Deep Learning for Computer Vision

Explaining CNNs: Visualization Methods

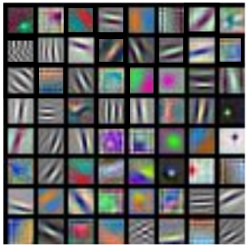
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

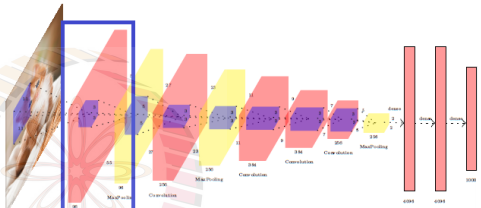
Acknowledgements

- Most of this lecture's slides are based on **Lecture 13** of **CS231n: Convolutional Neural Networks for Visual Recognition** course taught by Fei-Fei Li and others at Stanford University
- Some content (AlexNet CNN figures, etc) is also adapted from **Lecture 12** of **CS7015: Deep Learning** course taught by Mitesh Khapra at IIT Madras

Visualize Filters/Kernels (First Layer)[1]



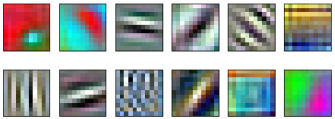
AlexNet:
64 x 3 x 11 x 11



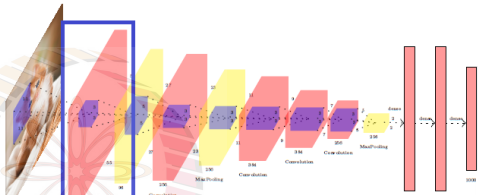
[1] Krizhevsky, One weird trick for parallelizing convolutional neural networks, 2014

Visualize Filters/Kernels (First Layer)[1]

AlexNet:
64 x 3 x 11 x 11



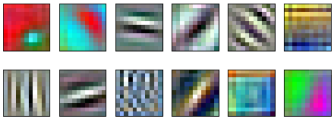
A Closer Look (AlextNet)
Fan et al.(arxiv:1904.05526)



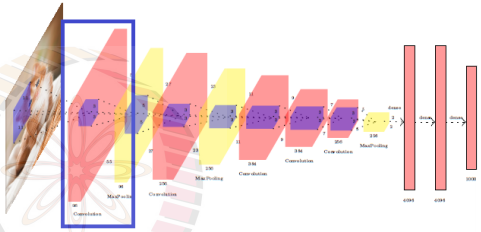
[1]Krizhevsky, One weird trick for parallelizing convolutional neural networks, 2014

Visualize Filters/Kernels (First Layer)[1]

AlexNet:
64 x 3 x 11 x 11



A Closer Look (AlextNet)
Fan et al.(arxiv:1904.05526)



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

[1] Krizhevsky, One weird trick for parallelizing convolutional neural networks, 2014

Visualize Filters/Kernels (First Layer)

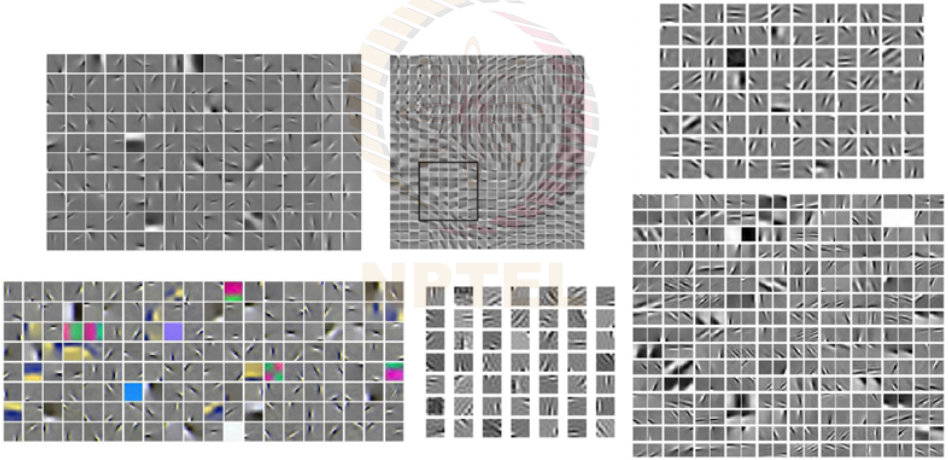
- You can visualize the kernels of higher layers but it is just not interesting

