하여 구성된 DFE에 F0을 통과시킨다. 이 때, SwinIR의 RSTB와 다른 점으로, 쿼리와 키에 추가 영역을 주는 것이 추가된다. (*추가 설명…)* 마지막으로 SFE와 DFE에서 추출한 feature들을 이용해, high-quality image인 I\_HQ를 reconstruct한다. *(추가 설명)*

*(Loss함수 설명)*

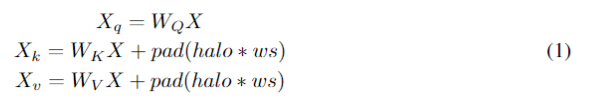


* 1. Neighbor Cardinal Position

그림2(b)에서 보이듯, 우리의 새로운 RSTB에는 ShinePost(가제)라고 하는 새로운 모듈이 추가된다.

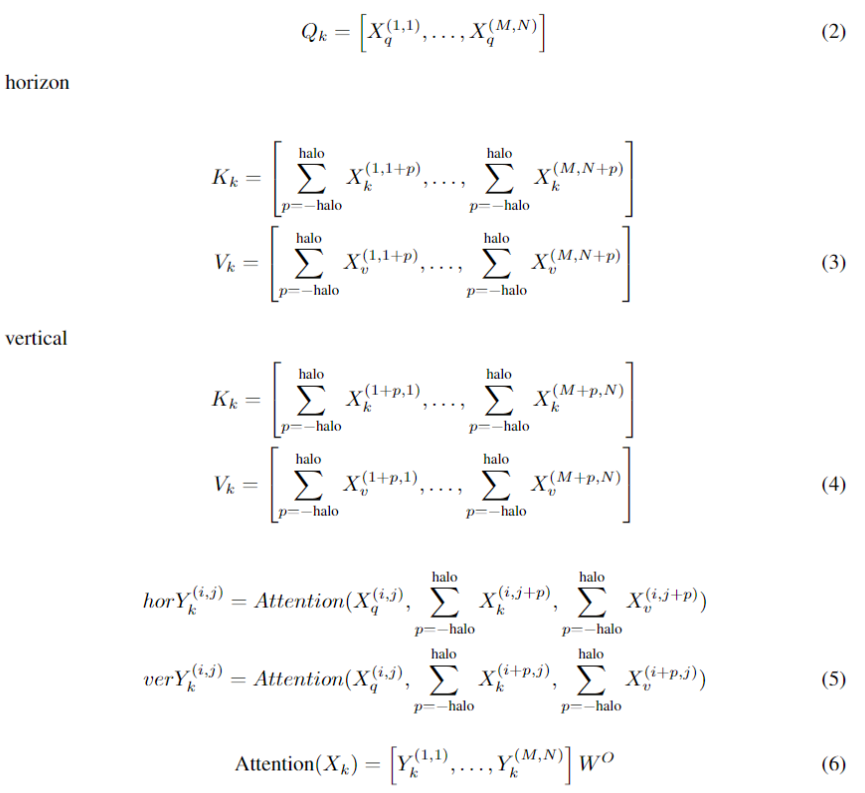
*Ngram과 뭐가 달라졌는지 서술?*

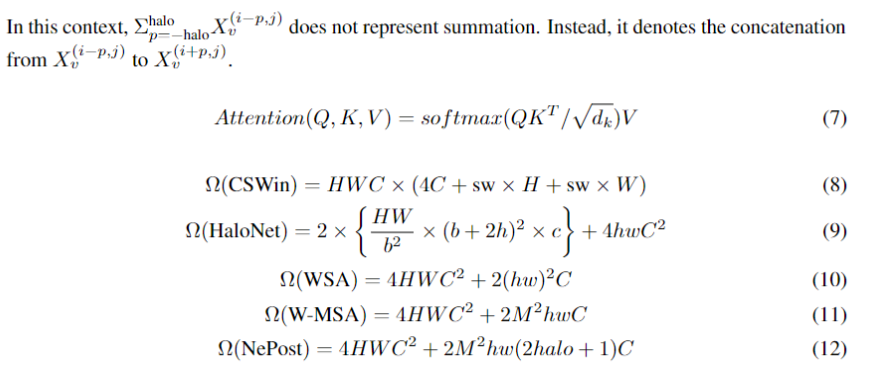
연산량 얘기



* 1. Largee-Halo Window Attention Variants

Halo를 늘릴수록 연산량과 성능이 어떻게 증가하는지(그림 6)





