

# 딥러닝 기반을 이용한 COVID-19 Low Dose CT 영상 복원

## COVID-19 Low Dose CT Image Restoration Using Deep Learning Based

허지혜<sup>1)</sup>、이수빈<sup>2)</sup>、정예지<sup>2)</sup>、민지원<sup>3)</sup>、임현영<sup>4)</sup>、정보경<sup>4)</sup>、하지원<sup>3)</sup>

경상국립대학교 정보통계학과<sup>1)</sup>、경상국립대학교 수학과<sup>2)</sup>

、상문고등학교<sup>3)</sup>、물금고등학교<sup>4)</sup>

멘토: 임동훈<sup>5)</sup>、김규동<sup>6)</sup>

경상국립대학교 정보통계학과<sup>5)</sup>、러닝스폰즈<sup>6)</sup>

고화질의 전산화단층촬영영상(CT)를 통해 정확한 병명을 진단할 수 있다. 하지만 전산화단층촬영영상은 방사선량을 사람 몸에 투과하여 영상을 얻어내기 때문에 몸에 좋지 않다. 이와 같은 단점 때문에 전산화단층 촬영 시 방사선량을 줄이면서 영상 화질을 개선하기 위한 연구가 많이 수행되었다. 본 연구에서는 3가지의 잡음을 기본 CT 영상에 추가하여 저선량 CT 영상을 가공한 후, 저선량 CT 영상을 딥러닝을 기반으로 한 모형에 넣어 고해상도 CT 영상을 얻는다. COVID-19로 진단이 된 CT 영상을 가공하여 비교를 쓰고 이를 정량적 평가로 모델의 성능을 비교한다.

High-quality computed tomography (CT) images can be used to accurately diagnose the disease. However, computed tomography is not good for the body because the radiation dose is transmitted through the human body to obtain an image. Because of these shortcomings, many studies have been conducted to improve image quality while reducing radiation dose during computed tomography. In this study, low-dose CT images are processed by adding three types of noise to the basic CT images, and then high-resolution CT images are obtained by putting the low-dose CT images into a model based on deep learning. The CT image diagnosed with COVID-19 is processed, the comparison is written, and the performance of the model is compared by quantitative evaluation.

키워드: CT, Deep Learning, super-resolution convolutional neural network, image quality

### 1. INTRODUCTION

2020년 1월, 집단 폐렴 원인 병원체인 COVID-19(코로나 바이러스 감염증-19; 코로나 19)가 처음으로 나타났다. 전염증세가 심하여 전 세계로 확산되면서 확진자가 약 4000만명, 사망자가 약 100만명을 넘어섰다. 전 세계적으로 확진자와 사망자가 기하급수적으로 증가

함에 따라 의료진 수 및 의료시설 부족 문제가 나타났다. 또한 진단 방법의 효율성에 대한 문제도 나타났다. 이 두가지 문제를 모두 해결할 방법과 전염 확산 방지를 위해 조기 의심 환자에 대한 신속하고 정확한 진단방법의 필요성이 대두되고 있다. 천식, 독감 등 폐 질환을 진단하기 위한 대표적인 검사로는 흉부 X선과 흉부 전산화단층촬영(CT; Computed Tomography)가 있다. 흉부 X선 검사는 다른

검사들보다 비교적 저렴한 검사 비용과 단시간 촬영이라는 장점이 있다. COVID-19 또한 폐질환의 일환으로서 CT 영상으로 판독이 가능하다. 전염병인 COVID-19의 특성상 동선이 겹칠시 감염 여부에 대한 검사를 자주 하게 되는데 이 때 방사선 피폭에 대한 위험성이 존재한다. 우리는 저선량 COVID-19 CT 영상을 이용하여 영상 화질을 복원하여 고해상도 CT 영상으로 만드는 연구를 진행해볼 것이다.

