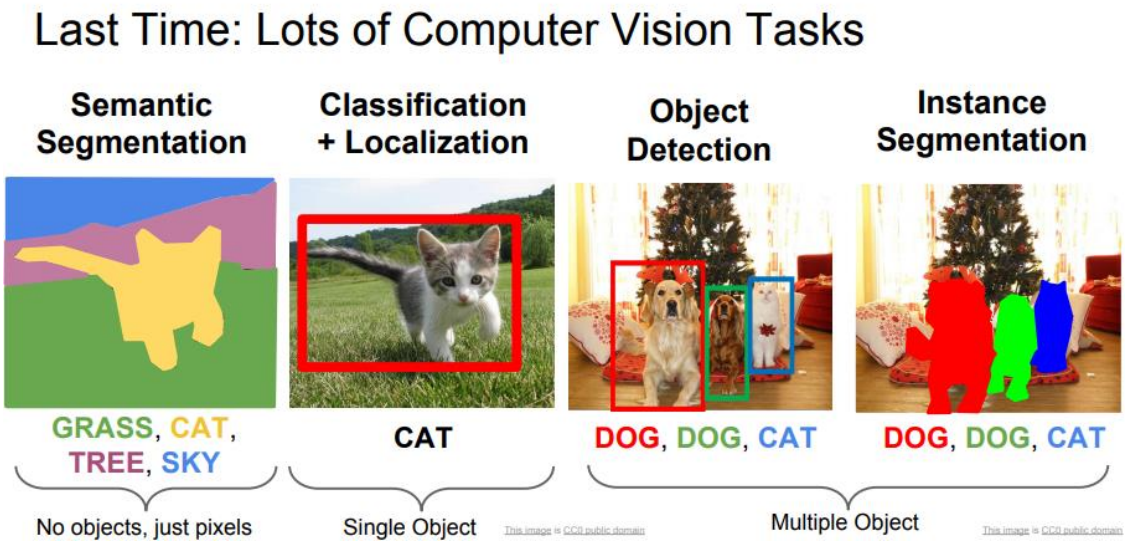
<영어 단어장>
administrative 관리상의, 행정상의
interpretable 이해가능한, 해석가능한
occulsion 폐색, 폐쇄, 맞물림

복습



2) classification + localization : 미리 정해둔 class, 고정된 수의 object 를 찾고 싶을때 (ex: pose recognition)
3) object detection : 미리 정해둔 class, 몇개의 object 를 찾을지는 모름 / R-CNN, Fast R-CNN, Faster R-CNN 을 이용
4) instance segmenation : semantic segmentation + object detection : 미리 정해둔 class, label the pixels beloning to each instance

12 강. visualizing and understanding

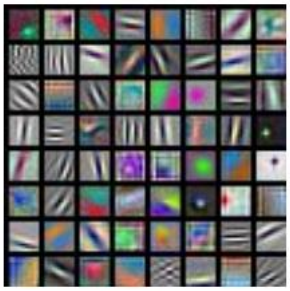
What's going on inside ConvNets?

CNN 이 비난? 받는 가장 큰 이유는 "Black Box Problem" 때문.
모델을 작성한 당사자도 CNN 안에서 어떤 과정을 거쳐서 학습이 되는지 그 내부 과정을 알 수 없다는 것.
따라서 12 강에서는 CNN 의 중관과정들을 좀 더 직관적으로 나타내는 방법들을 배운다.
그 중에서도 input, output 과 직접적인 연관이 있는 first & last layer 들 먼저 살펴보자.

First & Intermediate & Last

First Layer

First Layer: Visualize Filters

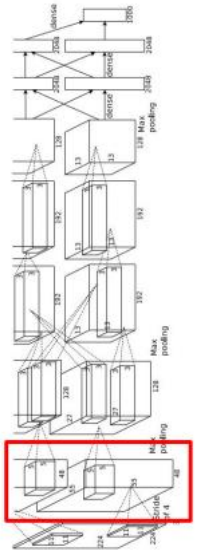


AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7

ResNet-101:
64 x 3 x 7 x 7

DenseNet-121:
64 x 3 x 7 x 7



Krizhevsky, "One weird trick for parallelizing convolutional neural networks", arXiv 2014
He et al, "Deep Residual Learning for Image Recognition", CVPR 2016
Huang et al, "Densely Connected Convolutional Networks", CVPR 2017

