

# Image Restoration

2023년도 동계 세미나

---



***Sogang University***

*Vision & Display Systems Lab, Dept. of Electronic Engineering*



***Presented By***

*Yoon Chan Nam*

# Outline

- Background
  - Vision transformer
- Uformer: A General U-Shaped Transformer for image Restoration
  - CVPR 2022
- Restormer: Efficient Transformer for High-Resolution Image Restoration
  - CVPR 2022
- Simple Baselines for Image Restoration
  - ECCV 2022

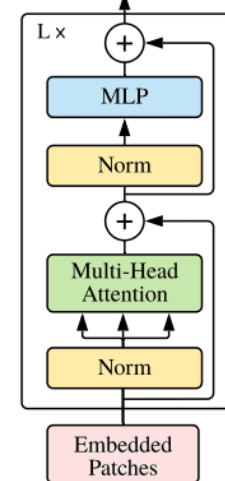
# Background

- Vision transformer

- CNN은 long range dependency를 효과적으로 모델링 하기 어려운 단점을 해결
  - CNN kernel은 content-independent 하기 때문
- 자연어 처리에서 사용되는 transformer를 vision task에 적용
  - 이미지를 patch 단위로 나누고 linear projection을 통해 patch embedding 생성
  - Patch embedding에 position embedding을 더해주고 transformer encoder의 입력으로 사용
  - Multi-head attention을 통한 patch embedding들 간의 관계를 global 하게 고려
  - 최종적으로 MLP를 통해 classification



Transformer Encoder



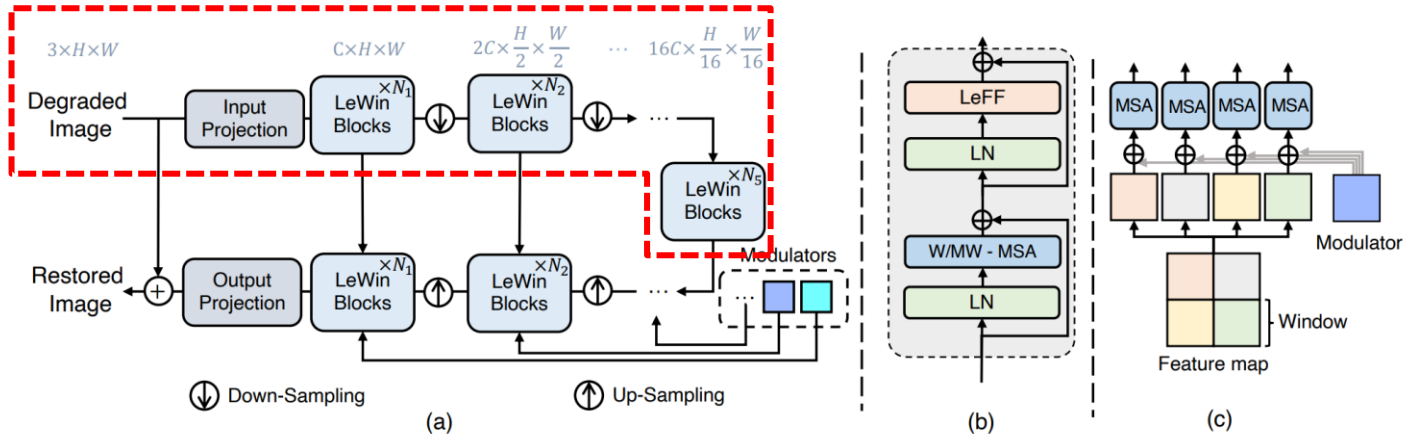
# Uformer: A General U-Shaped Transformer for image Restoration

# Introduction

- CNN 기반의 단점을 보완하고 더 많은 image detail 정보를 복구하는 것이 목표
  - 이미지 복원을 위한 효과적이고 효율적인 transformer-based Uformer 제안
    - Transformer block을 사용하여 계층적 encoder-decoder 구조 구축
    - 기본적인 U-Net을 기반으로 함
- Uformer는 크게 두가지 design으로 구성
  - Local enhanced window (LeWin) transformer block 소개
    - Global self-attention 대신 중복되지 않는 window-based self-attention 사용
      - ※ Local context capture하면서 high resolution feature map에서 계산 복잡도를 줄일 수 있음
    - 계층적 layer에 크기가 고정된 window 사용
      - ※ 더 많은 global dependency를 캡처 할 수 있음
    - Local context도 고려하기 위한 3x3 depth-wise convolution 추가
  - Learnable multi-scale restoration modulator 소개
    - Uformer decoder의 여러 layer에서 feature를 조정하기 위한 multi-scale spatial bias
      - ※ 다양한 image degradation을 처리하기 위함

# Method

- Overall pipeline
  - Skip connection이 있는 계층적 encoder-decoder 구조 사용
  - Degradation image를 input projection module을 통해 low level feature을 생성
    - $3 \times 3$  Conv와 LeakyReLU를 사용
    - ※  $I (3 \times H \times W) \rightarrow X_0 (C \times H \times W)$
  - 이후에 K-stage encoder 입력으로 들어 감
    - 각 encoder stage는 제안된 LeWin transformer block과 down-sampling layer로 이루어져 있음
    - ※ Transformer block을 통과 하여 flatten 된 feature는 down-sampling layer에서 2D feature로 재구성
    - ✓ Down-sampling layer:  $4 \times 4$  Conv, stride = 2
  - 마지막 stage에서도 LeWin transformer로 global dependency capture



# Method

- Overall pipeline

- 압축된 feature 들을 K-stage decoder의 입력으로 사용

- 각 decoder stage는 제안된 up-sampling layer와 LeWin transformer block으로 이루어져 있음

- ※ Transformer block을 통과 하여 flatten 된 feature는 up-sampling layer에서 2D feature로 재구성

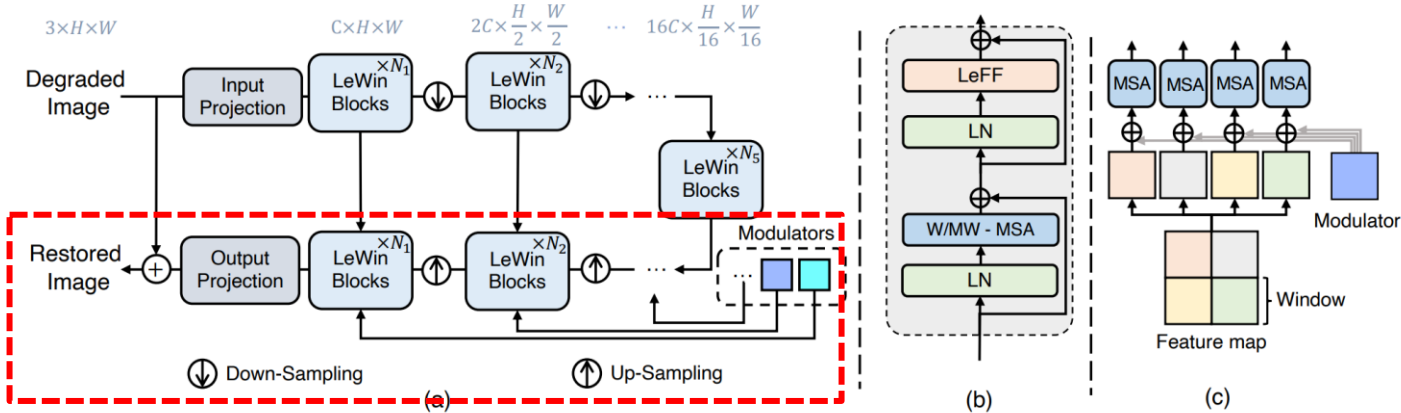
- ✓Up-sampling layer:  $2 \times 2$  Conv, stride = 2

- Decoder stage에서의 Lewin transformer는 skip connection feature와 up-sampled feature를 입력으로 받음

- 최종적으로 decoder를 통해 flatten된 feature를 다시 2D feature map으로 재구성

- Output projection을 통해  $C \times H \times W$  feature로 재구성

- 마지막으로  $3 \times 3$  Convolution layer를 사용하여  $3 \times H \times W$  의 output을 얻음