Visualize Filters/Kernels (First Layer)

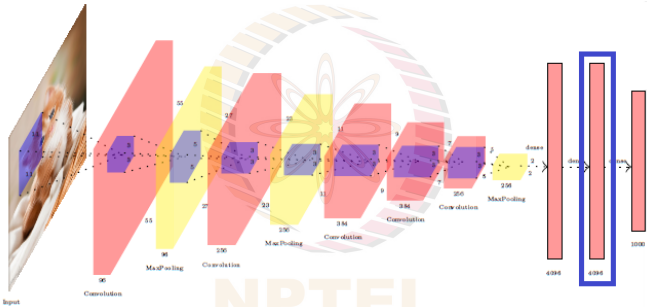
- You can visualize the kernels of higher layers but it is just not interesting
- Input to higher layers is no more the images we know or understand, so becomes difficult to interpret the filters beyond the first layer

Visualize Filters/Kernels (First Layer)

The Gabor-like filters fatigue



Visualize the Representation Space (Last Layer)



- 4096-dimensional feature vector for an image (layer immediately before the classifier)
- Run the network on many images, collect the feature vectors

Visualize the Representation Space (Last Layer)

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

Visualize the Representation Space (Last Layer)

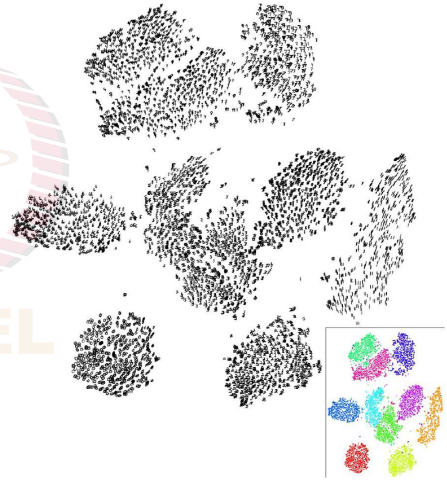
- Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions
- Use any dimensionality reduction algorithm

Visualize the Representation Space (Last Layer)

- Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions
- Use any dimensionality reduction algorithm
- Simple algorithm: Principal Component Analysis (PCA)

Visualize the Representation Space (Last Layer)

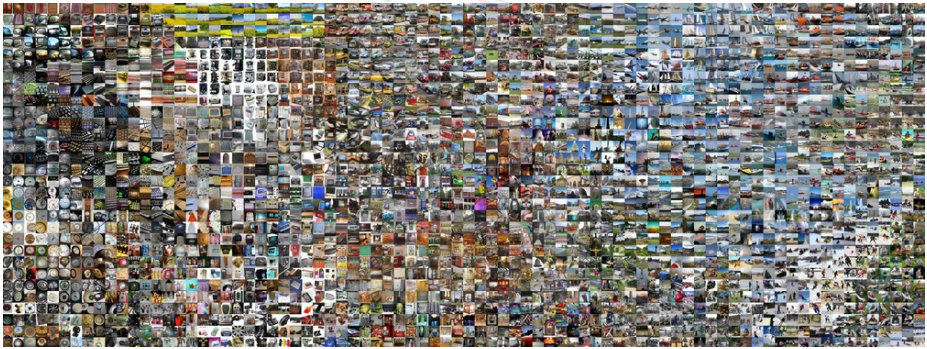
- Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions
- Use any dimensionality reduction algorithm
- Simple algorithm: Principal Component Analysis (PCA)
- More complex: **t-SNE** (right)



