

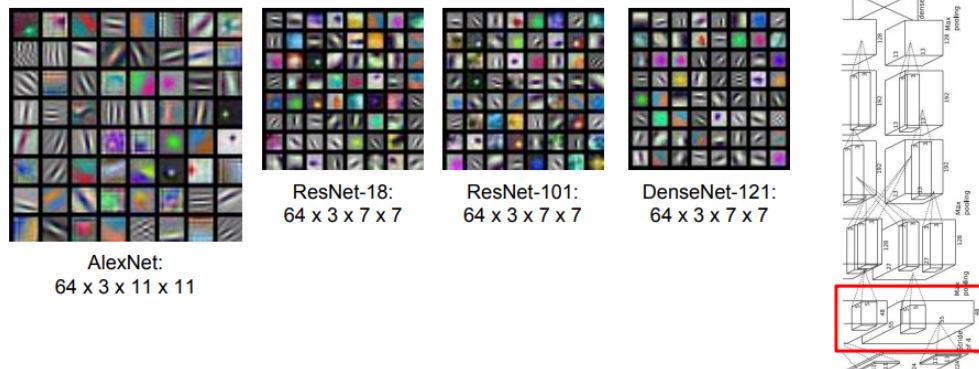
Visualizing이 중요한 이유

- 딥러닝이 잘 작동하는 이유를 시각화하여 설명하기 위해
- 딥러닝 내부 동작을 보여줌으로써 해석 가능한 부분이 있다는 것을 알려주기 위해

ConvNet 내부에서 어떤 일이 일어나고 있는지 시각화

First layer의 filter를 시각화

First Layer: Visualize Filters



First Layer는 Image raw pixel에 W 를 내적해서 feature map을 생성, W 는 $64 \times 3 \times 11 \times 11$ 사이즈.

W vector를 64개의 11×11 RGB 이미지로 변환하면 위와 같은 결과 나옴.

first layer의 Weight가 Edge와 Corner를 찾음.

Visualize the filters/kernels (raw weights)

We can visualize filters at higher layers, but not that interesting

(these are taken from ConvNetJS CIFAR-10 demo)



layer가 더 깊어지면서 합성곱이 이뤄지고 더 복잡해짐. => 직관적으로 해석할 수 있는 부분 적어짐.(두 번째 layer의 결과를 보면 첫 번째 layer의 결과를 input으로 넣고 필터 weight들의 내적을 시킨 값이기 때문)

Last Layer - Nearest Neighbor

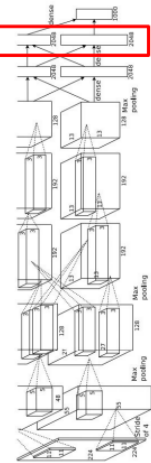
Last Layer: Nearest Neighbors

4096-dim vector

Recall: Nearest neighbors
in pixel space



Test image L2 Nearest neighbors in feature space



한 이미지에 4096 차원의 특징 벡터 존재.

각 이미지마다 특징 벡터들을 Nearest Neighbor Algorithm을 돌리게 되면, 유사한 이미지들이 검출됨.

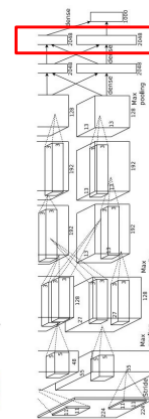
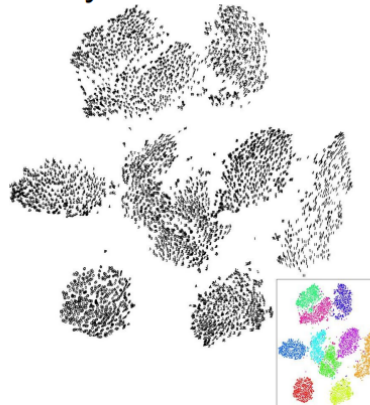
Last Layer - Dimensionality Reduction

Last Layer: Dimensionality Reduction

Visualize the “space” of FC7
feature vectors by reducing
dimensionality of vectors from
4096 to 2 dimensions

Simple algorithm: Principle
Component Analysis (PCA)

More complex: t-SNE

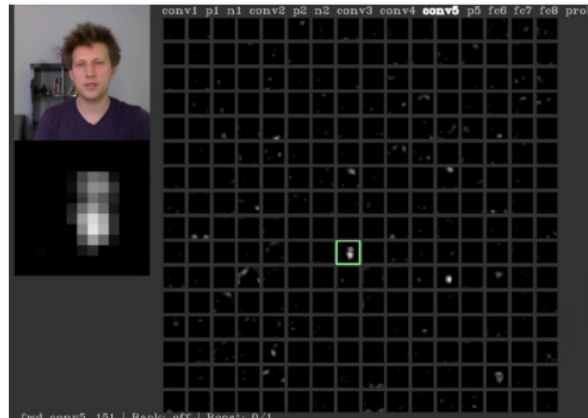


Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008.
Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

마지막 layer의 4096 vector를 PCA로 군집화. 같은 class들의 이미지들끼리 가깝게 분포함을 확인할 수 있음.

