1. 논문 소개

이번 글에서는 2016년 CVPR에 발표된 Context Encoders: Feature Learning by Inpainting 논문을 리뷰합니다.

리뷰는 논문 소개, 기존 방식과의 차이, 관련 연구, 컨텍스트 인코더, 결과 순서로 진행됩니다.

먼저 위쪽에 있는 이미지를 보겠습니다. 중앙 부분이 빠져 있지만 여기 계신 대부분의 사람은 주변 픽셀에서 해당 내용을 쉽게 상상할 수 있을 겁니다. 이 능력은 자연 이미지가 다양하더라도 건물 정면의 창문의 규칙적인 패턴처럼 고도로 구조화되어 있기 때문에 나타납니다.

본 논문에서는 최첨단 컴퓨터 비전 알고리즘이 이를 수행할 수 있는지에 대한 궁금증에서부터 시작되었습니다.

Image Inpainting은 이미지에서 훼손된 부분을 복원하거나 불필요한 문자나 특정 물체를 제거한 후 삭제된 영역을 자연스럽게 채우기 위해 널리 사용되는 기법입니다. 쉽게 말해, 이미지의 일부분이 누락되어 있어도 사람은 비워져있는 부분을 유추할 수 있기에 다음 사진과 같이 "딥러닝 기술을 응용하여 복원하겠다." 라는 의미입니다.

Context Encoders란 상황 기반 픽셀 예측을 이용한 이미지 학습 알고리즘으로 Inpainting을 사용하여 이미지의 빈 영역을 예측하고 채워 넣음으로써 주

변 픽셀의 Context 정보를 학습합니다.

논문에서 제안하는 Context Encoders는 Autoenoder와 유사한 형태인데요. 두 모델은 비지도 학습 방식을 사용하여 훈련됩니다. 즉, 레이블이 포함되지 않은 데이터에서 특징을 학습하고 복원하는 과정에서 모델이 스스로 학습합니다. 또한 인코더와 디코더로 구성된 네트워크를 가지고 있습니다. 이 구조는 입력 데이터를 저차원 특징으로 인코딩하고 다시 원본 데이터로 디코딩하는 역할을 합니다.

이 내용은 다음 장에서 자세히 다뤄보겠습니다.

2. 기존 방식과의 차이

먼저 오토 인코더는 단순히 입력을 출력으로 복사하는 신경망입니다. 하지만 간단한 신경망과 달리 네트워크에 여러가지 제약을 가함으로써 어려운 신경망으로 만드는 것을 의미합니다. Image Inpainting을 오토 인코더를 응용해서도 구현이 가능한데요. Autoenoder는 이미지를 입력되면 작은 차원인 병목레이어를 통과시키는 방식으로 Image를 재구성합니다. 하지만 영상의 의미론적인 특징을 파악하지 못하고 단순히 영상을 압축 시킬 수 있다는 문제점이 있습니다.

그래서 나온 것이 Donising Autoenoder입니다. Donising은 잡음 제거를 의미하는데요. 입력할 이미지에 noise를 주고 네트워크 내에서 그 noise를 해결하는 방식으로 특징을 파악하는 시스템이 있습니다. 하지만 이미지에 noise를 주는 과정이 low-level에서 진행되서 번거롭고 원상태로 복구할 때도 생각보다 의미있는 정보를 요구하지 않아 특징을 잘 찾는다고 말하기 힘들다는 단점이 있습니다.

대조적으로 이 논문에 기재된 Context Encoders는 특징을 잘 파악해서 아주 힘든 일도 할 수 있다고 주장합니다. 예를 들어 주변 픽셀로부터 힌트를 얻기 힘든 영상임에도 불구하고 누락된 영역을 채울 수 있다는 것입니다.

(c) Context Encoder ($L2$ loss)　　(d) Context Encoder ($L2$ + Adversarial loss)

Autoenoder에서 나타나는 문제는 재구성 손실과 적대적 손실을 최소화함으로써 제거 하였다고 주장하였는데요. L2라고 불리는 재구성 손실은 누락된 영역의 전체 구조를 주변 상황과 관련해서 포착하는데 쓰이고 적대적 손실은 누락된 부분을 채우고 일관성을 유지시키는 여러 방법 중 특정 기법을 선택하는 것과 관련이 있습니다. 단순히 재구성 손실만 사용하면 (c) 이미지처럼 흐릿한 결과를 반환하지만 재구성 손실과 적대적 손실을 모두 최소화하면 (d) 이미지와 같이 더 깔끔한 결과를 반환할 수 있습니다.