van der Maaten and Hinton, Visualizing High-Dimensional Data Using t-SNE, Journal of Machine Learning Research, 2008

t-SNE Visualization

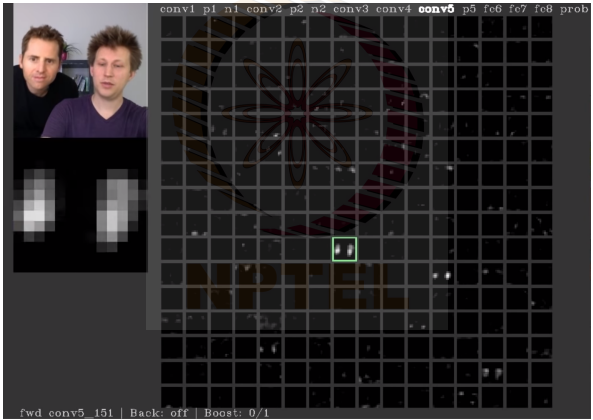
- Images that are nearby each other are also close in the CNN representation space, which implies that the CNN "sees" them as being very similar



- Notice that the similarities are more often class-based and semantic, rather than pixel and color-based.

Visualize Activations

Conv5 feature map is $128 \times 13 \times 13$; Visualize as 128 13×13 grayscale images

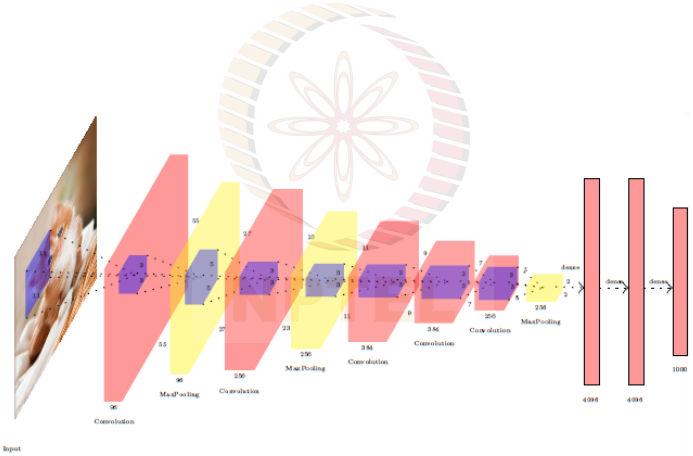


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

Figure copyright: Jason Yosinski, 2014.

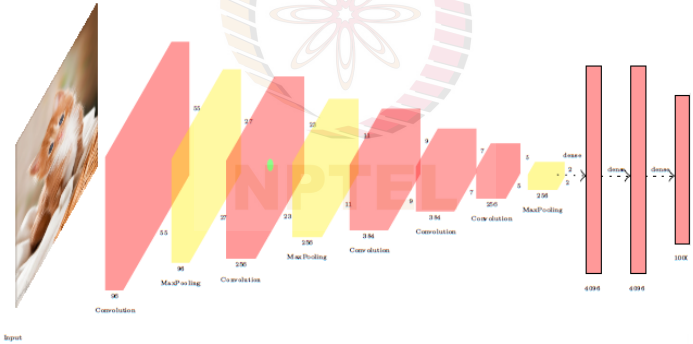
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers



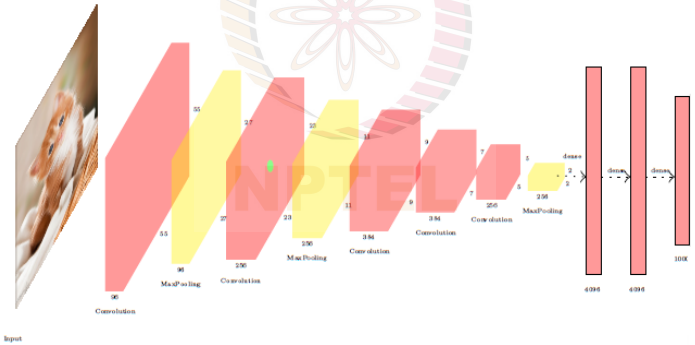
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers



