

CNN 기반의 Image Denoising 구현

김새롬, 이보림, 전경훈
학점교류, 소프트웨어학부, 수학과

Implementation of CNN-based Image Denoising

Kim Saerom, Lee Bolim, Jeon Kyunghoon

Github : https://github.com/bo-lim/DnCNN_CAU_CV

I. 서론

Abstract

이미지에서 noise를 제거하는 Image Denoising은 이미지를 해석하거나 인식하는 등의 기술의 성능에 긍정적인 영향을 준다. 기존 연구에는 필터링 등의 통계 기반 방법들이 주를 이뤘지만 CNN을 이용한 방법을 소개하는 Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising [1] 논문이 발표되었다. DnCNN이 나오음으로써 이미지에서 노이즈를 찾아내는 방식으로 바뀌며 더 정교한 denoising을 보여주었다. 본 논문에서는 이러한 DnCNN에 대한 구체적인 소개와 설명 및 구현과 실험을 진행하였다. 이미지에 추가하는 각 가우시안 노이즈 레벨에 따른 batch size, optimizer, 레이어 수를 변경해가며 최적의 조합을 찾고 기존 논문 결과와 비교 및 해석해보았다.

Image Denoising이란 이미지에서 나타날 수 있는 noise를 제거하여 원래의 고해상도 이미지로 복원하는 과정을 의미한다. 인간은 시각을 통해 사물을 인식하는 과정에서 많은 정보를 얻는다. 여러 디지털 매체가 발전한 만큼 디지털 이미지를 시각 자료로써 사용하여 많은 정보를 얻기도 한다.

그러나 이미지에 불규칙한 이상 패턴, 균일하지 않은 픽셀 분포와 같은 noise가 적용이 되어 있다면 시각적으로 이미지를 해석하기 어려운 상황이 발생할 수 있고, 시각적인 정보를 처리하고 해석하는 컴퓨터비전 분야에서 구현하고자 하는 이미지 분석, 패턴 인식, 객체 감지와 같은 기술의 적용에도 부정적인 영향을 미칠 수 있다.

이와 같은 문제를 해결하고자 noise를 최소화하고 이미지의 세부 정보를 최대한 보존하면서 시각적인 품질을 개선하고자 하는 Image Denoising 기술이 개발되었다.

Image Denoising의 초창기 연구는 딥러닝 모델을 활용한 구조 보다는 필터링(filtering), 잡음 모델링 및 통계 기반 방법들이 주를 이뤘다. 필터링의 방법으로는

타겟과 비슷한 intensity를 가지는 주변 픽셀을 고려하여 Edge-preserving에 강한 Bilateral Filtering, 여기에 location 정보를 추가한 Non-local means filter 등이 있다.

그러나 딥러닝 기반의 방법은 이러한 한계를 극복하였으며, 더 정교한 denoising이 가능하다는 것을 보였다.

따라서 본 논문에서는 Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising [arXiv, 2016]에서 제안한 컨볼루션 신경망 기반의 denoising 방법인 DnCNN(Denoising Convolutional Neural Network)를 소개하고 직접 구현해보고자 한다.

II. 본론

본 장에서는 잔차 연결, 배치 정규화 그리고 여러 층의 합성 신경망을 활용을 하여 이미지의 잡음을 제거하는 방법을 소개한다.

2.1 Problems for Existing Denoising Methods

모델의 설명에 앞서 본 장에서는 DnCNN 이전의 기법들의 문제와 한계점을 제시한다. Bilateral Filtering, Non-local means filter와 같은 기존의 방법들은 특정 통계적 가정에 기반하며, Denoising 성능과 원본 이미지의 세부 정보 보존 사이의 균형을 조절하는 것이 한계였다. 또한 계산이 복잡하며, 많은 시간이 소요되며, 수동적으로 파라미터를 설정해야 된다는 한계가 존재했다.

