# ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
Ilya Sutskever
Geoffrey E. Hinton
University of Toronto

# ImageNet Dataset

ImageNet

- Over 15 million images
- 22000 image categories (types of objects)
- Labeled by humans
  - using Amazon's Mechanical Turk

ILSVRC

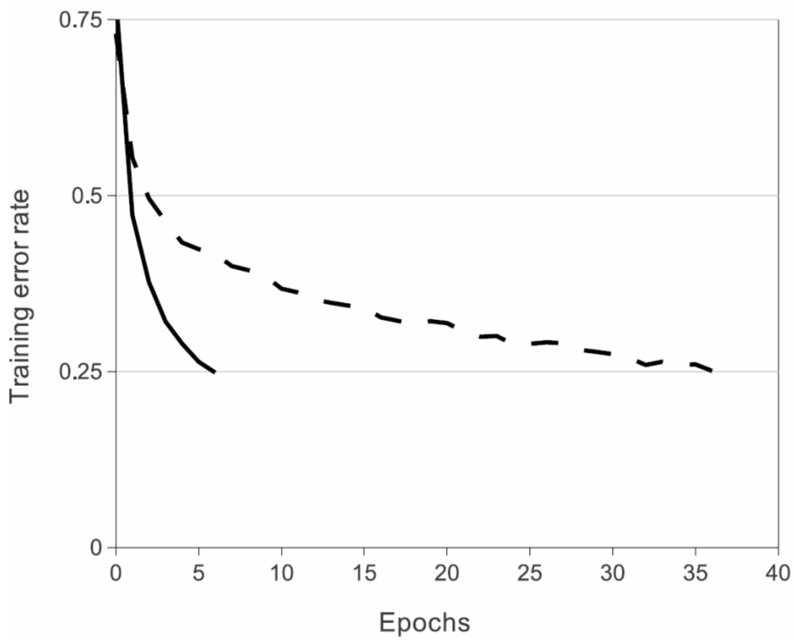
- Subset of ImageNet
- 1000 objects with about 1000 images each
- 1.2 million training images, 50k validation images
- 100k test images (no labels)
- Resized to 256x256

# NN Architecture

- ReLU (Rectified Linear Unit) Neuron

$$\varphi(x) = \max(0, x)$$

  - Converges faster than logit or tanh



# NN Architecture

- Convolutional layers
  - Convolve input with a “kernel” of weights
- Max-pooling layer
  - Find local max on a local neighborhood