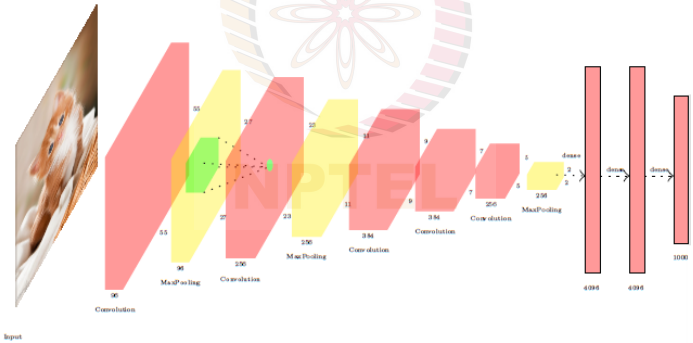
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers
- Feed images to the CNN and identify all images which cause that particular neuron to fire



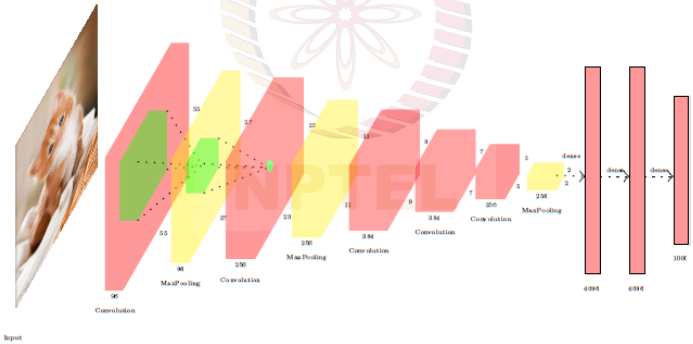
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers
- Feed images to the CNN and identify all images which cause that particular neuron to fire
- We can then easily trace back to the patch in the image which causes that neuron to fire



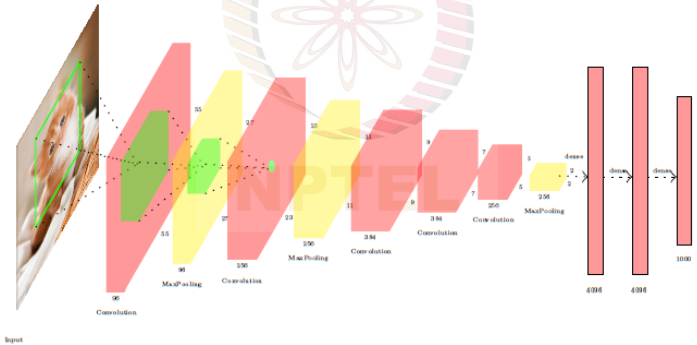
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers
- Feed images to the CNN and identify all images which cause that particular neuron to fire
- We can then easily trace back to the patch in the image which causes that neuron to fire



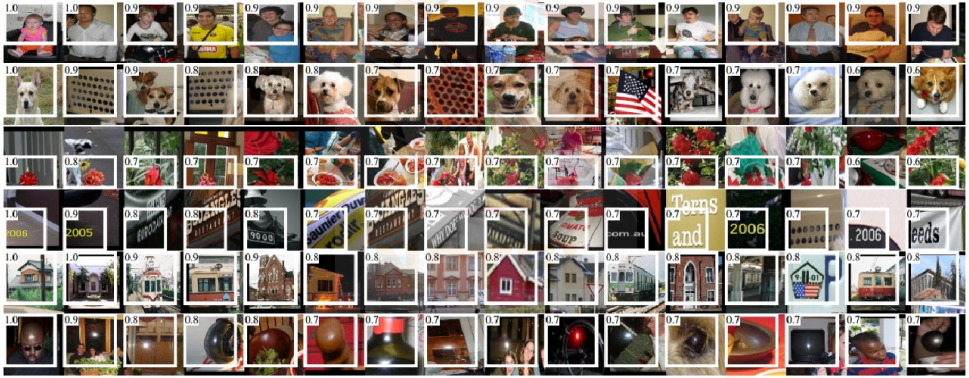
Visualize Maximally Activating Image Patches

