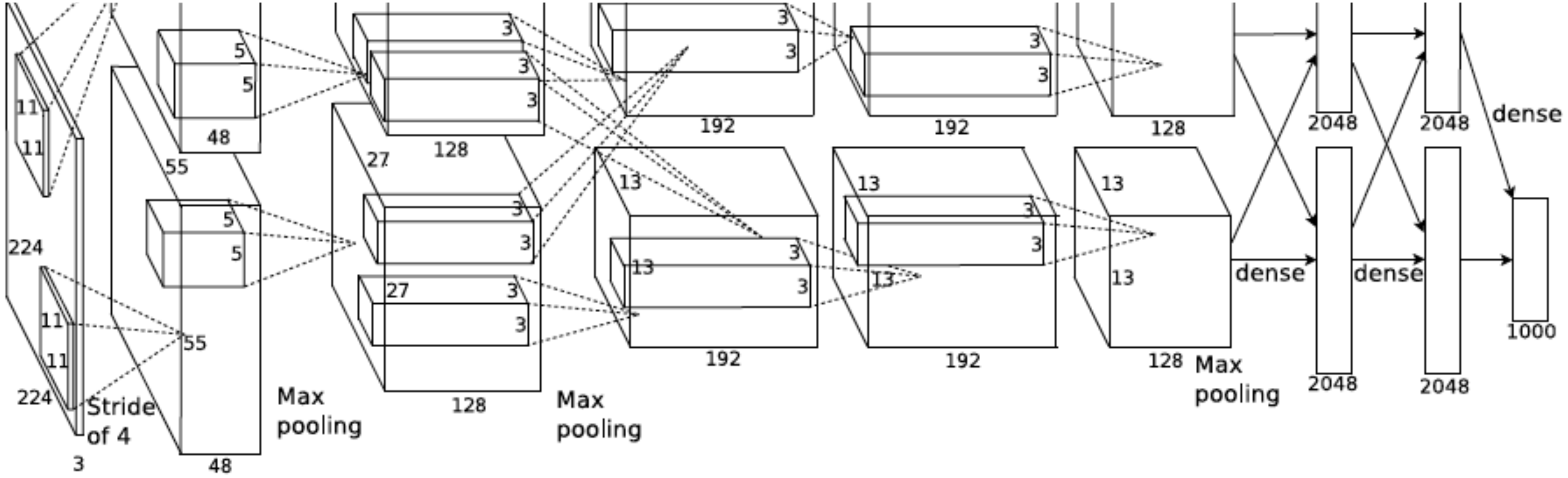
AlexNet



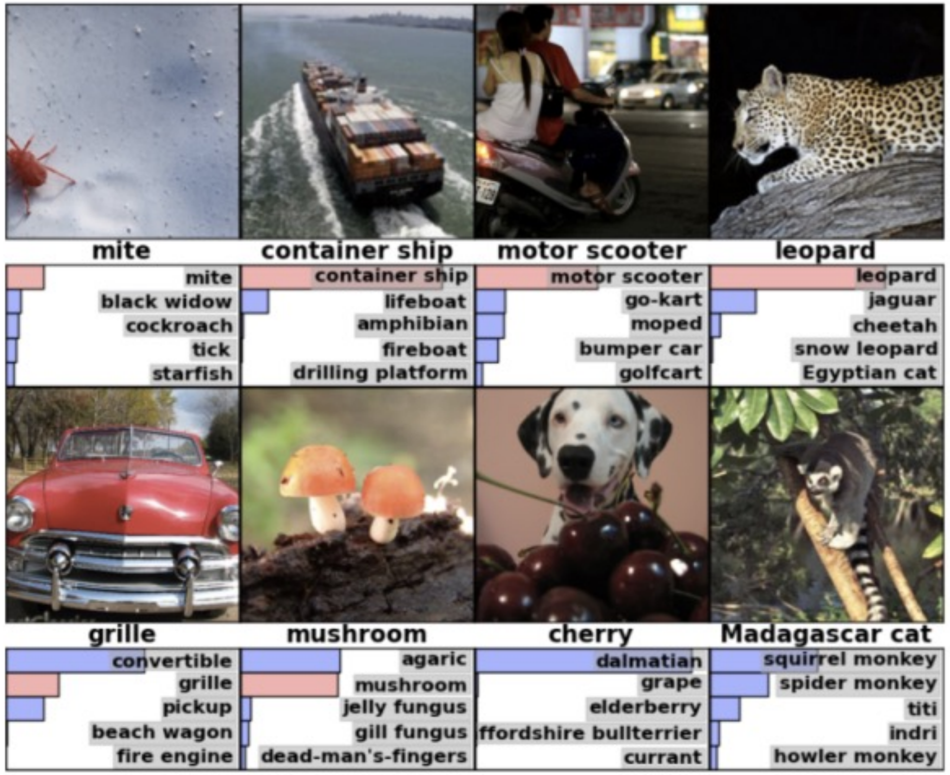
Many slides from Lana Lazebnik, Rob Fergus, Andrej Karpathy

Outline

- AlexNet (2012-2013)
- ImageNet Results
- Pre-trained CNNs as excellent feature extractors

