Visualizing and Understanding CNNs

Matthew D. Zeiler, Rob Fergus

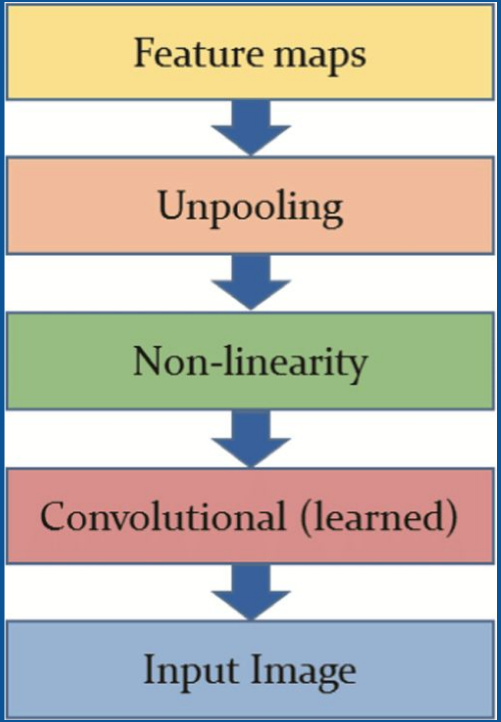
Presented by Jack Morris

CNNs are sooo good!

- Krizhevsky, A., Sutskever, I., and Hinton, G.E. *Imagenet classification with deep convolutional neural networks.* In NIPS, 2012.
  - Error rate of 26.1% → 16.4% percent (***37% reduction***)
- Reasons for this:
  - [1] larger training sets became available (ImageNet)
  - [2] powerful GPU implementations (CUDA)
  - [3] better model regularization, like Dropout (Hinton, 2012)
- Current ImageNet leaderboard

But how?

- Little insight into how CNNs work, or why
- Related work done to visualize features, but only on the first layer
- We want to visualize higher layers
- Idea: use **deconvnets** (Zeiler, 2010)

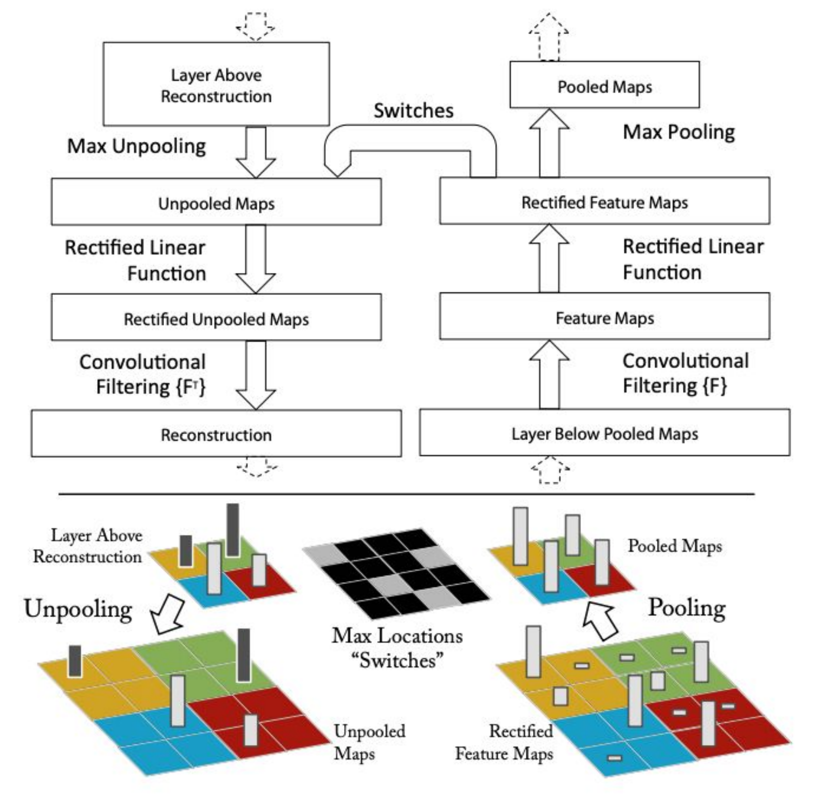


Deconvolutional networks

Deconvolutional networks basically do the same operations that CNNs do, but in reverse

- Convolutional neural networks: convolutions, nonlinearities, pooling
- De-convolutional neural networks: de-convolutions, unpooling, rectification
    - **Unpooling:** record locations of maxima using switch variables, reconstruct pre-pooling feature maps
    - **Rectification:** use ReLU to make sure feature maps are positive
    - **Filtering**: use filters (transposed) to map back to pixel space

Deconvolutional networks



Training Details

- ConvNet model, same architecture as AlexNet (Krizhevsky et al., 2012) for ImageNet classification
  - Some minor tweaks-- gained .1% top-5 accuracy
- Trained on ImageNet 2012: 1.3m images, 1000 classes
- Preprocessing by resizing, cropping, and using 10 different sub-crops

Minor detour: ImageNet

- Over 14-million hand-annotated images with over 20,000 categories
- Ongoing project
- Very standard dataset for deep learning
- Other famous image datasets I know about: CalTech-256, MNIST, SVHN, Celebrity face recognition dataset:

ImageNet tSNE: https://cs.stanford.edu/people/karpathy/cnnembed/

Human ImageNet

- ImageNet classification is really hard, can a human even do it well?
- There are 1000 classes
- There are somewhere around 120 breeds of dogs
- Karpathy got **5.1%** top-5 error [apparently some people have done slightly better]

http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

Section 4: ConvNet Visualization

- The Big Idea:

  Use the deconvnet to visualize feature activations on the ImageNet validation set.

