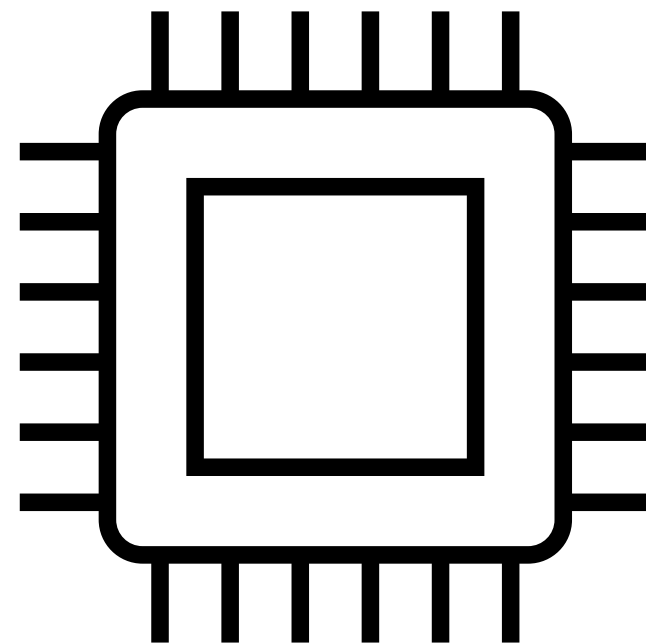


# Image Restoration (이미지 복원)

2025.07.09.

Copyright©2025 by 고재균



# Image Restoration [1]

- Image Restoration

- 이미지 복원은 손상된 이미지를 알고리즘을 통해 다시 원래 품질로 추정하는 Task를 의미
- 입력과 출력 모두 이미지
- 따라서 이미지 분류 및 객체 탐지와는 다르게 Low-Level Vision Task로 분류



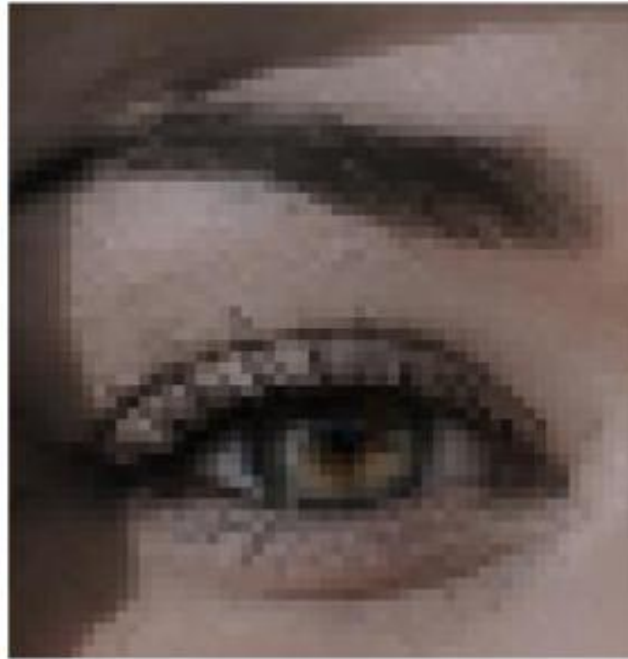
# Image Restoration [2]

- Image Degradation

- Image의 화질을 떨어뜨리는 현상
- Image Degradation에는 다음과 같이 여러 다양한 종류가 존재



Noise



Low-resolution



Blur

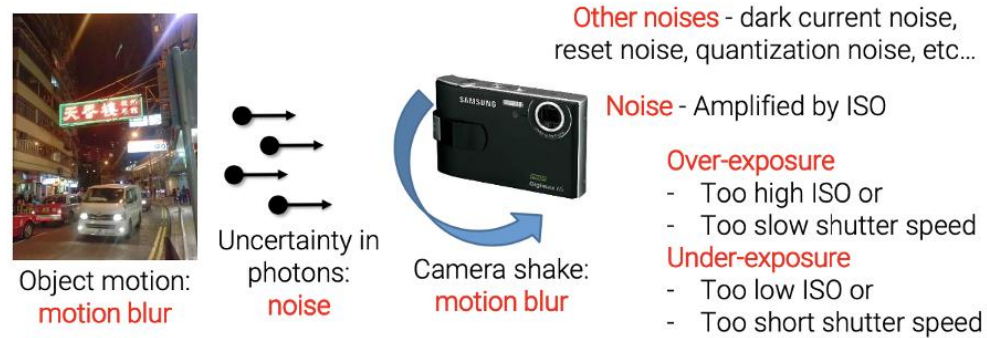
# Image Restoration [3]

## • Image Degradation

- 만약 빛이 적은 환경에서 사진을 촬영한다면 이미지 상에 알 수 없는 모양이 형성
- 이러한 것을 Noise라고 하며, 이는 어두운 환경에서 높은 ISO로 촬영할 때 주로 발생
- 우측의 이미지와 같이 Motion Blur 현상도 발생 가능
  - ▶ 보통 motion blur는 카메라가 흔들려서 생기는 현상
  - ▶ 우리가 사진을 찍고자 할 때는 충분한 Exposure Time이 필요
  - ▶ Exposure time이 길어질수록 손 떨림에 의해 카메라가 흔들릴 확률도 높아짐
  - ▶ 빛이 적은 환경에서도 Exposure Time이 길어야 하는데, 이때도 카메라가 흔들릴 확률이 높아짐
  - ▶ 이러한 환경에 Motion Blur가 발생
- Defocus Blur, Fog, Low-Resolution, Compression Artifact, Under or Over-exposure 등 다양한 Degradation이 존재
- 이러한 현상들은 사람의 눈으로 보았을 때 이미지의 퀄리티를 떨어뜨리고, Object Tracking이나 Image Segmentation 등의 시각 인식 Algorithm (High-Level Vision)의 성능을 저하

# Image Restoration [4]

## • Image Degradation Process

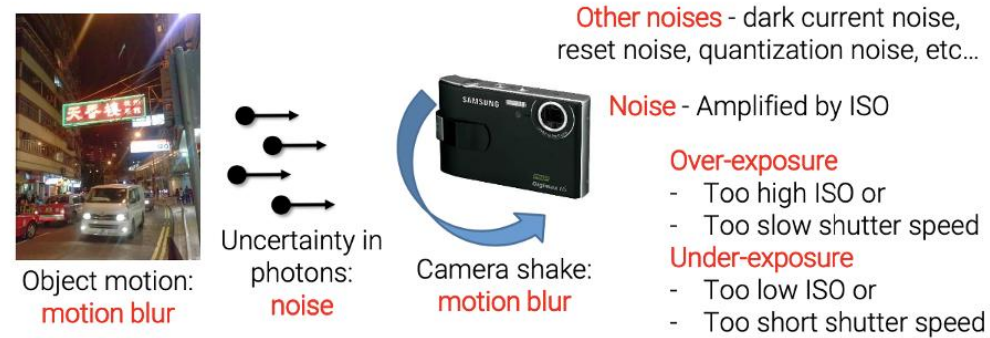


### ■ 사진을 촬영할 때 발생하는 Image Degradation의 과정

- ▶ Image degradation이 발생하기 쉬운 어두운 시간대에 밖에 나가서 사진을 찍은 것
- ▶ 길가에는 움직이는 차가 있는데 이 차의 움직임 때문에 Motion Blur가 발생
- ▶ 카메라는 Exposure Time 동안 카메라에 들어오는 광자의 양을 측정해주는 도구
- ▶ 그러나 광자는 카메라에 규칙적으로 들어오지 않아서 Noise가 발생
- ▶ 그리고 Exposure Time 동안에 손 떨림에 의해서 카메라가 흔들림