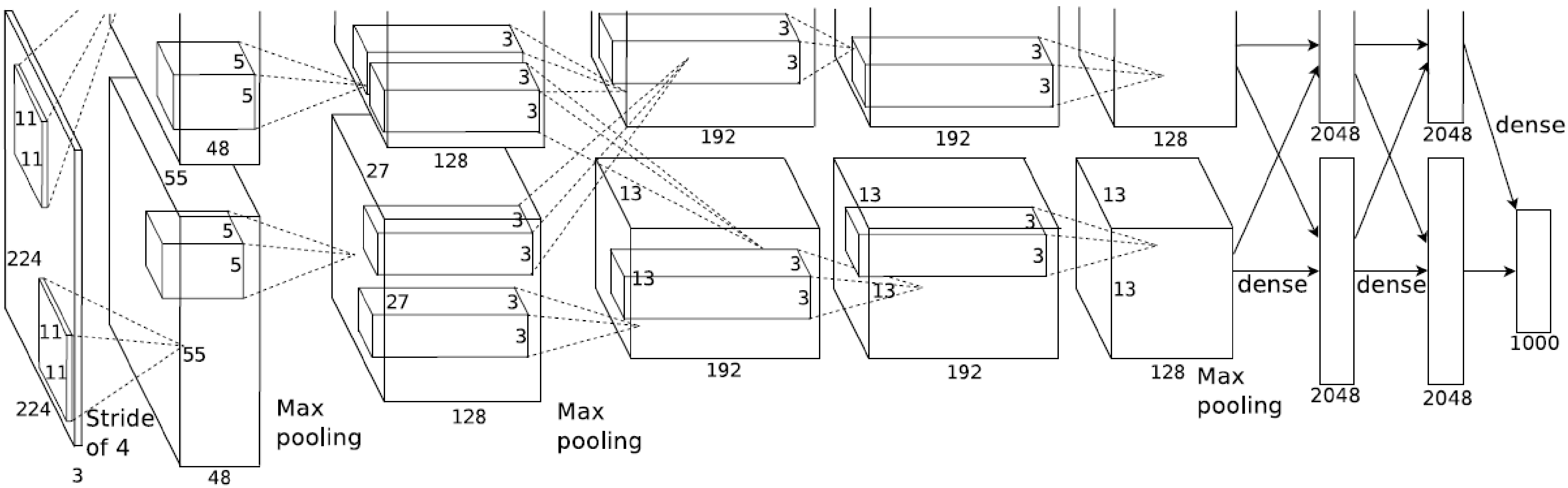



Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

# Reducing Overfitting

- NN has 60 million parameters!

Two main approaches

- Data Augmentation
  - Cropping 224x224 random patches of the images
    - and their reflection
  - Changing the RGB colors of the images
    - Using PCA to find main modes of color variation
- Dropout layer
  - Setting to zero each neuron output with probability 0.5
    - Those neurons are not part of backpropagation
  - Makes the network not rely on any neuron output too much
  - Doubles the number of iterations to converge

# Training Details

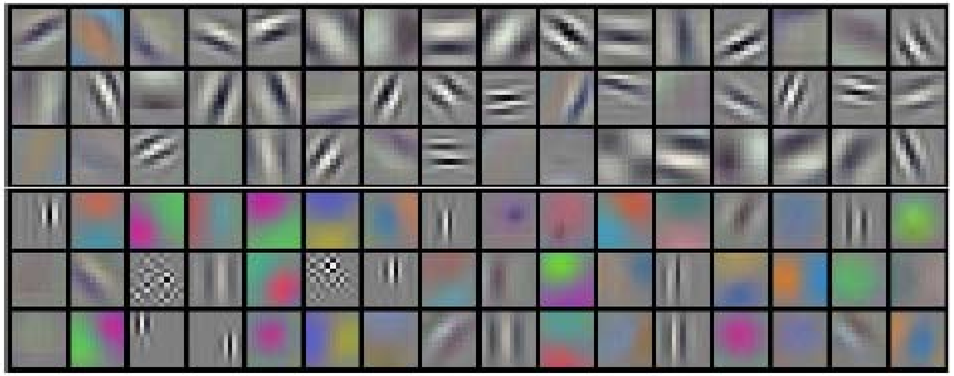


Figure 3: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer on the $224 \times 224 \times 3$ input images.

- Batch size 128 examples
- Momentum 0.9
- Shrinkage 0.0005
- Equal learning rate for all layers
  - Adjusted manually
  - Initially 0.01
  - Divided by 10 when no improvement with current rate
  - Three such reductions done