ImageNet Challenge

ILSVRC



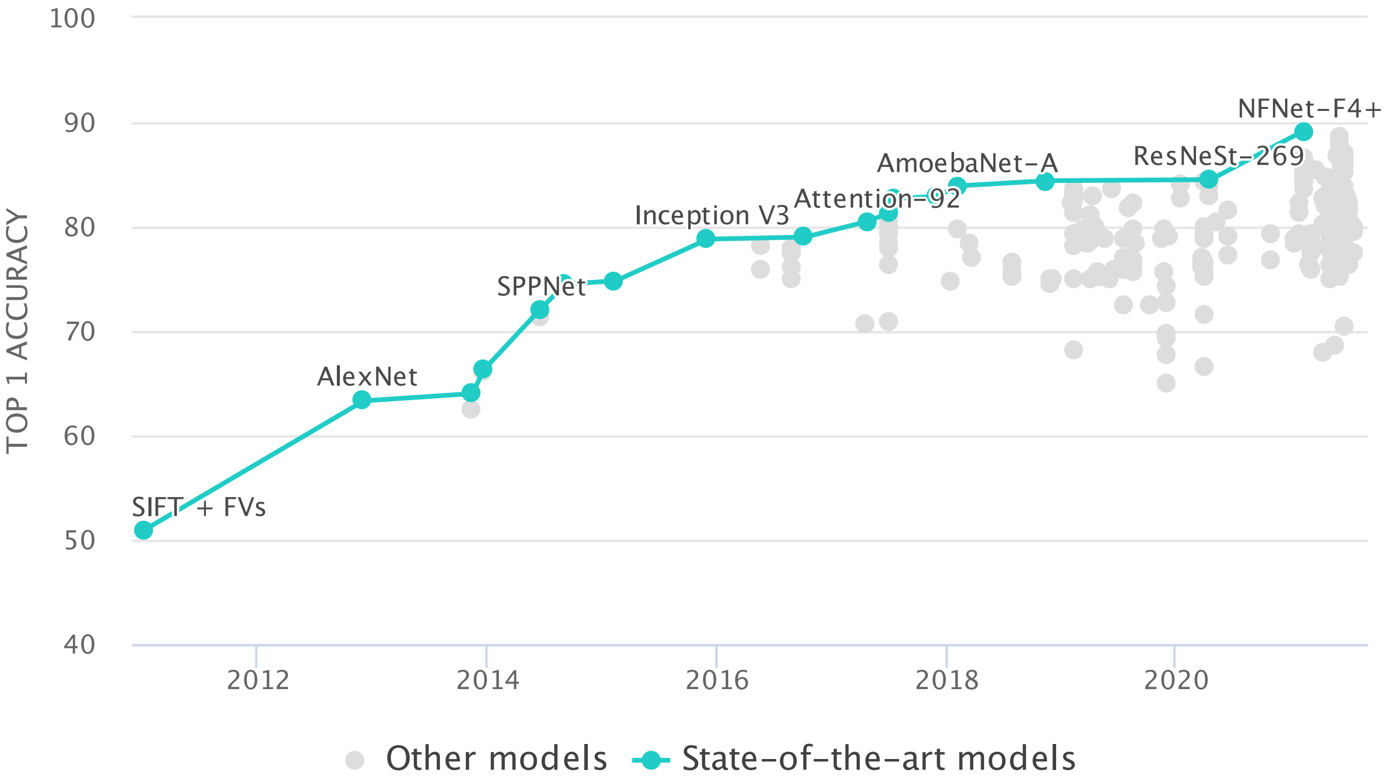
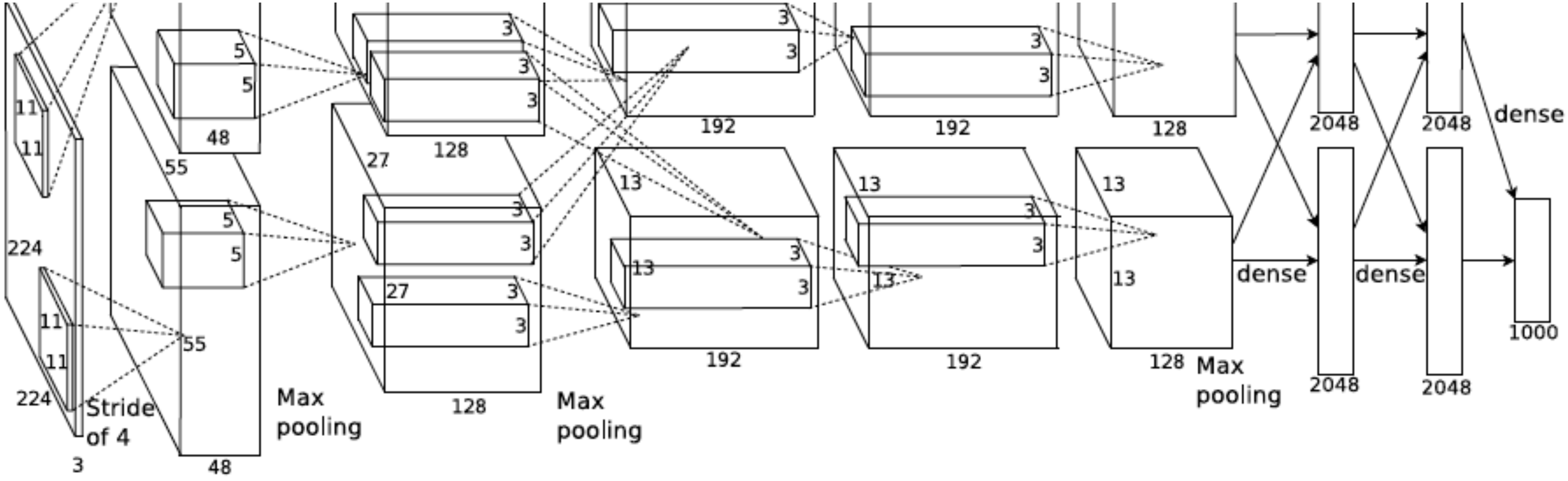
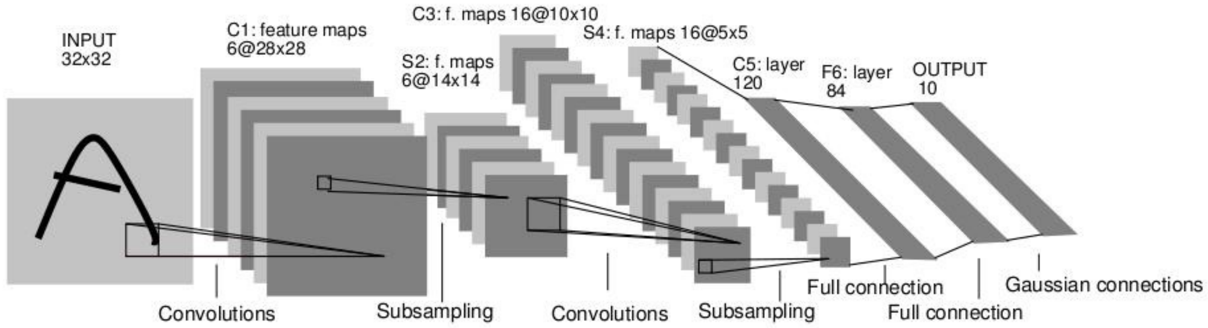