# Image Restoration [5]

## • Image Degradation Process

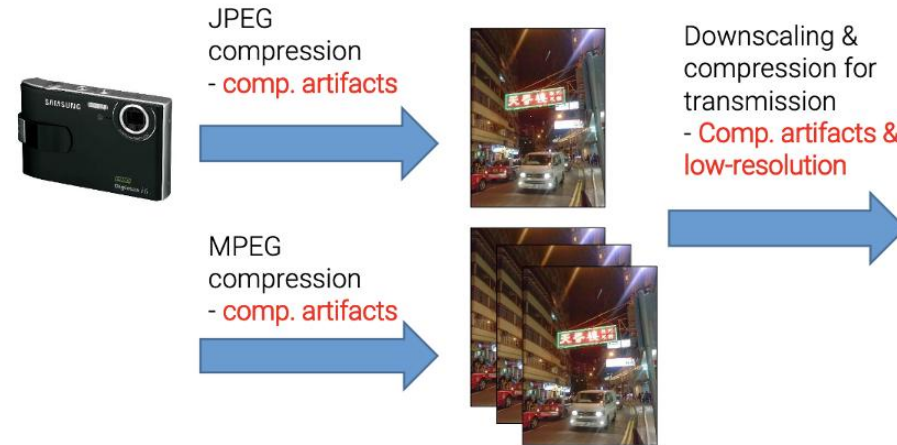


### ■ 사진을 촬영할 때 발생하는 Image Degradation의 과정

- ▶ 그래서 촬영된 사진을 보면 Motion Blur가 발생할 가능성 증가
- ▶ 더불어 추가적인 Noise들이 발생하게 되고, 이러한 모든 Noise들은 ISO 설정에 따라 증폭
- ▶ ISO를 높게 설정한다면 촬영된 사진에서 모든 Noise들이 증폭

# Image Restoration [6]

- Image Degradation Process



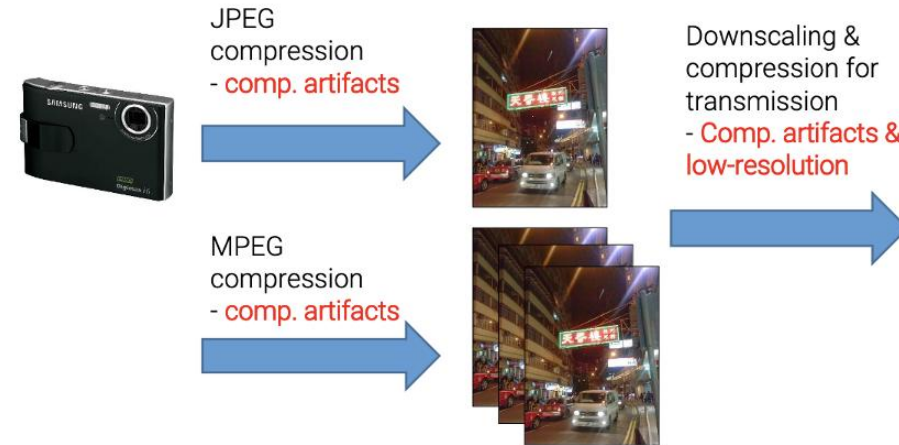
- 사진을 촬영할 때 발생하는 Image Degradation의 과정

- ▶ 마지막으로 사진을 촬영하고, 해당 사진은 ISP (Image Signal Processing) Pipeline에 의해서 처리 (RAW → sRGB)
- ▶ 이미지는 압축이 되고 JPEG 형식으로 저장
- ▶ 압축 과정 동안 몇가지 정보들이 손실이 되고 Compression Artifact가 추가 (JPEG Compression)
- ▶ 만약 동영상을 촬영한다고 하면 MPEG 형식으로 저장되고 여기서도 압축 과정에서 Compression Artifact가 생성 (MPEG Compression)



# Image Restoration [7]

- Image Degradation Process



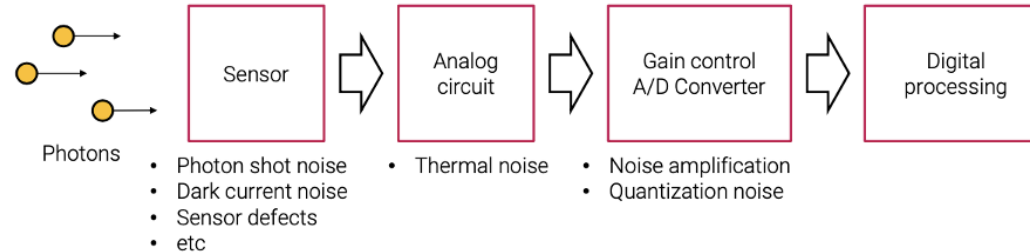
- 사진을 촬영할 때 발생하는 Image Degradation의 과정

- ▶ 카메라에 의해서 촬영되어 얻어진 사진은 예를 들어 4000×3000으로 초기에는 굉장히 큰 해상도를 가짐
- ▶ 이렇게 불필요하게 큰 경우에는 어딘 가에 업로드를 하거나 누군가에게 전송을 할 때 Downsampling과 Compression 과정이 되어 진행
- ▶ 이러한 경우에도 Compression Artifact가 생기게 되고 원래의 해상도를 잃어버림
- ▶ 그래서 여러 종류의 Image Degradation이 이 과정 중에 발생



# Image Restoration [8]

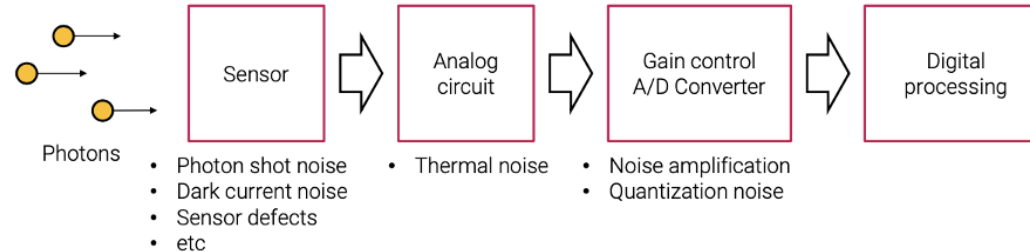
## • Camera Noise



- 다양한 종류의 Image Degradation이 있지만, 이 중에서 가장 대표적인 것은 카메라에 의해 생기는 Noise
- Image Noise는 종종 이미지를 포착할 때 생기게 되고 사진을 찍을 때 Noise의 종류 또한 다양함
- 카메라는 들어오는 광자의 양을 포착하는 도구이고 이렇게 모인 광자들은 센서 안에서 전하로 바뀜
- 다시 전하는 Analog Signal로 바뀌게 되고, 다시 Digital Signal로 바뀜 (A/D Converter)
- 촬영을 진행할 때 광자의 수는 시간에 따라 변동되고 무작위로 바뀌기 때문에 Noise가 발생
- 이러한 과정에서 발생하는 Noise를 Photon Shot Noise라고 칭함
- Dark Current Noise나 Sensor Defects와 같은 현상도 발생
- 센서에 도달하는 광자가 없을지라도 센서 자체는 약간의 전하를 가지고 있고 Dark Current Noise를 발생