저해상도 영상에서 고해상도 영상으로 변환 시키는 문제를 우리는 SR(Super Resolution)이라고 부른다. SR의 접근 방법은 매우 다양하다. 우리는 최근 컴퓨터 비전 분야에서 주목하고 있는 심층신경망(DNN; Deep neural network)기술을 저선량 CT 영상에 적용하여 고해상도 CT 영상으로 복원하는 모형을 비교하는 연구를 진행해볼 것이다. 비교할 모형은 다음과 같다. 우선 저선량 CT 영상을 입력으로 받아 고해상도 이미지로 출력하는 SRCNN(Super Resolution Using Deep Convolutional Networks)을 사용한다. SRCNN에 사용된 합성곱신경망(CNN; Convolutional neural network)은 저해상도, 고해상도 영상 간 매핑을 직접 학습한다. 이때 매핑이란, 저해상도 영상을 입력으로 받아 고해상도 영상을 출력하는 Deep CNN으로 표현한다. 영상의 공간 정보를 유지한 상태로 모형 학습이 가능하기 때문에 영상 내 세부정보를 유지할 수 있는 장점을 갖는다. CNN은 딥러닝 방법 중 하나로 이미지 처리에 적합한 특성을 갖는다. CNN 구조 내에서 합성곱 계층은 입력데이터의 특징 지도(feature map)를 추출할 수 있으며 이러한 합성곱 계층을 여러 계층으로 연결하여 각 합성곱 계층을 지날 때마다 저수준의 특징 지도가 점차 고수준의 특징 지도로 만들어진다. 그리고 최종적으로 전결합층(fully connected layer)을 통해 결과를 학습함으로써 영상화 모형 구현이 가능하다. 이러한 SRCNN에서는 연산량이 많다는 단점이 있다. 이러한 단점을 FSRCNN (Fast Super-Resolution Convolutional Neural Network)[1], ESPCN(Efficient Sub-pixel Convolutional Neural network)[2]와 같은 방법으로 보완하기 시작했다. FSRCNN과 ESPCN은 SRCNN의 convolu-

tion neural network의 마지막 layer를 transposed layer로 변경함으로써, 저선량 영상을 CNN에 입력하여 고해상도 영상을 출력한다. 이 방법으로 CNN 전체 연산량을 감소하여 시간을 항상 시켰으며, 고해상도 출력 영상의 화질 저하 또한 최소화 하였다. 이러한 딥러닝 기반 모델들을 사용하여 각 모델을 통해 출력된 CT 영상의 화질 분석을 통해 모형별 미치는 영향을 비교, 분석 하고자 한다.

## 2. MATERIAL

### 1. 데이터 준비

모형의 학습과 테스트를 위해 COVID-19 datasets와 COVID-CT-Dataset을 사용하였다. COVID-CT-Dataset에는 216명의 환자로부터 COVID-19의 임상 결과를 포함하는 349개의 CT 이미지가 있고 NON-COVID-19의 결과를 포함하는 397개의 CT 이미지가 있다. 저선량 CT 영상에서 고해상도 CT 영상으로 복원하는 모형을 만들기 위해 일반 CT 영상(GT; Ground truth)을 output으로 설정하였다. SRCNN은 일반 CT 영상에 Gaussian Noise를 추가하여 down sampling 후 bicubic interpolation 작업을 거친 CT 영상을 저선량 CT 영상(LR; Low resolution image)으로 모형의 input에 사용하였다. 데이터를 살펴보았을 때 저선량 CT 영상이 같이 있는 영상은 찾아보기 힘들어, 임의로 저선량 CT 영상을 만들어주었다. FSRCNN, ESPCN에 들어가는 일반 CT 영상에 Gaussian Noise를 추가한 저선량 CT 영상을 input에 사용하였다. 동일한 과정을 s&p Noise를 추가한 CT 영상과 poisson Noise를 추가한 CT 영상에 적용하여 어떠한 Noise가 추가되었는지 위 딥러닝 모형이 잘 복원을 하는지 볼 것이다. SRCNN, FSRCNN, ESPCN 영상의 size는  $299 \times 299$ 로 유지하였다.