평가도 인코더와 디코더를 독립적으로 실시하는데요. 인코더는 이미지 패치(잘게 자른 이미지로 생각됨)의 컨텍스트를 인코딩하면 resulting feature(결과로 얻은 특징으로 생각됨)을 얻는데 이 feature와 인접한 컨텍스트를 검색하면 원래 패치와 의미적으로 유사한 패치가 생성된다고 합니다. 이러한 작업이 곧 이미지 이해이기 때문에 이 이해도에 대해서 평가합니다. 디코더는 컨텍스트 인코더가 누락된 영역을 채울 수 있다는 것을 보여주는데요. 큰 누락 영역에 대해서 합리적인 결과를 제공할 수 있는지 평가합니다.

3. 관련 연구

이 논문에서는 Context Encoders와 관련된 각 분야에 대한 관련 연구를 간단히 소개해주었는데요.

첫 번째는 비지도 학습입니다.
비지도 학습이란 정답이 주어지지 않는 학습 전략입니다. 어떤 입력에 대한 올바른 결과가 무엇인지 알 수 없다는 특징을 가지고 대표적인 예로는 군집화와 스케일링이 있습니다. 그리고 심층 비지도학습의 연구 중 Autoenoder가 있는데요. 그 중 노이즈를 제거할 수 있는 오토 인코더는 이미지 재구성 능력이 뛰어납니다. Context Encoders도 노이즈를 제거할 수 있는 Autoenoder라고 생각할 수 있지만 모델의 입력에 적용된 손상이 훨씬 커서 회복되기에 더 많은 정보가 필요하다고 합니다.

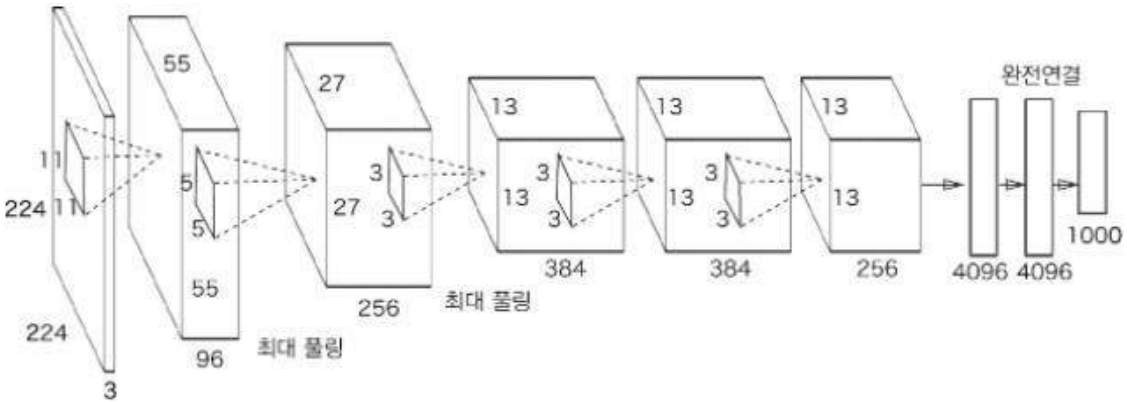
두 번째는 약한 지도학습입니다.
지도 학습이란 정답이 주어지는 학습 전략인데요. 어떤 input에 대한 올바른 output이 무엇인지 알 수 있다는 전제를 가집니다. 따라서 지도 학습을 위해서는, 데이터 셋과 데이터 각각에 대한 정답을 제공받아야 합니다. 그러나, 약한 지도 학습 환경에서는 주어지는 정답에 대한 정보가 제한됩니다. 비지도 학습과 같이 아무 정보도 주어지지 않는 경우와는 다르지만 일부에 대한 정보만 제공받아 학습하고, 그러한 학습을 통해 제공받지 않은 정보를 예측해내야 합니다. 영상 인식에서 객체에 대한 클래스 정보만을 제공받았지만, 영상 내의 객체 위치를 예측해내는 학습 모델을 예로 들 수 있습니다. 그리고 컴퓨터 비젼 분야에서 필요한 라벨링 작업은 주로 인적 자원을 이용하여 이루어지므로 많은 시간적, 경제적 비용이 소모되는데 CNN 기반 약한 지도 학습을 이용하면 비용 면에서 효과적이라고 합니다.