# Method

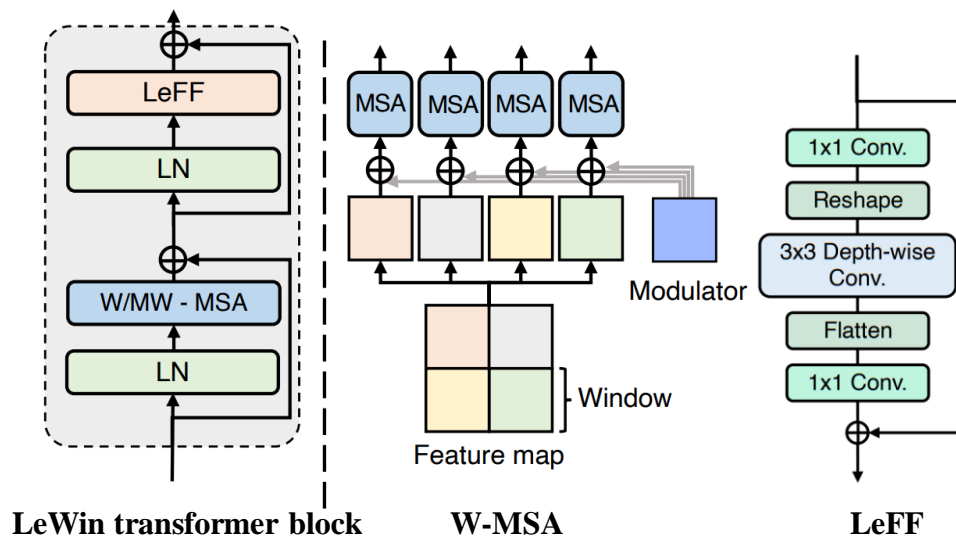
- LeWin (Local Enhanced Window) transformer block

- Non-overlapping Window-based Multi-head Self-Attention (W-MSA)

- High resolution feature map에서 기존에 사용하던 global self-attention은 비 효율적
    - Overlapping 되지 않는 local window 내에서 self-attention수행
      - ※ 계산 비용을 크게 줄일 수 있음

- Locally-enhanced Feed-Forward Network (LeFF)

- Transformer가 local dependency를 다루는데 한계점이 있다는 기존 연구
    - LeWin transformer block 마지막 layer에 사용
      - ※ 유용한 local context를 캡처 할 수 있음





# Method

- Non-overlapping Window-based Multi-head Self-Attention (W- MSA)

- $C \times H \times W$  인 feature map를 겹치지 않게  $M \times M$  window로 분할
- 각 분할된 window마다 self-attention 수행

$$\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^N\}, \quad N = HW/M^2,$$

$$\mathbf{Y}_k^i = \text{Attention}(\mathbf{X}^i \mathbf{W}_k^Q, \mathbf{X}^i \mathbf{W}_k^K, \mathbf{X}^i \mathbf{W}_k^V), \quad i = 1, \dots, N,$$

$$\hat{\mathbf{X}}_k = \{\mathbf{Y}_k^1, \mathbf{Y}_k^2, \dots, \mathbf{Y}_k^N\},$$

-  $N$ : window의 개수

-  $k$ : multi-head attention의 head 수

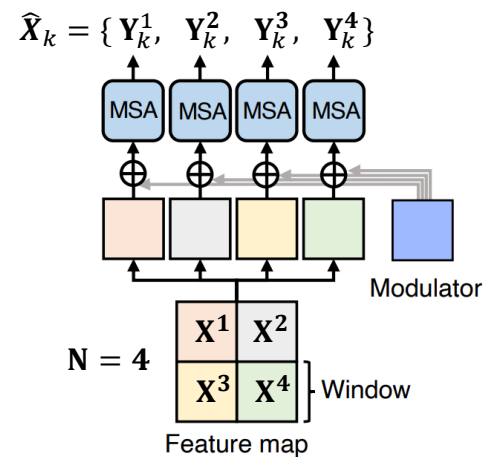
-  $\mathbf{W}_k^Q, \mathbf{W}_k^K, \mathbf{W}_k^V$ : 각각  $k$  번째 head에 대한 queries, keys, values

- 최종적으로 모든 head에 대한 출력을 concatenation 한 다음 linear projection 함

- 기존 global self-attention에 비해 계산 비용을 크게 줄일 수 있음

- 기존 global self-attention:  $O((HW)^2 C)$

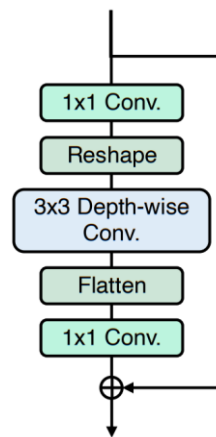
- Window-based self-attention:  $O\left(\frac{HW}{M^2} M^4 C\right) = O(M^2 HWC)$



# Method

- Locally-enhanced Feed-Forward Network (LeFF)
  - Normal transformer의 feed-forward network(FFN)는 local context를 활용하는데 제한적
    - 이웃 pixel 정보를 가져오는 것은 image restoration에서 중요한 문제
  - Transformer의 FFN에 depth-wise convolution block을 추가
    - Linear projection을 통해 확장된 feature를 2D feature로 reshape
    - Depth-wise  $3 \times 3$  Convolution layer로 local 정보를 캡처
    - 이후에 다시 flatten 후 linear layer를 통해 입력 feature의 dimension과 맞춰 줌

※ 각 convolution, linear layer에 GELU를 activation function으로 사용



Locally-enhanced Feed-Forward Network (LeFF)

# Method

- Multi-Scale Restoration Modulator

- Light-weight multi-scale restoration modulator 제안

- 다양한 image degradation을 처리하기 위해 사용
    - Feature를 보정하고 더 많은 세부 정보를 복원

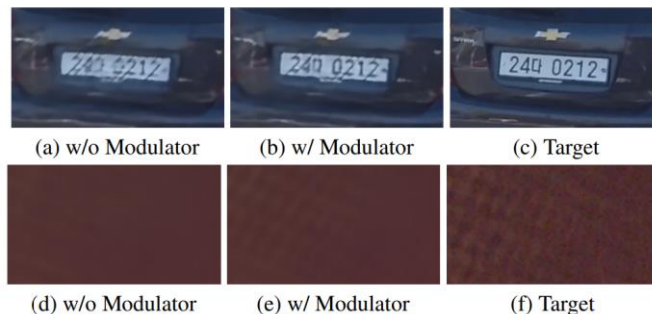
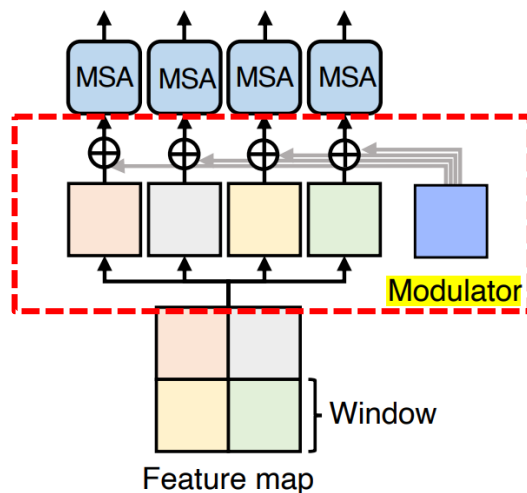
- Uformer의 decoder stage에서 multiple modulator를 적용

