

# 신경망의 처리과정에 대한 해석의 필요성

- 신경망의 구조와 학습에 대한 일반론적 설명은 가능
- 신경망이 이미지를 처리하는 구체적인 방식은 여전히 블랙박스
- 신경망의 이미지 처리에 대해 구체적 이해가 가능하다면 모형 개선에 도움
- 인공지능의 설명가능성(explainability)가 현재 주요한 이슈
- 사업적, 정책적 의사결정에도 중요

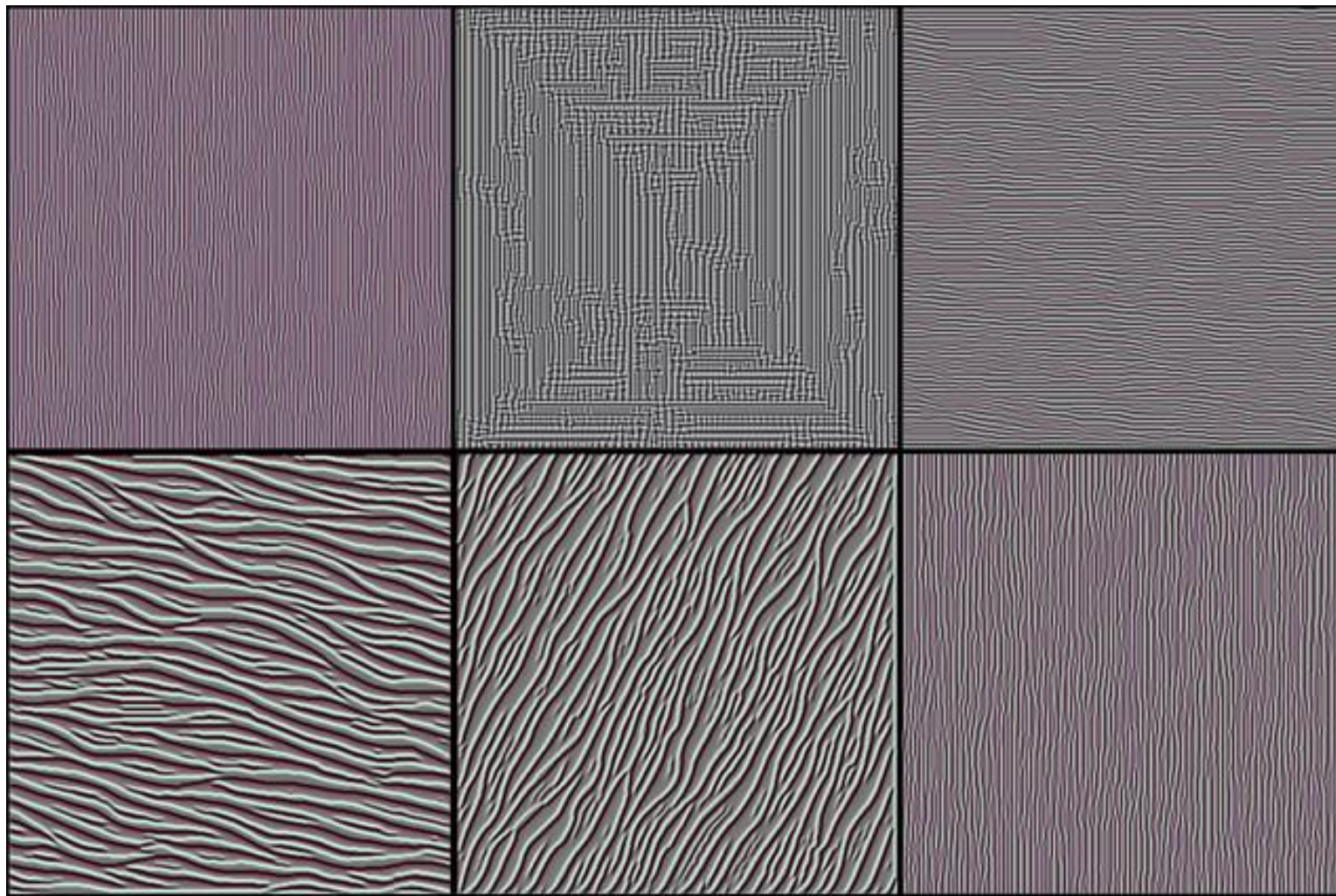
# CNN 레이어의 시각화

- CNN의 각 레이어는 특징(feature)을 추출
- 단순한 특징에서 시작하여 점점 더 복잡한 특징을 추출
- 각 레이어에서 어떠한 특징을 추출하는지 시각화