# Image Restoration [9]

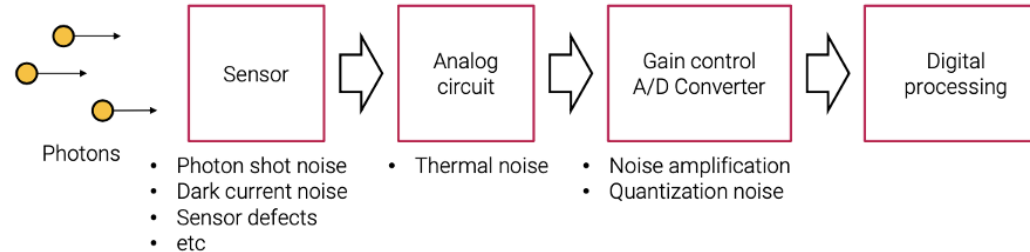
## • Camera Noise



- Thermal Noise는 Analog Circuit에서 발생하는 Noise
- 센서가 광자를 모은 후에는 광자들이 Analog Signal로 전환되고 이는 카메라에 있는 Analog Circuit에서 처리
- 만약 기온이 높다면 높은 온도가 Analog Circuit에 어떠한 영향을 주고 Analog Signal이 무작위로 변동
- 그러면 이러한 변동으로 인해서 Thermal Noise가 발생

# Image Restoration [10]

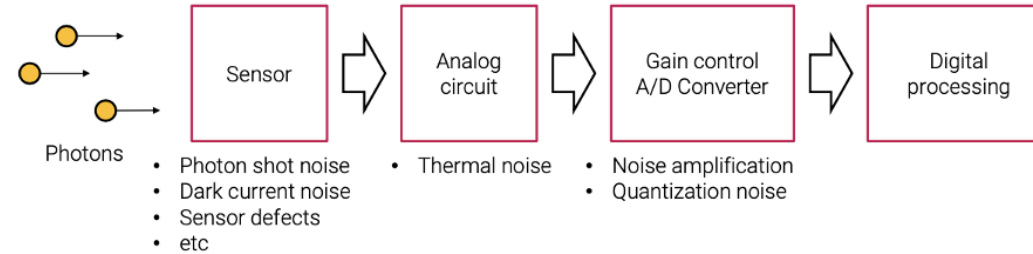
## • Camera Noise



- 그러면 이러한 모든 Noise들은 Gain Control A/D Converter에서 증폭
- Analog Signal이 이 Converter에 의해서 Digital Signal로 전환이 되는데, 이러한 전환 과정 도중에 Quantization Noise가 발생
- Quantization 전에는 3.1이었던 Signal이 후에는 3.0으로 0.1이 사라짐
- 원래의 Signal과 0.1이라는 차이가 발생하게 되고, 이 차이가 또한 Noise를 가지고 있다고 간주함
- 해당 Noise를 Quantization Noise라고 정의

# Image Restoration [11]

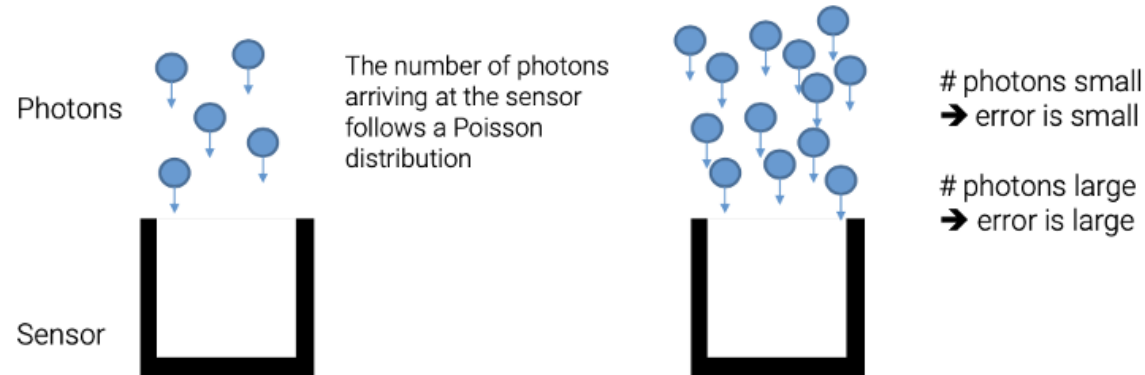
## • Camera Noise



- Pre-ADC Noise는 A/D Converter 이전의 모든 Noise를 말하고, Photosensor의 Signal이 증폭될 때 증폭
- ISO-Level은 Digital 카메라에서 입력 Signal이 얼마나 증폭되는지를 조절
- 이러한 Noise의 증폭은 빛이 적은 환경에서 특히 발생하는 문제
- 빛이 적은 환경에서는 높은 ISO 값을 사용

# Image Restoration [12]

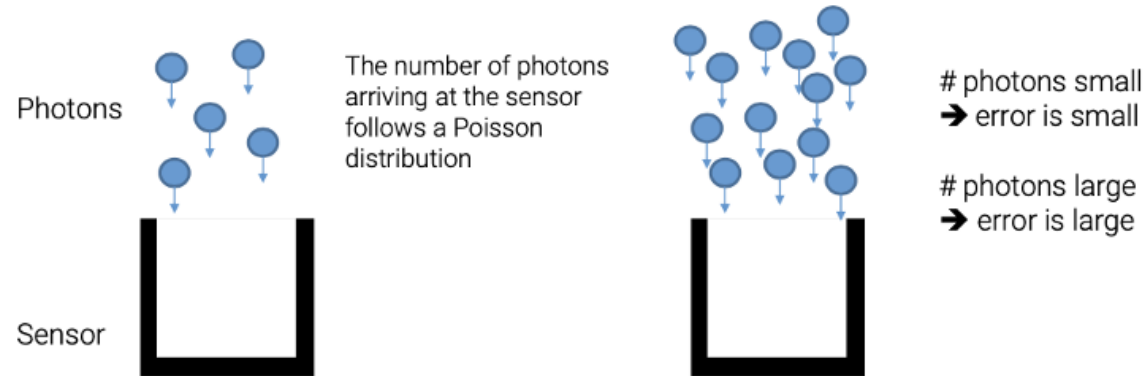
## • Shot Noise



- 사진을 촬영해서 이미지를 얻을 때까지 다양한 Noise들이 발생
- Digital Image에서 가장 대표적인 Noise는 이중에서도 Shot Noise
- 카메라에는 Sensor Plane이 존재하고, 이 Sensor Plane에는 굉장히 많은 Sensor들이 존재
- 여기서 각 센서들을 Photodiode라고 하고 Photodiode는 들어오는 광자들을 모음
- 이들의 Signal Strength는 센서를 때리는 광자들의 수에 비례함
- 센서를 때리는 광자들은 무작위이고, Shot noise 혹은 Photon Shot Noise는 무작위로 들어오는 광자들의 총 개수로 인하여 발생한 Noise