- $M \times M \times C$  크기의 학습 가능한 tensor

※  $M$ : window size,  $C$ : feature map channel size

- Self-attention 이전에 각 window에 shared bias의 형태로 더함

- Decoder에 각 단계 추가했을 때 detail 복원 성능이 오르는 것을 실험적으로 알 수 있음



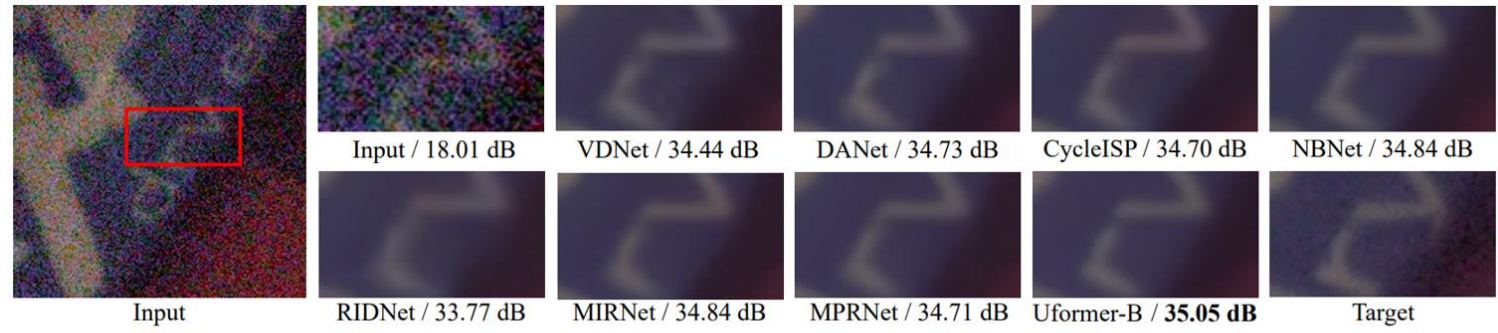
	GoPro [45] Uformer-T	SIDD [1] Uformer-B	SPAD [61] Uformer-B
Modulator	-    ✓	-    ✓	-    ✓
PSNR ↑	29.11 <b>29.57</b>	39.86 <b>39.89</b>	47.43 <b>47.84</b>

Table 8. Effect of the multi-scale restoration modulator.

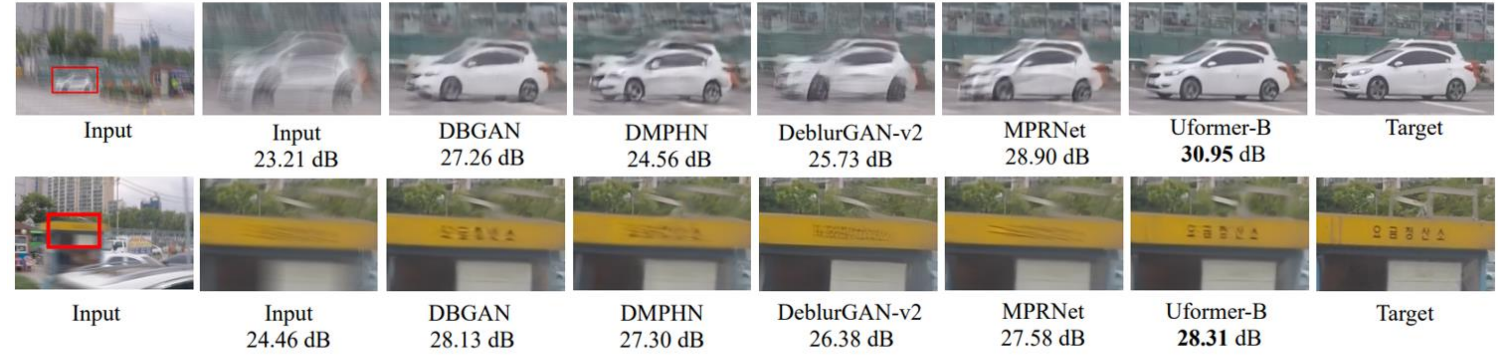
# Results

- Qualitative comparison

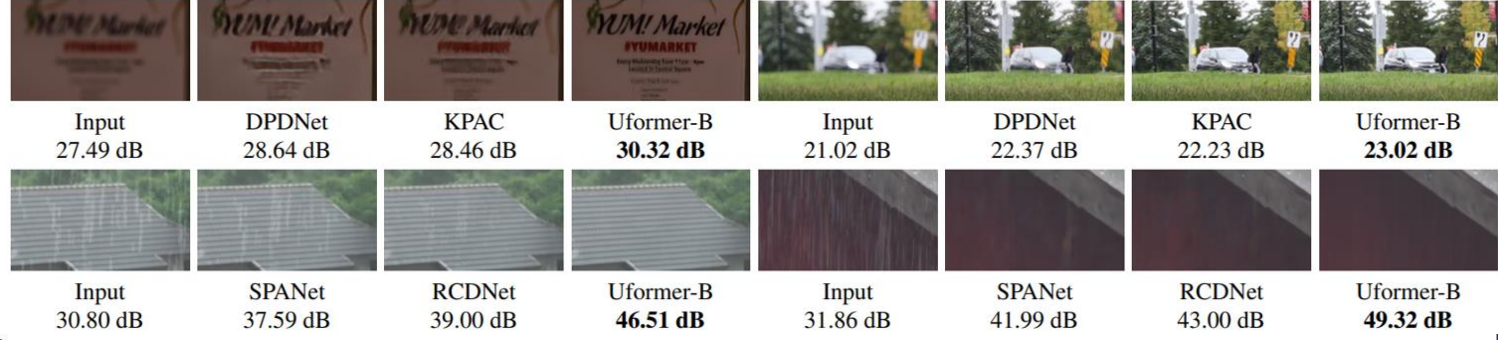
### Denoising



### Motion blur



### Real rain



# Results

- Quantitative comparison

Method	SIDD [1]		DND [46]	
	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$
BM3D [13]	25.65	0.685	34.51	0.851
RIDNet [4]	38.71	0.914	39.26	0.953
VDN [73]	39.28	0.909	39.38	0.952
DANet [74]	39.47	0.918	39.59	0.955
CycleISP [75]	39.52	0.957	39.56	<b>0.956</b>
MIRNet [76]	39.72	<u>0.959</u>	39.88	<b>0.956</b>
MPRNet [77]	39.71	0.958	39.80	0.954
NBNet [9]	<u>39.75</u>	<u>0.959</u>	<u>39.89</u>	0.955
<b>Uformer-B</b>	<b>39.89</b>	<b>0.960</b>	<b>40.04</b>	<b>0.956</b>

**Denoising**

	DMENet [30]	JNB [53]	DPDNet [3]	KPAC [54]	<b>Uformer-B</b>
PSNR $\uparrow$	23.41	23.84	25.13	<u>25.24</u>	<b>26.28</b>
SSIM $\uparrow$	0.714	0.715	0.786	<u>0.842</u>	<b>0.891</b>

**Defocus blur removal**

Method	GoPro [45]		HIDE [52]		RealBlur-R [48]		RealBlur-J [48]	
	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$
Nah <i>et al.</i> [45]	29.08	0.914	25.73	0.874	32.51	0.841	27.87	0.827
DeblurGAN [28]	28.70	0.858	24.51	0.871	33.79	0.903	27.97	0.834
Xu <i>et al.</i> [67]	21.00	0.741	-	-	34.46	0.937	27.14	0.830
DeblurGAN-v2 [29]	29.55	0.934	26.61	0.875	35.26	0.944	28.70	0.866
DBGAN [80]	31.10	0.942	28.94	0.915	-	-	-	-
SPAIR [47]	32.06	0.953	30.29	0.931	-	-	<u>28.81</u>	<u>0.875</u>
†Zhang <i>et al.</i> [79]	29.19	0.931	-	-	35.48	0.947	27.80	0.847
†SRN [56]	30.26	0.934	28.36	0.915	35.66	0.947	28.56	0.867
†DMPHN [78]	31.20	0.940	29.09	0.924	35.70	0.948	28.42	0.860
†MPRNet [77]	32.66	<u>0.959</u>	<b>30.96</b>	<u>0.939</u>	35.99	<u>0.952</u>	28.70	0.873
<b>Uformer-B</b>	<b>33.06</b>	<b>0.967</b>	<u>30.90</u>	<b>0.953</b>	<b>36.19</b>	<b>0.956</b>	<b>29.09</b>	<b>0.886</b>

**Motion blur removal**

	GMM [34]	RESCAN [33]	SPANet [61]	JORDER-E [71]	RCDNet [60]	SPAIR [47]	<b>Uformer-B</b>
PSNR $\uparrow$	34.30	38.11	40.24	40.78	41.47	<u>44.10</u>	<b>47.84</b>
SSIM $\uparrow$	0.9428	0.9707	0.9811	0.9811	0.9834	<u>0.9872</u>	<b>0.9925</b>