# 경사상승법 gradient ascent

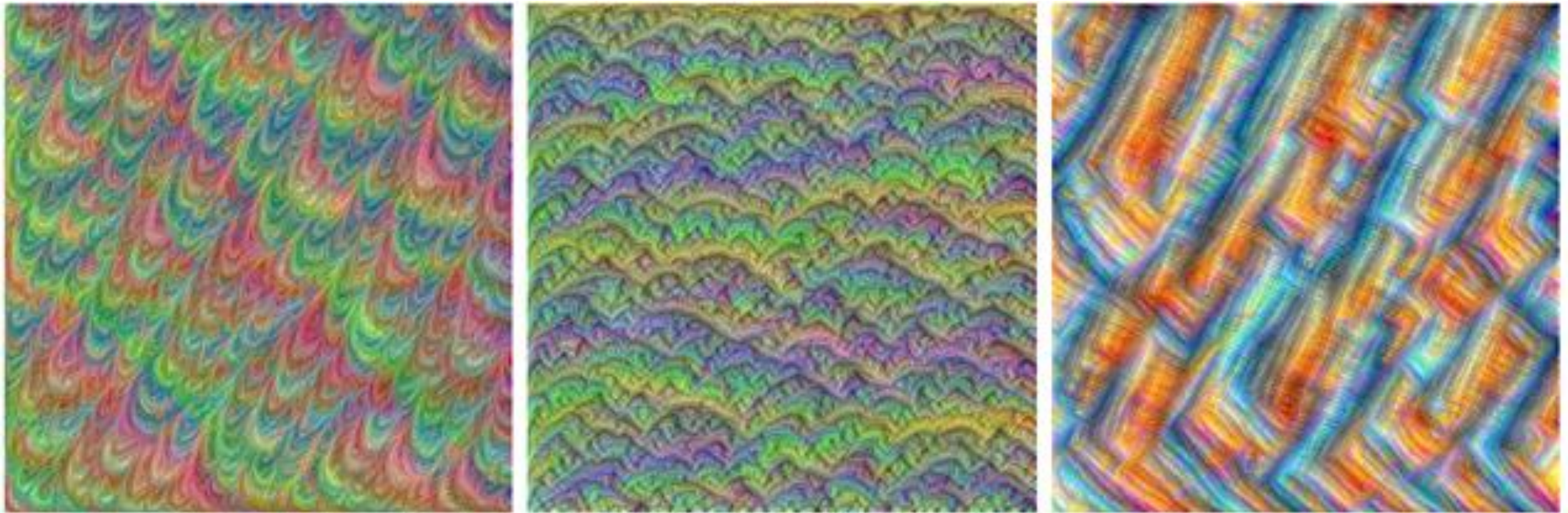
- 경사상승법을 통해 특정 레이어를 가장 많이 활성화시키는 이미지를 만들어 내면 그 레이어가 추출하는 특징을 시각화 가능
- 경사상승법: 목표함수의 값을 극대화시키는 방향으로 경사를 따라 이동