# Image Restoration [13]

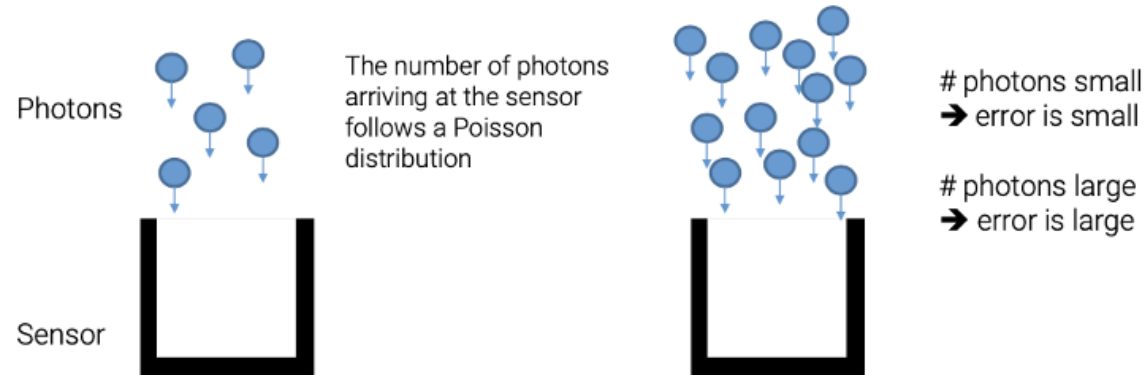
## • Shot Noise



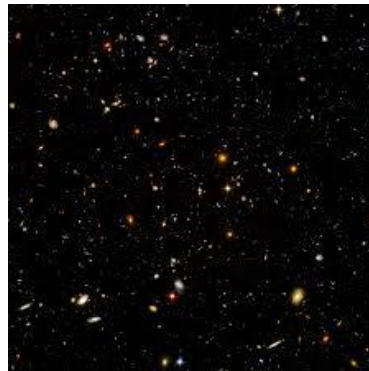
- Shot Noise는 Poisson Distribution을 따르고, 광자의 수가 많을 때는 Poisson Distribution이 Gaussian Distribution으로 근사 될 수 있음
- 이러한 것이 Image Denoising Method가 Noise를 Gaussian Noise로 가정하는 이유
- 결론적으로 광자는 Poisson Distribution을 따르는 Noise이고, Shot Noise는 센서 크기가 작아지고 Pixel 패킹이 촘촘해 질수록 더 심해지는 현상이 발생
- 이는 결국 빛이 적은 환경에서 사진을 촬영하는 것과 유사
- 센서의 크기가 작은 상태에서 동일한 Exposure Time 동안 광자를 받을 때 기존보다 적은 개수를 받게 됨

# Image Restoration [14]

- Shot Noise



- 센서의 크기가 작다는 것은 동일한 Exposure Time 동안 적은 광자를 받게 됨
- 이는 빛이 적은 환경에서 사진을 촬영하는 것과 같은 구조이고, 그래서 Noise는 더 심해지게 됨

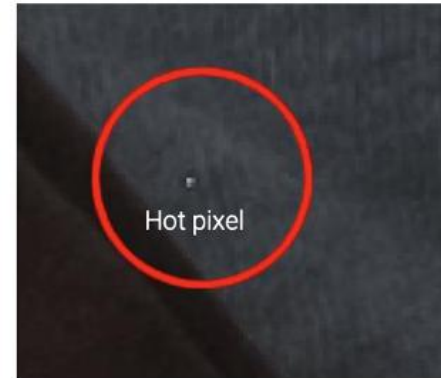


**Hubble Ultra-Deep Field**  
(About 11 Days of Exposure Time)



# Image Restoration [15]

- Salt-and-Pepper Noise (S&P Noise)



- Salt-and-Pepper Noise는 전형적인 카메라 Noise
- Replacement Noise라고도 불리고 이는 Pixel 값들의 일부가 무작위 값으로 대체되기 때문
- Salt Noise와 Pepper Noise가 합쳐진 형태로 Salt Noise는 무작위로 Pixel이 하얀색인 경우를 의미하고 Pepper Noise는 무작위로 Pixel이 검정색인 경우를 의미
- Analog circuit이나 A/D converter에 어떠한 문제가 생기게 되면 이러한 Noise가 발생

# Image Restoration [16]

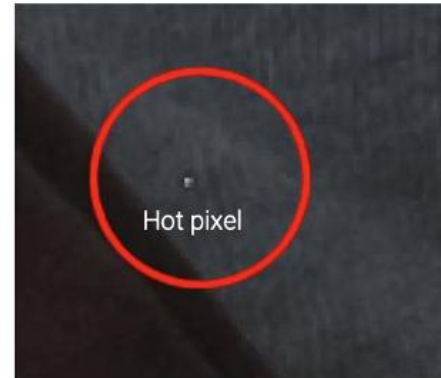
- Salt-and-Pepper Noise (S&P Noise)



- 카메라에는 Sensor Plane이 존재하고 많은 수의 Photodiode가 존재
- 어떠한 문제가 이 센서나 처리 과정에서 발생하게 되면 제대로 작동하지 않은 Photodiode들이 생기게 됨
- 이 Photodiode는 들어오는 광자에 대해서 덜 Sensitive해져서 제대로 동작하지 않음
- Dead Pixel이라는 것은 Photodiode가 제대로 동작하지 않은 것을 의미
- 만약 Dead Pixel이 존재하게 되면 드물게 검은색을 보이는 Pixel이 생김
- 반대로 Hot Pixel은 Photodiode가 들어오는 광자에 대해서 매우 Sensitive해서 해당하는 Pixel은 매우 밝은 색으로 보임

# Image Restoration [17]

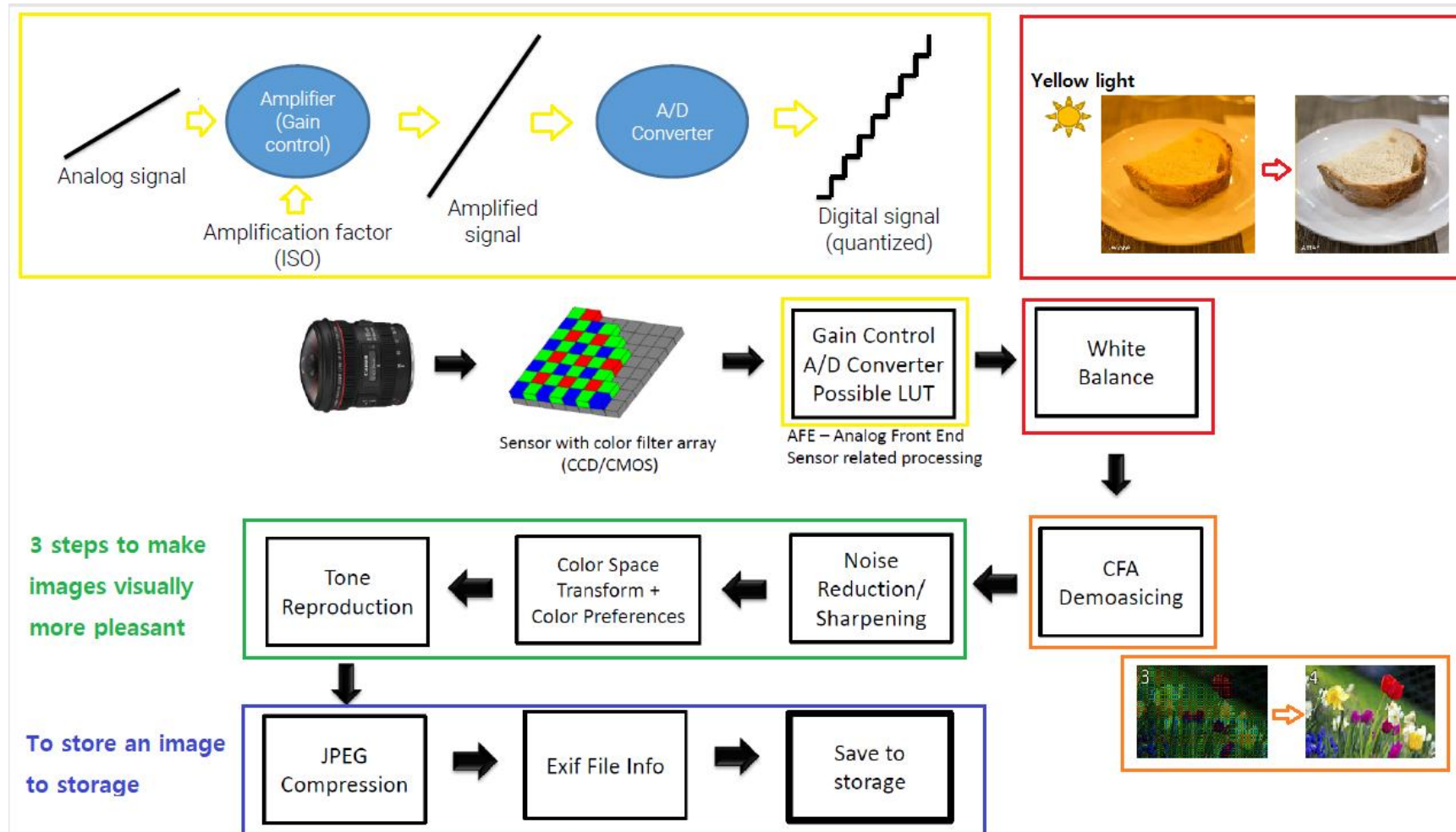
- Salt-and-Pepper Noise (S&P Noise)