### 2. 데이터에 추가한 Noise

우리 연구에서 CT 영상에 추가한 Noise는 Gaussian Noise, s&p Noise, poisson Noise이다. Gaussian Noise는 Gaussian 분포를 따라 이루어져 있다. Gaussian 분포는 모든 분야 등에서 가장 보편적인 분포로 자연 현상을 가장 잘

표현하는 이상적인 확률 모형이며 실험 오차 등에 많이 쓰이는 분포이다. 평균이 0이고 표준 편차가  $\sigma$ 인 1차원 Gaussian 분포는 다음 식처럼 나타낸다.

$$N(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp - \frac{1}{2\sigma^2} (x - \mu)^2$$

영상을  $N(0, \sigma^2)$ 를 따르는 정규 분포의 수를 생성해 낸 후 기존 영상에 Gaussian Noise를 더하여 Gaussian Noise가 추가된 영상을 얻는다.

s&p Noise(salt and pepper)의 경우 대표적으로 임펄스 잡음이다. 영상의  $I_{ij}$ 에  $dnlgks$  화소는 다음과 같이 표현 가능하다.

$$X_{i,j} = \begin{cases} Z, & \text{with probability } P \\ S, & \text{with probability } 1 - P \end{cases}$$

Z는 s&p Noise에 의해 훼손된 영상의 잡음값을 의미하고 잡음값은 두 가지의 극값을 가진다는 특성이 있다. S는 잡음에 훼손되지 않은 원 영상의 화소값을 의미한다.

poisson Noise는 변동(fluctuation)의 정도를 결정하는 중요한 특징이 평균값에서 다소 멀리 떨어져 있는 변동의 표준편차인데 이는 평균값의 제곱근과 같다. 수가 커지면 정규 분포에 근접하다. 영상의 pixel에 분산값을 곱해주고 나눠서 추가 해주면 된다.

### 3. METHOD

#### 1. SRCNN

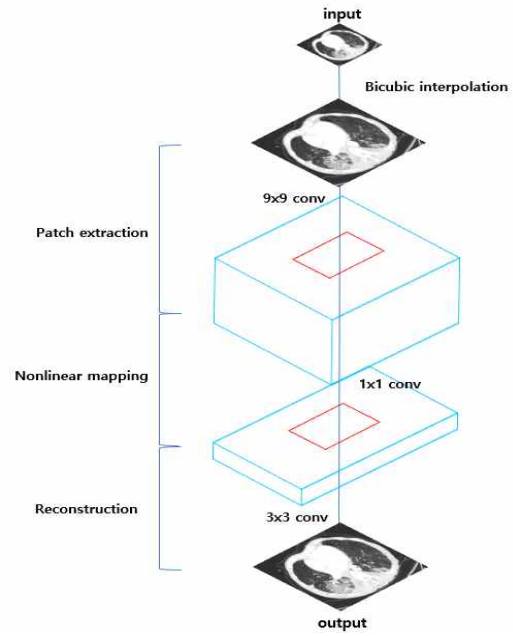


Fig 1. Architecture for SRCNN

SRCNN(Super resolution convolution nural network)은 Fig 1.과 같이 3개의 합성곱 계층으로 구성되어 있다. 먼저 모형에 들어가기 앞서 영상을 별도로 처리하는 과정을 거친다. 학습을 시킬 때 원본 영상을 정답 label로 두고 원본 영상을 down sampling 한 후 bicubic interpolation을 진행하여 input 으로 만든다. 첫 번째 층은 input 에서 여러 patch를 추출하고 feature map으로 나타 낸다. kernel size는  $9 \times 9$ , 추출한 feature map의 개수를 64개로 하였으며 다음 식으로 feature map을 추출한다.

$$F_1(Y) = \max_{\mathbb{R}}(0, W_1 * Y + B_1)$$

$$\begin{aligned} W_1 &: \text{Filter} \\ B_1 &: \text{Bias} \end{aligned}$$

두 번째 층은 한 vector 내에 다른 vector로 비선형 매핑을 한다. kernel size는  $3 \times 3$ , feature map의 개수를 32개로 하였다.

$$F_2(Y) = \max_{\mathbb{R}}(0, W_2 * F_1(Y) + B_2)$$

$$\begin{aligned} W_2 &: \text{Filter} \\ B_2 &: \text{Bias} \end{aligned}$$

세 번째 층은 만들어진 patch를 모아 고해상도 영상을 만드는 층이다. kernel size 는  $5 \times 5$ , feature map을 1로 설정하였다.

