Pretrained CNN Features and Fine-Tuning

Danna Gurari
University of Colorado Boulder
Fall 2022

Review

- Last lecture:
    - Regularization
    - Parameter norm penalty
    - Early stopping
    - Dataset augmentation
    - Dropout
    - Batch Normalization
- Assignments (Canvas)
    - Lab assignment 2 due Wednesday
- Questions?

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

Today’s Topics

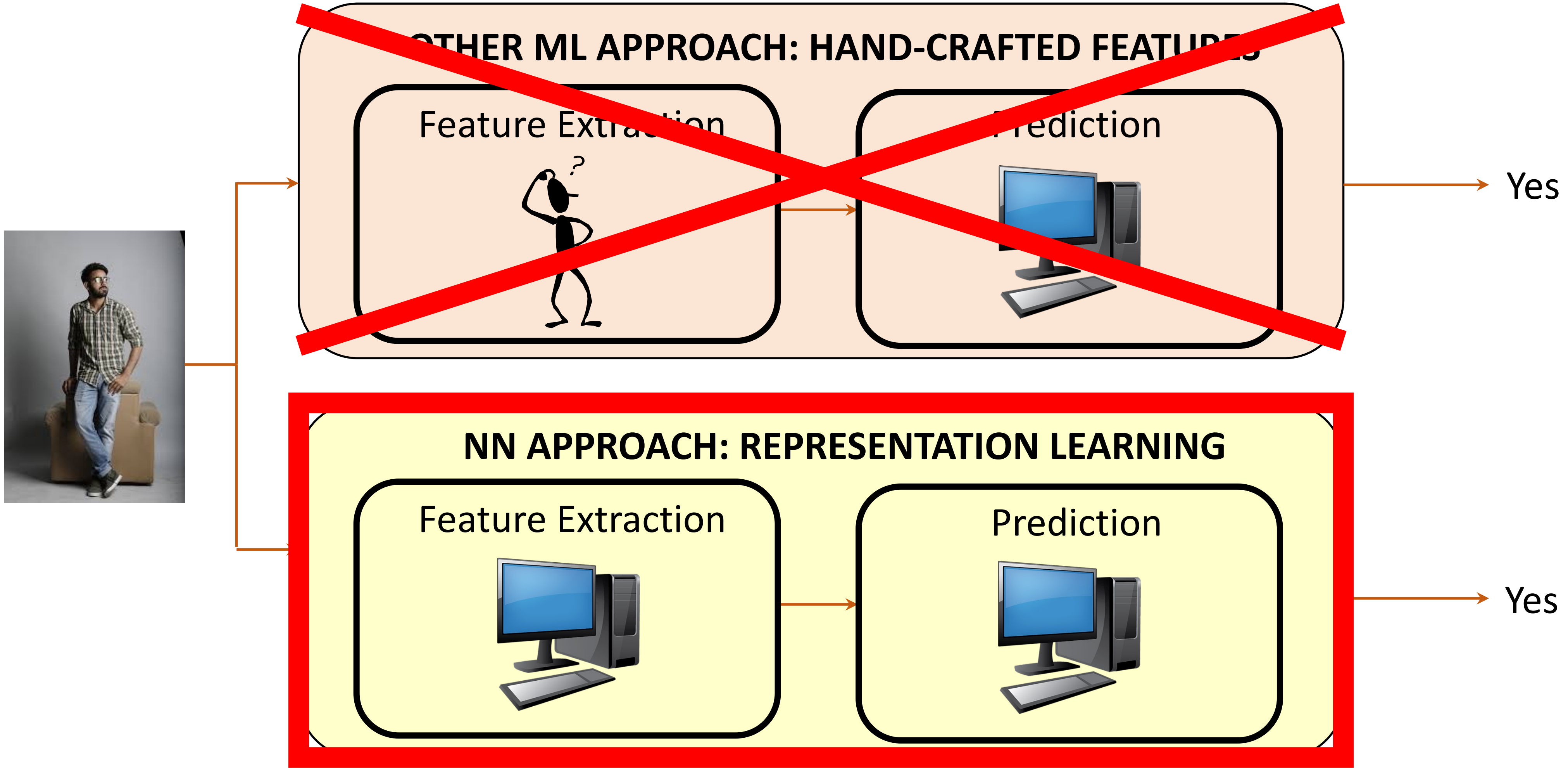
- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

Recall: Motivation for Neural Networks (NNs) Over Other Machine Learning (ML) Approaches

INPUT

e.g., Is a person present?