# VGG16의 block1\_conv1



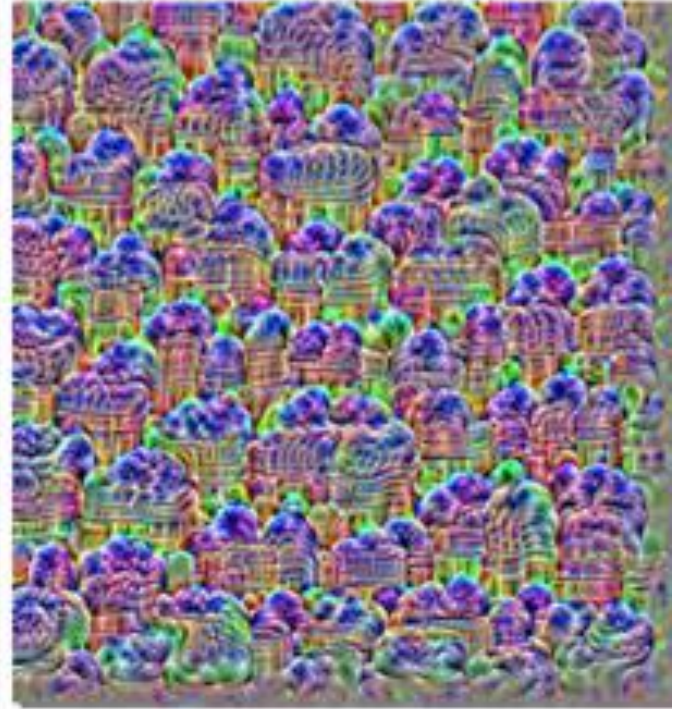
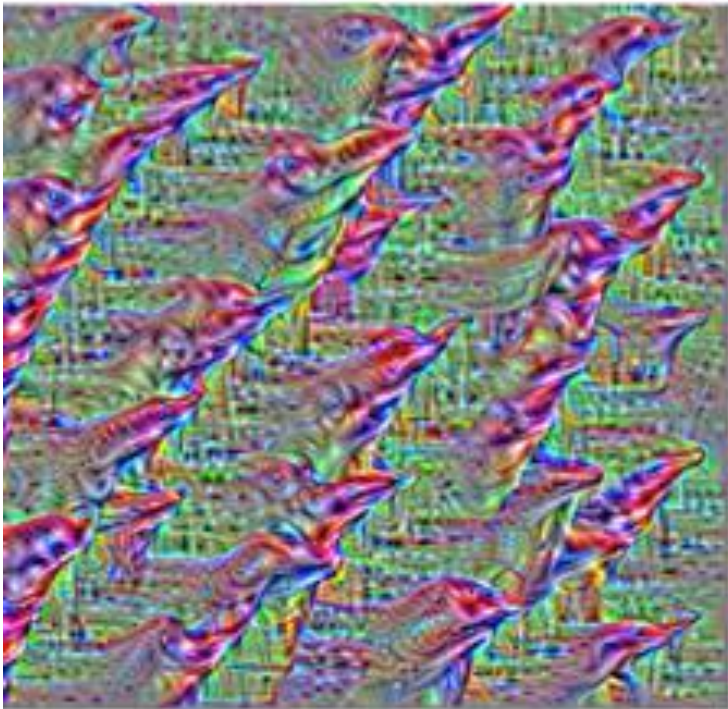


# block3\_conv2





# block5\_conv3



# 적대적 사례 adversarial example

- 앞의 시각화에서 새의 부리와 머리 모양에 반응하는 것을 볼 수 있음
- 만약 새의 부리와 머리 부분을 가렸을 때도 새를 새로 인식할 수 있을까?
- 적대적 사례: 신경망의 처리 능력을 떨어트리기 위해 의도적으로 고안된 사례