Visualizing Activation

conv5 feature map is 128x13x13; visualize as 128 13x13 grayscale images

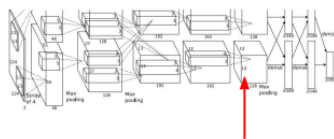


intermediate layer들은 input에 직접적으로 연결되어있는게 아니기 때문에 사람이 해석하기 쉽지 않다고 했음. 그러나 Activation Map을 통과시킨 Layer의 모습을 시각화하기 되면 사람의 얼굴이 있는 부분이 활성화된 layer가 존재한다는 것을 확인.

Maximally Activating Patches

: input 이미지의 어떤 patch가 neuron을 가장 활성화시키는지 확인하는 방법

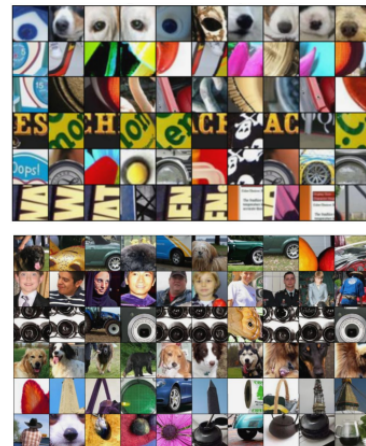
Maximally Activating Patches



Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17/128

Run many images through the network, record values of chosen channel

Visualize image patches that correspond to maximal activations



conv layer는 128x13x13의 feature map을 output.

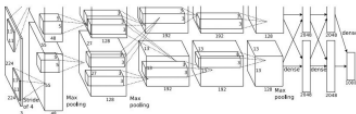
이중 임의로 하나의 채널을 고르면 13x13의 feature map이 나옴. input에서부터 network를 통과시키며 conv의 layer에서 가장 높은 값을 나타낸 위치를 찾고, 그 지점에서부터 receptive field를 거슬러 올라 input에서 나타난 결과.


Occlusion Experiments

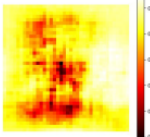
: 이미지의 어떤 부분을 가렸을 때 예측 성능이 얼마나 줄어드는지 heatmap으로 나타낸 것

Occlusion Experiments


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

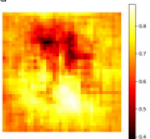



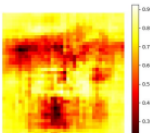

schooner



African elephant, *Loxodonta africana*


go-kart

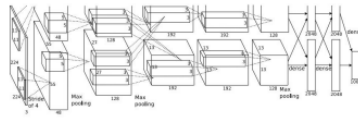
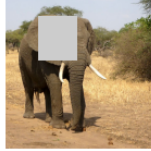


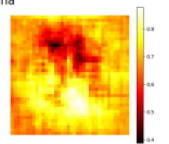
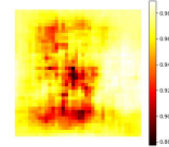
[Boat image](#) is [CC0 public domain](#)
[Elephant image](#) is [CC0 public domain](#)
[Go-Karts image](#) is [CC0 public domain](#)

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

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



schooner

A three-masted schooner with white sails is sailing on a body of water under a clear blue sky. The ship is viewed from a side-on perspective, showing its full length and the arrangement of its sails.

heatmap에서 색이 진할수록 예측 확률이 떨어짐. 진한 부분일수록 예측에 critical

Saliency Maps

: 이미지의 어떤 **pixel**이 **classification**에 영향을 줬는지 체크
 각 픽셀별로 결과에 얼마나 영향을 끼치는지 **Gradient Descent** 방식으로 접근해 영향력이 큰 **pixel**을 찾아냄.

Intermediate Features via guided BackProp

: 중간의 뉴런을 골라 이미지의 어떤 **patch**가 영향을 크게 줬는지 확인. **patch** 내부에서도 어떤 픽셀이 크게 영향을 줬는지 알 수 있음.

Visualizing CNN features : Gradient Ascent

: 어떤 **weight**이 주어졌을 때 그 **neuron**을 활성화시키는 **generalized**한 **image**를 알아보기 위해 **gradient ascent** 진행

Gradient Ascent : loss가 최대가 되는 parameter 찾는 방법

고정된 W 에 대해 input image의 pixel value를 gradient ascent로 바꿔가면서 neuron이나 class score를 극대화.

Fooling Images / Adversarial Examples

: 하나의 class가 다른 class로 classify되도록 image를 계속 update할 경우 노이즈 발생

DeepDream : Amplify existing features

: feature들이 무엇을 찾는지 대략적으로 파악, neuron activation 증가

Feature Inversion

: CNN에서 구한 feature들만으로 input 이미지를 생성.

input x^* 는 주어진 feature와 새롭게 생성할 이미지의 feature간의 간극을 최소화 시키는 방향으로 gradient ascent