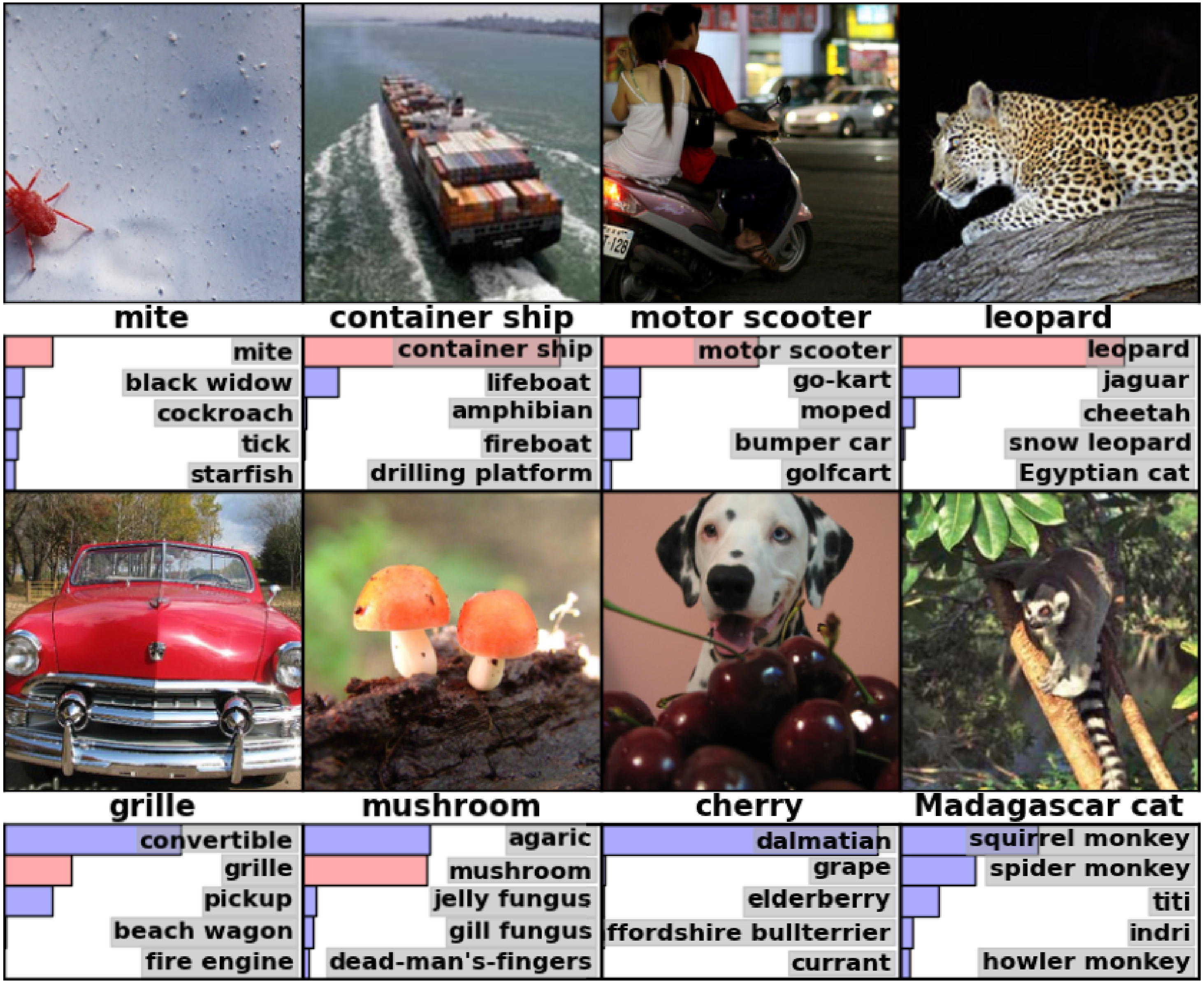
# Results

- Training takes about 5 days
- Result on ImageNet 2009:
  - 8.9 million images
  - 10,184 categories
  - Top 1:67.4%,  top 5:40.9%
  - Next best results are 78.1%, 60.9%

| Model | Top-1 | Top-5 |
|---|---|---|
| *Sparse coding [2]* | *47.1%* | *28.2%* |
| *SIFT + FVs [24]* | *45.7%* | *25.7%* |
| CNN | **37.5%** | **17.0%** |

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

# Results


mite
mite
black widow
cockroach
tick
starfish
container ship
container ship
lifeboat
amphibian
fireboat
drilling platform
motor scooter
motor scooter
go-kart
moped
bumper car
golfcart
leopard
leopard
jaguar
cheetah
snow leopard
Egyptian cat
grille
convertible
grille
pickup
beach wagon
fire engine
mushroom
agaric
mushroom
jelly fungus
gill fungus
dead-man's-fingers
cherry
dalmatian
grape
elderberry
ffordshire bullterrier
currant
Madagascar cat
squirrel monkey
spider monkey
titi
indri
howler monkey