- ISP 과정에서 카메라는 그 자체로 Dead Pixel과 Hot Pixel을 구분
- 그래서 이미지로부터 자체적으로 Dead Pixel과 Hot Pixel을 다뤄서 문제를 해결
- Raw Image나 ISP 과정이 없다면 우리가 촬영한 이미지로부터 Hot pixel과 Dead Pixel들을 발견할 수 있음

# Image Restoration [18]

- Camera Imaging Pipeline for JPEG



# Image Restoration [19]

- Noise Degradation 실습

Scikit-Image를 활용한 노이즈 생성

(Google Colab. 환경)

# Image Restoration [20]

- Quantitative Evaluation

- Peak Signal-to-Noise Ratio (PSNR)

$$PSNR = 10 \log \frac{s^2}{MSE}$$

- ▶ Pixel 단위의 오차 값을 계산

- ▶  $S \rightarrow$  이미지의 최댓값 (255 for 8-bit Image)

- ▶ 한계점

- ▷ PSNR은 intensity 의 값들을 위 식에 의해 종합하여 평가하는 방식이기 때문에, 실제로 사람이 봤을 때 느끼는 것과 다른 점수를 산출 할 때가 있음

- ▷ 분모에 MSE가 들어가 있기 때문에 정답과 유사하지 않은 Blurry한 이미지를 측정했을 때 높은 PSNR 점수가 산출 가능함

# Image Restoration [21]

- Quantitative Evaluation

- Structural Similarity (SSIM)

- ▶ PSNR 의 한계를 극복하기 위해, 개발된 Metric
    - ▶ 사람의 지각 능력과 Metric을 일치시키는 목적에 개발
    - ▶ 사람은 영상에서 구조 정보를 반영하여 영상을 바라보게 되는데, 영상이 얼마나 그 구조 정보를 변화시키지 않았는가를 살펴보는 Metric
    - ▶ 원본 이미지 (x)와 왜곡 이미지 (y)의 Luminance (l), Contrast (c), Structure (s)를 비교

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma$$

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$



# Image Restoration [22]

- Denoising 실습

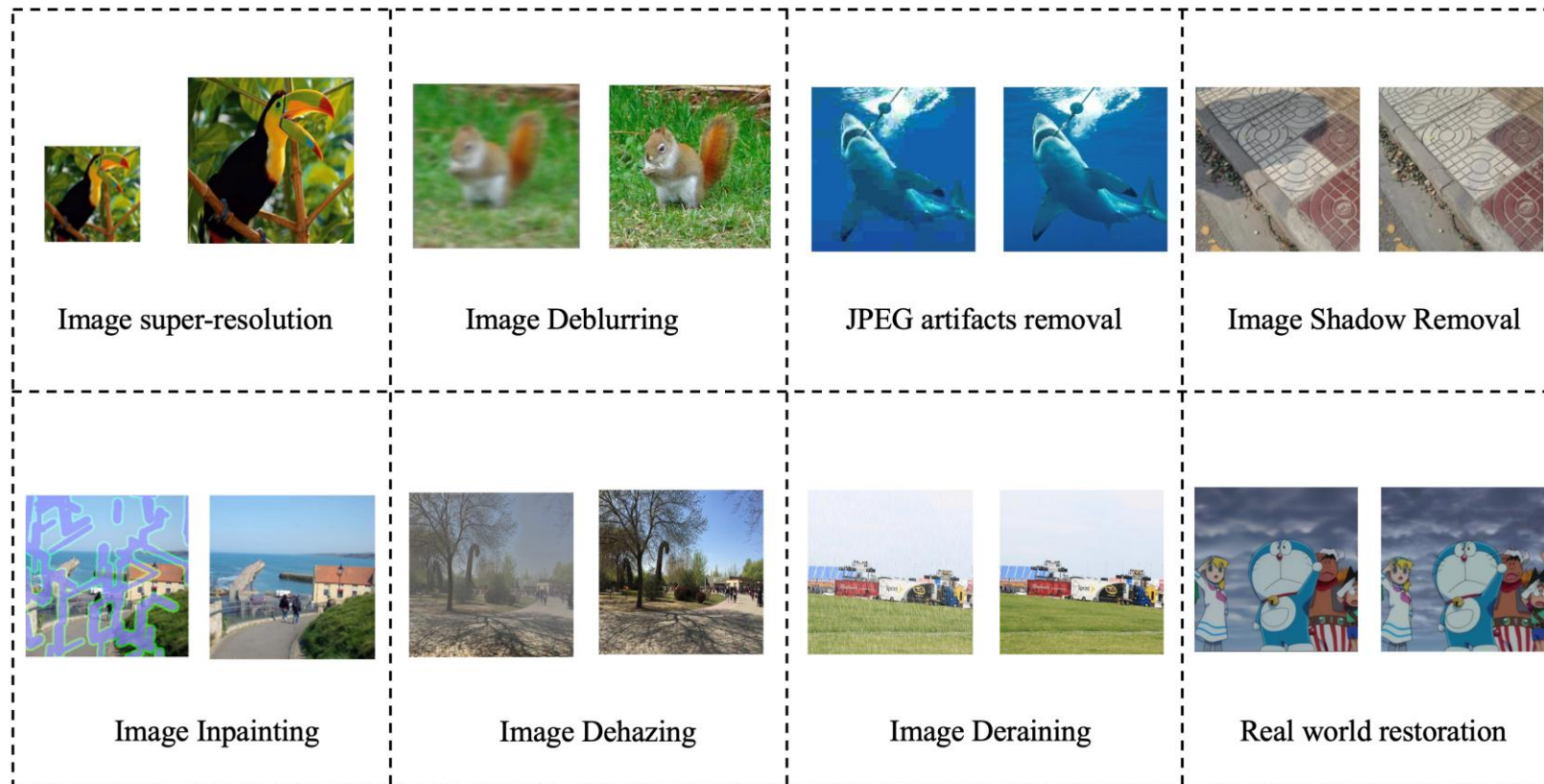
**Scikit-Image를 활용한 디노이징**

(Google Colab. 환경)

# Image Restoration [23]

- Image Restoration Task

- Image Denoising (CBSD / SIDD / ...)
- Super-Resolution (DIV2K / Urban100 / ...)
- Image Deblurring