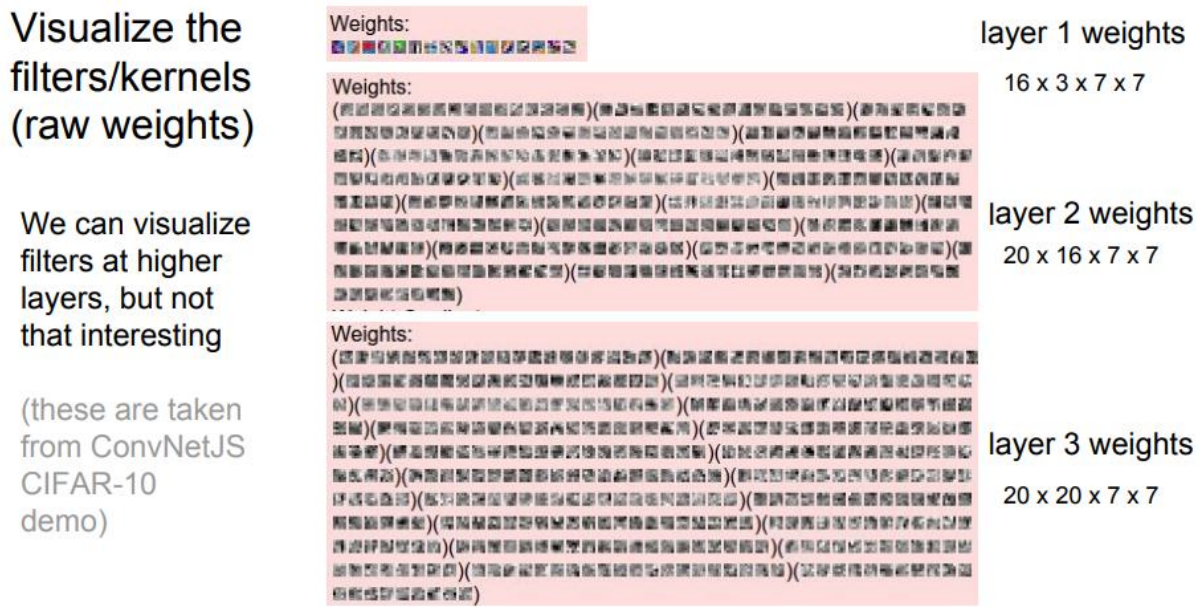
이미지의 pixel 이랑 weight of conv layer 를 바로 inner product 하기에
filter 가 어떻게 생겼는지를
해당 filter 의 learned weights 를 visualize 해서 알 수 있다.

첫번째 사진에서 AlexNet 의 first layer 에서의 64 개의 filter 가 3*11*11 짜리 이미지로 visualize 됐다.

AlexNet 뿐만 아니라 다른 것들도 살펴보면,
항상 first layer 에서는 각 이미지를 보면 다양한 각도와 다양한 위치에 직선과 점들이 있는 모습.
1 강에서 배웠는데 human visual system 의 early layer 도 물건을 감지할때 oriented edge 로 감지했음.
그것과 유사한 것을 확인 가능.

Q. 왜 [filter 를 visualize 한 것] = [filter 가 무엇을 보는지] 인가?
A. filter 를 원본 이미지에 내적한다. 이때 filter 에서 값이 높은 부분이 output 에서도 값이 높음.

Intermediate CONV layer



중간 단계들도 visualize 할 수는 있지만,
이 예제에서도 볼 수 있듯 16*7*7 이라 이걸 바로 이미지화 할 수는 없음.
위의 사진에서는 16 개의 7*7 gray scale 이미지로 visulaize 했음.
이런 16 장의 이미지가 20 개 있어야함. 20*16*7*7 이니까.

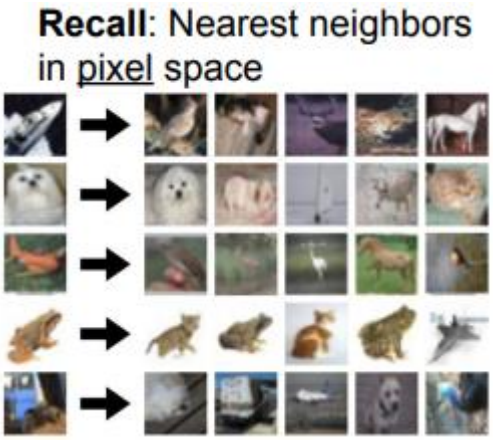
정리하자면, 16*3*7*7 은 16 장의 channel 3 으로 color 인 7*7 이미지로,
20*16*7*7 은 20 장 * (16 장 * channel 1 으로 7*7) 으로. 근데 별로 좋은 직관을 주진 못함.

