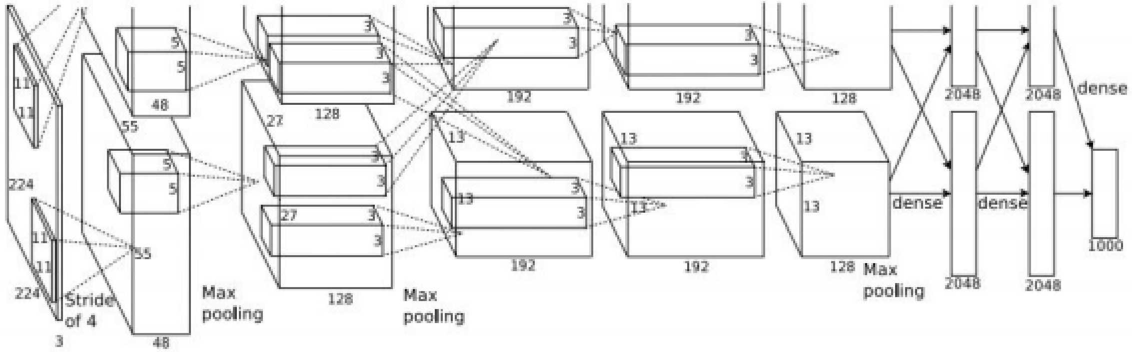
CS231N 12강
Visualizing and Understanding

This image is CC0 public domain

Input Image:
3 x 224 x 224



Class Scores:
1000 numbers

What are the intermediate features looking for?

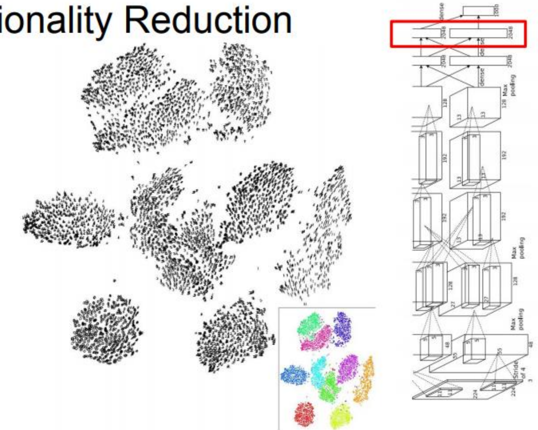
First layer: Image raw pixel에 W를 내적하여 feature map 생성
Last layer: 보통 FC-layer로 dimension을 class 개수로 축소해줌

Last Layer: Dimensionality Reduction

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

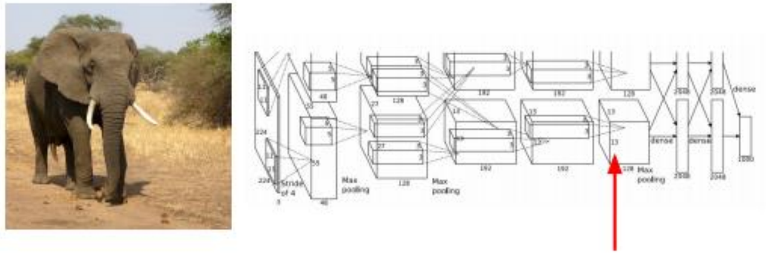
Simple algorithm: Principle Component Analysis (PCA)

