

## Image Completion with inpainting

2020-08-24 노정현

기간 : 2020-08-19 ~ 2020-08-23

### ● Image Completion과 Inpainting

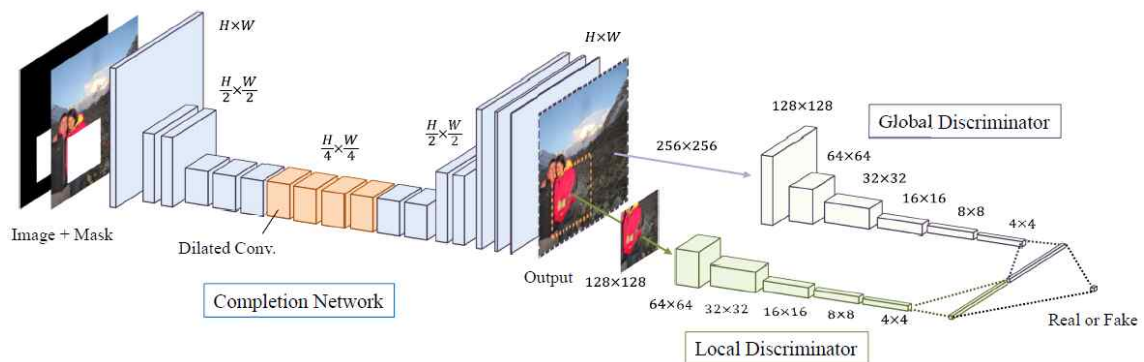
Image Completion

- 원하는 이미지의 부분에 다른 내용을 채우는 기법
- 원하지 않는 부분을 잘라내는데 사용

Image Inpainting

- 이미지의 특징들 추출해 가려지거나 보이지 않는 부분을 복원, 생성

### ● 구조



Completion Network

- CNN을 사용하며 이미지를 완성

Context Discriminators (GAN 사용)

- Completion Network에서 만들어진 영상을 이용하여 진짜인지 가짜인지 구분하며 학습을 진행

#### - global discriminator

전체 이미지를 입력으로 받아 전체의 일관성을 인식

#### - local discriminator

채워진 부분 주위만 입력으로 받아 세부 모습의 품질을 평가

### ● 특징

- 임의의 이미지 크기와 마스크 사용가능
- local consistency 유지
- 새로운 물체들을 생성 가능

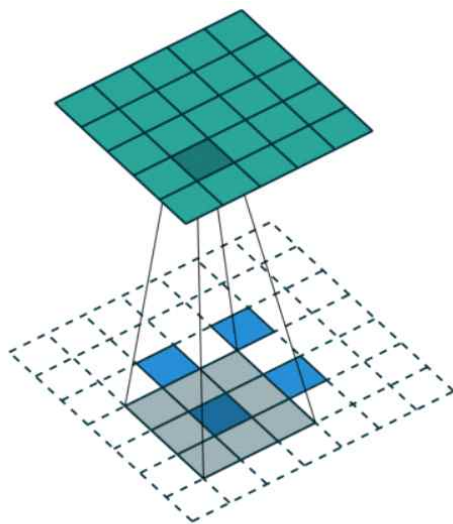
● Completion Network

Type	Kernel	Dilation ( $\eta$ )	Stride	Outputs
conv.	$5 \times 5$	1	$1 \times 1$	64
conv.	$3 \times 3$	1	$2 \times 2$	128
conv.	$3 \times 3$	1	$1 \times 1$	128
conv.	$3 \times 3$	1	$2 \times 2$	256
conv.	$3 \times 3$	1	$1 \times 1$	256
conv.	$3 \times 3$	1	$1 \times 1$	256
dilated conv.	$3 \times 3$	2	$1 \times 1$	256
dilated conv.	$3 \times 3$	4	$1 \times 1$	256
dilated conv.	$3 \times 3$	8	$1 \times 1$	256
dilated conv.	$3 \times 3$	16	$1 \times 1$	256
conv.	$3 \times 3$	1	$1 \times 1$	256
conv.	$3 \times 3$	1	$1 \times 1$	256
deconv.	$4 \times 4$	1	$1/2 \times 1/2$	128
conv.	$3 \times 3$	1	$1 \times 1$	128
deconv.	$4 \times 4$	1	$1/2 \times 1/2$	64
conv.	$3 \times 3$	1	$1 \times 1$	32
output	$3 \times 3$	1	$1 \times 1$	3

Conv. : 11개, Dilated Conv. : 4개, Deconv. : 2개 => 총 17개

- encoder-decoder 구조를 따르며 메모리와 계산을 줄여줌
- 다른 모델들과는 다르게 pooling을 사용하지 않고 strided convolution을 사용하여 이미지를 1/4로 축소

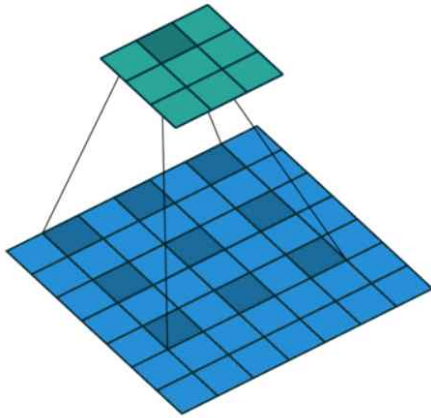
strided conv. : convolution 연산 시 일정 칸씩 띄우면서 연산