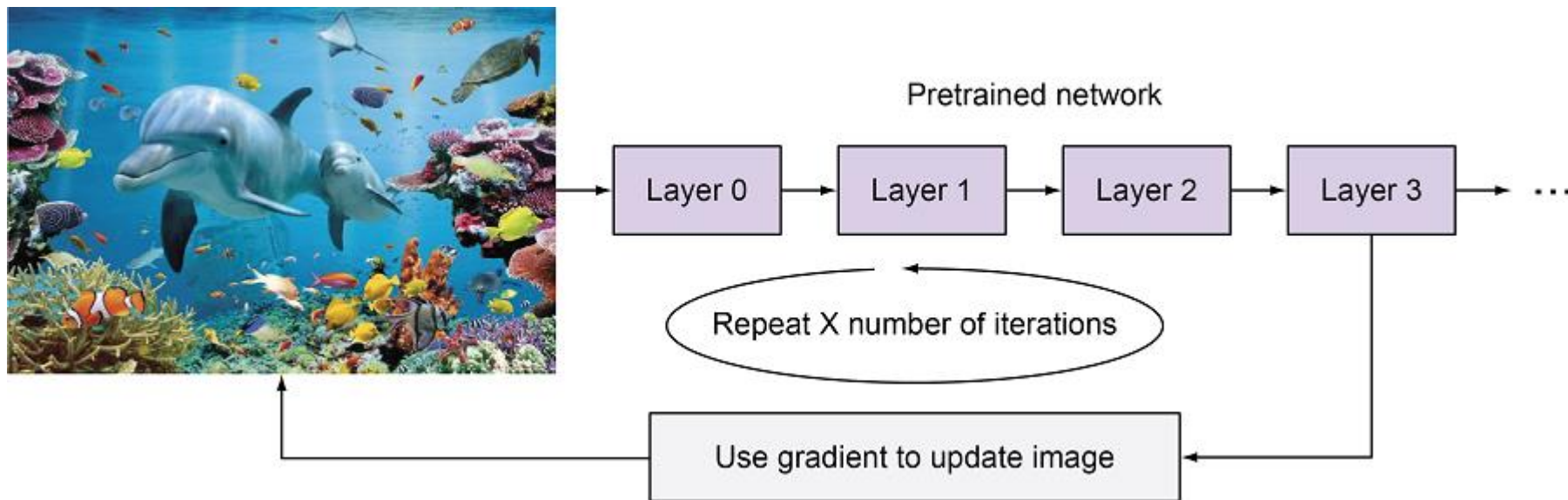


# 딥드림 DeepDream





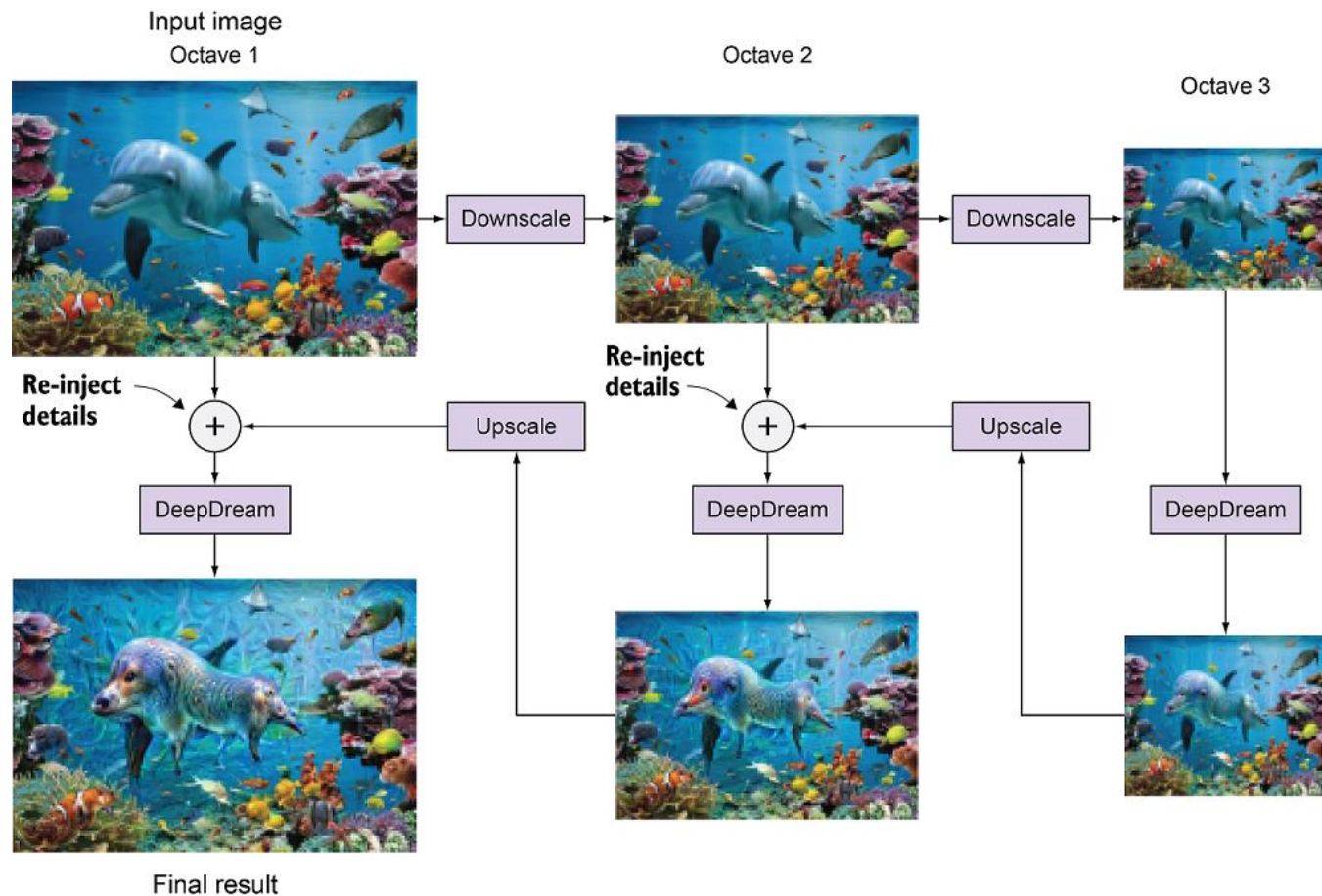
# 딥드림 알고리즘



# 옥타브 Octaves

- DeepDream은 사전 학습된 신경망을 활용
- 대부분의 사전 학습된 신경망은 ImageNet 등 작은 크기의 데이터에 학습
- DeepDream의 입력 이미지는 크기가 큼
- 이 문제를 해결하기 위해 옥타브(octave)라는 방법을 사용

# 옥타브를 사용한 딥드림





# Style Transfer



# Style Transfer의 손실 함수

- 내용 손실 *content loss*: 입력된 이미지와 합성된 이미지의 내용의 차이