* 1. Discussion

**Difference to NGram**

1. Experiments
   1. Experimental Setup

우리는 우리의 NepoSwinT(가제)를 DIV2K 데이터셋에 있는 800장의 이미지를 이용하여 훈련한다.

* 1. Ablation Study (SRFormer 또는 HNCT)

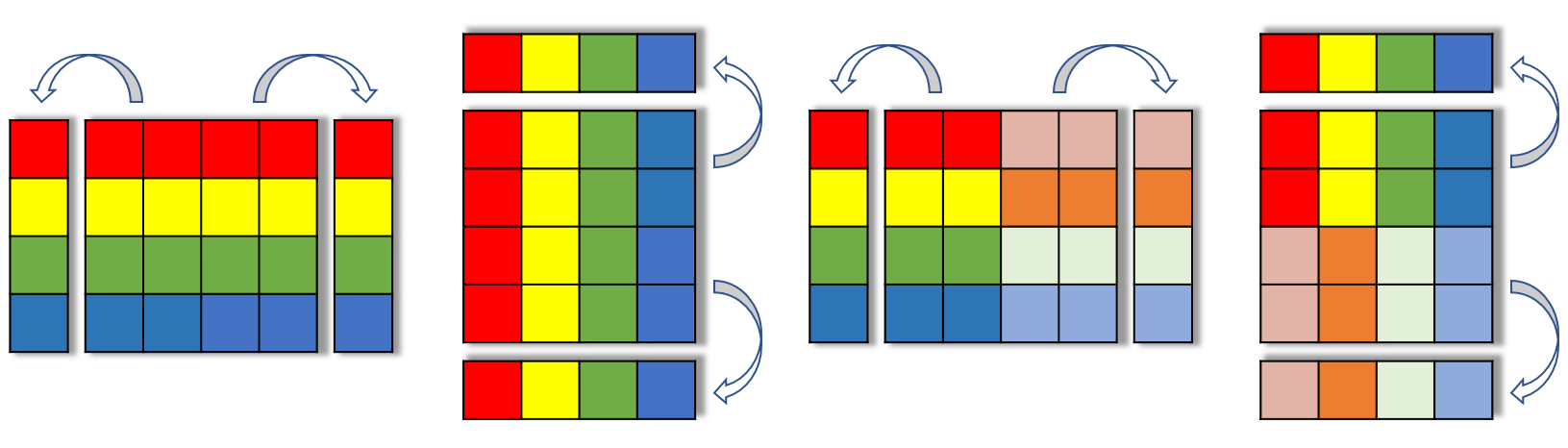
우리는 우리의 NepoSwinT가 얼마나 성능이 오르는지, HNCT와 SRFormer에 삽입하는 Ablation Study를 통해서 보여준다.