Q. weight 값이 0~255 가 아니라면 어떻게 visualize?
A. 0~255 값으로 scale 해서 visualize.

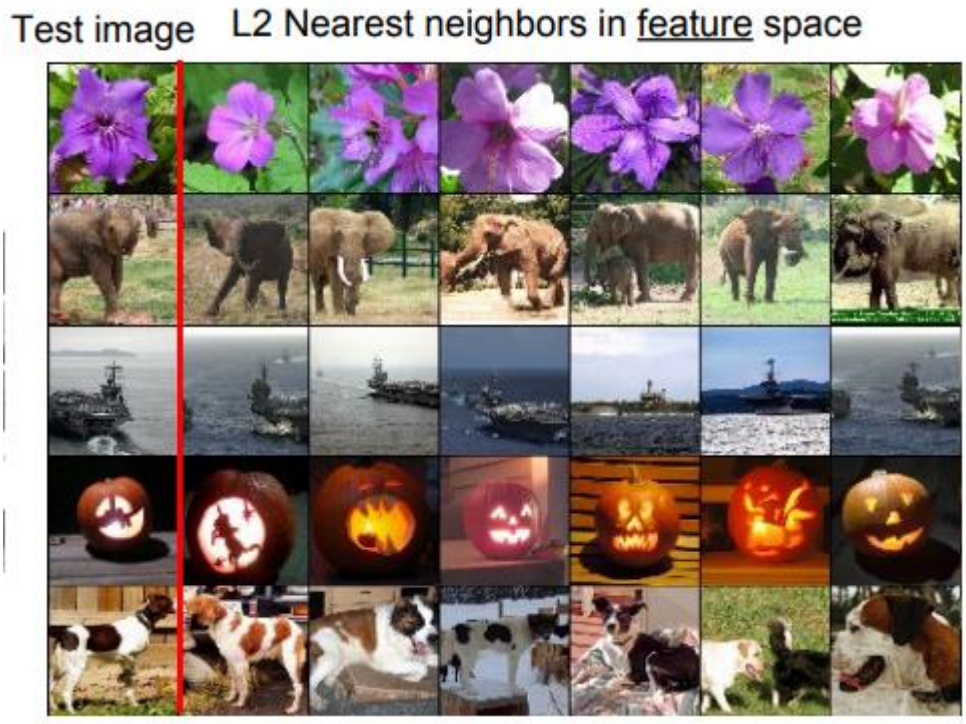
Last Layer - (1) Nearest Neighbor

1000 개의 class 가 있으면
1000 개의 score 가 있었고
그 앞에 4096*1000 개의 fully connected layer 가 있었음.

이렇게 input 이미지가 여러 layer 를 거치고 마지막에 4096-D feature space 가 나온다.
여기에 L2 Nearest Neighbor 를 적용시키기 위해서
각 input 이미지를 넣고 각 이미지에 대해 4096-D feature space 를 뽑아서
그것끼리의 NN 을 진행해서 비슷할 때
그 이미지들을 나열한게 아래 두번째 사진.



cf) pixel space 에 NN.



feature space 에 L2 Nearest Neighbor

pixel space 에 NN 한 것과 다름!
pixel 은 조금 다를 지언정 image 의 semantic content 가 동일함!
두번째 행의 코끼리를 보면, test image 는 오른쪽보는 코끼리인데 왼쪽보는 코끼리들도 결과로 나옴!
왼쪽 보는 코끼리 & 오른쪽 보는 코끼리는 pixel 은 완전 달라도 feature space 에서는 비슷하다고 판단된 것!
즉, last layer 에서는 image 의 semantic content 를 capture 한 것!