Figure source

AlexNet: ILSVRC 2012 winner



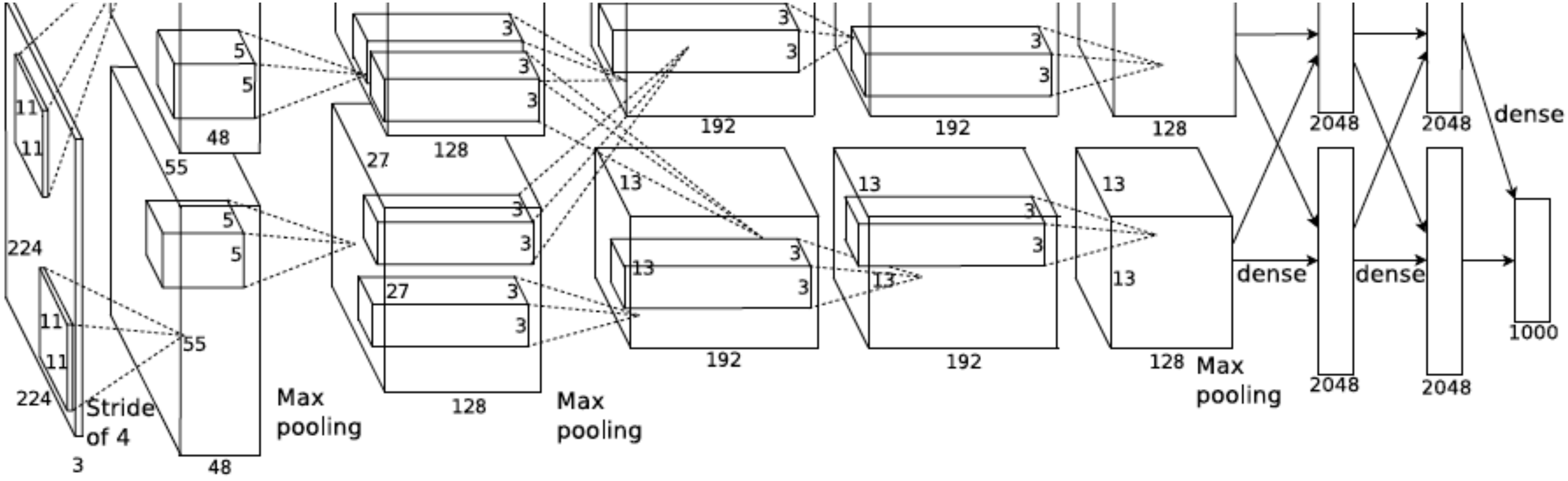
- Successor of LeNet-5, but with a few crucial changes



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, [Gradient-based learning applied to document recognition](), Proc. IEEE 86(11): 2278–2324, 1998

A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](), NIPS 2012

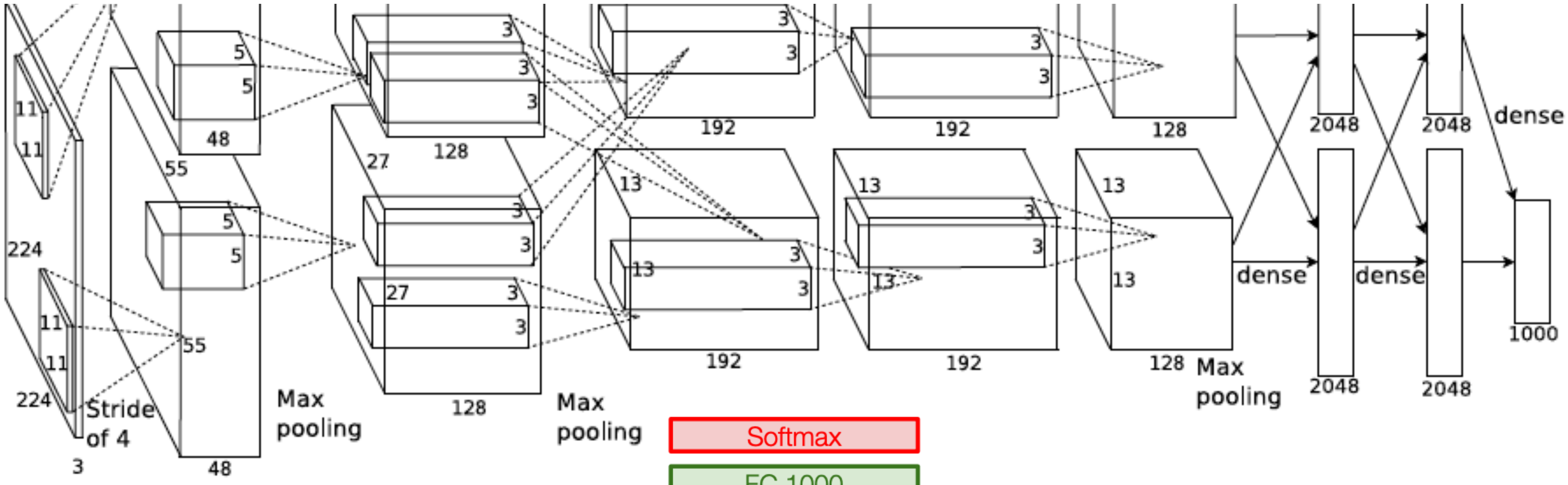
AlexNet: ILSVRC 2012 winner

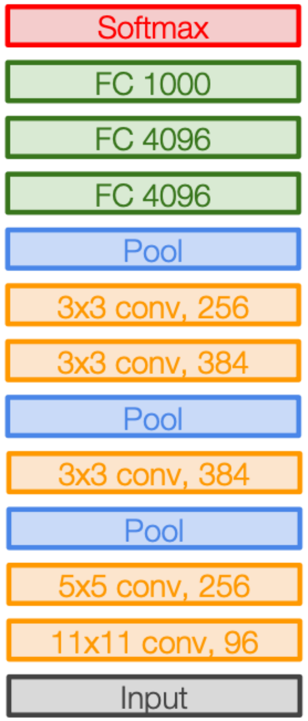


- Successor of LeNet-5, but with a few crucial changes
  - Max pooling, ReLU nonlinearity
  - Dropout regularization
  - More data and bigger model (7 hidden layers, 650K units, 60M params)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week