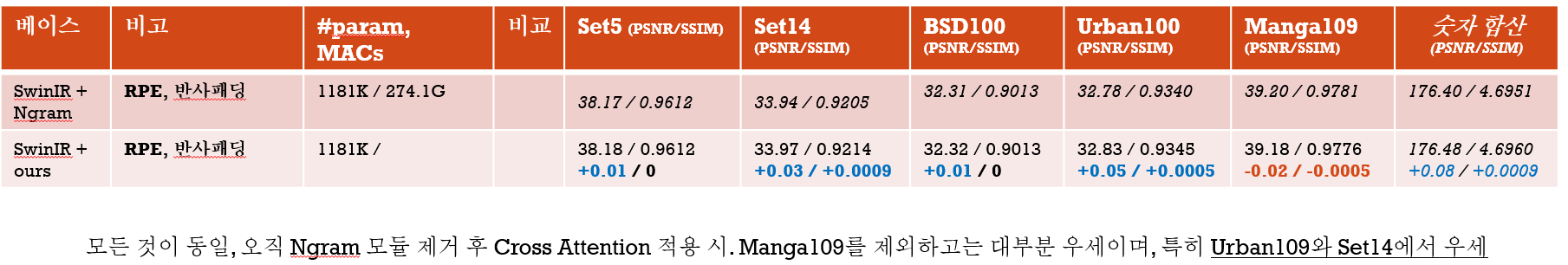
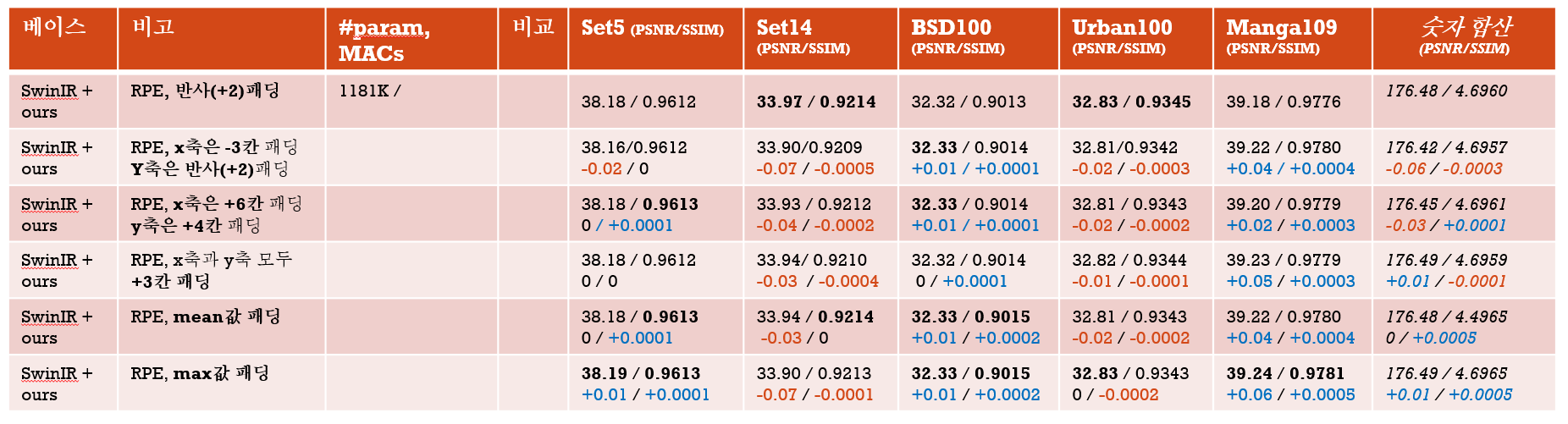
* 1. Comparison with other State-of-the-art methods

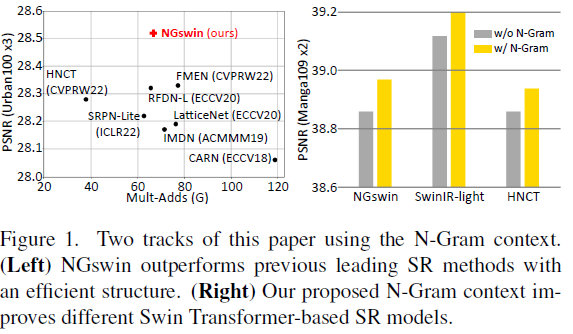
1. Conclusion

CNN-based 모델은 놀라운 성능의 증가를 보여주었지만, CNN을 이용한 super-resolution 아키텍처는 근본적인 한계점을 가지고 있다. 대표적으로, CNN은 제한된 커널 사이즈로 인해 오직 이미지의 local region에 대해서만 정보의 처리가 가능하다. 이는 곧 CNN을 활용한 아키텍처는 long-range dependency modeling에 대한 효율성을 달성하기 어려움을 뜻한다.