$$F(Y) = \max(0, W_3 * F_2(Y) + B_3)$$

$W_3$  : Filter

$B_3$  : Bias

활성화 함수(Activation function)는 ReLU 함수를 이용하였다. ReLU 함수의 식은 다음과 같다.

$$f(x) = ReLU = \max(0, x)$$

x는 영상의 각 픽셀값을 나타낸다. 손실 함수(Loss function)은 이진 분류(Binary Classification)이기 때문에 BCELoss를 사용하였다. BCELoss 함수의 식은 다음과 같다.

$$BCE(x) = -\frac{1}{N} \sum_{i=1}^N y_i \log(h(x_i; \theta)) + (1 - y_i) \log(1 - h(x_i; \theta))$$

y는 영상의 각 픽셀에 대한 예측값이며, N은 전체 픽셀의 개수이다. SRCNN 모형의 최적화를 위해 Adam(Adaptive moment estimation)을 사용하였다. learning rate는 0.01로 설정하여 진행하였다. kernel이 움직이는 정도를 나타내는 stride는 모든 층에  $1 \times 1$ 로 적용하였고, epoch당 한 번에 학습하는 data size인 batch size는 64로 설정하였으며 Jupyter Notebook 환경에서 Python과 pytorch framework를 이용하여 SRCNN 모형을 학습시켰다.

## 2. FSRCNN

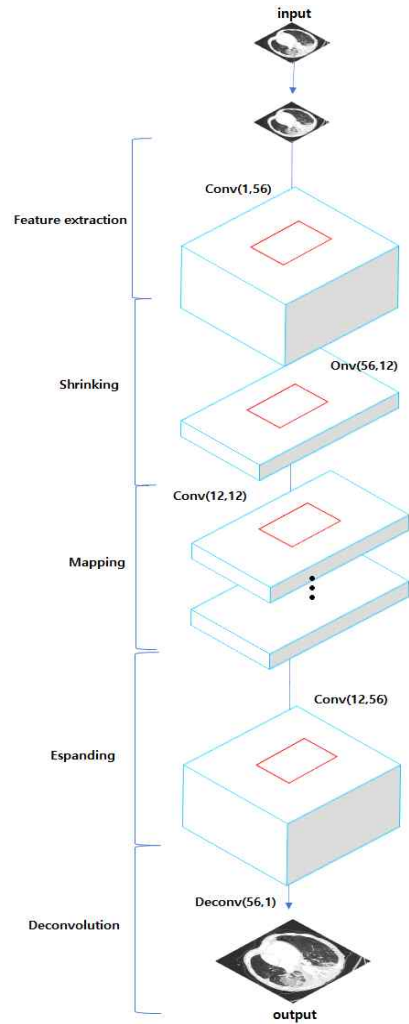


Fig 2. Architecture for FSRCNN

SRCNN에서 속도를 높이는 방법으로 고안된 FSRCNN 모형은 SRCNN에서의 input에 들어갈 영상의 작업을 별도로 하지 않아 계산량이 늘어나지 않는다. FSRCNN(Fast SRCNN)은 parameter가 많을수록 속도는 줄어들지만 그와 동시에 성능은 좋아진다. FSRCNN은 Fig 2.처럼 5가지의 단계로 진행된다. 첫 번째 층은 kernel size는 SRCNN처럼 input 영상의 size가 크지 않아  $5 \times 5$  이고 Feature map은 56으로 설정하였다. 두 번째 층은 convolution 연산을 수행시 연산 비용이 커지는 것을 방지하기 위해  $1 \times 1$  kernel size로 channel의 수를 줄인다. 따라서 feature map은 12로 설정하였다. 세 번째 층은 SRCNN의 두 번째 층과 동일한 역할인 비선형 매핑을 한다. kernel size는  $3 \times 3$ 이