Transposed 2D convolution with no padding, stride of 2 and kernel of 3

strided가 convolution과 downsampling을 적용 시 pooling보다 효과적  
pooling과 strided는 같은 효과를 보이지만 pooling이 전통적인 방식

- Deconvolution layer를 통해 원래의 resolution으로 복원
- Dilated convolutional layer 사용 (Dilate : 팽창)



파란색 : input, 녹색 : output

진한 파란부분 : weight 나머진 0

필터 내부에 zero padding을 추가해 강제로 receptive field를 늘림  
동일한 계산으로 더 넓은 시야를 가짐  
기존정보의 손실이 적고 특성을 유지  
대부분이 weight가 0이므로 연산 효율성

## ● Context Discriminators

- Generative Adversarial Networks(GAN)를 기반으로 한 Local, Global Discriminator가 Completion Network에서 만들어진 이미지를 이용하여 학습

### - GAN (생산적 적대 신경망)

generative model를 CNN을 이용하여 비지도학습하는 신경망으로

generative model은 discriminator라 불리는 보조 네트워크를 이용하여 만들어진 이미지가 진짜인지 가짜인지를 학습하는 모델

=> generator network가 discriminator network를 속이도록 학습

(ex. 위조지폐범과 경찰)

generator network의 손실함수와 GAN의 손실함수의 조합으로 가중치 학습

GAN은 학습 중 불안정하다는 단점 => 순수한 generative model을 사용하지 않고 두 개의 discriminator 사용(Global and local)

(a) Local Discriminator				(b) Global Discriminator			
Type	Kernel	Stride	Outputs	Type	Kernel	Stride	Outputs
conv.	$5 \times 5$	$2 \times 2$	64	conv.	$5 \times 5$	$2 \times 2$	64
conv.	$5 \times 5$	$2 \times 2$	128	conv.	$5 \times 5$	$2 \times 2$	128
conv.	$5 \times 5$	$2 \times 2$	256	conv.	$5 \times 5$	$2 \times 2$	256
conv.	$5 \times 5$	$2 \times 2$	512	conv.	$5 \times 5$	$2 \times 2$	512
conv.	$5 \times 5$	$2 \times 2$	512	conv.	$5 \times 5$	$2 \times 2$	512
FC	-	-	1024	conv.	$5 \times 5$	$2 \times 2$	512
				FC	-	-	1024

(c) Concatenation layer

Type	Kernel	Stride	Outputs
concat.	-	-	2048
FC	-	-	1

- Global Discriminator

전체 이미지를 256 x 256로 rescale 후 6개의 convolution layer와 하나의 fully-connected layer를 통해 1024 차원의 벡터로 반환

- Local Discriminator

Completion network에 의해 완성된 지역을 중심으로 128 x 128의 이미지를 input으로 하고 나머지는 Global Discriminator과 동일

처음부터 128 x 128의 이미지를 사용하기 때문에 conv. layer가 하나 적음  
학습 시 한 번에 하나의 지역만 고려

- Concatenation Layer

Global, Local에서 나온 각각의 1024 차원의 벡터를 연결하여 2048 차원의 벡터를 만들고 하나의 fully-connected layer를 이용하여 0과 1 사이의 값을 결과로 하여 이미지가 진짜인지 가짜인지 구분

- 프로그램 동작 과정

1. 학습이 완료된 모델을 읽어온다.
2. 테스트하기 위한 이미지와 마스크를 크기가 4의 배수가 되게 조정한다.
3. 이미지와 마스크를 torch에서 사용하기 위해 tensor로 변환한다.
4. 이미지에 마스크 부분이 검은색이 되도록 채널별로 적용한다.
5. 마스크 처리된 이미지와 마스크를 input으로써 모델에 입력한다.
6. 결과는 마스크 부분에 해당하는 이미지만 있으므로 마스크 처리된 이미지에 더해 결과를 반환한다.

● 결과 (1083, 658) - 14.6s





- 결과 (960, 540) - 10.6s



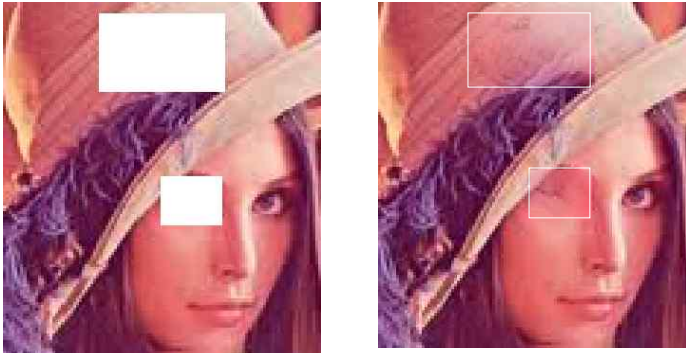
- **모델의 장점 및 한계점**

- **장점**

- 기존의 다른 모델들과 비교하여 입력 영상의 크기가 정해지지 않음
- 지역 일관성을 유지
- 새로운 물체를 생성 가능
- CNN을 사용하여 자체적으로 특징점 검출
- 두 개의 GAN을 사용하여 이미지의 완성도를 높임

- **한계점**

- 원본 이미지와 마스크가 필요함
- 인물의 얼굴에 적용 시 한계가 있음 => 얼굴 특징에 대한 학습이 없기 때문



- 복잡한 구조를 갖는 부분에 대한 한계