Vision Transformer 기반 모델은 self-attention 매커니즘을 디자인하여 context간의 global interaction을 캡쳐함으로써 CNN의 문제점을 해결한다. SwinIR은 Swin Transformer 기술을 super-resolution에 적용함으로써 CNN 기반의 모델보다 높은 성능을 달성한다. 그러나





NGSwin : Super-Resolution 결과만 있고, 두 개의 모델 생성

SwinIR, SRFormer : DIV2K(+Flickr2K)로 실험(**classical image SR**) + DIV2K로 실험(**lightweight image SR**)

SFNet: Learning Object-aware Semantic Correspondence

Abstract

저희는 시맨틱 대응 문제, 즉 동일한 객체 또는 장면 범주의 서로 다른 인스턴스를 묘사하는 이미지 간에 조밀한 흐름 필드를 설정하는 문제를 다룹니다. 저희는 이 작업을 위해 합성 기하학적 변형을 받고 이진 전경 마스크로 주석을 단 이미지를 사용하여 컨볼루션 신경망(CNN)을 훈련할 것을 제안합니다. 이러한 마스크를 감독 신호의 일부로 사용하면 훈련 데이터의 양이 수동으로 점 대응을 선택하는 비용으로 제한되는 시맨틱 흐름 방법과 이미지 간의 단일 전역 기하학적 변환의 회귀가 배경 클러터와 같은 이미지별 세부 사항에 민감할 수 있는 시맨틱 정렬 방법 간에 좋은 절충점을 제공합니다. 저희는 이 아이디어를 구현하는 SFNet이라는 새로운 CNN 아키텍처를 제안합니다. 마스크 및 흐름 일관성과 평활성 용어를 결합한 손실이 있는 엔드투엔드 훈련을 위한 argmax 함수의 새롭고 차별화 가능한 버전을 활용합니다. 실험 결과는 표준 벤치마크에서 최신 기술을 크게 능가하는 접근 방식의 효과를 보여줍니다.

1. 소개

이미지 전반에 걸쳐 조밀한 대응을 설정하는 것은 컴퓨터 비전 [5, 30, 36]의 기본 작업 중 하나입니다. 초기 연구는 비디오 시퀀스에서 동일한 장면의 다른 보기(스테레오 매칭 [19, 36]) 또는 인접 프레임(광학 흐름 [4, 5])을 처리하는 데 중점을 두었습니다. 시맨틱 대응 알고리즘 (예: SIFT Flow [30])은 한 걸음 더 나아가 동일한 객체 또는 장면 범주의 서로 다른 인스턴스를 묘사하는 이미지 사이의 조밀한 흐름 필드를 찾습니다. 이는 특히 외관/장면 레이아웃 및 배경 클러터의 큰 변화가 있는 경우 매우 어렵습니다. 시맨틱 대응에 대한 고전적인 접근 방식 [3, 20, 26, 30, 47]은 일반적으로 충실도 및 정규화 용어를 포함하는 객관적인 함수를 사용합니다. 충실도 용어는 수공예 특징 (예: SIFT [32], HOG [7], DAISY [45])이 이미지 사이의 조밀한 흐름 필드를 따라 일치하도록 장려하고, 정규화 용어는 불연속성을 객체 경계에 정렬하면서 부드럽게 만듭니다. 객체 인식 [10, 30], 시맨틱 분할 [26], 공동 분할 [44], 이미지 편집 [8] 및 장면 파싱 [26, 50]을 포함한 다양한 컴퓨터 비전 작업에서 유용한 것으로 입증되었지만, 수공예 특징은 고수준 시맨틱 (예: 외관 및 모양 변형)을 캡처하지 않으며, 이미지별 세부 사항 (예: 텍스처, 배경 클러터, 폐색)에 강건하지 않습니다.