고 feature map은 12로 설정하였다. 네 번째 층은 줄였던 channel의 수를 늘리기 위한 층이다. kernel size는  $1 \times 1$  이고 feature map은 56으로 설정하였다. 다섯 번째 층은 feature map을 확장시키는 upsampling의 기능을 가졌다. transposed convolution 연산을 하여 image의 size를 키워준다. kernel size는  $9 \times 9$ 이고 feature map은 1로 설정하였다. FSRCNN의 활성화 함수 (Activation function)는 ReLU 함수를 이용하였다. 손실 함수(Loss function)은 이진 분류(Binary Classification)이기 때문에 BCE Loss를 사용하였다. FSRCNN 모형의 전반적인 진행 방향은 SRCNN과 동일하게 하였다. FSRCNN의 최적화를 위해 Adam(Adaptive moment estimation)을 사용하였다. learning rate는 0.01로 설정하여 진행하였다. kernel이 움직이는 정도를 나타내는 stride는 모든 층에  $1 \times 1$ 로 적용하였고, epoch당 한 번에 학습하는 data size인 batch size는 16으로 설정하였으며 Jupyter Notebook 환경에서 Python과 pytorch framework를 이용하여 FSRCNN 모형을 학습시켰다.

### 3. ESPCN

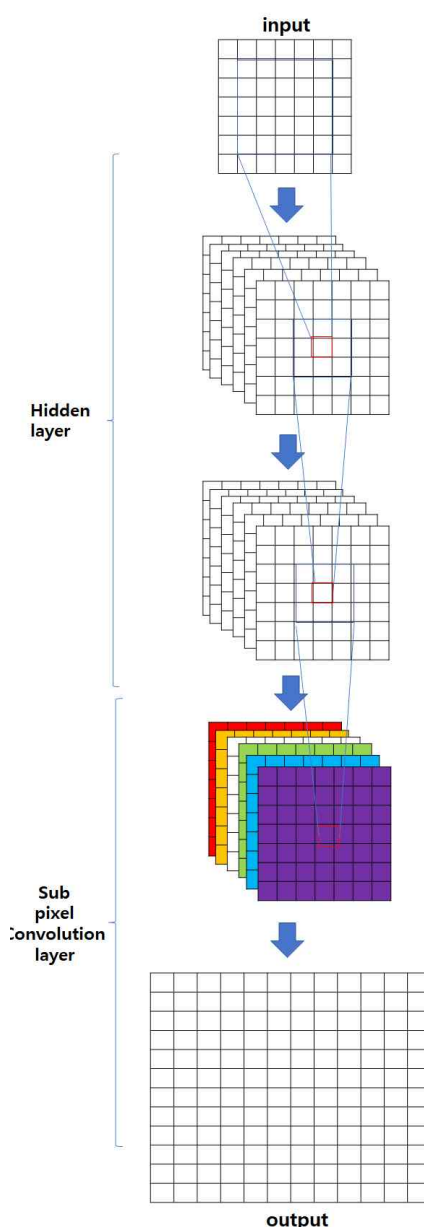


Fig 3. Architecture for ESPCN

ESPCN은 합성곱 계층 3층으로 이루어진 네트워크를 사용했다. 먼저 저선량 CT 영상에 바로 1개의 CNN을 적용한 후 저선량 CT 영상의 feature map을 upscale 해서 1개의 고해상도 CT 영상을 만들기 위해 sub-pixel convolution layer를 적용했다. Deconvolution layer는 Max-pooling 등과 같이 down sampling layer로부터 해상도를 복원할 때 자주 쓰는 방법 중 하나이다. 하지만 Deconvolution layer는 계산량이 많다. 저선량 CT 영상의 해상도를 높이

기 위한 다른 방법인 Efficient sub-pixel convolutional layer이 있다.