More complex: **t-SNE**



Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008

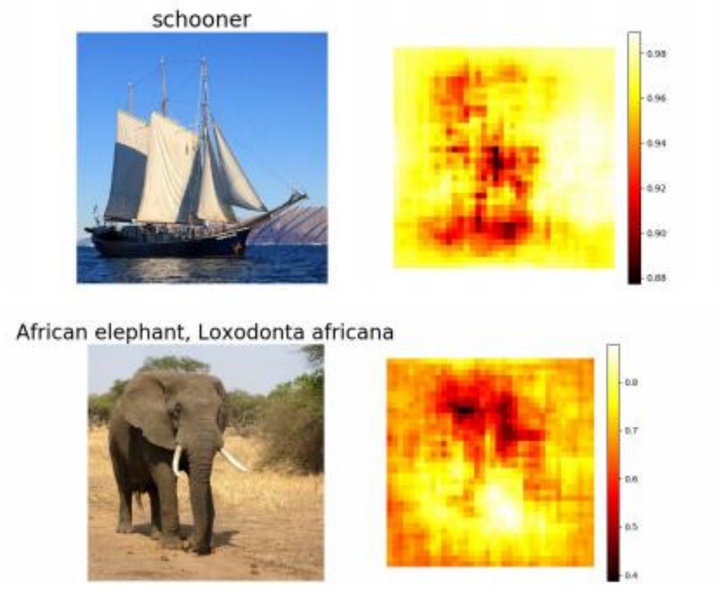
Maximally Activating Patches



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

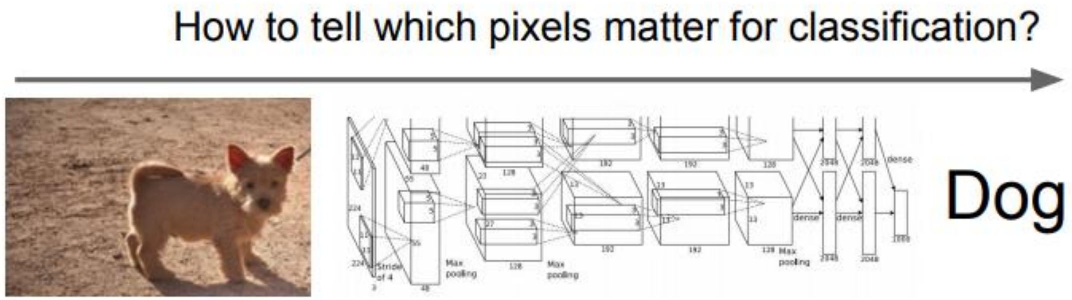
Visualize image patches that correspond to maximal activations



Maximally Activating Patches: input image의 어떤 부분(patch)이 뉴런을 활성화 시키는지 확인하는 방법

Occlusion Experiment: input image의 어떤 부분을 가렸을때 예측성능이 얼마나 줄어드는지 heatmap으로 나타낸것
*heatmap에서 색이 진할수록 예측확률이 떨어지는것으로, 예측에 중요한 부분임.

Saliency Maps



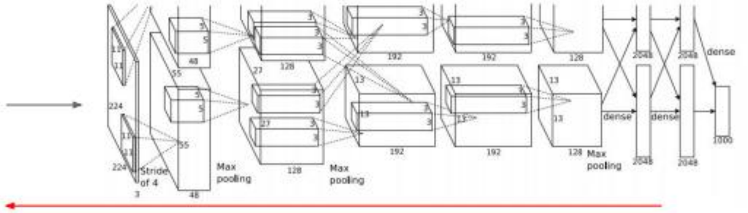
Saliency Maps: input image의 어떤 pixel이 classification에 영향을 주었는지 확인,
각 pixel에 gradient descent방식으로 접근