# Closest Images in Last Layer

# ILSVRC 2012

- Classification task:
  - 1000 object categories
  - 1.2 million images train, 50k validation, 100k test
  - Error= top 5 error

| **Model** | **Top-1 (val)** | **Top-5 (val)** | **Top-5 (test)** |
|---|---|---|---|
| *SIFT + FVs [7]* | — | — | *26.2%* |
| 1 CNN | 40.7% | 18.2% | — |
| 5 CNNs | 38.1% | 16.4% | **16.4%** |
| 1 CNN* | 39.0% | 16.6% | — |
| 7 CNNs* | 36.7% | 15.4% | **15.3%** |

Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release.

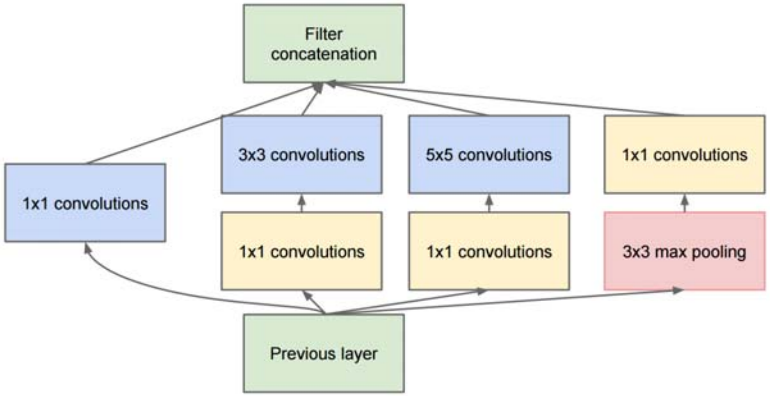
# ILSVRC 2014-2017

- Classification task:
  - Same data as ILSVRC-2012
  - 1000 object categories
  - 1.2 million images train, 50k validation, 100k test
  - Error= top 5 error
- Deep networks:
  - 16-150 layers

| Team Name | Error % | Layers |
|---|---|---|
| 2017 | | |
| WMW | 2.25 | |
| 2016 | | |
| Trimps-Soushen | 2.99 | |
| 2015 | | |
| MSRA | 3.56 | 150+ |
| Trimps-Soushen | 4.58 | |
| Qualcomm | 4.87 | |
| 2014 | | |
| GoogLeNet | 6.66 | 22-27 |
| VGG | 7.33 | 16-19 |
| MSRA | 8.06 | |

# Other Recent Layers

- Maxout layer
  - Maximum between different channels
  - Like an “OR” i.e. alternative representations
- Distance Transform Layer
  - A generalization of the max pooling layer
  - Penalizes parts that are far from prescribed locations
- Inception Layer
  - Parallel channels with different complexities
- Batch Normalization Layer
  - Normalize the responses of each batch
- Lorenz loss layer
  - More robust to labeling noise
  - Easier to train than softmax



# Popular CNN Packages

- Matconvnet (Oxford, Matlab)
  - Easy to use and modify
- Tensorflow (Google, Python)
  - Scalable, but hard to modify
- Caffe (Berkeley, C++)
  - Very popular in the vision community
- CNTK (Microsoft, C++)
  - Fast and scalable to multi-CPU
- Theano (Montreal, Python)
  - Symbolic differentiation, steep learning curve
- MxNet
  - Python, Matlab, R
- Torch (Facebook)

# Conclusions

- Deep Neural Network
  - ReLU neuron for fast convergence
  - GPU implementation for speed
  - Tricks to avoid overfitting:
    - Data augmentation
    - Dropout
    - Inception
- Results
  - Large datasets with millions of images and thousands of categories
  - Training takes a few days on a single machine
  - Results better every year