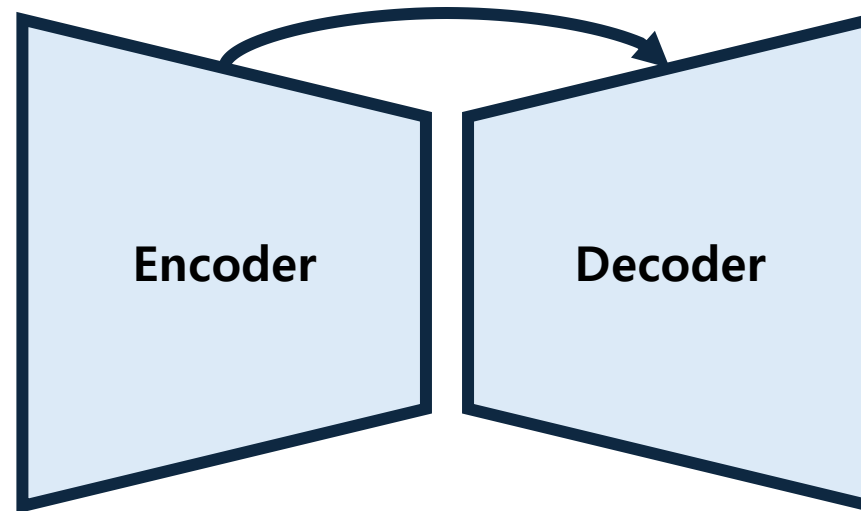
# Image Restoration [24]

- **Image Restoration Task**

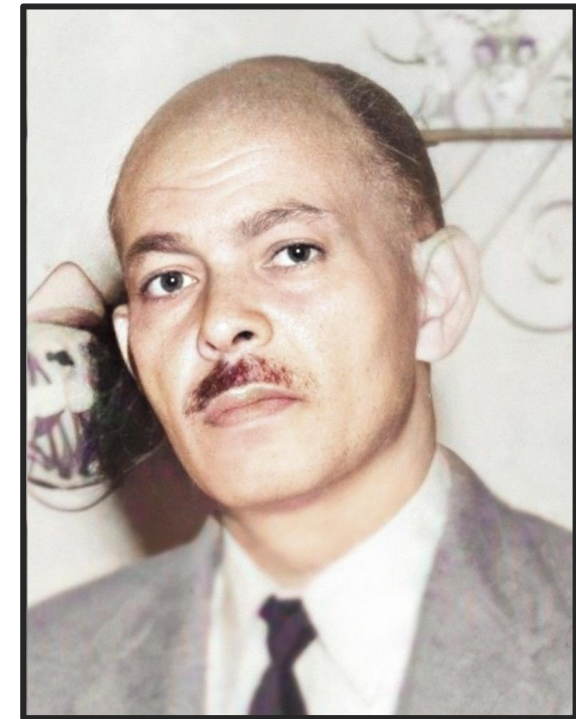
- Same as Regression Task!
- Train Neural Network to Predict (Estimate) Pixel Value!



**Input**  
(Degraded Image)



**Restoration Model**  
(Regression Model)



**Output**  
(Restored Image)

# Image Restoration [25]

- Image Restoration Model

- Autoencoder Architecture
  - ▶ Receptive Field 확장
  - ▶ LQ  $\rightarrow$  HQ 도메인 변환
  - ▶ Skip Connection (Encoder와 Decoder를 연결)
  - ▶ 노이즈 억제·정보 압축(Bottleneck 효과)
  - ▶ 멀티-스케일 특징 융합
  - ▶ 자기 지도 / 사전학습 용이
  - ▶ Residual 학습과의 궁합
  - ▶ 모듈화 및 확장성
  - ▶ 파라미터 및 메모리 효율
  - ▶ 가변 해상도 처리

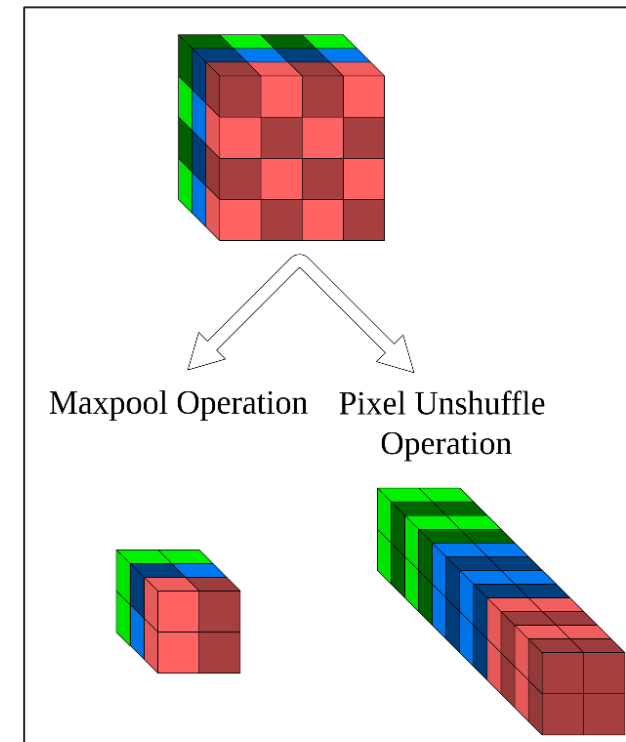
# Image Restoration [26]

## • Image Restoration Model

### ▪ Autoencoder Architecture

#### ▶ Encoder

- ▷ Convolutional Layer와 Downsampling을 통한 정보 추출 및 차원 압축을 진행
- ▷ Downsampling은 Feature Map의 Spatial 크기를 줄이는 과정 ( $H \times W \rightarrow H/2 \times W/2$ )
- ▷ Downsampling은 다음과 같은 기법을 통하여 진행
  - ◆ Learnable Convolutional Layer with Stride=2
  - ◆ Max / Average Pooling Layer
  - ◆ Pixel Unshuffle
    - ❖  $H \times W \times C \rightarrow H/2 \times W/2 \times 4C$



# Image Restoration [27]

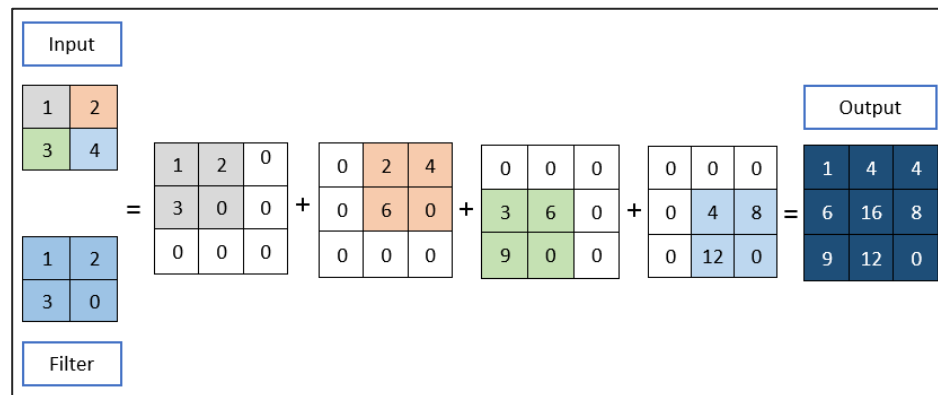
## • Image Restoration Model

### ▪ Autoencoder Architecture

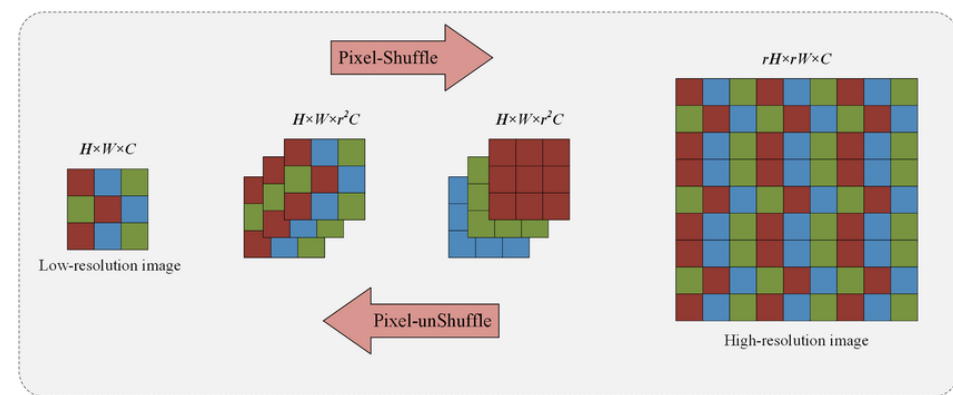
#### ▶ Decoder

- ▷ Convolutional Layer와 Upsampling을 통한 정보 추출 및 차원 복원을 진행
- ▷ Downsampling은 Feature Map의 Spatial 크기를 늘리는 과정 ( $H/2 \times W/2 \rightarrow H \times W$ )
- ▷ Downsampling은 다음과 같은 기법을 통하여 진행
  - ◆ Interpolation + Convolutional Layer
  - ◆ Transposed Convolutional Layer
  - ◆ Pixel Shuffle

$$\diamond H/2 \times W/2 \times C \rightarrow H \times W \times C/4$$



Transposed Conv.