그러나 딥러닝 기반의 방법은 이러한 한계를 극복하였으며, 더욱 정교한 Denoising이 가능하다는 것을 보였다. DnCNN의 경우 이미지에서 잡음을 분리하는 작업과 더불어 잔차 연결, 배치 정규화 그리고 여러 층의 합성 신경망을 활용하여, 성능 향상을 이끌어냈으며, 간단한 구조를 통해서 빠른 연산 속도를 이룩하였다.

2.2 Model Architecture

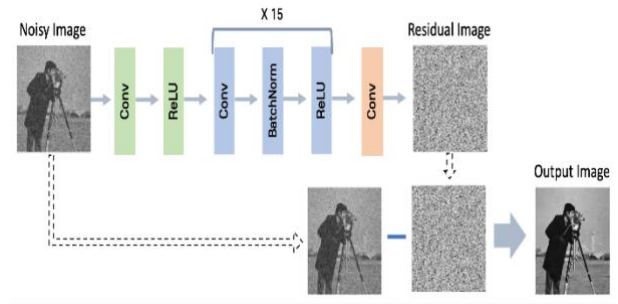


그림 1. 모델 아키텍처

잔차 연결과 여러 층의 합성 신경망이 중심인 DnCNN 모델의 구조는 간단하다. DnCNN은 다음과 같은 구조로 이루어져 있다.

- 입력 레이어: Gray Scale 이미지의 형태를 가지며, 고정 크기의 패치(patch) 형태로 사용된다.
- 합성곱 레이어: 이미지의 특징을 추출하기 위해 N개의 합성곱 레이어를 사용한다. 이때 N은 15, 18, 23로 설정하였다. 이는 작은 크기의 필터를 사용하여 이미지의 지역적 특징을 파악한다. 합성곱 레이어의 개수(N)는 모델의 깊이를 결정하며, 더 깊을수록 복잡한 특징을 학습할 수 있다.
- 활성화 함수: ReLU를 활용한다. ReLU의 경우 입력값이 0보다 작을 때 0을 반환, 0보다 크거나 같을 경우 그대로 출력한다. 비선형 활성화 함수이므로, 모델은 비선형성을 학습하여, 더욱 복잡한 특징을 추출할 수 있다.

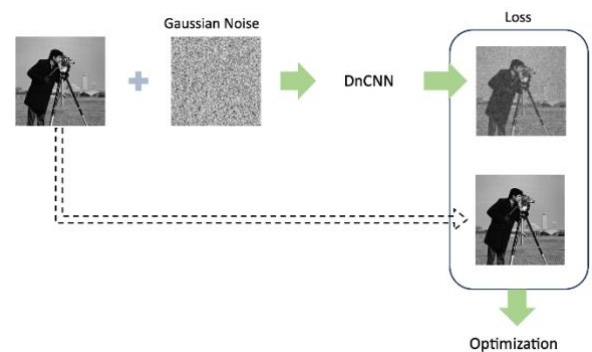


그림2. 모델의 흐름

DnCNN은 기본적으로 CNN을 활용한다. 먼저 정답 이미지를 활용하여 가우시안 잡음을 Noisy Image를 만든다. 결과로 만들어진 Noisy Image를 DnCNN 모델을

거쳐 passed image를 만들어 기존의 정답 이미지에서 passed image를 빼는 연산을 거쳐 Residual Image를 만든다. 이제 정답 이미지와 결과로 나온 Residual Image의 MSE Loss를 계산하여 최적화를 진행하는 것이다.

2.3 Residual Learning and Batch Normalization for Image Denoising

본 논문에서는 모델의 효과적인 학습을 위해 잔차 학습(Residual Learning formulation)을 사용하였으며, 더욱 빠른 학습과 향상된 잡음 제거 수행을 위해 배치 정규화(Batch normalization)을 활용하였다.