- Consider a CNN, and a single neuron in any of its intermediate layers
- Feed images to the CNN and identify all images which cause that particular neuron to fire
- We can then easily trace back to the patch in the image which causes that neuron to fire



Visualize Maximally Activating Image Patches

- Repeating this for others neurons in the CNN shows us a pattern



Rich feature hierarchies for accurate object detection and semantic segmentation by Ross Girshick et al.

Visualize Maximally Activating Image Patches

- Repeating this for others neurons in the CNN shows us a pattern

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

Occlusion Experiments[2]

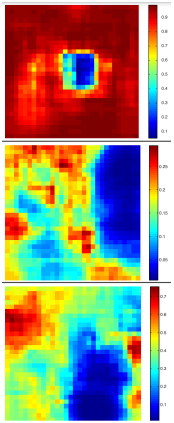
- Typically, we are interested in understanding which portions of an image are responsible for maximizing probability of a certain class

[2] Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014

Occlusion Experiments[2]

- Typically, we are interested in understanding which portions of an image are responsible for maximizing probability of a certain class
- Occlude (gray out) different patches in the image (centered on each pixel), and see effect on predicted probability of the correct class $\implies$ gives you a probability for each pixel

Input image



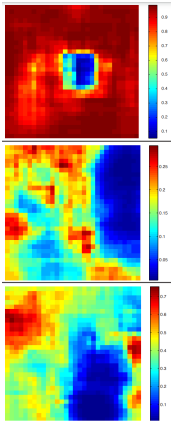
Probability of correct class

[2] Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014

Occlusion Experiments[2]

- Typically, we are interested in understanding which portions of an image are responsible for maximizing probability of a certain class
- Occlude (gray out) different patches in the image (centered on each pixel), and see effect on predicted probability of the correct class $\Longrightarrow$ gives you a probability for each pixel
- For example, the first heat map (top) shows that occluding the face of the dog causes a maximum drop in the prediction probability

Input image

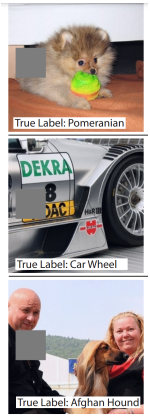


Probability of correct class

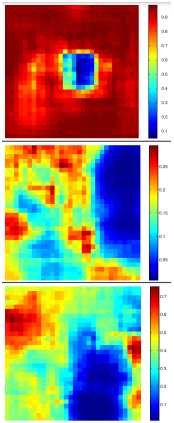
[2] Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014

Occlusion Experiments[2]

- Typically, we are interested in understanding which portions of an image are responsible for maximizing probability of a certain class
- Occlude (gray out) different patches in the image (centered on each pixel), and see effect on predicted probability of the correct class $\implies$ gives you a probability for each pixel
- For example, the first heat map (top) shows that occluding the face of the dog causes a maximum drop in the prediction probability
- Similar observations for other images



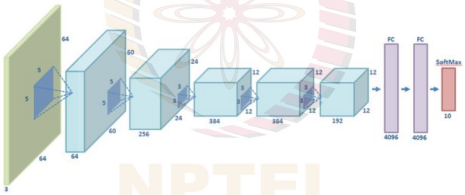
Input image



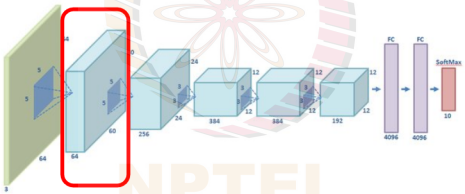
Probability of correct class

[2] Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014

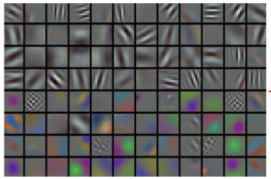
Summary: 'Don't-disturb-the-model' methods