A. Krizhevsky, I. Sutskever, and G. Hinton, [ImageNet Classification with Deep Convolutional Neural Networks](), NIPS 2012

AlexNet: ILSVRC 2012 winner

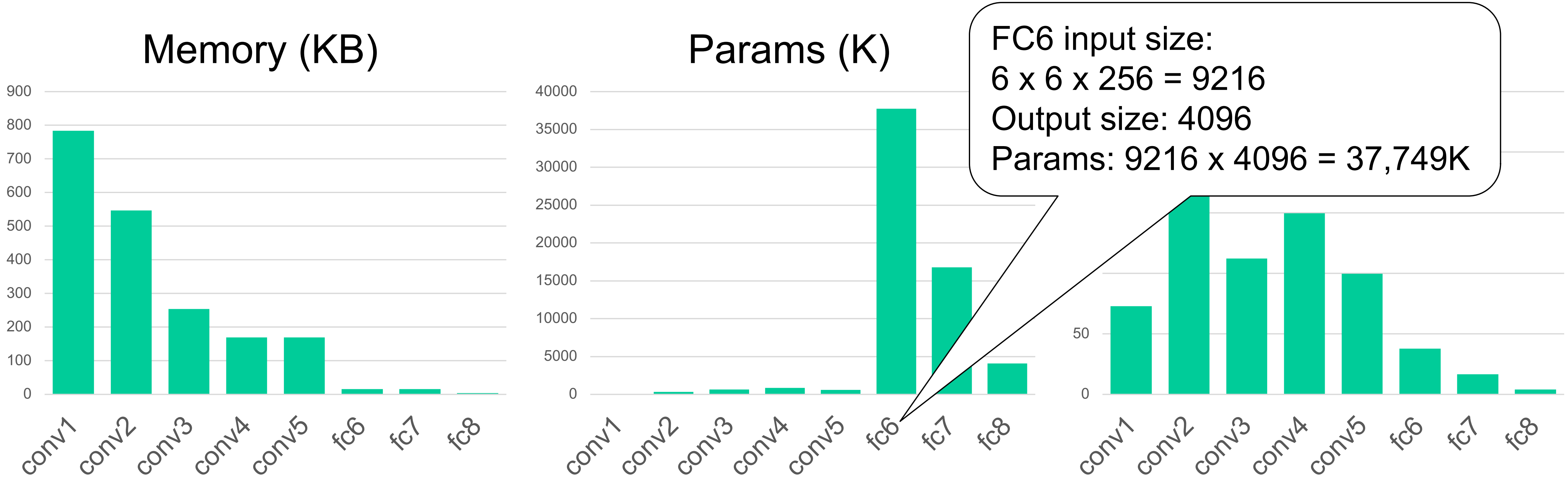




AlexNet (modified): Stats

	Input size		Layer				Output size		Receptive Field	Effective Stride	Effective Padding
Layer	C	H / W	filters	kernel	stride	pad	C	H / W			
conv1	3	227	64	11	4	2	64	56	11	4	2
pool1	64	56		3	2	0	64	27	19	8	2
conv2	64	27	192	5	1	2	192	27	51	8	18
pool2	192	27		3	2	0	192	13	67	16	34
conv3	192	13	384	3	1	1	384	13	99	16	50
conv4	384	13	256	3	1	1	256	13	131	16	66
conv5	256	13	256	3	1	1	256	13	163	16	66
pool5	256	13		3	2	0	256	6	195	32	66
flatten	256	6					9216		259	32	66
fc6	9216		4096				4096		259	32	66
fc7	4096		4096				4096		259	32	66
fc8	4096		1000				1000		259	32	66