잔차 학습의 경우 파라미터를 추가하는 것이 아닌, 입력과 출력 사이에 새로운 연결을 만드는 것이므로 계산의 복잡도가 올라가지 않는다. 나아가 모델이 깊어져도 일반화를 잘 시키며, 정확도를 높이는 효과를 가진다.

배치 정규화의 경우 역전파시 파라미터의 범위에 덜 민감하여 학습률을 크게하여 빠르게 학습할 수 있다. 또한 자체적인 규제화를 하여 낮은 민감도를 갖는다.

2.4 Evaluation Indicator for Image Denoising

실험 결과 소개에 앞서, 모델을 평가하는 지표를 소개한다. 본 논문에서는 이미지의 품질을 평가하는데 PSNR과 SSIM, 두 가지 지표를 활용한다. [2]

먼저 PSNR(Peak Signal-to-Noise Ratio)의 경우 신호가 가질 수 있는 최대 전력에 대한 잡음의 전력을 의미한다. 두 이미지 간의 PSNR값은 재구성된 이미지가 원본 이미지와 얼마나 유사한지를 나타내며, 높은 PSNR 값은 높은 품질을 의미한다.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M * N}$$

PSNR은 일반적으로 데시벨(dB) 단위로 표현되며, 값이 30dB 이상이면 이미지의 품질이 매우 좋다고 간주된다. 그러나 PSNR의 경우 인간의 시각 체계와 일치하지 않을 수 있다. 그리하여 SSIM이라는 평가 지표를 소개한다.

SSIM의 경우 재구성된 이미지와 원본 이미지 간의

구조적 유사성을 측정하여 이미지 품질을 평가한다. SSIM은 인간의 시각적 인식과 관련하여 밝기, 대비와 구조를 활용한다.

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

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x \sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}$$

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i, \quad \sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2}, \quad \sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$$

위의 식과 같이 세가지 요소를 고려하면, SSIM의 값은 0부터 1 사이의 값을 지니게 된다. 이때 높은 SSIM의 값은 높은 구조적 유사성과 이미지 품질을 의미한다. 일반적으로 0.9보다 높은 경우 이미지의 품질이 매우 좋다고 간주된다. SSIM은 PSNR과 다르게 인간의 시각적 인식과 일치하는 지표이므로 이미지의 구조적 특징을 활용한 품질 평가에 용이하다.

III. 실험

3.1 Dataset

Train, validation dataset은 Berkeley Segmentation Dataset을 사용하였고, test dataset은 Set12, Set68 dataset을 사용하였다. Set12, Set68 dataset은 영상 복원 이미지 업스케일링 알고리즘의 성능 평가를 위해 사용하기도 한다.

3.2 실험 환경

본 실험에서는 Google이 제공하는 클라우드 기반의 Jupyter 노트북 환경인 Google Colaboratory를 통해 웹 브라우저 상에서 실험을 진행하였다. 딥러닝 프레임워크는 PyTorch를 사용하였다.

Colaboratory 내부 환경에서는 텍스트 기반의 데이터 교환 형식으로 사용되는 json 파일을 통해 파라미터를 관리하고, 모델의 학습 과정 및 실험 결과를 시각화하여 정리하기 위해 Weight and Biases 플랫폼을 활용하였다.

3.3 실험 내용

본 실험에서는 이미지의 Noise Level 별로 코드에

적용된 하이퍼파라미터를 변경하면서 PSNR과 SSIM 평가 지표에 따른 결과를 비교하고자 하였다. PSNR은 원본 이미지와 노이즈가 제거된 이미지 간의 품질 차이를 수치화 한 평가 지표이고, SSIM은 인간 시각 시스템의 특성을 모델링하여 이미지 간의 시각적 화질 차이를 파악할 수 있는 평가지표이다. 이미지에 noise가 적용된 정도를 의미하는 Noise Level은 σ 로 정의하며, $\sigma=10$, $\sigma=25$, $\sigma=50$ 로 나누었다. 변경할 하이퍼파라미터로는 Batch Size, Optimizer, CNN layer의 수가 있다.