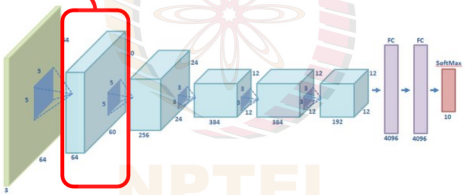
Summary: 'Don't-disturb-the-model' methods



Summary: 'Don't-disturb-the-model' methods

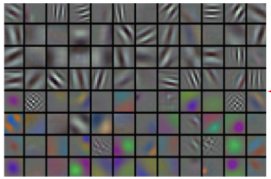


Visualizing the filter/kernels
(raw weights)



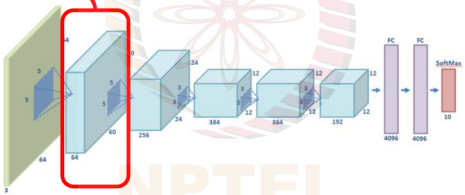
Conv

Summary: 'Don't-disturb-the-model' methods



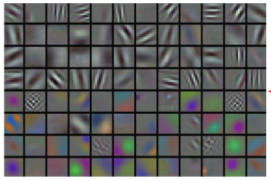
Visualizing the filter/kernels
(raw weights)

Only interpretable at the first layer!
Not interesting enough for higher layers!



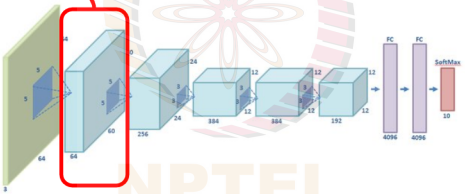
Conv

Summary: 'Don't-disturb-the-model' methods

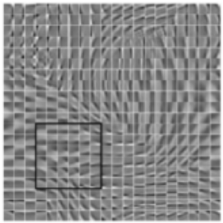


Visualizing the filter/kernels
(raw weights)

Only interpretable at the first layer!
Not interesting enough for higher layers!

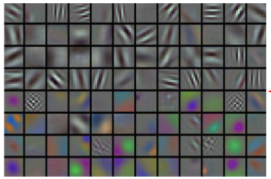


Conv

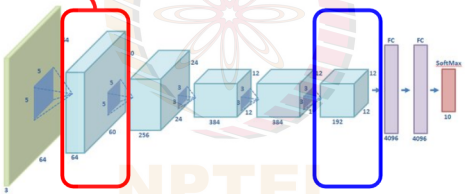


The Gabor-like filter fatigue

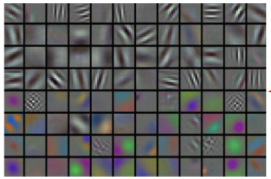
Summary: 'Don't-disturb-the-model' methods



Visualizing the filter/kernels
(raw weights)

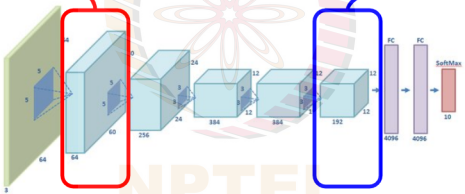


Summary: 'Don't-disturb-the-model' methods



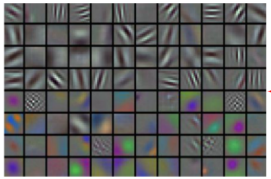
Visualizing the filter/kernels
(raw weights)