세 번째는 자기 지도 학습입니다.
자기 지도 학습은 비지도학습의 연구 주제 중 하나입니다. 비지도학습이기 때문에 레이블이 존재하지 않는 데이터만 이용합니다. 이때 사용되는 데이터는 이미지가 될 수도 있고, text, speech, video 등 다양한 종류의 데이터가 될 수 있습니다. 그래서 결국 라벨이 없는 데이터를 이용해서 스스로 학습 후 분류하는 것을 의미합니다. 이것이 가능하게 하려면 사전 학습이 필요합니다. 예를 들어 회전한 영상을 입력으로 주고 회전한 각도를 맞추게 하는 사전 학습 과제가 있다면 이 과정에서 기계는 상의 고품질 성 벡터를 추출하게 되고 이 추출한 것이 추후 분류에도 유리하게 작용할 것이라는 아이디어입니다. 이러한 학습 또한 비전에서 라벨링하는 작업을 줄이기 때문에 비용 면에서 효과적이라고 합니다.

4. Context Encoders

이제 컨텍스트 인코더에 대해 자세히 알아보겠습니다.
먼저 일반 아키텍처의 개요와 학습 절차에 대한 세부 정보를 제시하며, 마지막으로 이미지 영역 제거를 위한 세가지 전략 순서로 진행하겠습니다.

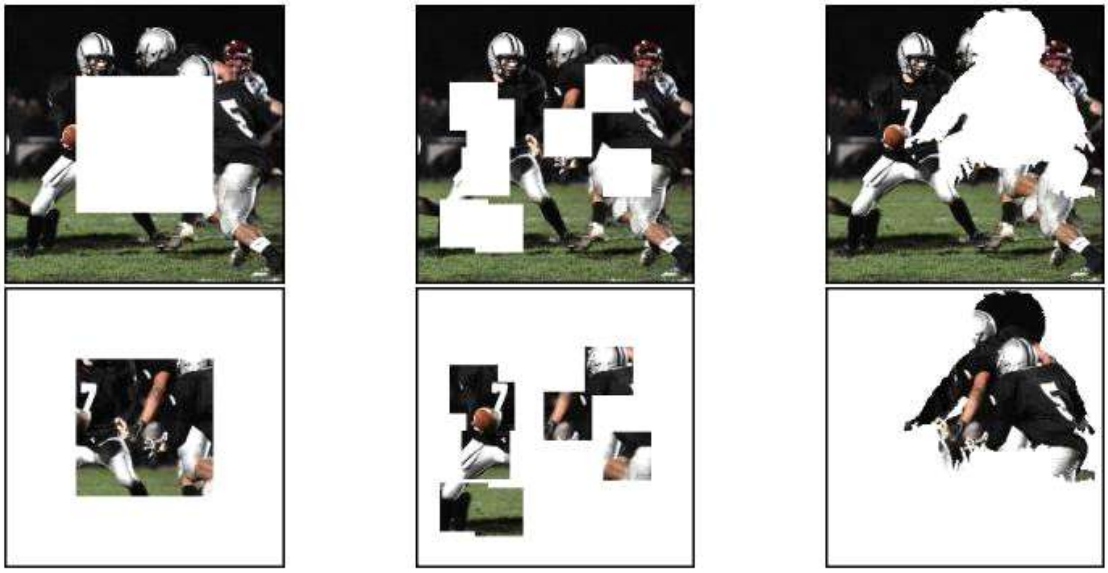
컨텍스트 인코더의 전체적인 아키텍처는 간단한 인코더-디코더 파이프라인입니다. 인코더는 누락된 영상으로부터 잠재 표현을 생성하고, 디코더는 누락된 이미지의 컨텐츠를 생성합니다. 이 둘은 채널 별로 Fully Connected Layer로 돼있고 이렇게 하였을 때 디코더가 전체 이미지 내용을 추론할 수 있습니다. 즉 인코더가 추론한 것을 가져올 수 있겠다고 이해한 것입니다.



인코더는 AlexNet 아키텍처에서 파생되었으며, 입력 이미지 크기가 227x227이며 첫 다섯 개의 컨볼루션 레이어와 그 뒤에 오는 pooling 레이어(pool5)를 사용하여 추상적인 6 × 6 × 256 차원의 특징 표현을 계산합니다. AlexNet과 달리 해당 모델은 ImageNet 분류를 위해 훈련되지 않았으며 대신에 네트워크는 무작위로 초기화된 가중치로부터 "from scratch"에서 컨텍스트 예측을 위해 훈련되었습니다.
그러나 인코더 아키텍처가 컨볼루션 레이어로만 제한된 경우 특징 맵의 한 모서리에서 다른 모서리로 직접 정보가 전파되지 않습니다. 왜냐하면 컨볼루션 레이어는 모든 특징 맵을 함께 연결하지만 특정 특징 맵 내의 모든 위치를 직접 연결하지 않기 때문입니다. 이 정보 전파는 주로 완전 연결된 레이어나 inner product 레이어에서 처리됩니다. 이 아키텍처에서는 인코더와 디코더의 잠재적인 특징 차원이 각각 6 × 6 × 256 = 9216임을 언급하고 있습니다. 오토 인코더와 달리 컨텍스트 인코더는 원본 입력을 재구성하지 않기 때문에 더 작은 병목을 갖을 필요가 없습니다. 그러나 인코더와 디코더를