	exp1	exp2	exp3	exp4	exp5	exp6
Noise Level (σ)	10	10	10	10	10	10
Batch Size	64	64	64	64	64	64
Optimizer	Adam	Adam	Adam	SGD	SGD	SGD
Number of layers	17	20	25	17	20	25

표 1. $\sigma=10$ 일 때의 실험 구성

	exp7	exp8	exp9	exp10	exp11	exp12
Noise Level (σ)	10	10	10	10	10	10
Batch Size	128	128	128	128	128	128
Optimizer	Adam	Adam	Adam	SGD	SGD	SGD
Number of layers	17	20	25	17	20	25

표 2. $\sigma=10$ 일 때의 실험 구성

$\sigma=25$ 와 $\sigma=50$ 의 경우에도 [표 1], [표 2]와 동일한 실험 환경에서 진행하였다.

3.4 실험 결과

	Batch Size	Optimizer	Number of layers	PSNR (dB)	SSIM
$\sigma=10$	128	Adam	17	32.58	0.944
$\sigma=25$	64	Adam	25	34.47	0.968
$\sigma=50$	64	Adam	25	24.79	0.858

표 3. Noise Level에 따른 결과

$\sigma=10$ 일 때는 layer를 깊게 쌓지 않아도 성능을 확보할 수 있었으며, 오히려 과도한 학습은 과적합 (overfitting)의 문제를 발생시킬 가능성이 있었다. $\sigma=25$, $\sigma=50$ 일 때는 이미지에서 제거해야 할 noise의 양이 많다보니 epoch을 거듭할 수록 성능이 향상되는 것을 확인하였다. 또한 작은 Batch Size를 사용했을 때, 데이터의 변동성을 잘 따라가 수렴 속도를 높일 수 있었던 것으로 예상된다. Optimizer는 모든 Noise Level에서 Adam이 우수한 성능을 보였다.

또한 기존 참고 논문에서의 결과와 수치적으로 비슷한 값을 나타냈으며, 각 Noise Level에 대해 PSNR, SSIM 모두 비슷한 변화 양상을 보였다. 다만 epoch를 더 증가시켜 실험을 한다면 참고 논문 저자들의 실험 결과와 더 유사한 값을 나타낼 수 있을 것으로 예상된다.

IV. 결론 및 향후 연구 방향

본 연구에서는 잔차 연결과 배치 정규화를 통한 여러 층의 합성곱 신경망을 통해 이미지의 잡음을 제거하는 방법을 채택하였다. 제안한 방법을 데이터셋에 대하여 실험한 결과, PSNR과 SSIM에서 평균 30.61dB 그리고 0.923의 값이 확인되었다. 이는 본론에서 언급한 높은 품질의 이미지의 범주에 들어가는 수치이다. 향후 연구는 효과적인 noise 제거 방법을 통해 이미지의 품질을 높일 수 있는 방법을 찾는 방향으로 발전할 것이며, 본문에서 제시된 noise 제거 기법에 더해 다양한 산업에서 활용할 수 있을 것이다.

부록

Our Github : <https://github.com/bo->

[lim/DnCNN_CAU_CV](https://github.com/bo-lim/DnCNN_CAU_CV)

Github: <https://github.com/cszn/DnCNN>

Blog: <https://velog.io/@danielseo/Computer-Vision-DnCNN>

Berkeley Segmentation Dataset:

<https://github.com/smartboy110/denoising-datasets/tree/main/BSD400>

Set12dataset:

https://www.researchgate.net/figure/Twelve-test-images-from-Set12-dataset_fig1_334308775

Set68dataset:

<https://github.com/claumichele/CBSD68-dataset>

참고문헌

- [1] Kai Zhang, Yunjin Chen, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", IEEE Transactions on Image Processing vol.26, 2016
- [2] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi; "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network "The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4681-4690