**Real rain removal**

# Conclusions

- Transformer block을 기반으로 하는 image restoration을 위한 Uformer architecture 제안
  - LeWin transformer block을 이용하여 기존 Convolution network 기반 구조의 단점 해결
    - Local context를 처리 할 수 있을 뿐만 아니라 long-range dependency를 효율적으로 캡처 가능
  - Multi-scale restoration modulator 제안
    - Uformer의 decoder에 추가되어 다양한 image degradation을 처리하고 복원 성능 향상
  - 광범위한 실험을 통해 image restoration 분야에서 state-of-the-arts 성능 달성
    - Denoising, motion deblurring, defocus deblurring and deraining
  - 계산 비용이 적고, 모델 parameter가 적기 때문에 기존 U-Net 구조를 능가함
    - 같은 depth의 normal U-Net과의 실험 비교

	GMACs	# Param	PSNR ↑
UNet-T	15.49G	9.50M	39.62
UNet-S	34.76G	21.38M	39.65
UNet-B	86.97G	53.58M	39.71
ViT	8.83G	14.86M	38.51
Uformer-T	12.00G	5.23M	39.66
Uformer-S	43.86G	20.63M	39.77
Uformer-B	89.46G	50.88M	<b>39.89</b>

---

# Restormer: Efficient Transformer for High-Resolution Image Restoration

# Introduction

- Transformer-based model의 단점
  - Resolution에 따라 계산 복잡도 증가
  - High resolution image에 대한 복원작업에 적용이 어려움
    - Long-range pixel의 관계를 다루기 어려움
      - ※ Uformer와 같이 window 범위를 제한하는 것은 long-range pixel 관계를 다루는데 있어 모순됨
- 다양한 image restoration 분야에서 적용 가능한 restoration transformer (Restormer) 제안
  - Multi-scale encoder-decoder 구조에 transformer block 사용
    - Long-range pixel representation을 캡처 할 수 있음
  - Multi-Depth-wise convolution head Transposed Attention block (MDTA) 제안
    - Depth-wise convolution으로 spatial local context를 강조할 수 있음
    - Pixel 간의 global relationship을 모델링 할 수 있음
  - Gated-Depth-wise convolution Feed-forward Network (GDFN) 제안
    - Spatial local context를 강조할 수 있음
    - 후속 layer에 더 정제된 image feature를 전송

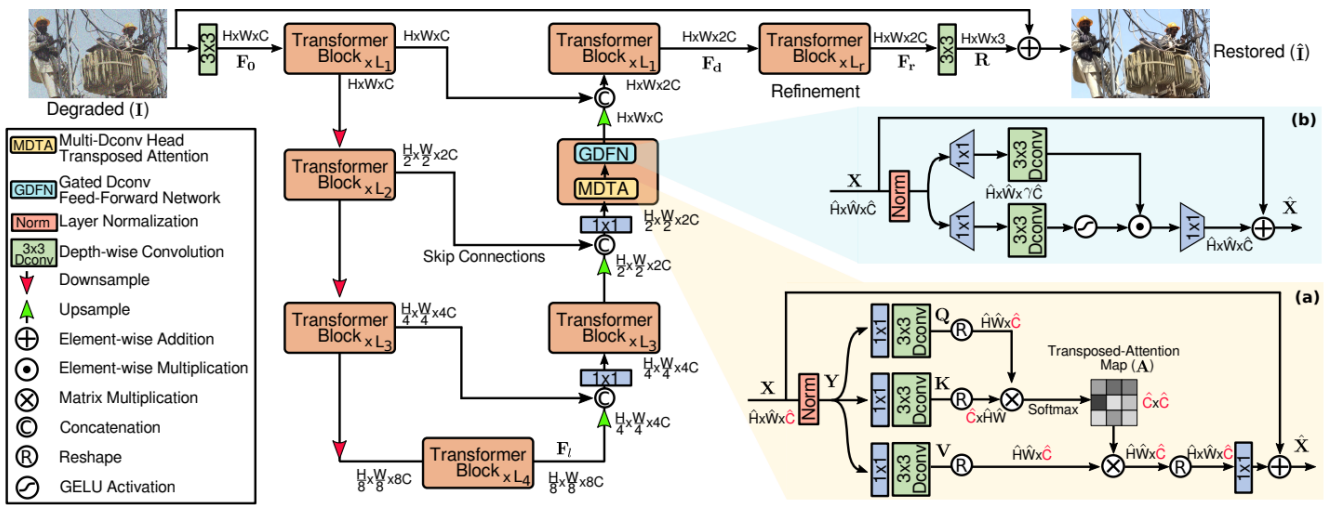


# Method

- Overall pipeline

- Skip connection이 있는 4-level의 계층적 encoder-decoder 구조 사용
    - 각 level에는 여러 layer의 transformer block 포함
  - $3 \times 3$  Conv block을 통과하여 low-level feature를 얻음
    - 이후에 각 계층의 transformer block의 입력으로 사용
- ※ Transformer block은 제안한 MDTA와 GDFN으로 구성

- Decoder를 통과한 feature는 다시 transformer block을 통과하여 refinement
- 최종적으로  $3 \times 3$  Conv block을 통과하여 output map을 생성함
  - Global skip connection을 이용하여 input image를 더해 줌



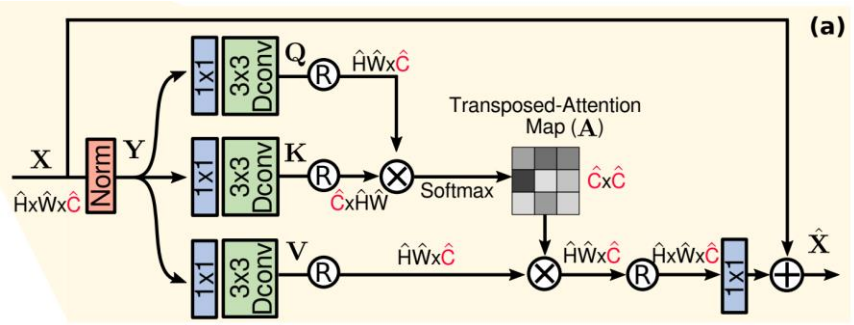
# Method

- Multi-Dconv(Depth-wise convolution) Head Transposed Attention (MDTA)
  - 일반적인 self-attention의 경우 계산 복잡도가 image 해상도에 따라 quadratically 증가
    - 고해상도의 image에 대해 일반적인 self-attention을 적용시키는 것은 어려움
  - Channel 전반에 걸쳐 self-attention을 적용하는 방식을 제안
    - Layer normalized feature( $\hat{H} \times \hat{W} \times C$ )가  $1 \times 1$  Conv,  $3 \times 3$  depth-wise Conv 통과하여 각각 Q, K, V 생성
      - ※  $3 \times 3$  Conv는 local context 정보를 얻기 위함
    - Q, K로 transposed-attention map을 생성하고, V와 element-wise 곱하여 최종 output feature map 생성
      - ※ 기존 regular attention map ( $\hat{H}\hat{W} \times \hat{H}\hat{W}$ ) 대신 transposed-attention map ( $\hat{C} \times \hat{C}$ ) 생성
        - ✓ Feature의 channel 수를 multi-head self-attention의 head 개수로 나눠 병렬적으로 학습
        - ※ 최종적으로 resize 후  $1 \times 1$  Conv 통과

$$\hat{\mathbf{X}} = W_p \text{Attention}(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}) + \mathbf{X},$$

$$\text{Attention}(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}) = \hat{\mathbf{V}} \cdot \text{Softmax}(\hat{\mathbf{K}} \cdot \hat{\mathbf{Q}} / \alpha)$$

- $\hat{\mathbf{Q}} = W_d^Q W_p^Q \mathbf{Q}$ ,  $\hat{\mathbf{K}} = W_d^K W_p^K \mathbf{K}$ ,  $\hat{\mathbf{V}} = W_d^V W_p^V \mathbf{V}$
- $W_d$ :  $3 \times 3$  depth-wise convolution
- $W_p$ :  $1 \times 1$  convolution
- $\hat{\mathbf{X}}$ : output feature
- $\alpha$ : learnable scale parameter



# Method

- Gated-Dconv Feed-forward Network (GDFN)