모델의 구조는 크게 총 3가지 층으로 나누어져 있다. 첫 번째 층은 Convolution층이다. input 영상의 크기는 input 영상의 channel의 개수(input channels)와 같다. kernel size는  $5 \times 5$  이고 데이터의 손실을 막기 위해 kernel size인 5에 해당하는  $\text{mod}(5/2)$ 만큼의 padding 작업을 해주었다. 층의 끝에는 활성화 함수로 Hyperbolic Tangent(tanh) 함수를 이용하였다. 두 번째 층 또한 Convolution층이다. kernel size는  $3 \times 3$  이고, 첫 번째 층과 마찬가지로 padding 작업을 해주었다. kernel size인 3에 맞추어  $\text{mod}(3,2)$ 값만큼 패딩을 해주었다. 활성화 함수는 마찬가지로 Tangent(tanh) 함수를 이용하였다. 마지막 층인 세 번째 층에서는 convolution과 Pixel-Shuffle과정이 함께 진행된다. Convolution의 경우 kernel size는  $3 \times 3$ 이고 padding 또한 위의 층과 동일하게  $\text{mod}(3,2)$ 로 진행해준다. 여기서 output channel이 Pixel-Shuffle을 진행하기 위해 영상의 channel의 개수(input channels)  $\times$  scale<sup>2</sup>이 된다. 본 연구에서 scale의 경우 3으로 사용하였다. Pixel-Shuffle연산은  $H \times W \times C \cdot r^2$  텐서를  $rH \times rW \times C$  텐서로 바꿔주는 연산이다. 수식으로는 아래와 같이 나타낼 수 있다.

$$PS(T)_{x,y,z} = T_{[x/r], [y/r], c \cdot r \cdot \text{mod}(y/r) + c \cdot \text{mod}(x/r)}$$

마지막 conv layer에는 nonlinearity를 부여하지 않고 이 layer로 저선량 CT 영상 feature maps에서 바로 고해상도 영상을 만든다. 이 연산은 각각 독립적으로 병렬처리 될 수 있기 때문에 빠르다. 전체 연산은 아래 수식처럼 표현할 수 있다.

$$I^{SR} = f^L(I^{IR}) = PS(W_L^* F^{L-1}(I^{LR}) + b_L)$$

#### 4. 평가 척도

학습된 모델을 이용하여 획득한 CT image가 잘 예측되었는지 확인하는 방법에는 여러가지가 있다. 우리 논문에서는 PSNR(Peak signal to noise ratio; 최대 신호 대 잡음비)로 측정하였다. PSNR은 영상의 복제값 중 최대값과 잡음의 비로 다음의 식으로 나타낼 수

있다.

$$PSNR = 20 \times \log_{10}(MAX_y) - 10 \times \log_{10}(MSE_y)$$

$MAX_y$ 는 모델을 통해 획득한 영상의 최대 복제값을 나타낸다.  $MSE_y$ 는 영상의 각 복셀에 대한 예측값과 실제값의 차이의 제곱을 나타낸 것이다.

#### 4. RESULT

Gaussian Noise를 추가한 저선량 CT 영상을 이용하여 각각의 모델에 영상 복원을 시도하였다. 결과는 다음과 같다.

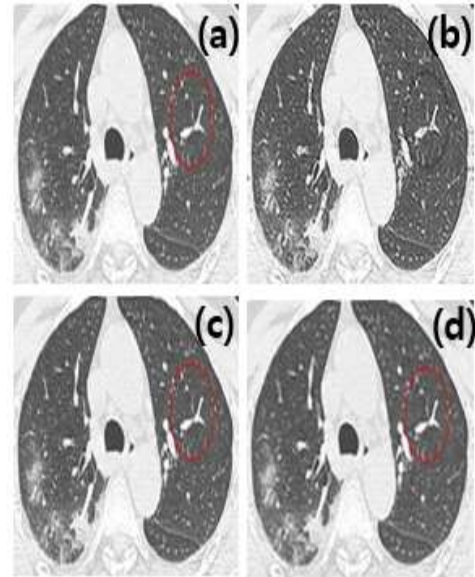


Fig 4. (a) Gaussian Noise 추가한 저선량 CT 영상 (b) SRCNN 모델로 만들어진 고해상도 영상 (c) FSRCNN 모델로 만들어진 고해상도 영상 (d) ESPCN 모델로 만들어진 고해상도 영상

COVID-19 환자의 CT 영상을 가지고 저선량 CT 영상으로 변환한 뒤, 각각의 모델의 결과를 살펴보자면, 판단에 혼돈을 주는 NOISE가 잘 제거된 모습이 보인다. Fig 3. 에 (a)는 원본 CT 영상에 Gaussian Noise를 추가한 저선량 CT 영상이다. 이를 이용하여 각각의 모델에 맞는 형태로 변환하여 input으로 넣어준