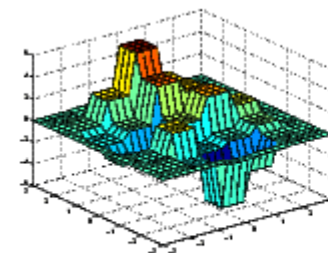
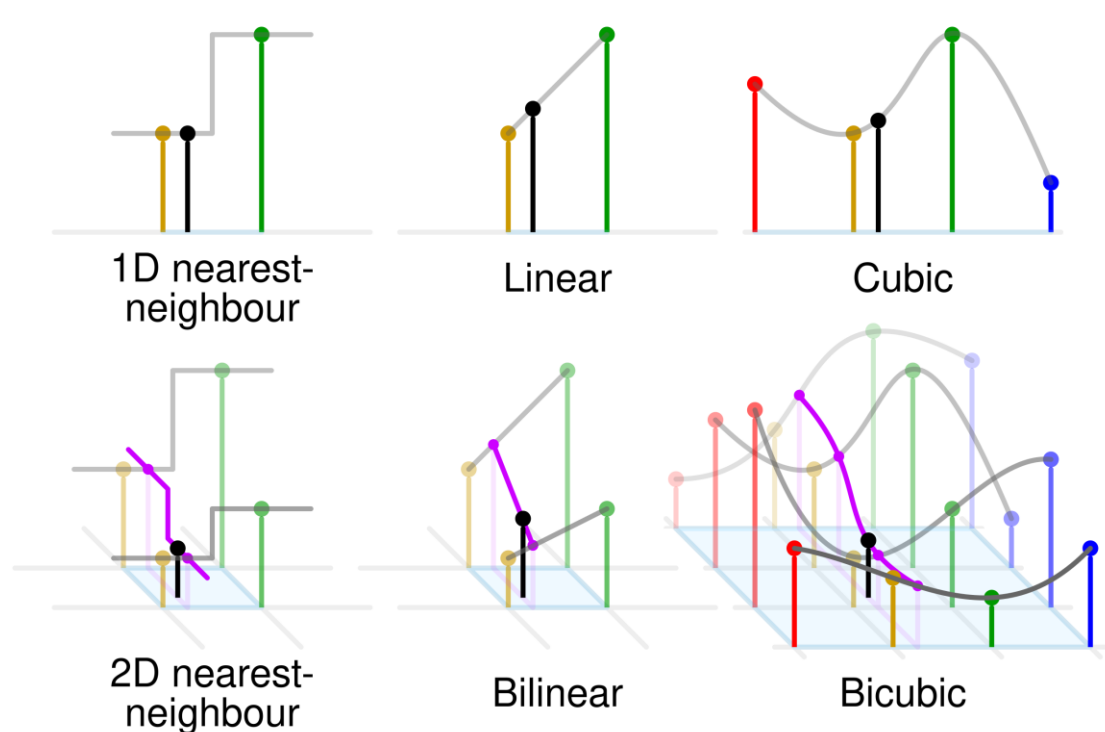
Pixel Shuffle

# Image Restoration [28]

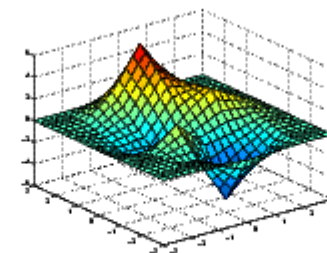
- Image Restoration Model

- Autoencoder Architecture

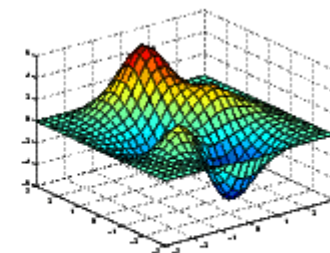
- Linear Interpolation



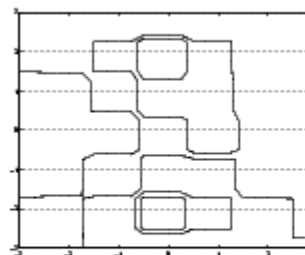
```
surf(xi,yi,zi1)  
% nearest
```



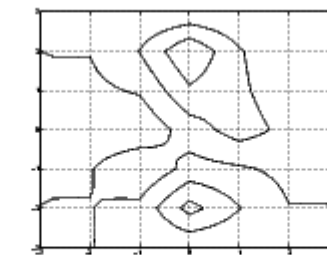
```
surf(xi,yi,zi2)  
% bilinear
```



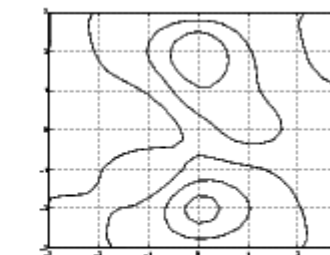
```
surf(xi,yi,zi3)  
% bicubic
```



```
contour(xi,yi,zi1)  
% nearest
```



```
contour(xi,yi,zi2)  
% bilinear
```



```
contour(xi,yi,zi3)  
% bicubic
```



# Image Restoration [29]

- **Image Restoration Model**

- Autoencoder Architecture

- ▶ Decoder

- ▷ Why use Linear Interpolation + Convolutional Layer Combination?
      - ▷ It reduces artifacts induced by Transposed Convolutional Layer!

<https://distill.pub/2016/deconv-checkerboard/>

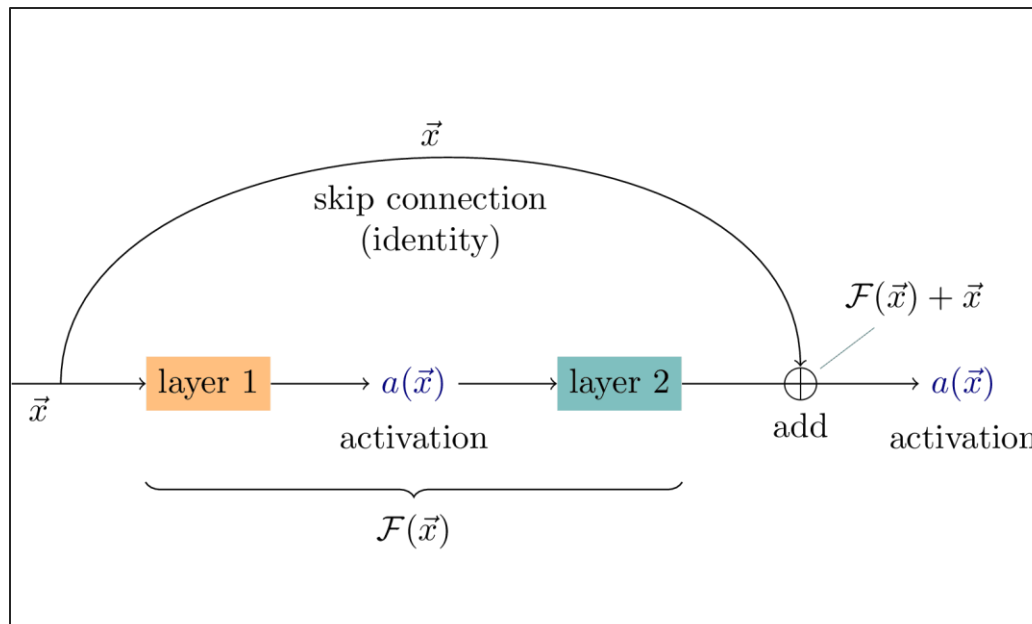
# Image Restoration [30]