AlexNet (modified): Analysis



Most of the memory usage is in the early convolution layers

Nearly all parameters are in the fully-connected layers

Most floating-point ops occur in the convolution layers

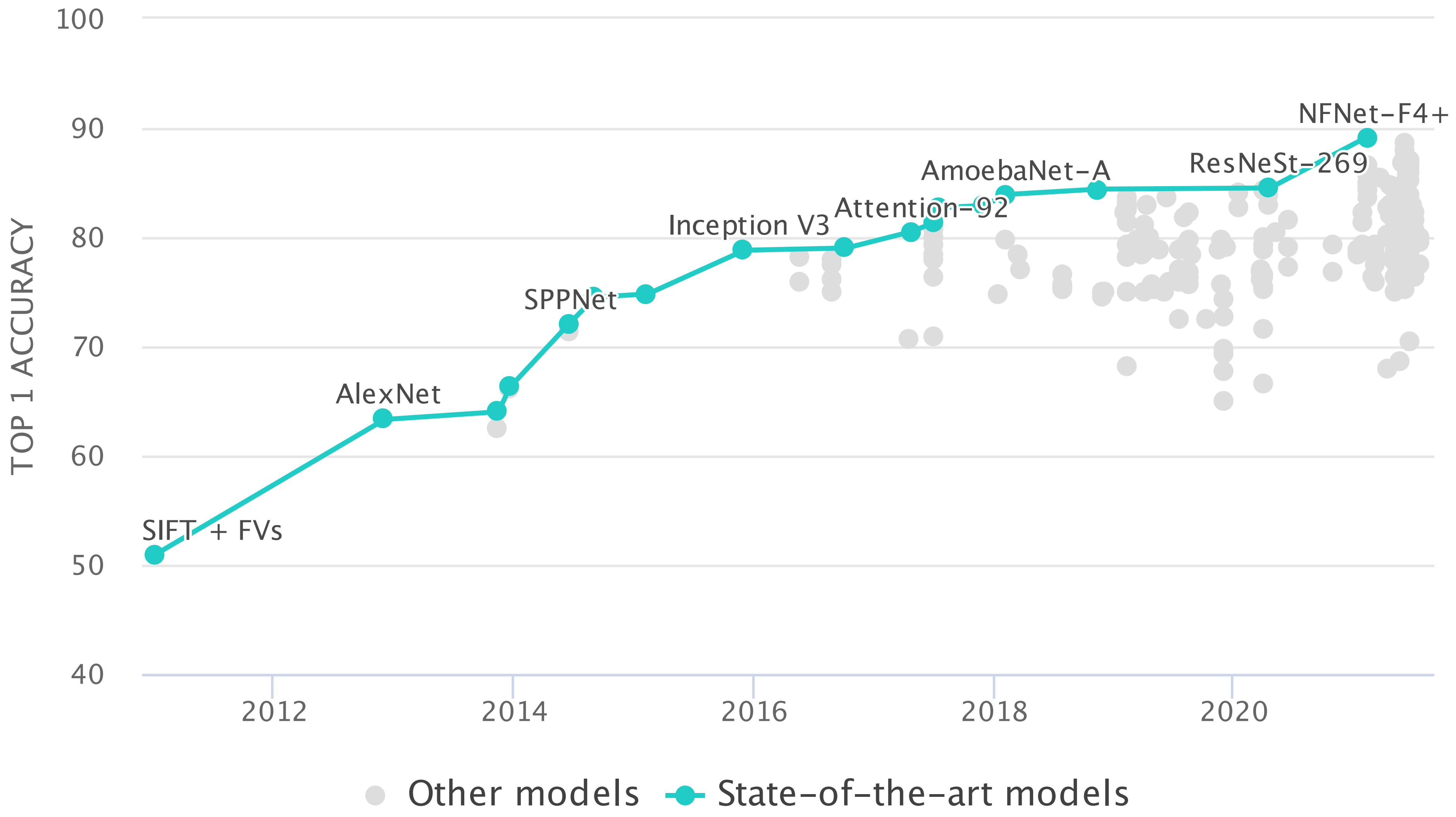
Source: [J. Johnson]

ImageNet Challenge 2012-2014

Team	Year	Place	Error (top-5)	External data
XRCE	2011		25.8%	no
SuperVision – Toronto (7 layers)	2012	-	16.4%	no
SuperVision	2012	1st	15.3%	ImageNet 22k