Visualizing CNN features: Gradient Ascent

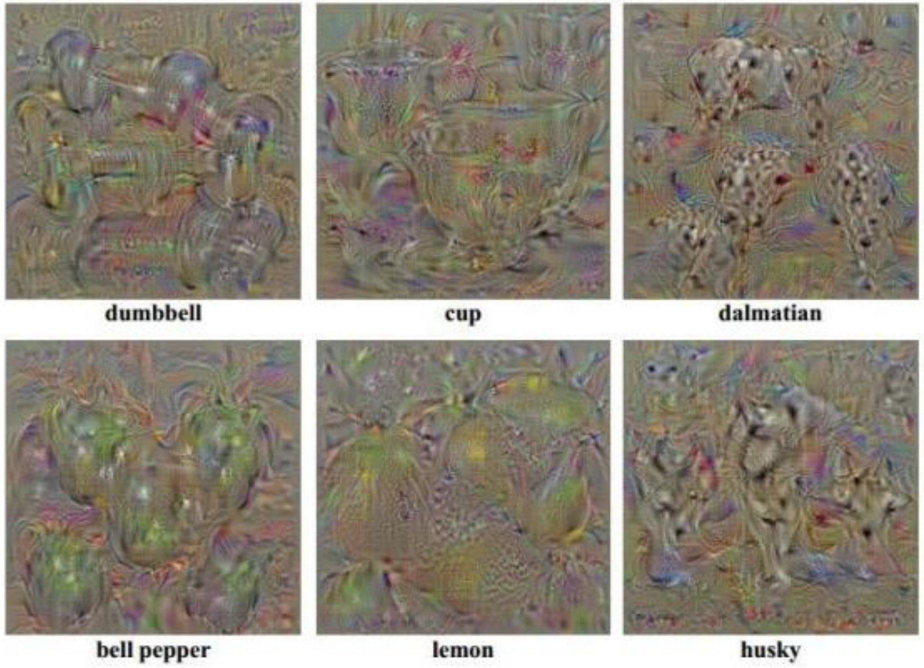
$$\arg\max_{I} \boxed{S_c(I)} - \lambda\|I\|_2^2$$

score for class c (before Softmax)

1. Initialize image to zeros



Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

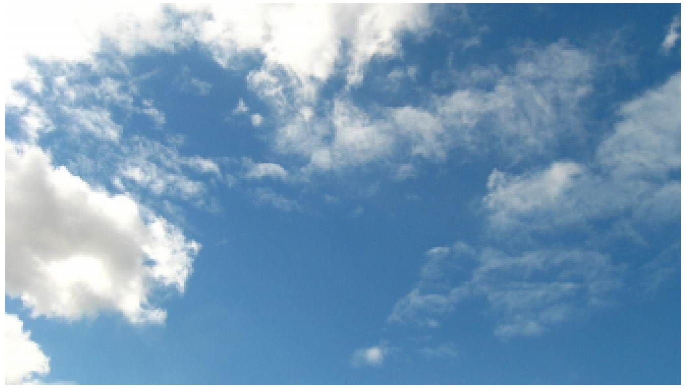


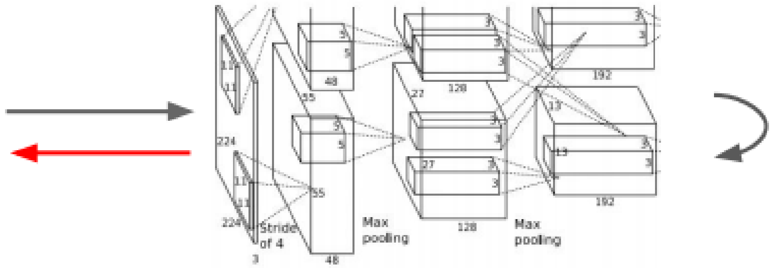
Gradient Ascent: Loss가 최대가 되는 Parameter 찾는 방식
Regularization: Gaussian Blur Inage, Clip pixels with small values to 0, small gradient to 0

Deep Dream: neuron activations를 증가시키는 방향으로 시각화

DeepDream: Amplify existing features

Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network





Choose an image and a layer in a CNN; repeat:

1. Forward: compute activations at chosen layer
2. Set gradient of chosen layer *equal to its activation*
3. Backward: Compute gradient on image
4. Update image

Feature Inversion: CNN에서 구한 feature들만으로 통해 역으로 input이미지를 생성하는 방법

Feature Inversion

Given a CNN feature vector for an image, find a new image that:

- Matches the given feature vector
- “looks natural” (image prior regularization)

Given feature vector

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^{H\times W\times C}}{\operatorname{argmin}} \ \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda\mathcal{R}(\mathbf{x})$$