다. (b)는 SRCNN 모델을 통해 나온 고해상도 CT 영상을 나타낸다. SRCNN N input으로 넣어 주기 위해 우선 channel을 1로 grayscale 시킨 후 output에 color화 시켜주었는데 색이 돌아오지 않아 (a) 영상과 색의 차이를 보여준다. (c)는 FSRCNN 모델을 통해 나온 고해상도 CT 영상이다. (d)는 ESPCN을 통해 만들어진 고해상도 CT 영상이다. 시각적인 비교 뿐만 아니라 정량적 판단 척도인 PSNR 척도를 통해 살펴보자면 다음과 같다.

모형	PSNR
SRCNN	30.79
FSRCNN	36.02
ESPCN	30.11

Table 1. Gaussian Noise CT image PSNR

Gaussian Noise를 추가하여 만든 저선량 CT 영상과 모델을 통해 나온 output에 대한 PSNR 값은 FSRCNN이 가장 복원을 잘 했고 SRCNN, ESPCN 순서임을 확인할 수 있다.

다음은 s&p Noise를 추가한 저선량 CT 영상을 가지고 모형들의 결과를 비교해 보았다.

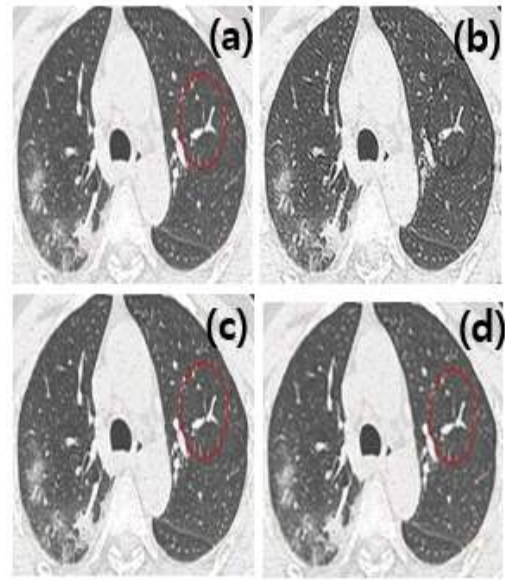


Fig 5. (a) s&p Noise 추가한 저선량 CT 영상 (b) SRCNN 모형으로 만들어진 고해상도 영상 (c) FSRCNN 모형으로 만들어진 고해상도 영상 (d) ESPCN 모형으로 만들어진 고해상도 영상

Gaussian Noise를 추가한 저선량 CT 영상보다 s&p Noise를 추가한 저선량 CT 영상이 소금을 뿌린 것 처럼 조금 더 잡음이 선명하게 보이는 것을 알 수 있다. 변환 과정은 전부 동일하고 (b)는 SRCNN을, (c)는 FSRCNN을, (d)는 ESPCN을 지난 OUTPUT을 나타낸다. 이 또한 PSNR을 확인해보면 Table2처럼 나온다.

모형	PSNR
SRCNN	30.13
FSRCNN	31.10
ESPCN	27.71

Table 2. s&p Noise CT image PSNR

s&p Noise를 추가한 저선량 CT 영상도 마찬가지로 FSRCNN의 PSNR 값이 가장 높음을 알 수 있다.

poisson Noise를 추가하여 만든 저선량 CT 영상을 가지고 모형들의 결과를 비교해 보았다.