합성곱 신경망(CNN)은 지난 몇 년 동안 의미론적 대응에서 괄목할 만한 발전을 허용했다. CNN[6, 16, 23, 24, 27, 35, 40, 41, 42, 51]을 사용하는 최근 방법은 클래스 내 변형에 불변하는 풍부한 의미론적 특징의 이점을 활용하여 최첨단 결과를 달성한다. 의미론적 흐름 접근법[6, 16, 27, 35, 51]은 개별 픽셀 또는 패치에 대한 대응을 찾으려 시도한다. 비강체 변형의 영향을 크게 받지 않지만 배경 클러터로 인해 쉽게 주의를 산만하게 한다. 또한 훈련을 위해서는 지상 진실 대응을 가진 많은 양의 데이터가 필요하다. 픽셀 수준의 의미론적 대응은 매우 강력한 제약을 부과하지만 수동으로 주석을 다는 것은 매우 노동 집약적이고 다소 주관적이어서 사용 가능한 훈련 데이터의 양이 제한된다[14]. 대안은 기능 설명자만 학습하거나 렌더링 엔진이 제공하는 3DCAD 모델을 활용하는 것이다[51]. 반면에 의미론적 정렬 방법[23, 24, 40, 41, 42]은 의미론적 대응을 기하학적 정렬 문제로 공식화하고 글로벌 변환 모델(예: 아핀 및 박판 스플라인)의 매개변수를 직접 회귀하여 열두 이미지로 만든다. 이는 무작위 변환을 사용하여 지상 진실 매개변수가 합성적으로 생성되는 자체 지도 학습을 활용하지만 비강체 변형에 대한 더 높은 민감도를 가진다. 또한 배경 클러터는 개별 객체에 초점을 맞추는 것을 방지하고 변환 매개변수를 추정하는 것을 방해한다. 이러한 문제를 극복하기 위해 최근 방법은 이상을 세거나 주의 프로세스[42]를 계산하여 주의를 산만하게 하는 영향을 완화한다.

본 논문에서는 객체 인식 시맨틱 흐름을 확립하기 위한 새로운 접근 방식을 제시하고 이진 전경 마스크를 감독 신호로 활용할 것을 제안합니다(그림 1). 저희의 접근 방식은 이미지 간의 고품질 대응을 통해 공통 객체를 배경에서 분할할 수 있다는 통찰력을 기반으로 합니다. 이 아이디어를 구현하기 위해 하위 픽셀 수준에서 시맨틱 흐름 필드를 출력하는 SFNet이라는 새로운 CNN 아키텍처를 소개합니다. 저희는 마스크/흐름 일관성 및 평활도 용어와 함께 새롭고 차별화 가능한 버전의 argmax인 커널 소프트 argmax를 활용하여 SFNet 엔드 투 엔드(end-to-end)를 훈련하여 산만한 세부 정보를 필터링하면서 객체 인식 대응을 설정합니다. 저희의 접근 방식에는 다음과 같은 이점이 있습니다: 첫째, 마스크는 대규모 데이터 세트에 사용할 수 있고, 좋은 제약 조건 세트를 제공하기 때문에 현재의 의미 흐름과 정렬 방법 사이의 좋은 절충안입니다. 훈련을 위해 이진 전경 마스크를 명시적으로 활용하면 중요한 객체와 장면 요소 간의 대응 학습에 집중할 수 있습니다. 테스트 시간에 마스크가 필요하지 않습니다. 둘째, 저희의 방법은 파라메트릭 회귀(즉, 의미론적 정렬)보다 비강체 변형에 더 강력한 조밀한 비파라메트릭 흐름 필드(즉, 의미론적 흐름)를 설정합니다. 마지막으로, 커널 소프트 argmax는 전체 네트워크를 엔드투엔드로 훈련할 수 있도록 하므로 저희의 접근 방식은 의미론적 대응 작업에 특화된 높은 수준의 의미론을 통해 추가적인 이점을 얻습니다. 이 논문의 주요 기여는 다음과 같이 요약할 수 있습니다:

• 픽셀 수준의 지상 실측보다 더 쉽게 주석을 달 수 있고 널리 사용할 수 있는 이진 전경 마스크를 직접 활용하여 손실 함수에 통합하여 의미 흐름을 학습할 것을 제안한다.

• 우리는 커널 소프트 argmax를 도입하여 서브 픽셀 수준에서 차별화 가능한 흐름 필드를 제공하면서 다중 모드 분포에 덜 민감하게 만든다.