완전히 연결하면 매우 많은 파라미터가 발생하여 현재의 GPU에서 효율적인 훈련이 어렵게 될 것입니다. 이 문제를 완화하기 위해 채널별 fully-connected 레이어를 사용하여 인코더의 특징을 디코더에 연결합니다.
Convolution layers가 서로 fully conntected 되어 있으므로 가중치를 모두 공유하는 형태가 됩니다. feature map은 말 그대로 특징만 추출했고 모든 위치를 직접 연결하지는 않기에 정보가 feature map의 한 모서리에서 다른 모서리로 전파할 수 있는 방법이 없습니다. 따라서 이 모든 정보 전파를 fully connected로 처리합니다.
일반적인 방식으로 Encoder와 Decoder를 완전히 연결하면 parameter의 수가 훈련이 어려울 정도로 증가할 수 있습니다. 이에 채널 방식의 Fully Conneted layer를 구성하여 Decoder에 연결해줍니다. 채널 방식의 Fully conneted layer는 입력 레이어에 크기가 n*n인 feature map이 m개 있으면 n*n 차원의 m개 feature map을 출력하여 Decoder에 전달합니다. 이 때, 각 feature map 마다 활성함수를 적용하며 전달하게 됩니다. fully connected Layer와의 차이점은 파라미터 개수가 mn^4(mn 제곱의 4)이며 feature map을 연결하는 파라미터가 존재하지 않고 feature map 내에서만 정보를 전파한다는 것입니다.

디코더는 파이프라인 후반부로 전달받은 특징 맵을 사용하여 이미지의 픽셀을 생성합니다. 디코더는 RELU 활성화를 거친 5개의 up-convolution layer로 이루어져 있습니다. up-convolution layer는 고해상도 이미지를 생성하는 역할을 합니다.



(a) Central region (b) Random block (c) Random region

다음은 이미지 영역 제거를 위한 세가지 전략에 대해 알아보겠습니다. 컨텍스트 인코더의 입력은 하나 이상의 영역이 제거된 이미지입니다. 여기서 제거는 해당 영역을 제로로 설정하는 것을 가정합니다. 제거된 영역은 어떤 모양이든 될 수 있으며, 이 논문에서는 세 가지 다른 전략이 제시되어 있습니다.

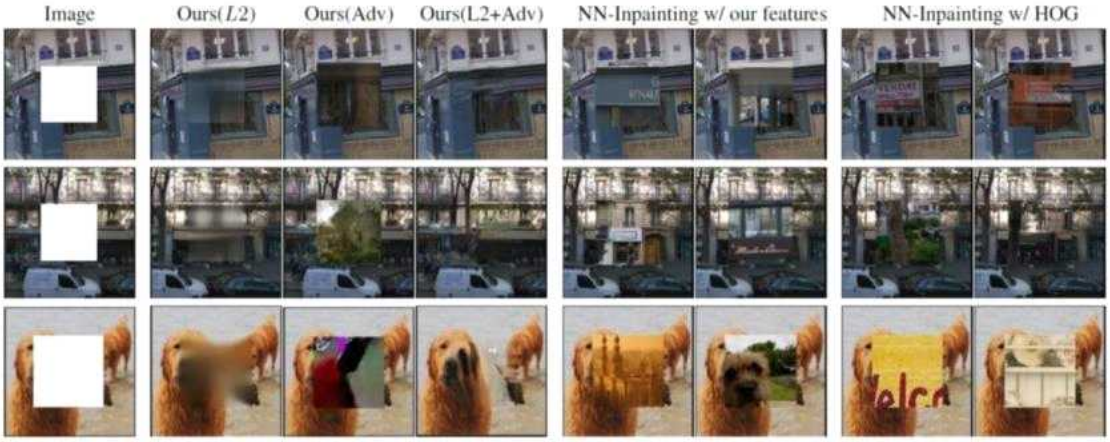
Central Region은 가장 간단한 형태인 이미지의 중앙에 있는 정사각형 패치입니다. 이는 inpainting에 잘 작동합니다. 그러나 네트워크는 중앙 마스크의 경계에 끌리는 낮은 수준의 이미지 특징을 학습합니다. 이러한 낮은 수준의 이미지 특징은 일반적으로 마스크가 없는 이미지에 일반화되지 않으며, 따라서 학습된 특징은 매우 일반적이지 않습니다.