poisson Noise를 추가한 저선량 CT 영상도

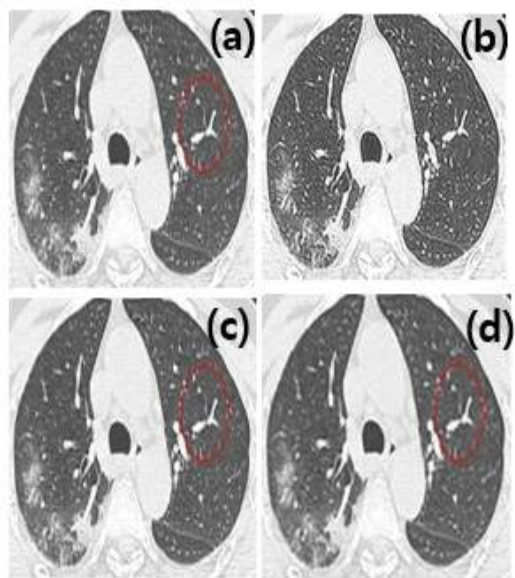


Fig 6. (a) poisson Noise 추가한 저선량 CT 영상 (b) SRCNN 모형으로 만들어진 고해상도 영상 (c) FSRCNN 모형으로 만들어진 고해상도 영상 (d) ESPCN 모형으로 만들어진 고해상도 영상

동일한 과정으로 진행하였고, PSNR을 구해보면 다음과 같다.

모형	PSNR
SRCNN	30.11
FSRCNN	35.74
ESPCN	29.77

Table 3. poisson Noise CT image PSNR

poisson Noise를 추가한 저선량 CT 영상도 FSRCNN의 PSNR 값이 가장 높음을 알 수 있다.

## 5. CONCLUSION

본 연구에서 적용한 딥러닝 기반 SRCNN, FSRCNN, ESPCN 모형은 CT 영상의 정량적 평가를 통해 화질이 향상시킬 수 있다는 것을 보여주고 있다. 화질을 향상시킨다는 것은 다른 말로 여러 Noise를 추가하여 비교하여 Noise를 영상 복원으로 제거할 수 있음이다. 본 연구를 조금 더 발전시켜 딥러닝 기반 CT 영상 복원 모형을 CT 데이터에 맞춰 여러 파

라미터를 조절한다면 조금 더 좋은 성능의 모형이 나올 것이다.

## 참고문헌

- [1] ] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," In Proceedings of the European conference on computer vision, Glasgow, United Kingdom, Springer, Cham, pp. 184-199, Sep. 2014.
- [2] Y. L. Cun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network", Advances in Neural Information Processing Systems, 1989.
- [3] Y. L. Cun, K. Kavukcuoglu, C. Farabet, "Convolutional networks and applications in vision", Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 253-256, 2010.
- [4] K. Umehara, J. Ota, T. Ishida, "Application of Super-Resolution Convolutional Neural Network for Enhancing Image Resolution in Chest CT ", Journal of Digital Imaging, Vol. 31, No. 4, pp. 441-450, 2018. <http://dx.doi.org/10.1007/s10278-017-0033-z>



- [5] Classification of COVID-19 pneumonia from chest CT images based on reconstructed super-resolution images and VGG neural network Cham, 2016.
- [6] I. Shiri, A.Alchavanallaf , A. Sanaat, Y. Salimi, D.Askari, "Ultra-low-dose chest CT imaging of COVID-19 patients using a deep residual neural network" In Proceedings of the European conference for computed tomography 1420-1431. Sep. (2021)
- [7] S. Zhang, G.Liang, S.Pan and L.Zheng, "A Fast Medical Image Super Resolution Method Based on Deep Learning Network", Proceedings of 2019 IEEE International Symposium,
- [8] 백지현, 김남호, "S&P 잡음 환경에서 표준 편차를 이용한 변형된 가중치 필터", 한국정보통신학회논문지 Vol.24, 4:474-480, Apr, 2020
- [9] N.Kibok, C.Jeonghyo, L.Seungwan, K.Burrryong, "Image Quality Evaluation in Computed Tomography Using Super-resolution Convolutional Neural Network", for Journal of the Korean Society of Radiology, Volume 14 Issue 3, Pages 211-220, (2020)
- [10] W.Shi, J.Caballero, F.Huszar, J.Totz, A.P.Aitken, R.Bishop, D.Rueckert, Z.Wang, "Real-Time Single Image and Video Super Resolution Using an Efficient Sub-Pixel Convolutional Neural Network" , for Computer vision and Pattern Recognition, CVPR 2016 paper
- [11] Dong, Chao, Chen Change Loy, and Xiaou Tang. "Accelerating the super-resolution convolutional neural network." European conference on computer vision . Springer,