• 우리는 의미론적 대응을 위한 표준 벤치마크에 새로운 기술을 적용하여 전경 마스크를 활용하는 접근 방식의 효과를 명확하게 보여주었다. 또한 절제 연구를 통해 광범위한 실험 분석을 제공한다.

우리의 코드와 모델은 비교와 향후 작업을 장려하기 위해 온라인으로 제공됩니다. https://cvlab- yonsei. github.io/projects/SFNet .

2. 관련업무

대응 문제는 컴퓨터 비전에서 스테레오, 동작 분석, 물체 인식 및 모양 일치를 포함한 광범위한 주제를 다룬다. 이 주제들에 대해 포괄적으로 검토하는 것은 이 논문의 범위를 벗어난다. 우리는 우리와 관련된 대표적인 작품들을 간략하게 검토한다.

고전적인 접근 방식은 예를 들어, 일치 [32]와 같은 희소 대응을 찾거나 동일한 장면/객체의 주변 뷰 사이의 밀도 높은 일치를 설정하는 데 중점을 두었습니다. 이와 달리, 의미론적 대응 방법은 동일한 객체 또는 장면 카테고리의 서로 다른 인스턴스를 포함하는 사진에 걸쳐 밀도 높은 일치를 추정합니다. 의미론적 대응에 대한 초기 연구는 그래픽 모델 [20, 26, 30] 또는 무작위 샘플링 [1, 47]을 사용한 공간 정규화와 함께 SIFT [3, 20, 26, 30], DAISY [47] 및 HOG [14, 44, 46]과 같은 수공예 기술자의 로컬 특징을 일치시키는 데 중점을 둡니다. 그러나 고수준 의미론을 고려하면서 수공예 특징을 설계하는 것은 매우 어렵고, 예를 들어, 클러터, 텍스처, 혼탁 및 외관 변형에 의해 이들 간의 컴퓨팅 유사성이 쉽게 분산됩니다. 객체 제안을 매칭[14, 46]을 위한 후보 영역으로 사용하거나 스케일 공간 [38]에서 매칭을 수행하여 객체/객체 부분 간의 배경 클러터 또는 스케일 변화에 강력한 대응을 추정하려는 많은 시도가 있습니다.

최근 CNN의 이미지 특징은 높은 수준의 의미론을 표현하는 강력한 용량과 모양 및 모양 변화에 대한 견고성을 보여주었습니다[17, 29, 43]. Long et al. [31]은 먼저 이미지 간의 의미론적 대응을 설정하기 위해 CNN을 적용합니다. SIFT Flow [30] 방법과 동일한 절차를 따르지만 픽셀 수준의 주석이 있는 훈련 데이터 세트가 부족하여 ImageNet 분류 작업에 대해 훈련된 기성 CNN 특징을 활용합니다. 이 문제는 3D 모델 [51]의 실측 대응을 합성하거나 보간을 사용하여 희소 키포인트 데이터 세트에서 일치 쌍의 수를 증가시킴으로써 완화될 수 있습니다. 보다 최근에는 PF 데이터 세트 [15]가 PASCAL 2011 키포인트 데이터 세트 [2]의 실측 주석과 함께 20개 이미지 범주의 1300+ 이미지 쌍을 제공하는 것으로 출시되었습니다. 이를 통해 의미론적 대응 작업에 특정한 로컬 특징 [16, 27, 35]을 학습할 수 있습니다. CNN 기능을 사용하는 이러한 접근 방식은 초기 방법보다 큰 차이로 성능이 뛰어나지만 훈련을 위한 손실 함수는 주로 흐름 필드의 차별화 가능성 부족으로 인해 공간 정규화를 포함하지 않습니다. 대조적으로, 우리의 흐름 필드는 차별화 가능하여 공간 정규화기 엔드투엔드로 전체 네트워크를 훈련할 수 있습니다.