## • Image Restoration Model

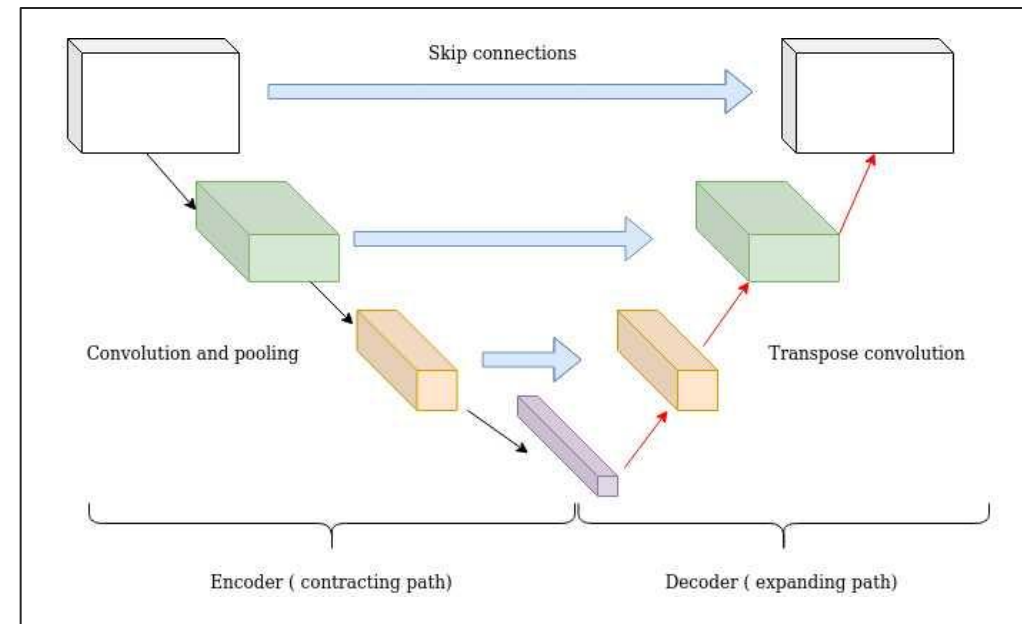
### ▪ Autoencoder Architecture

#### ► Skip Connection

- ▷ Connection between Encoder and Decoder!
- ▷ It passes Multiscale Features from Encoder to Decoder
- ▷ Helps to restore / reconstruct the output well



Residual Skip Connection



Autoencoder Skip Connection

# Image Restoration [31]

- Denoising 실습

DnCNN을 활용한 디노이징

(Google Colab. 환경)

# Image Restoration [32]

- Denoising 실습

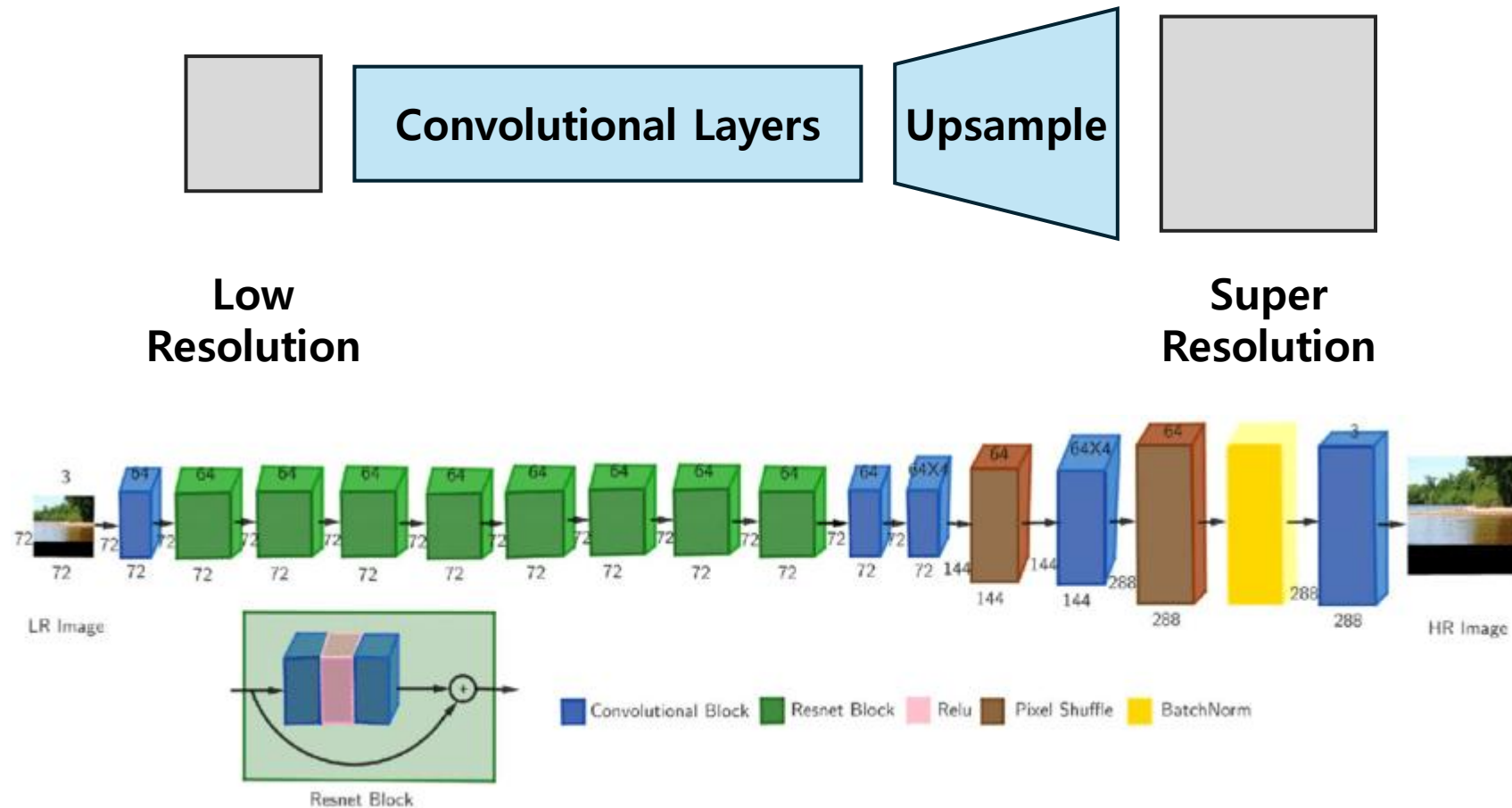
**Autoencoder를 활용한 디노이징**

(Google Colab. 환경)

# Image Restoration [33]

## • Image Restoration Model

- Super-Resolution Model Architecture (Upscaling)
  - Different from Autoencoder Architecture



# Image Restoration [34]

- Super-Resolution 실습

**Deep Learning 모델을 활용한 Super-Resolution 실습**

(Google Colab. 환경)