- Project top activations back into the pixel space to visualize them

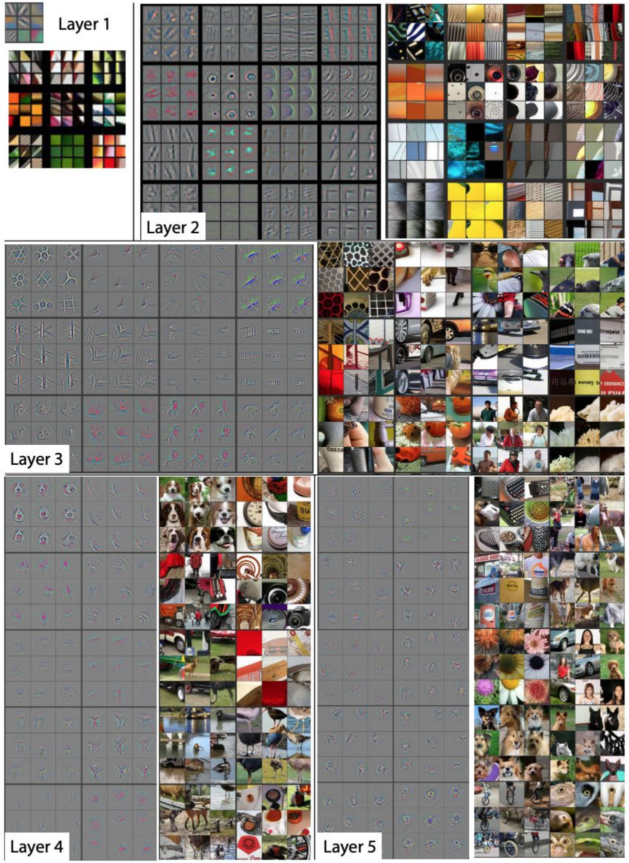


Figure 2. Visualization of features in a fully trained model. For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using our deconvolutional network approach. Our reconstructions are *not* samples from the model: they are reconstructed patterns from the validation set that cause high activations in a given feature map. For each feature map we also show the corresponding image patches. Note: (i) the the strong grouping within each feature map, (ii) greater invariance at higher layers and (iii) exaggeration of discriminative parts of the image, e.g. eyes and noses of dogs (layer 4, row 1, cols 1). Best viewed in electronic form.

Section 4: ConvNet Visualization

Observations:

- Projecting back into pixel space shows hierarchical features
    - Layer 2 shows corners and edges… layer 5 shows full objects with different variations in poses [figure 4]
- Activations seem to be pose invariant [figure 5]
    - Rotations and scaling have a large effect on the first layer but small effects on the last layer (generally)

Section 4.1: Architecture Selection

- Visualizing first and second layers shows that filters are based on extremely high- and low-frequency information, with little variation
- Also can see artifacts caused by large stride value (4) -- skips things

- How to fix this?
  - Reduce filter size
  - Decrease stride from 2 to 4

- New architecture produces much better feature maps

Section 4.2: Occlusion Sensitivity

- Natural question about image classifiers:

  Is my model identifying the truly important parts of the image, or just classifying based on the surrounding context?
  - Does this really look like a horse, or is it just standing in a field?

- We can systematically occlude different parts of the input image and monitor the output of a classifier

- When we cover up the strongest feature map, this changes the most probable class. Cool!

Section 4.3: Correspondence Analysis

- Measure the spacial layout of faces using a fancy equation measuring the relationship between different facial features
- Turns out, the feature representations do seem to represent the spatial relationships on a face (specifically between eyes and nose)

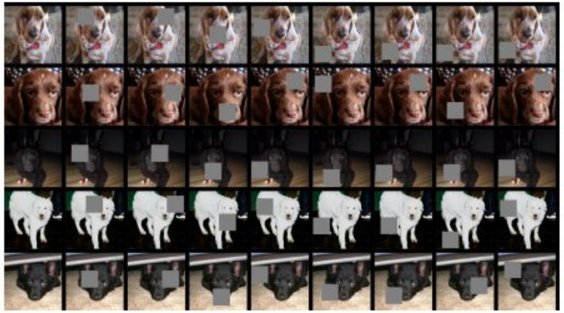


Figure 8. Images used for correspondence experiments. Col 1: Original image. Col 2,3,4: Occlusion of the right eye, left eye, and nose respectively. Other columns show examples of random occlusions.

Section 5: Experiments

- New model gets **14.8%**, **the best published performance on ImageNet**
- Slightly improved (only .2% or so) by using an ensemble of 6 different models
- Also improved scores on Caltech-101 and Caltech-256 datasets