- Gated-depth-wise convolution

- 두개의 병렬 경로로 이루어져 있으며 두 요소의 곱으로 나타낼 수 있음

- ※ 두 경로를 각각  $1 \times 1$  convolution으로 차원을 확장하고  $3 \times 3$  Dconv를 진행

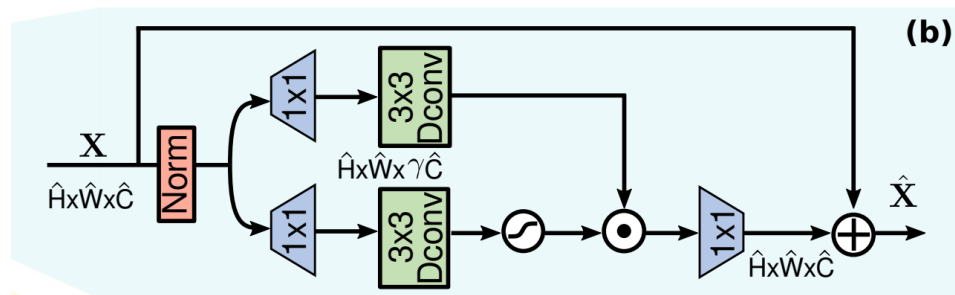
- ※ 하나의 경로에는 GELU activation을 사용함

- ※ 최종적으로 다시  $1 \times 1$  convolution을 통해 입력 channel과 동일

- Feature transformation을 효과적으로 수행 할 수 있음


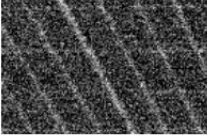
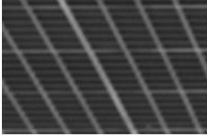
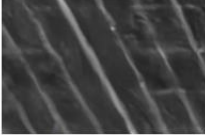
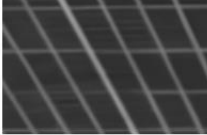
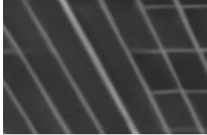
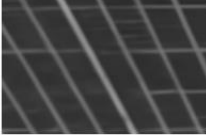
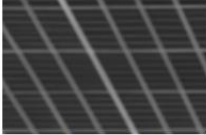

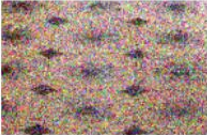





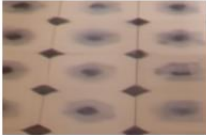





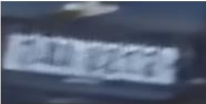





- Depth-wise convolution을 사용하여 인접한 pixel의 위치 정보를 공유할 수 있음

- ※ MDTA와는 다르게 local context feature를 중점적으로 강화하는데 목적



# Results

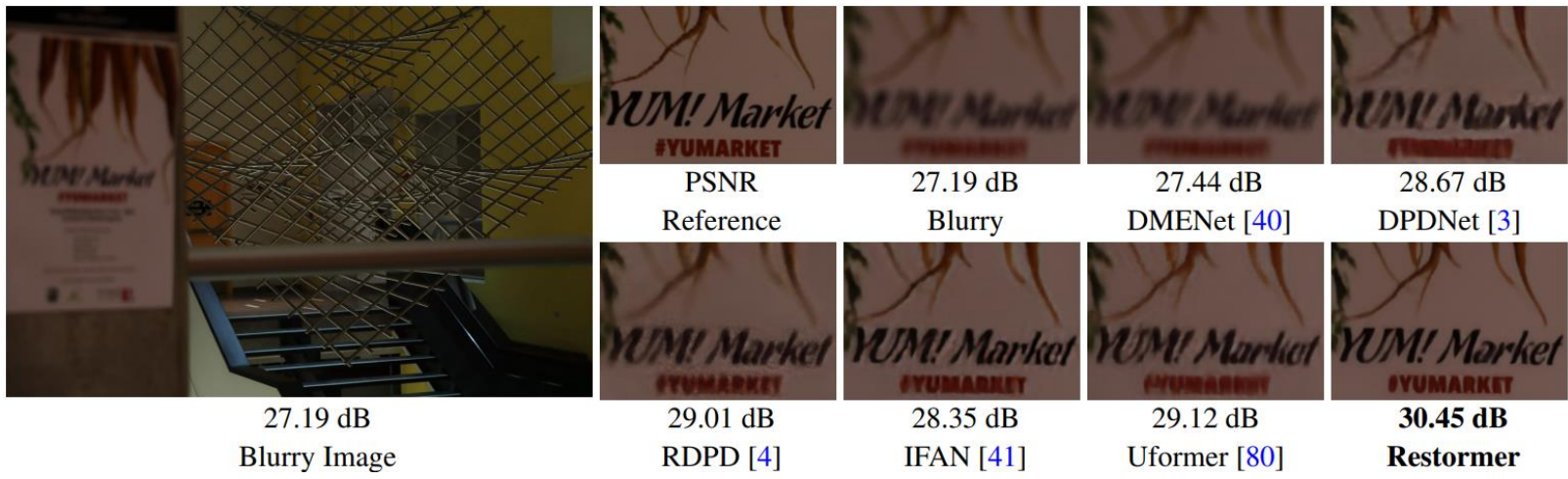
- Qualitative comparison

Denoising								
	Noisy Image	14.92 dB Noisy	PSNR Reference	27.61 dB DnCNN [101]	31.83 dB DRUNet [99]	30.12 dB DeamNet [63]	31.74 dB SwinIR [44]	<b>32.83 dB</b> <b>Restormer</b>
Denoising								
	Noisy Image	14.81 dB Noisy	PSNR Reference	33.83 dB FFDNet [103]	35.09 dB DRUNet [99]	34.86 dB IPT [13]	35.20 dB SwinIR [44]	<b>35.63 dB</b> <b>Restormer</b>
Motion Deblurring								
	19.45 dB Blurry Image		PSNR Reference	19.45 dB Blurry	23.85 dB Gao <i>et al.</i> [23]	23.56 dB DBGAN [100]	23.86 dB MTRNN [58]	
								
			24.85 dB DMPHN [94]	25.20 dB Suin <i>et al.</i> [71]	25.67 dB MPRNet [93]	24.33 dB MIMO-UNet+ [14]	<b>26.96 dB</b> <b>Restormer</b>	

# Results

- Qualitative comparison

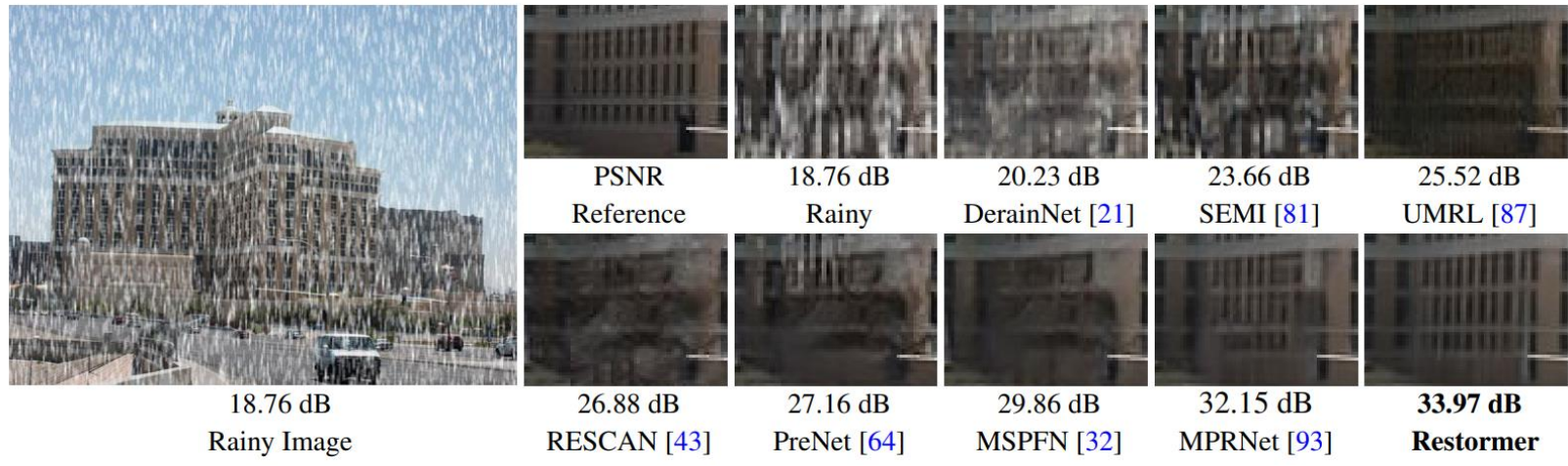
Defocus  
deblurring



27.19 dB  
Blurry Image

PSNR Reference	27.19 dB Blurry	27.44 dB DMENet [40]	28.67 dB DPDNet [3]
29.01 dB RDPD [4]	28.35 dB IFAN [41]	29.12 dB Uformer [80]	<b>30.45 dB</b> Restormer