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

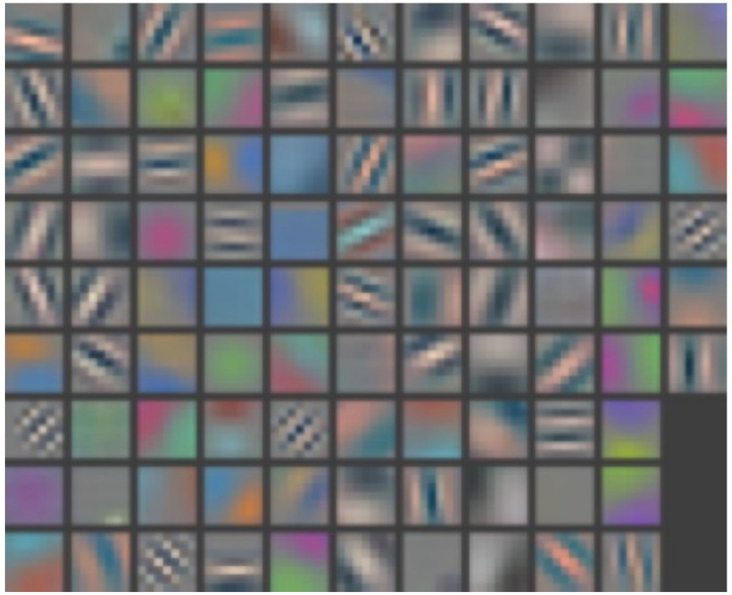
Breakthrough + Many different models



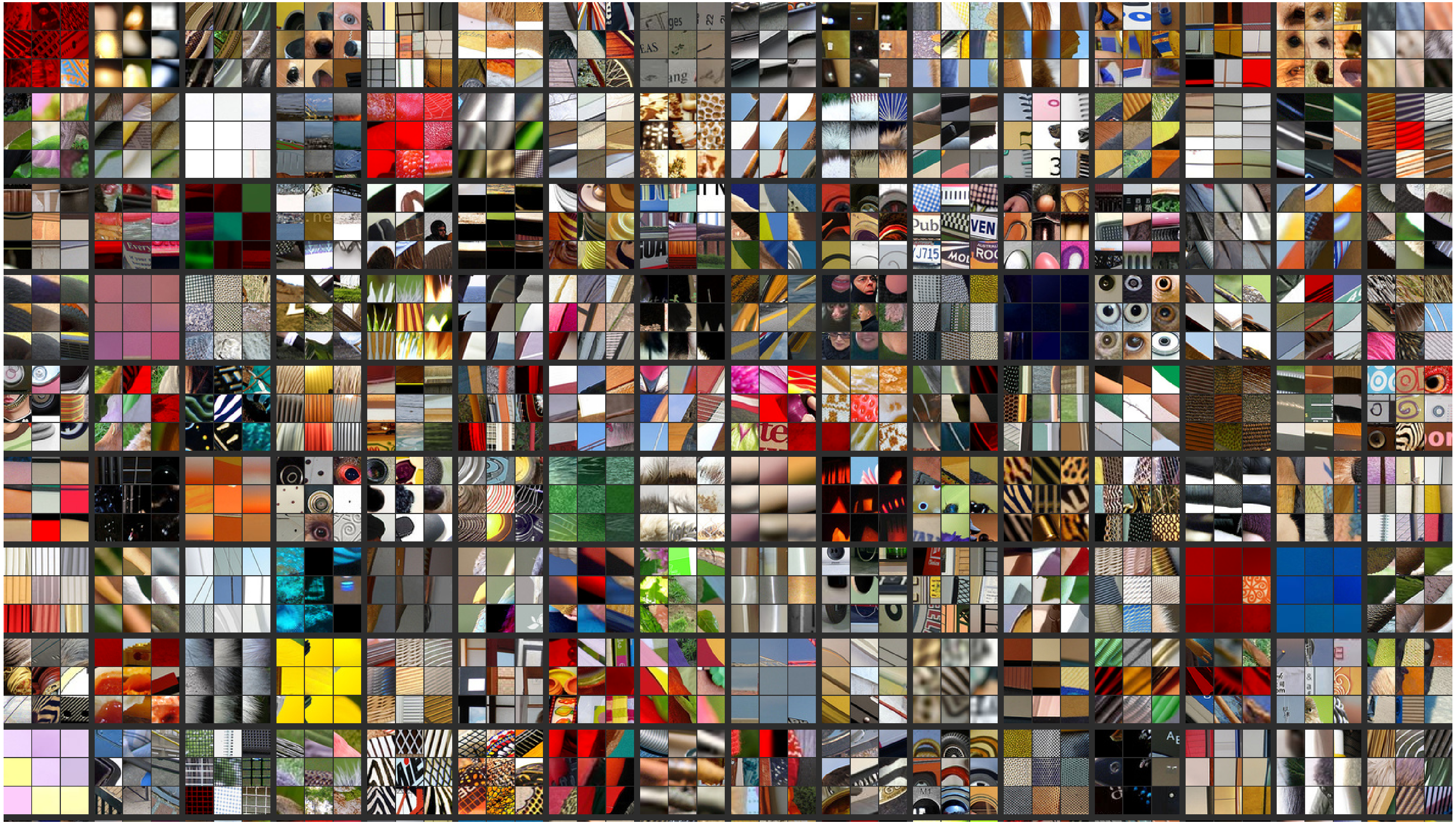
Source: https://paperswithcode.com/

Layer 1: Top-9 Patches

Layer 1 filters



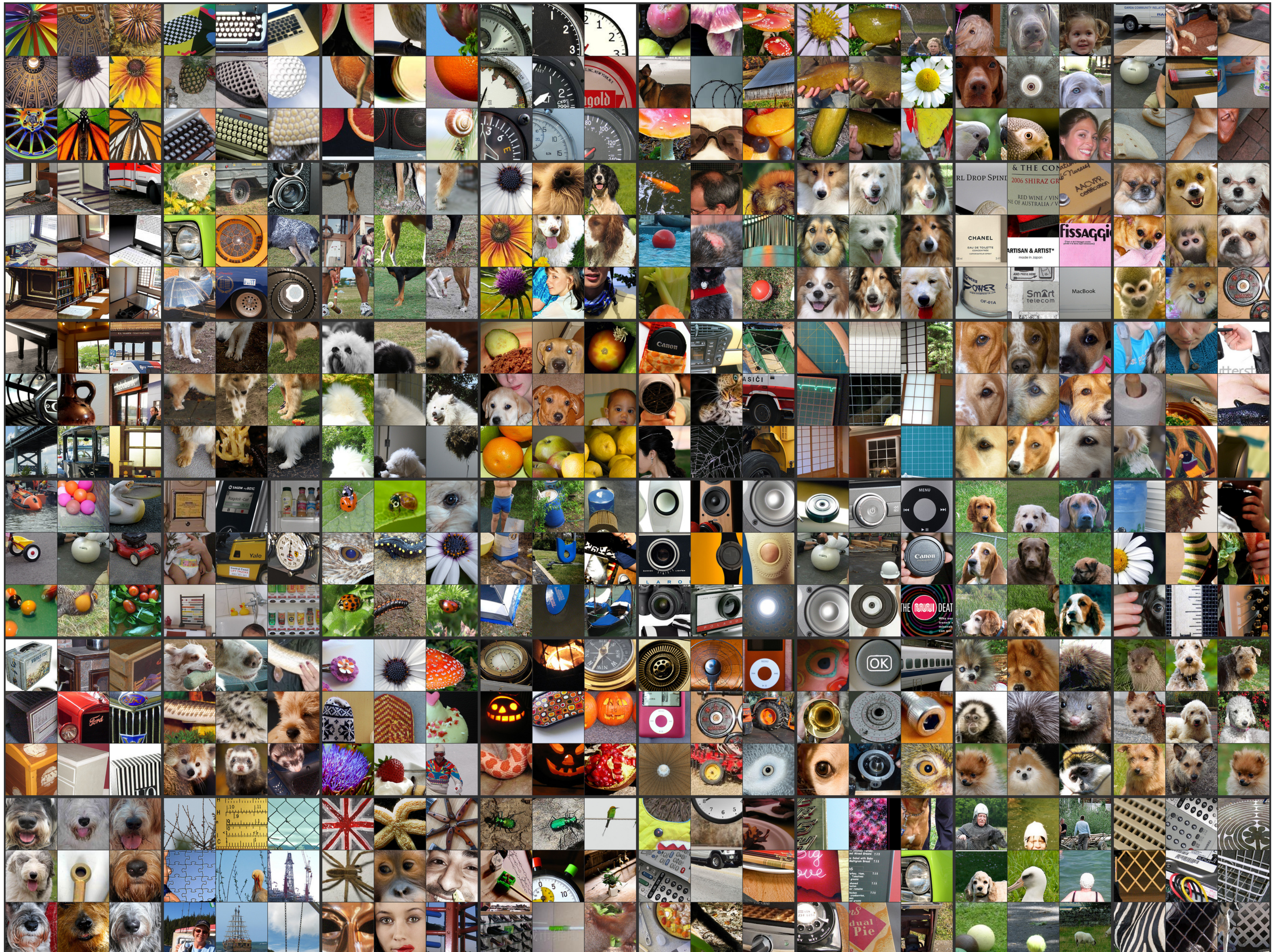
Layer 2: Top-9 Patches



- **Patches from validation images that give maximal activation of a given feature map**