- 스타일 손실 *style loss*: 입력된 이미지와 합성된 이미지의 스타일 차이
- 잡음 손실 *noise loss*: 합성된 이미지에 포함된 잡음
- Style Transfer의 총 손실 = 스타일 손실 + 잡음 손실 - 내용 손실

# 내용 손실

- 이미지의 내용은 신경망의 후반부에서 처리
- 신경망의 후반부 한 레이어의 활성화 패턴의 차이로부터 내용 손실을 계산
- 차이는 MSE로 계산



# 스타일 손실

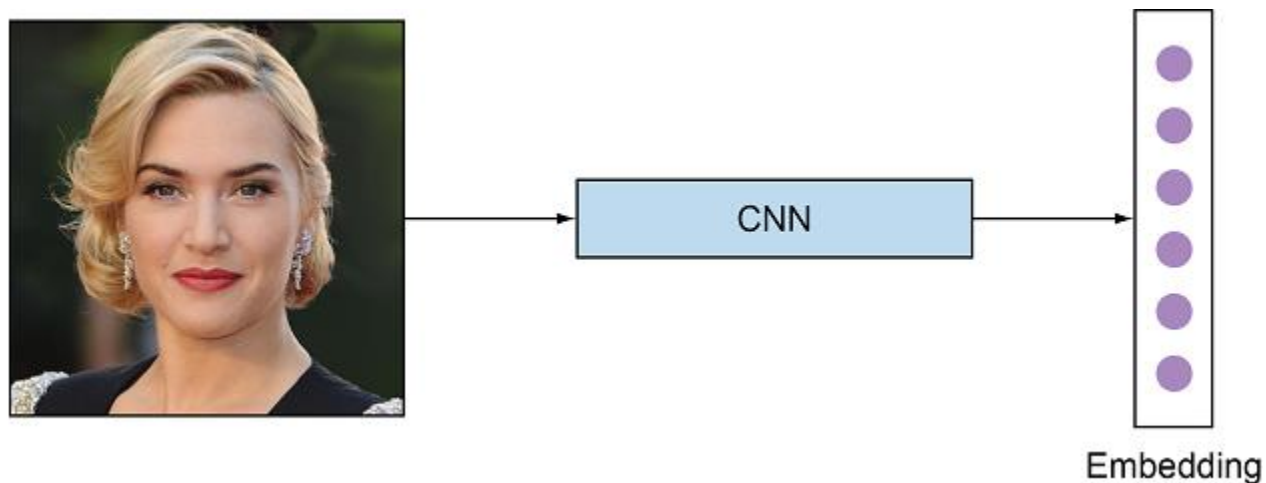
- 내용 손실은 한 레이어에서 계산하나 스타일 손실은 여러 레이어에서 계산
- 그람 행렬 Gram matrix를 이용해 손실을 계산