Deraining

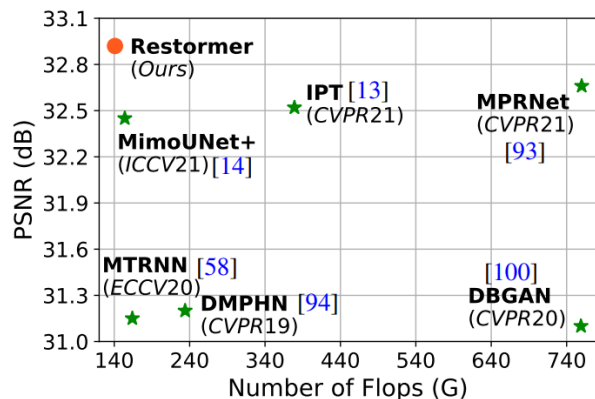


18.76 dB  
Rainy Image

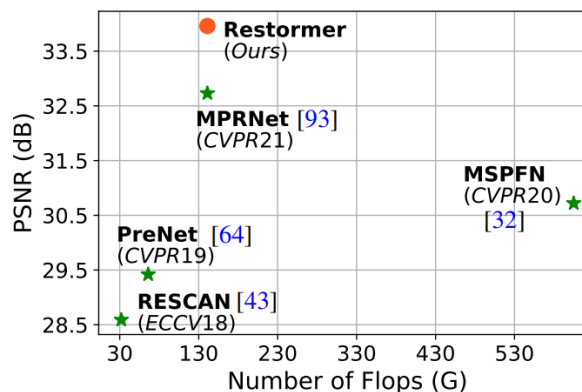
PSNR Reference	18.76 dB Rainy	20.23 dB DerainNet [21]	23.66 dB SEMI [81]	25.52 dB UMRL [87]
26.88 dB RESCAN [43]	27.16 dB PreNet [64]	29.86 dB MSPFN [32]	32.15 dB MPRNet [93]	<b>33.97 dB</b> Restormer

# Results

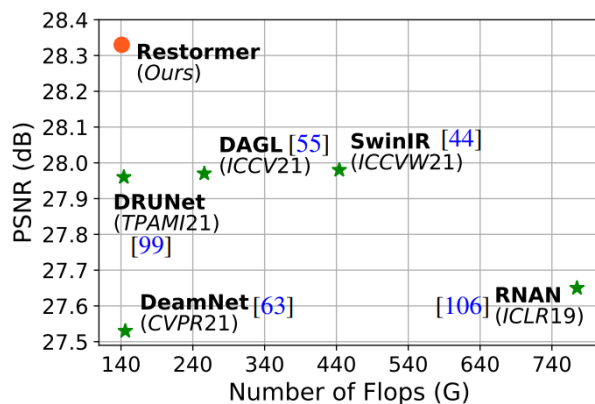
- Quantitative comparison



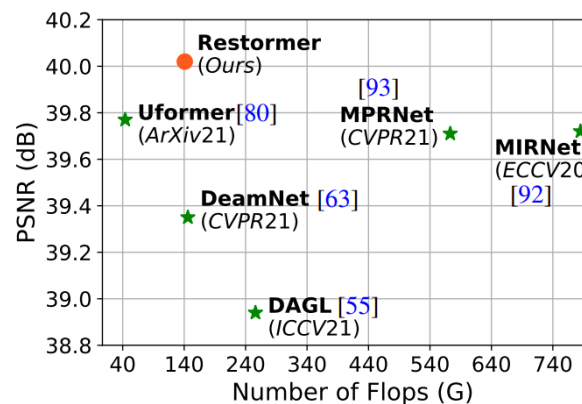
(a) Deblurring (Tab. 2)



(b) Deraining (Tab. 1)



(c) Gaussian Denoising (Tab. 4)



(d) Real Denoising (Tab. 6)

# Conclusion

- 고해상도 이미지를 처리하기에 효율적인 Restormer architecture 제안

- Transformer block의 구성요소 변형

- Multi-Dconv head Transposed Attention (MDTA) block제안

- ※ 기존 transformer model의 계산 복잡도를 완화하고 local context 정보를 효과적으로 다룸

Table 2. Computational comparison of Transformer-based image restoration models.

	Params (M)	FLOPs (B)	Time (s) 256×256 patch	PSNR (Denoising; Table 5) Urban100, $\sigma = 50$
IPT [8]	115.3	379	3.35	29.71
SwinIR [22]	11.50	444	1.80	29.82
Restormer	26.11	141	0.11	30.02

- Gated-Dconv Feed-forward Network (GDFN) 제안

- ※ Gate mechanism을 도입하여 feature transformation을 더 잘 control 할 수 있음

- ※ 마찬가지로 local context 정보를 효과적으로 다룸

- Transformer 구조에 CNN의 장점을 추가

- MDTA, GDFN 모두 depth-wise convolution을 사용하여 local context encoding 가능

## Simple Baselines for Image Restoration

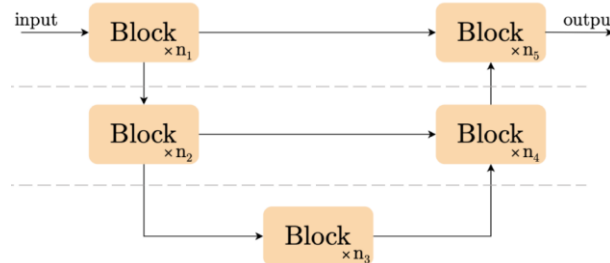


# Introduction

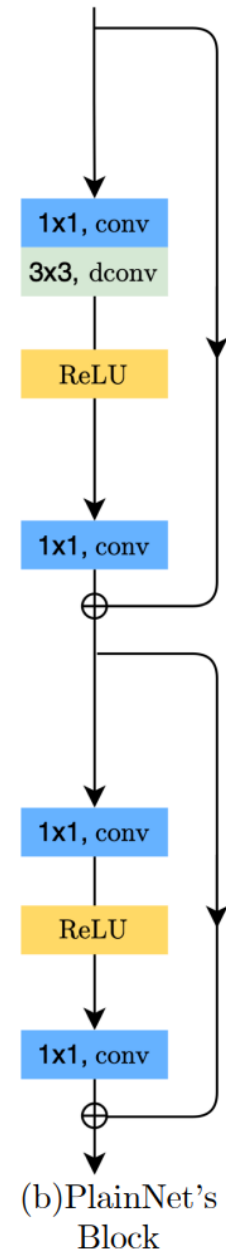
- Image restoration을 위한 딥러닝 방식이 발전을 이룸
  - 하지만 computational complexity가 증가하는 문제
- 기존 SOTA 방식의 성능을 능가하고 계산적으로 효율적인 baseline 제시
  - Complexity 관점에서 최적의 intra 또는 inter-block 구조를 고려함
  - SOTA 방식을 분해하여 필수 구성 요소만을 추출한 baseline 구축
    - 기존 SOTA 방식을 능가하고, computational cost 낮음
  - Baseline에서 Nonlinear Activation Free Network (NAFNet) 제안
    - Non-linear activation function을 제거하거나 대체하여 baseline을 더욱 단순화
    - Non-linear activation function이 image restoration 분야에 필요성이 적다는 것을 보여주는 첫번째 연구