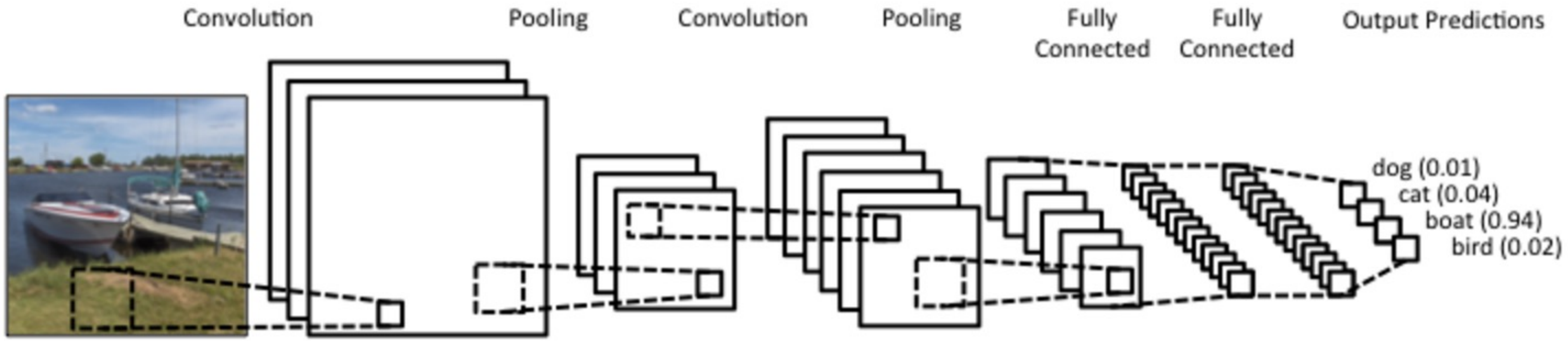
Layer 3: Top-9 Patches

Layer 4: Top-9 Patches



Layer 5: Top-9 Patches

Learned Representations are Useful in General



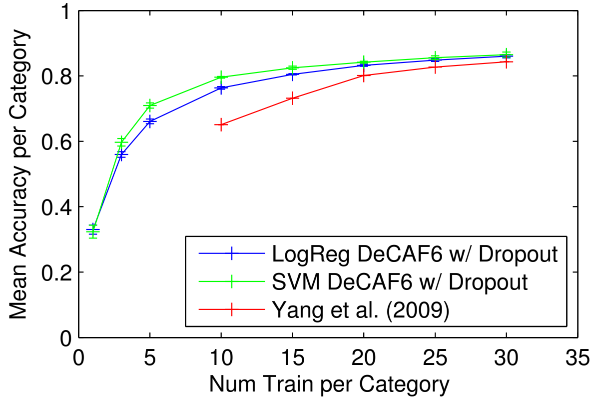
1. Features extracted from CNNs trained on ImageNet were effective for many CV tasks.
2. Furthermore, learned network weights serve as an excellent starting point for other tasks.

J. Donahue, Y. Jia et al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. ICML 2014

How to use a trained network for a new task?

	DeCAF_5	DeCAF_6	DeCAF_7
LogReg	63.29 ± 6.6	84.30 ± 1.6	84.87 ± 0.6
LogReg with Dropout	-	86.08 ± 0.8	85.68 ± 0.6
SVM	77.12 ± 1.1	84.77 ± 1.2	83.24 ± 1.2
SVM with Dropout	-	$\mathbf{86.91 \pm 0.7}$	85.51 ± 0.9
Yang et al. (2009)		84.3	
Jarrett et al. (2009)		65.5	

Caltech 101



Caltech 101

	Amazon → Webcam		
	SURF	DeCAF_6	DeCAF_7
Logistic Reg. (S)	9.63 ± 1.4	48.58 ± 1.3	53.56 ± 1.5
SVM (S)	11.05 ± 2.3	52.22 ± 1.7	53.90 ± 2.2
Logistic Reg. (T)	24.33 ± 2.1	72.56 ± 2.1	74.19 ± 2.8
SVM (T)	51.05 ± 2.0	78.26 ± 2.6	78.72 ± 2.3
Logistic Reg. (ST)	19.89 ± 1.7	75.30 ± 2.0	76.32 ± 2.0
SVM (ST)	23.19 ± 3.5	80.66 ± 2.3	79.12 ± 2.1
Daume III (2007)	40.26 ± 1.1	$\mathbf{82.14 \pm 1.9}$	81.65 ± 2.4
Hoffman et al. (2013)	37.66 ± 2.2	80.06 ± 2.7	80.37 ± 2.0
Gong et al. (2012)	39.80 ± 2.3	75.21 ± 1.2	77.55 ± 1.9
Chopra et al. (2013)		58.85	

Domain Adaptation

Method	Accuracy
DeCAF_6	58.75
DPD + DeCAF_6	**64.96**
DPD (Zhang et al., 2013)	50.98
POOF (Berg & Belhumeur, 2013)	56.78