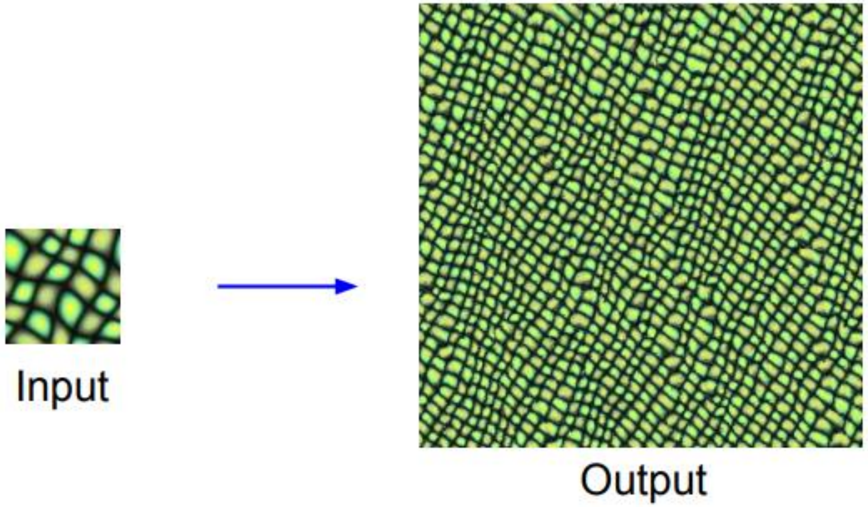
Features of new image

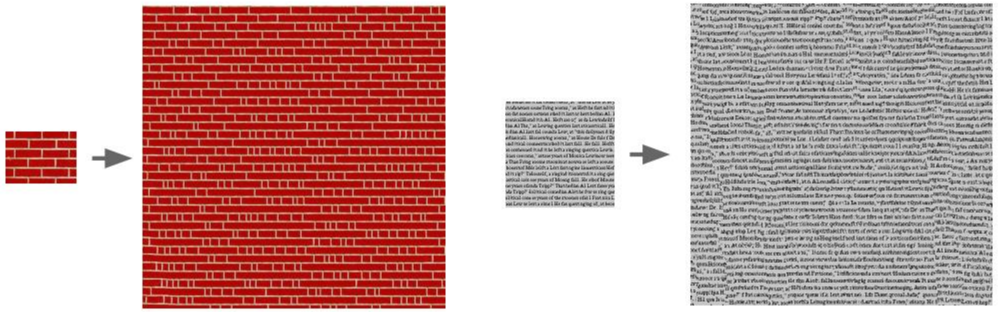
$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j}\left(\left(x_{i,j+1} - x_{ij}\right)^2 + \left(x_{i+1,j} - x_{ij}\right)^2\right)^{\frac{\beta}{2}}$$

Total Variation regularizer
(encourages spatial smoothness)

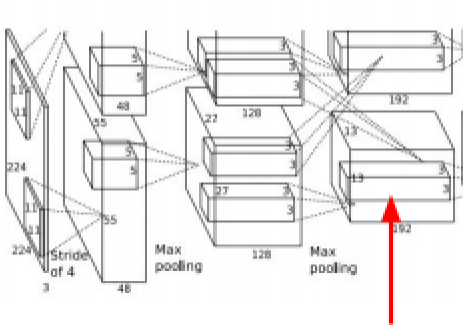
Texture Synthesis: 주어진 이미지를 같은 texture의 더 큰 이미지로 생성하는 방법
Nearest Neighbor 사용