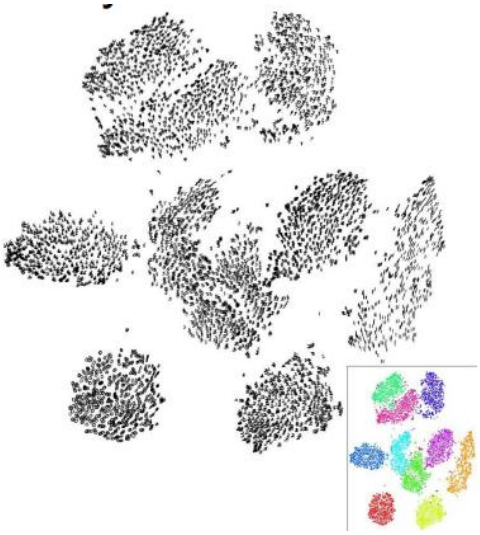
Last Layer - (2) Dimensionality reduction

마지막 layer 의 4096 vector 를 PCA(Principle Component Analysis)로 군집화하자.
PCA 는 차원축소의 일종으로
4096 차원이 vector 를 x,y 2 차원으로 축소해 좌표평면에 나타내어 visualize.

Last Layer - (3) Dimensionality reduction (t-SNE)

tSNE = t-distributed stochastic neighbor embedding
: 강력한 non-linear dimension reduction (차원 축소) 방법

: m-nest (손글씨 dataset)의 28*28 차원을 2 차원으로 축소하고 그것을 two-dimensional representation 으로 visualize.



large dataset 의 이미지를 넣고 돌리고, 그때마다 4096-D feature vector 를 기록해두고
거기에 t-SNE 차원축소를 진행. (4096D -> 2D)

grid 를 만들어두고 차원축소 결과에 해당되는 2D 좌표에 원래 이미지를 가져다가 붙여서 visualize 한게 오른쪽 사진.
가운데에 커다란 군집. 왼쪽밑에 초록색 군집...!

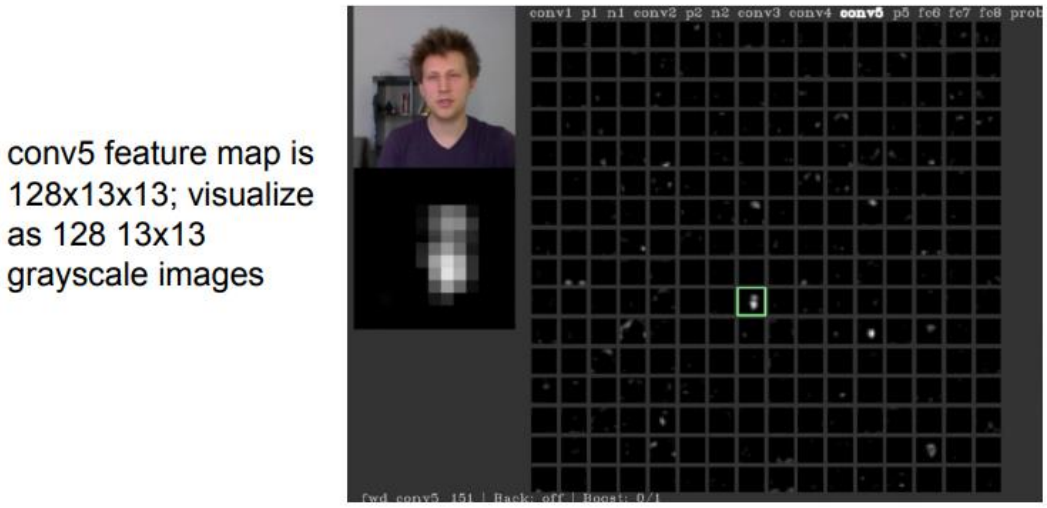
---

intermediate layer visualize 해봤자 직관적이지 않았음. 그럼 중간은 어떻게?!