# Method: Build a Simple Baseline

- Architecture
  - Inter-block complexity를 줄이기 위해 고전적인 U-Net 선정
- Plain block
  - Baseline을 구축 하기 전 block 내부를 최대한 단순하게 구성
    - Convolution, ReLU로 이루어짐
  - Transformer block을 사용하지 않음
    - Transformer는 computer vision에서 좋은 성능을 보이지만 일부 그렇지 않음
    - Self-attention보다 간단한 depth-wise convolution 사용
- Normalization
  - Layer normalization 사용
    - Transformer의 발전 과 함께 layer norm이 많은 SOTA방식에서 사용됨



(c) UNet Architecture(ours)



(b) PlainNet's Block

# Method: Build a Simple Baseline

- Activation

- GELU (Gaussian Error Linear Unit) 사용

- 기존 SOTA 방법에서는 ReLU activation을 GELU로 대체하는 경향이 있음

- Attention

- Self-attention 기법의 계산 복잡도는 feature map 크기에 quadratic하게 증가

- Spatial attention을 channel attention으로 변경하여 사용

- Channel attention으로 계산 효율성을 충족시키고 global 정보를 집계가능

- 여러 연구에서 image restoration 분야에 channel attention의 효과가 검증됨

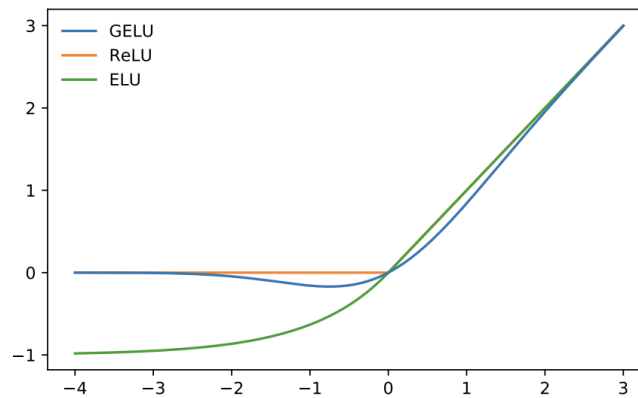
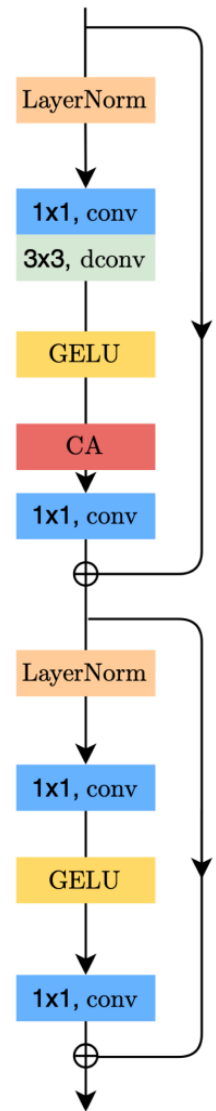


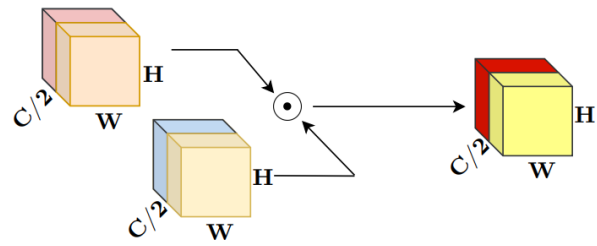
Figure 1: The GELU ( $\mu = 0, \sigma = 1$ ), ReLU, and ELU ( $\alpha = 1$ ).



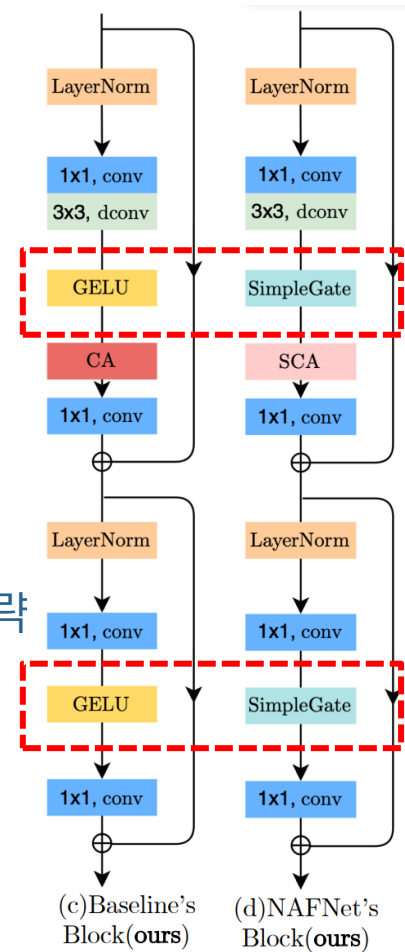
(c) Baseline's Block(ours)

# Method: NAFNet

- Nonlinear Activation Free Network (NAFNet) 제안
  - Baseline의 단순성을 유지하면서 성능을 더 향상
- Gated Linear Units (GLU)
  - GLU는 일부 SOTA image restoration 모델에서 성능을 올리기 위해 사용
    - Activation으로 사용한 GELU와 형태 유사
      - ※  $\text{Gate}(X, f, g, \sigma) = f(X) \odot \sigma(g(X))$
      - ※  $\text{GELU}(x) = x\Phi(x)$
    - $f, g$  를 identity function,  $\sigma(\cdot)$ 를  $\Phi(\cdot)$ 라고 할 수 있음
  - $f(X) \odot g(X)$  자체가 비 선형적이기 때문에 non-linear activation  $\sigma(\cdot)$  생략
    - 따라서 GLU를 기반으로 activation이 제거된 simple gate로 대체
      - ※  $\text{SimpleGate}(X, Y) = X \odot Y$
      - ※ X, Y는 각각 input feature를 2등분 한 feature



(c) Simple Gate



# Method: NAFNet

- Simplified Channel Attention (SCA)

- Baseline에서 사용하는 channel attention을 간단화

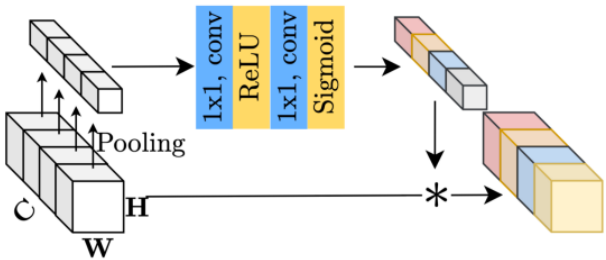
- Global 정보 집계와 channel 정보 상호작용 기능을 유지하며 간소화

- ⊛  $CA(X) = X * \sigma(W_2 \max(0, W_1 pool(X)))$

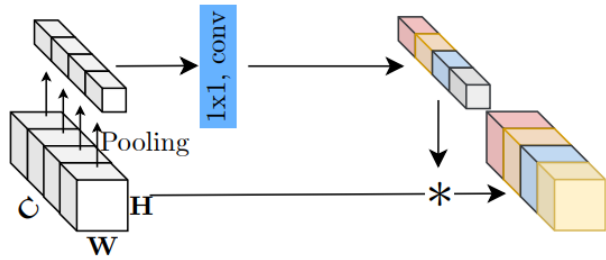
- ⊛  $CA(X) = X * \Psi(X)$

- $1 \times 1$  Conv와 activation layer를  $1 \times 1$  Conv 하나로 줄임

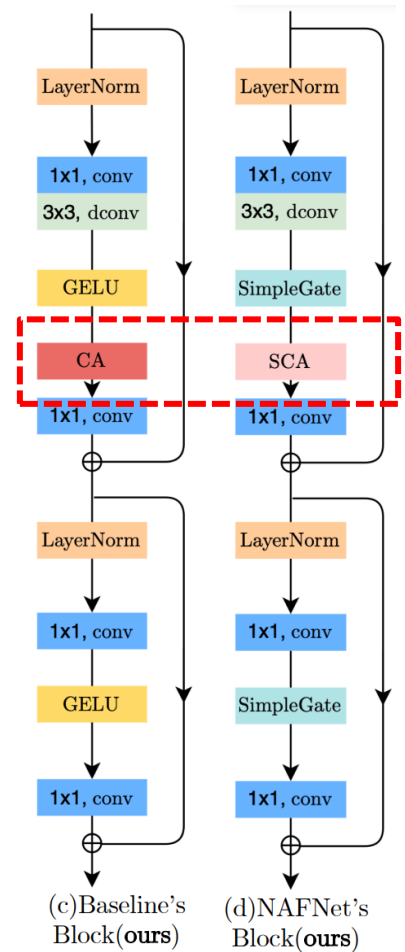
- ⊛  $SCA(X) = X * Wpool(X)$



(a) Channel Attention



(b) Simplified Channel Attention



(c) Baseline's Block(ours)