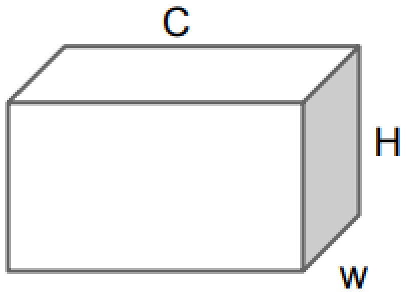


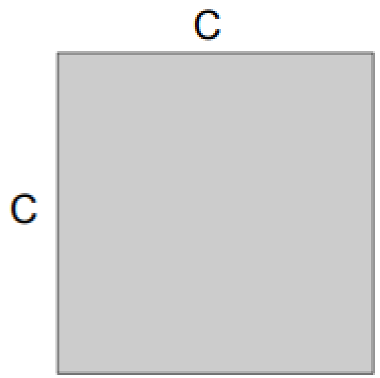


Gram Matrix: 서로 다른 공간 정보에 있는 Channel을 가지고 외적하여 새로운 Matrix를 만드는것

Neural Texture Synthesis: Gram Matrix







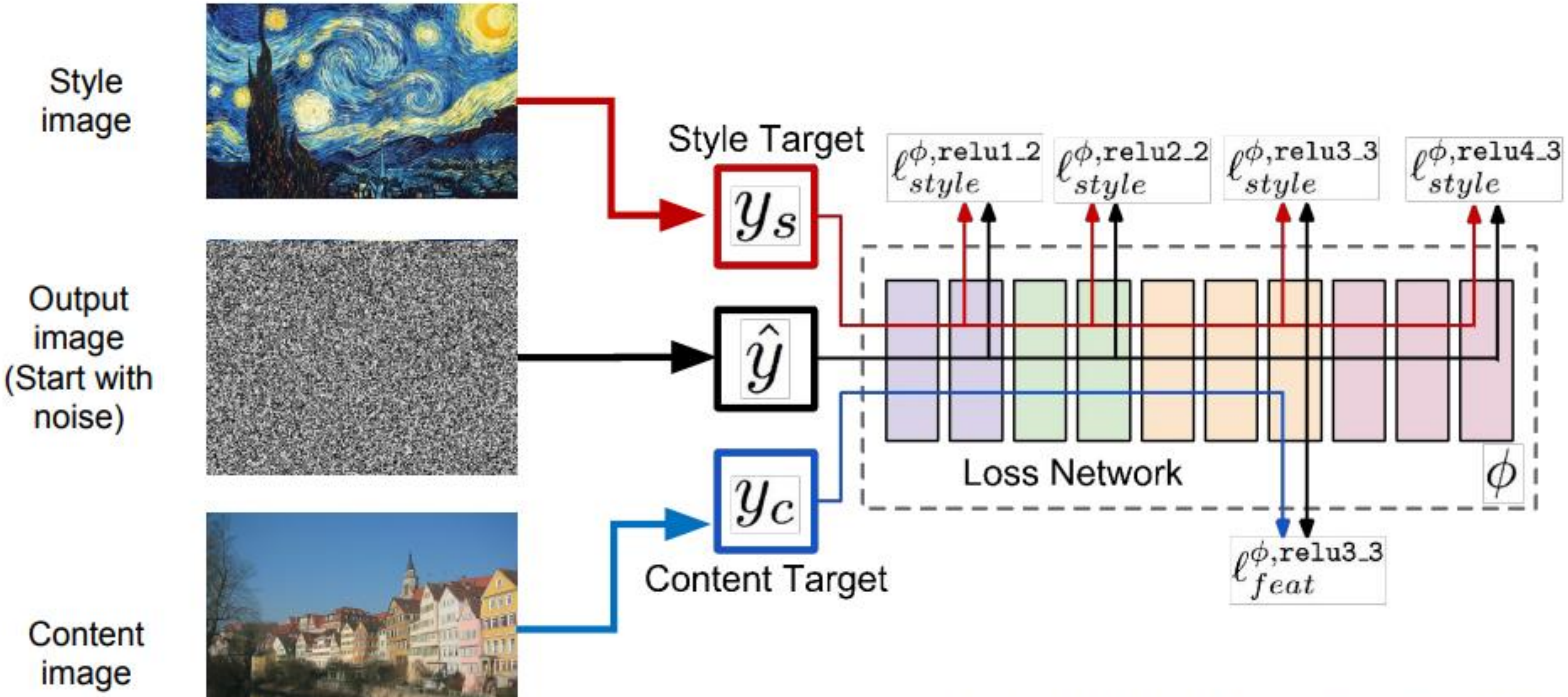
Gram Matrix

Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

Neural Style Transfer: Texture 합성을 예술에 적용한것으로, Gram Matrix를 재구성하는것과 Feature을 재구성하는것을 합하여 만들어진 이미지의 결과



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.