1. Visualizing Activations

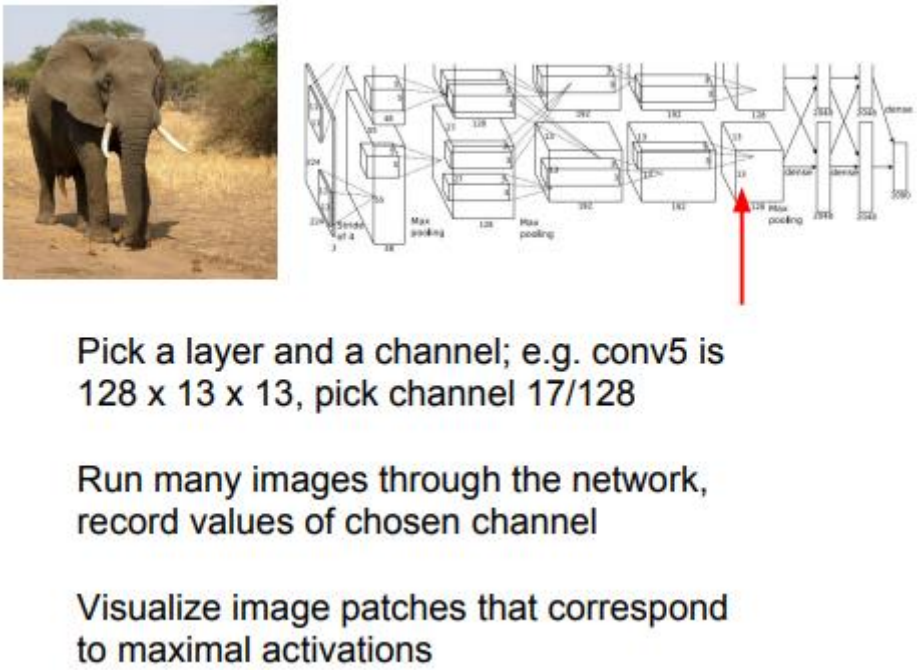
앞서 봤듯 intermediate layer 는 visualize 해봤자 직관적이지 않았음.

그런데 AlexNet 의 conv5 "activation map"에서 사람의 사진을 입력했을때 얼굴의 모양과 위치가 비슷한 image 확인 가능.



Q. image 가 K*K 이면 activation map 도 K*K 인가?
A. 항상은 아님. sub-sampling 도 하니까.

2. Visualizing by Maximally Activationg Patches



Visualizing what types of patches from input image.
Maximally Activation Patches 는 input 이미지의 어떤 patch(부분, crop)이 neuron 을 가장 활성화 시키는지를 확인하는 방법.
128 개 중에 channel 하나 고름. ex) 17 번째
그러면 13*13 feature map 이 나옴.

CONV 이므로 한 channel 에 있는 모든 뉴런은 같은 weight 를 이용.

각각은 이미지전체를 보는게 아니라 이미지의 subset 을 보고있음.

input 의 network 를 쭉 통과시키며 conv5 17 번째 layer 에서 가장 높은 값을 나타낸 위치를 찾고
그 지점에서 receptie field 들을 쭉 거슬러올라와서 input 을 나타내면 아래와 같음.

Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015;
reproduced with permission.

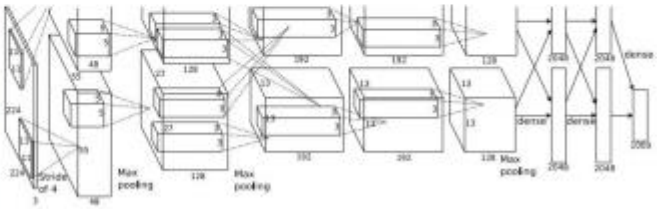
사진 1. conv5, 사진 2. 더 upper layer 라서 더 큰 object 들이 보임.

3. Occlusion Experiments

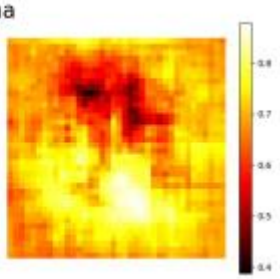
Occlusion Experiment 는 이미지의 어떤 부분을 가렸을때
예측 성능이 얼마나 줄어드는지를 heatmap 으로 나타낸 것.

왼쪽같이 코끼리 이미지에서
이마, 귀 부분을 가리면 예측 성능에 어떤 변화가 있을지
가리는 부분을 옮겨가면서 측정을 해서 heatmap 으로 나타내보면 다음과 같다.

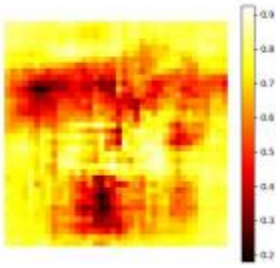
Mask part of the image before feeding to CNN, draw heatmap of probability at each mask location



African elephant, Loxodonta africana



go-kart



heatmap 에서 색깔이 진할 수록 예측확률이 떨어지는것을 의미.
따라서 진한 부분일 수록 예측에 critical 한 역할을 한다는 것.

29:32 이어서...