Visualize patches that
maximally activate neuron



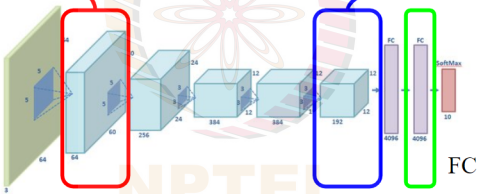
Conv Pool

Summary: 'Don't-disturb-the-model' methods

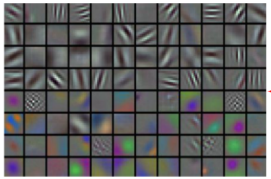


Visualizing the filter/kernels
(raw weights)

Visualize patches that
maximally activate neuron

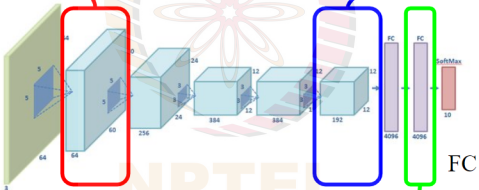


Summary: 'Don't-disturb-the-model' methods



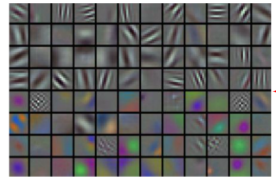
Visualizing the filter/kernels
(raw weights)

Visualize patches that
maximally activate neuron



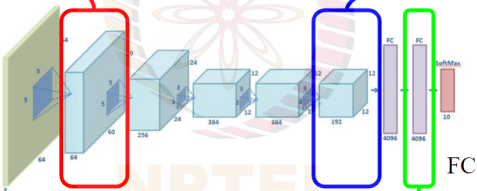
Visualizing the representation

Summary: 'Don't-disturb-the-model' methods



Visualizing the filter/kernels
(raw weights)

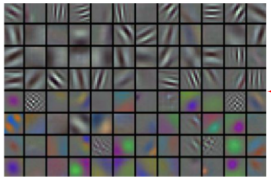
Visualize patches that
maximally activate neuron



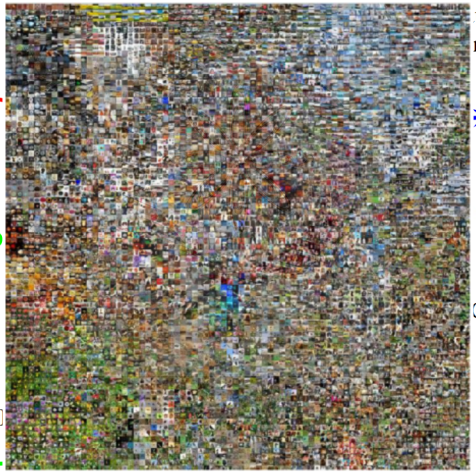
t-SNE

Visualizing the representation

Summary: 'Don't-disturb-the-model' methods



Visualizing the filter/kernels
(raw weights)



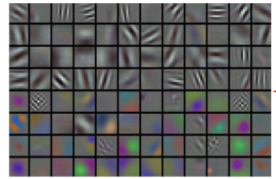
t
ron

C

t-SN

Visualizing the representation

Summary: 'Don't-disturb-the-model' methods

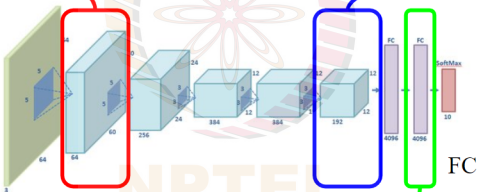


Visualizing the filter/kernels
(raw weights)