(d) NAFNet's Block(ours)

# Results

- Ablation

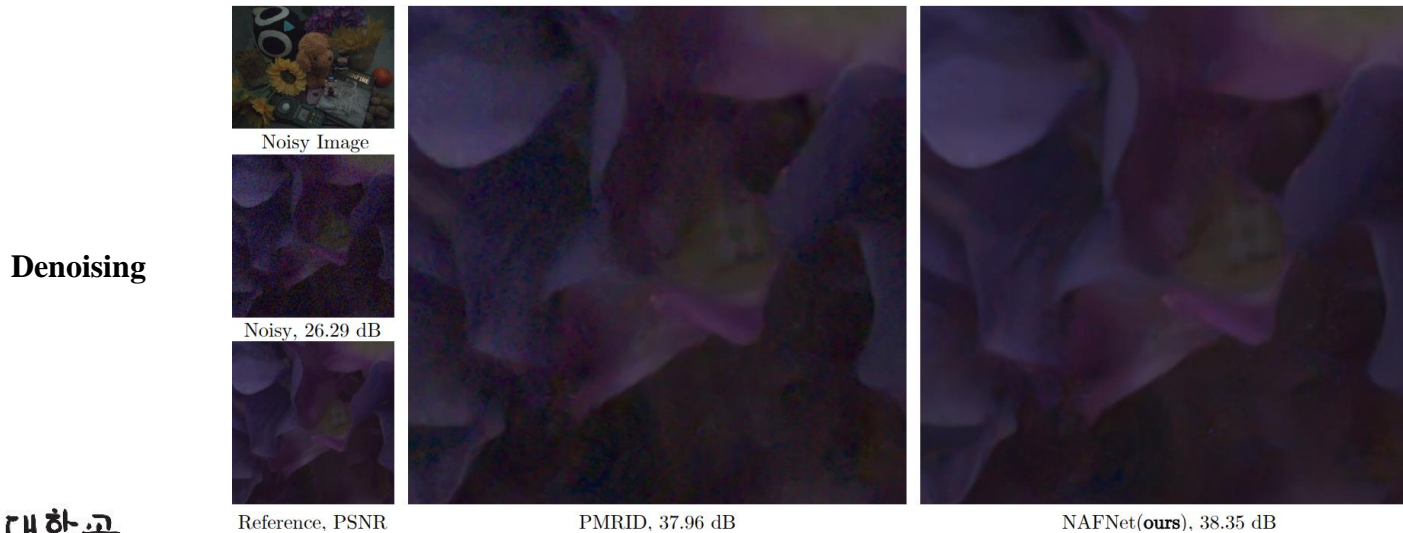
- PlainNet을 기반으로 GELU 및 channel attention (CA)을 적용한 것을 baseline으로 둠
- Baseline을 기반으로 simple channel attention (SCA)과 simple gate (SG)의 유효성을 검증

	lr	LN	ReLU→GELU	CA	SIDD		GoPro	
					PSNR	SSIM	PSNR	SSIM
PlainNet	$1e^{-4}$				39.29	0.956	28.51	0.907
PlainNet*	$1e^{-3}$				-	-	-	-
	$1e^{-3}$	✓			39.73	0.959	31.90	0.952
	$1e^{-3}$	✓	✓		39.71	0.958	32.11	0.954
Baseline	$1e^{-3}$	✓	✓	✓	39.85	0.959	32.35	0.956

	GELU→SG	CA→SCA	SIDD		GoPro		speedup
			PSNR	SSIM	PSNR	SSIM	
Baseline			39.85	0.959	32.35	0.956	1.00×
	✓		39.93	0.960	32.76	0.960	0.98×
		✓	39.95	0.960	32.54	0.958	1.11×
NAFNet	✓	✓	39.96	0.960	32.85	0.960	1.09×

# Results

- Qualitative comparison



# Results

- Quantitative comparison

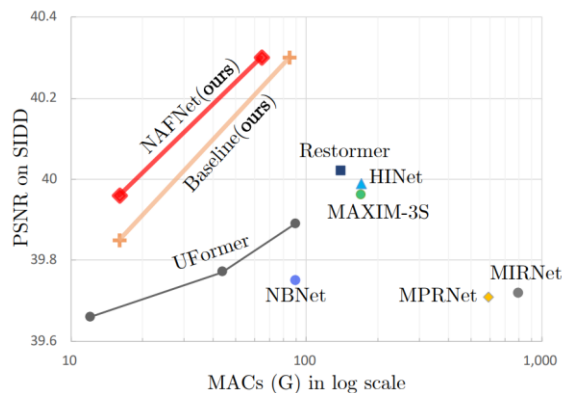
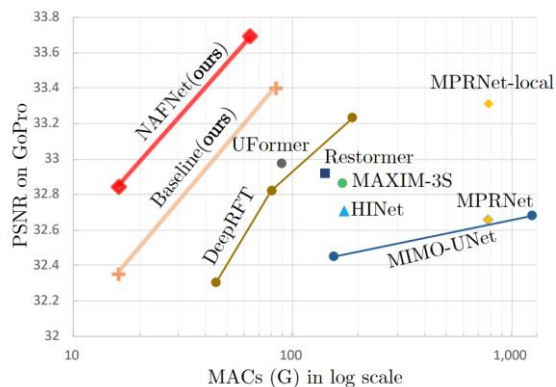


Table 6: Image Denoising Results on SIDD[1]

Method	MPRNet [37]	MIRNet [40]	NBNet [6]	UFormer [36]	MAXIM [32]	HINet [5]	Restormer [39]	Baseline <b>ours</b>	NAFNet <b>ours</b>
PSNR	39.71	39.72	39.75	39.89	39.96	39.99	40.02	40.30	40.30
SSIM	0.958	0.959	0.959	0.960	0.960	0.958	0.960	0.962	0.962
MACs(G)	588	786	88.8	89.5	169.5	170.7	140	84	65

Table 7: Image Deblurring Results on GoPro[26]

Method	MIMO-UNet [7]	HINet [5]	MAXIM [32]	Restormer [39]	UFormer [36]	DeepRFT [25]	MPRNet -local[8]	Baseline <b>ours</b>	NAFNet <b>ours</b>
PSNR	32.68	32.71	32.86	32.92	32.97	33.23	33.31	33.40	33.69
SSIM	0.959	0.959	0.961	0.961	0.967	0.963	0.964	0.965	0.967
MACs(G)	1235	170.7	169.5	140	89.5	187	778.2	84	65

Table 8: Raw image denoising results on 4Scenes[35]

Method	PSNR	SSIM	MACs(G)
PMRID[35]	39.76	0.975	1.2
NAFNet(ours)	40.05	0.977	1.1

Table 9: Image deblurring results on REDS-val-300[27]

Method	PSNR	SSIM	MACs(G)
MPRNet[37]	28.79	0.811	776.7
HINet[5]	28.83	0.862	170.7
MAXIM[32]	28.93	0.865	169.5
NAFNet(ours)	29.09	0.867	65



# Conclusion

- SOTA 방법을 분해하여 필요한 구성요소만으로 baseline 구축
  - Image denoising, deblurring에서 SOTA 달성
- Baseline을 기반으로 nonlinear activation function을 대체하거나 제거한 NAFNet 제안
  - 더 단순화 되었지만 성능은 baseline보다 좋거나 같음
- 제안한 baseline으로 image restoration 분야의 idea 검증에 도움을 줌
  - 간단하고 효율적인 baseline 제시
- 미래의 computer vision 모델 설계에 영향
  - Image restoration 분야에서 SOTA 성능을 달성하는데 non-linear activation의 불 필요성을 보임

감사합니다