최근의 여러 방법[23, 24, 40, 41, 42]은 매개 변수 모델을 사용하여 의미론적 대응을 기하학적 정렬 문제로 공식화한다. 특히, 이러한 방법은 먼저 이미지 간의 특징 상관 관계를 계산하고, 이미지를 정렬하기 위해 전역 변환 모델의 매개 변수(예: 아핀, 호모그래피 및 박판 스플라인)를 추정하기 위해 회귀 계층에 공급된다. 이를 통해 합성적으로 생성된 데이터를 사용하여 자체 지도 학습[24, 40, 41, 42]을 활용하고 전체 CNN을 엔드투엔드로 훈련할 수 있다. 이러한 접근 방식은 모든 픽셀에 동일한 변환을 적용하며, 이는 암시적 공간 정규화의 효과를 가지며, 원활한 일치를 제공하고 종종 의미론적 흐름 방법을 능가한다[6, 14, 16, 27, 51]. 그러나 특징 쌍 간의 상관 관계는 노이즈가 많고 이상치(예: 서로 다른 배경 간)를 포함하기 때문에 배경 클러터 및 폐색[24, 40]에 의해 쉽게 분산된다. 이러한 맥락에서, 우리의 방법은 특징 상관 관계에서 직접 의미론적 대응을 설정함으로써 이 문제를 방지한다.

우리와 유사하게, 많은 방법[23, 27, 50, 51]은 의미론적 대응을 학습하기 위해 객체 경계 상자 또는 전경 마스크를 활용한다. 그러나 이들은 객체 위치를 손실 함수에 사전에 명시적으로 통합하지 않는다. 대신 사전을 훈련 샘플 전처리에 사용합니다. 예를 들어, 양성/음성 훈련 쌍 [23, 27]을 생성하거나 [50, 51]을 일치시키기 위한 후보 영역을 제한한다. 대조적으로, 우리는 사전을 손실 함수에 직접 통합하여 네트워크를 훈련하여 최신 기술을 크게 능가한다.

1. 접근

이 섹션에서는 네트워크 아키텍처(섹 3.1) 및 손실 함수(섹 3.2)를 포함한 객체 인식 의미 대응을 설정하는 접근 방식을 설명한다. 이 방법의 개요는 그림 2에 나와 있다.

3.1. 네트워크 아키텍처

저희 모델은 완전 컨볼루션이며 주로 세 부분으로 구성되어 있습니다(그림 2). 먼저 각 하위 네트워크가 공유 매개변수를 사용하여 동일한 구조를 갖는 샴 네트워크를 사용하여 소스 및 대상 이미지에서 특징인 Is 및 It을 추출합니다. 그런 다음 두 이미지에서 모든 로컬 특징 쌍 간의 일치 점수를 계산하고 커널 소프트 argmax를 통해 각 특징에 가장 적합한 일치를 할당합니다. 모든 구성 요소를 구별할 수 있으므로 전체 네트워크를 엔드 투 엔드로 훈련할 수 있습니다. 다음에서는 소스 대 타겟 매칭을 위한 네트워크 아키텍처에 대해 자세히 설명합니다. 소스 매칭에 대한 대상도 유사하게 계산됩니다.

특징 추출 및 매칭 : 저희는 특징 추출을 위해 이미지넷 분류[9]를 위해 훈련된 ResNet 101[17]을 활용합니다. 이러한 CNN 기능은 풍부한 의미론을 제공하지만 일반적으로 분류를 위해 매우 차별적인 부분을 사용합니다. 이는 세분화된 로컬라이제이션을 위해 공간 변형을 캡처해야 하는 특징 매칭에는 덜 적합할 수 있습니다. 따라서 저희는 추가 적응 레이어를 사용하여 의미론적 대응 작업에 특정한 특징을 추출하여 외관과 공간 컨텍스트 모두에서 매우 차별적인 w.r.t로 변환합니다. 이는 d차원 로컬 특징의 hxw 격자에 해당하는 각 이미지에 대한 크기 hxwxd의 특징 맵을 제공합니다. 그런 다음 개별 d차원 특징에 L2 정규화를 적용합니다. 실험에서 볼 수 있듯이 적응 레이어는 매칭 성능을 크게 향상시킵니다.

풍선