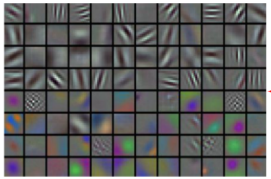
Visualize patches that
maximally activate neuron

t-SNE

Visualizing the representation



Summary: 'Don't-disturb-the-model' methods

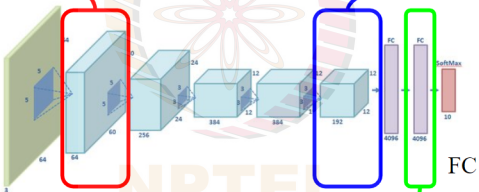


Visualizing the filter/kernels
(raw weights)

t-SNE

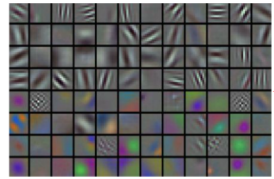
Visualizing the representation

Visualize patches that
maximally activate neuron

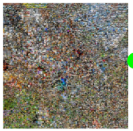


Occlusion Experiment

Summary: 'Don't-disturb-the-model' methods



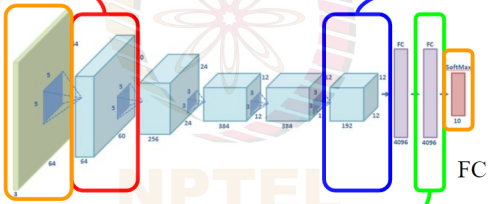
Visualizing the filter/kernels
(raw weights)



t-SNE

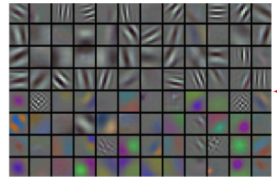
Visualizing the representation

Visualize patches that
maximally activate neuron

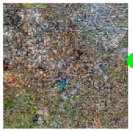


Occlusion Experiment

Summary: 'Don't-disturb-the-model' methods



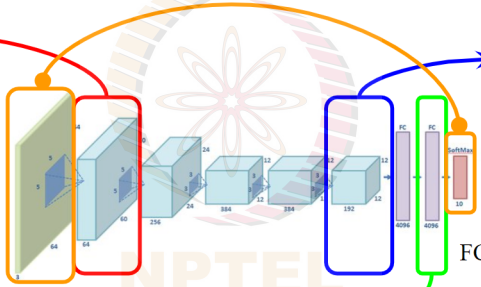
Visualizing the filter/kernels
(raw weights)



t-SNE

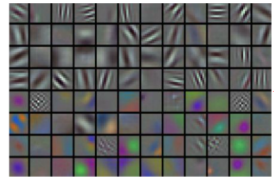
Visualizing the representation

Visualize patches that
maximally activate neuron

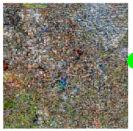


Occlusion Experiment

Summary: 'Don't-disturb-the-model' methods



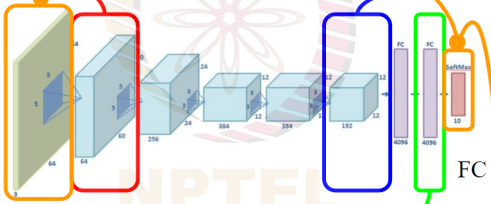
Visualizing the filter/kernels
(raw weights)

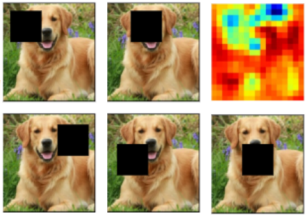


t-SNE

Visualizing the representation

Visualize patches that
maximally activate neuron





Occlusion Experiment

Readings

Summary of Visualizing CNNs

- Lecture Notes of CS231n, Stanford

Miscellaneous

- Deep Visualization Toolkit demo video and webpage by J. Yosinski et al.
- To see high-resolution t-SNE visualizations, visit here
- To know more about t-SNE, visit here

References

Laurens van der Maaten and Geoffrey E. Hinton. "Visualizing Data using t-SNE". In: 2008.

Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Nov. 2013).

Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *ECCV*. 2014.

Jost Tobias Springenberg et al. "Striving for Simplicity: The All Convolutional Net". In: *CoRR* abs/1412.6806 (2015).