# 잡음 손실

- 이미지를 오른쪽으로 한 픽셀 이동시켰을 때의 픽셀 간 차이
- 이미지를 아래로 한 픽셀 이동시켰을 때의 픽셀 간 차이

# 임베딩 embedding

- 유사성을 보존하면서 원래보다 낮은 차원의 벡터로 표현하는 것
- 또는 그렇게 표현된 벡터





# 임베딩의 응용 (1)

- Face identification
  - DB에 저장된 얼굴 이미지들 중에서 찾기
  - 예: 출입 시스템
- Face verification
  - 특정 얼굴과 일치 여부를 확인
  - 예: 얼굴로 잠금 해제

## 임베딩의 응용 (2)

- Image recommendation systems
  - 검색 또는 구매한 상품과 시각적으로 비슷한 상품을 추천
- object re-identification
  - CCTV 등에서 특정 대상을 추적할 경우 서로 다른 이미지에서 동일 대상을 식별할 수 있어야 함

# 임베딩의 필요성

- HD 해상도  $1280 \times 720 = 92$ 만개의 픽셀
- 4K 해상도  $3840 \times 2160 = 829$ 만개의 픽셀
- 원본 이미지를 바탕으로 검색할 경우 하나의 이미지와 비교하기 위해 수 십만~수 백 만 개의 값을 계산 필요
- 고해상도 이미지에는 불필요한 디테일을 많이 포함



# 이미지를 단순히 축소하면?

- 단순히 크기만 줄이는 것으로는 불충분
- 이미지의 '의미'를 보존해야

# 임베딩 구현 방법

(1) crossentropy loss

(2) contrastive loss

(3) triplet loss

# crossentropy loss

- 이미지 분류 모형을 학습 (crossentropy를 최소화)
- 신경망의 마지막 예측 레이어를 제거
- 신경망의 특정한 레이어를 출력으로 사용
- 해당 레이어가 추출하는 특징을 임베딩으로 사용

# contrastive loss

- 이미지를 2개씩 짝지음
- 임베딩을 출력하는 모형에 두 이미지를 각각 입력
- 두 이미지가 같은 클래스인 경우 임베딩이 가깝게
- 두 이미지가 다른 클래스인 경우 임베딩이 멀어지게



# triplet loss

- contrastive loss의 확장
- $a$ ,  $p$ ,  $n$  3개의 이미지를 사용
- $a$ ,  $p$ 는 같은 대상의 이미지,  $n$ 은 다른 대상의 이미지
- $a$ 와  $p$ 의 출력은 가깝게,  $a$ 와  $n$ 의 출력은 멀도록 수정

# 적절한 삼중항 찾기

- $a$ 와  $n$ 이 비슷한 점이 있어야 학습이 효율적
- hard data mining: 신경망이 잘 구분하지 못하는 사례들에 대해서만 학습
- 문제점: 다른 사람인데 같은 사람으로 잘못 표시된 이미지와 바르게 표시된 이미지가 있을 경우, 전자에 더 많이 학습될 수 있음
- semi-hard data: 너무 쉽거나 너무 어렵지 않은 정도의 사례들