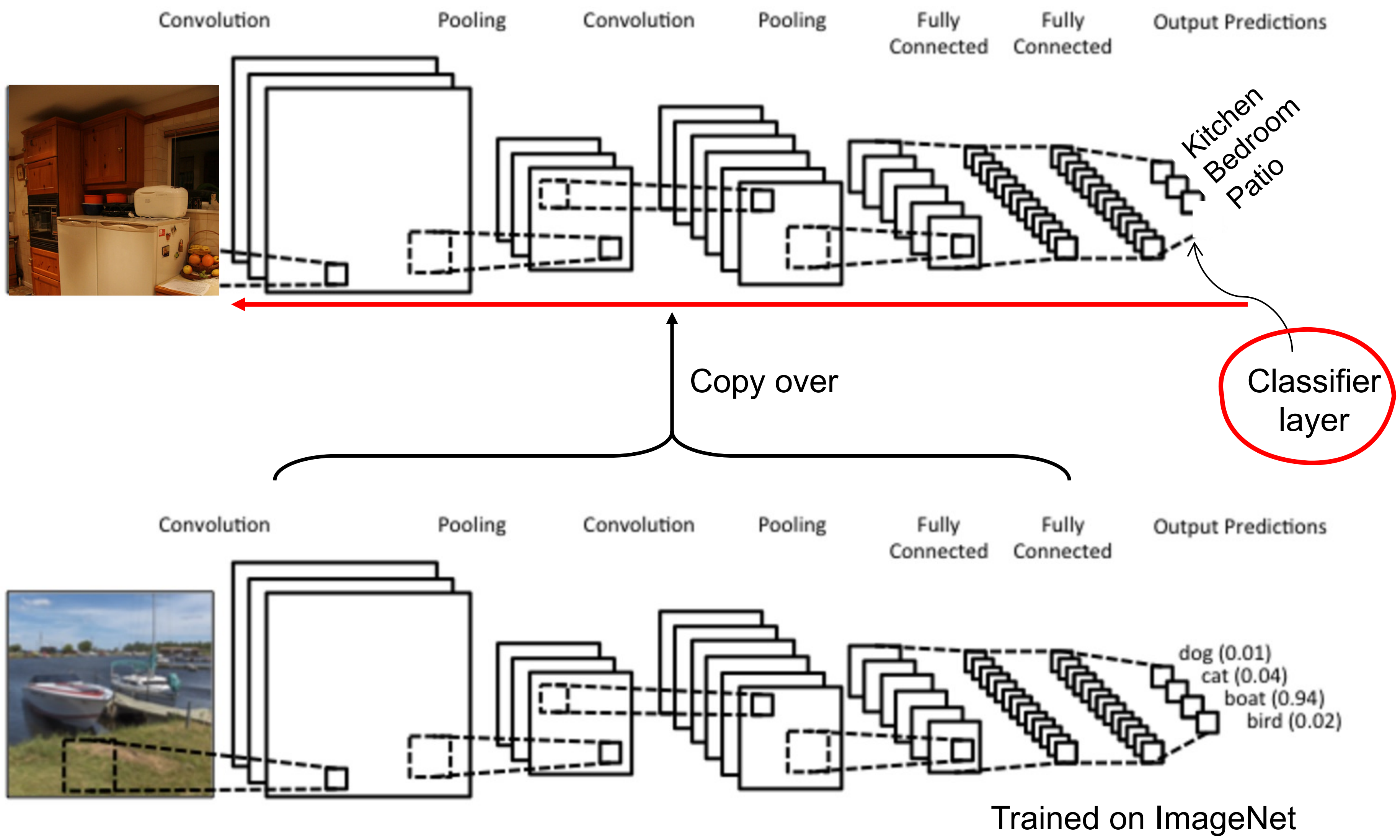
Fine-grained Classification

	DeCAF_6	DeCAF_7
LogReg	$\mathbf{40.94 \pm 0.3}$	40.84 ± 0.3
SVM	39.36 ± 0.3	40.66 ± 0.3
Xiao et al. (2010)	38.0	

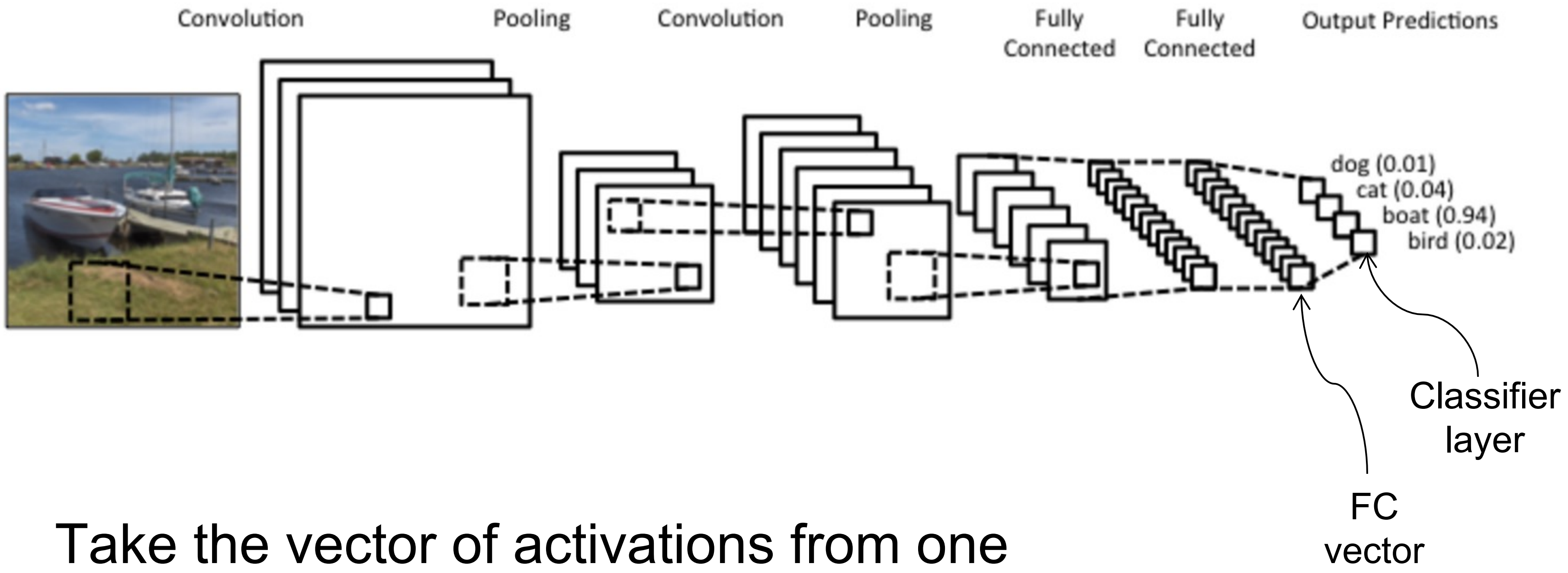
Scene Classification

J. Donahue, Y. Jia et al. [DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition](). ICML 2014

How to use a trained network for a new task?



How to use a trained network for a new task?



- Take the vector of activations from one of the fully connected (FC) layers and treat it as an off-the-shelf feature
  - Train a new classifier layer on top of the FC layer
- *Fine-tune* the whole network