Random Block은 네트워크가 마스킹된 영역의 일정한 경계에 끌리는 것을 방지하기 위해 마스킹 프로세스를 랜덤화합니다. 고정된 위치에서 단일 대형 마스크를 선택하는 대신 여러 작은 중첩 마스크를 제거하여 이미지의 최대 1/4를 커버합니다. 그러나 Random Block 마스킹은 여전히 날카로운 경계의 특징에 의해서 저수준의 특징밖에 찾지 못합니다.

마지막으로 Random Region은 이미지로부터 임의의 모양이 제거된 형태입니다. 앞서 Central Region과 Random Block이 유사한 일반적인 특징을 찾아내는 것보다 훨씬 좋은 특징을 잘 찾습니다. 따라서 Random Region

dropout 은 특징을 찾는 용도로 많이 사용됩니다.

5. 결과



다음은 제안한 방법론의 실험 결과를 살펴보겠습니다.
Semantic Inpainting은 이미지의 일부 영역이 누락되거나 손상된 경우, 해당 영역을 예측하고 복원하는 작업을 말합니다. 먼저 학습된 모델이 공백을 얼마나 잘 생성하는지 살펴보겠습니다.

봐야 할 포인트는 L2 / Adv / L2+Adv의 비교입니다. L2로 학습한 모델이 생성한 부분은 대충 맞긴 맞는데 Blurry한 모습입니다. 반면 Adv로 학습한 모델이 생성한 부분은 진하고 실제 같긴 한데, 문맥에 맞지 않는 생뚱 맞은 그림을 만들어낸 모습입니다. 마지막으로 이 둘을 결합했을 때 비로소 그럴듯한 이미지를 생성하는 모습을 볼 수 있습니다.

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	78.2%	56.8%	48.0%
Random Gaussian	initialization	< 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Wang *et al.* [39]	motion	1 week	58.7%	47.4%	-
Doersch *et al.* [7]	relative context	4 weeks	55.3%	46.6%	-
Ours	context	14 hours	56.5%	44.5%	30.0%

다음으로는 제안한 방법으로 Pretraining한 뒤 Transferability Test 성능을 살펴보겠습니다.
주어진 정보에 따르면, 실험에서는 Classification), Detection, Semantic Segmentation에 대한 자기 지도 학습 방식의 성능이 기존 다른 방식들에 비

해 낮게 나타났다는 것을 확인할 수 있습니다. 이러한 결과는 여러 측면에서 해석될 수 가 있는데요.
자기 지도 학습 방식은 지도 학습 방식과 비교했을 때 성능이 낮게 나타났다고 설명되어 있습니다. 이는 현재 실험에서 사용된 자기 지도 학습 방식이 다른 전통적인 지도 학습 방식에 비해 효과적이지 않다는 것을 나타낼 수 있습니다. 하지만 자기 지도 학습은 상대적으로 최근에 주목을 받는 기술 중 하나이며, 지속적인 발전과 연구가 이루어지고 있다고 합니다. 성능이 다른 방식에 비해 낮게 나왔다 하더라도, 앞으로의 연구에서 더 나은 성능을 달성할 수 있는 가능성이 있습니다.
Inpainting을 사용한 자기 지도 학습은 참신한 방법으로 소개되었고, 이것이 실험에서 성능이 낮게 나왔더라도 새로운 시각에서의 실험 및 학습 방법을 제시한 의미가 있습니다. 실험 결과를 통해 Inpainting이 자기 지도 학습에 적용될 수 있다는 가능성이 제시되었을 것입니다.
최종적으로, 실험 결과는 그 자체로 중요한 통찰력을 제공하며, 더 나은 성능을 추구하기 위한 향후 연구의 기반으로 활용될 수 있다고 생각됩니다.

마지막으로 이 논문은 자기 지도 학습과 인페인팅을 결합하여 딥 러닝 모델이 특성을 효과적으로 학습하는 방법을 제시하는 중요한 논문입니다. 이를 통해 레이블이 제한된 상황에서도 의미 있는 특성을 추출할 수 있으며, 이는 다양한 응용 분야에서 유용하게 활용될 수 있다고 생각됩니다.