OUTPUT



What Neural Networks Learn

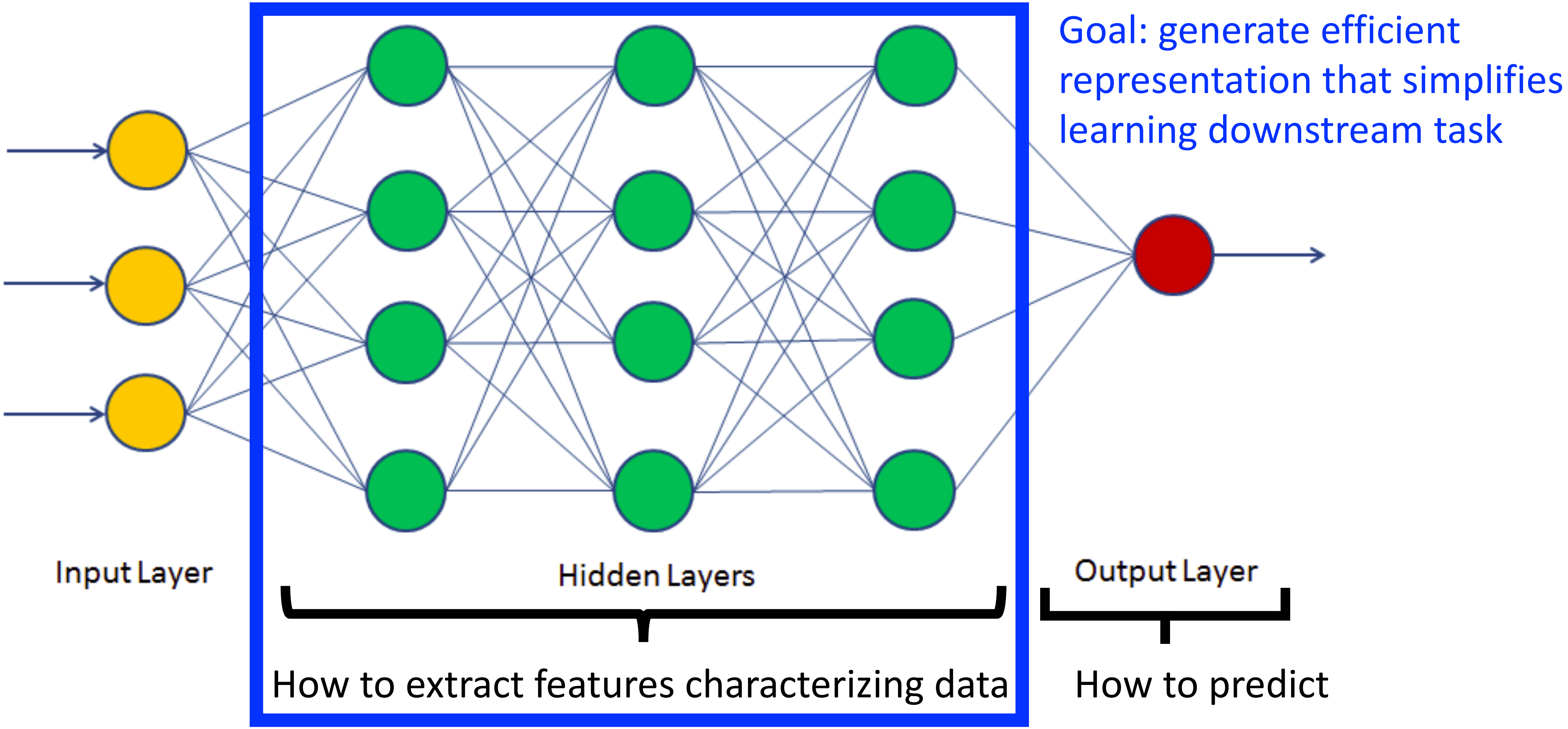
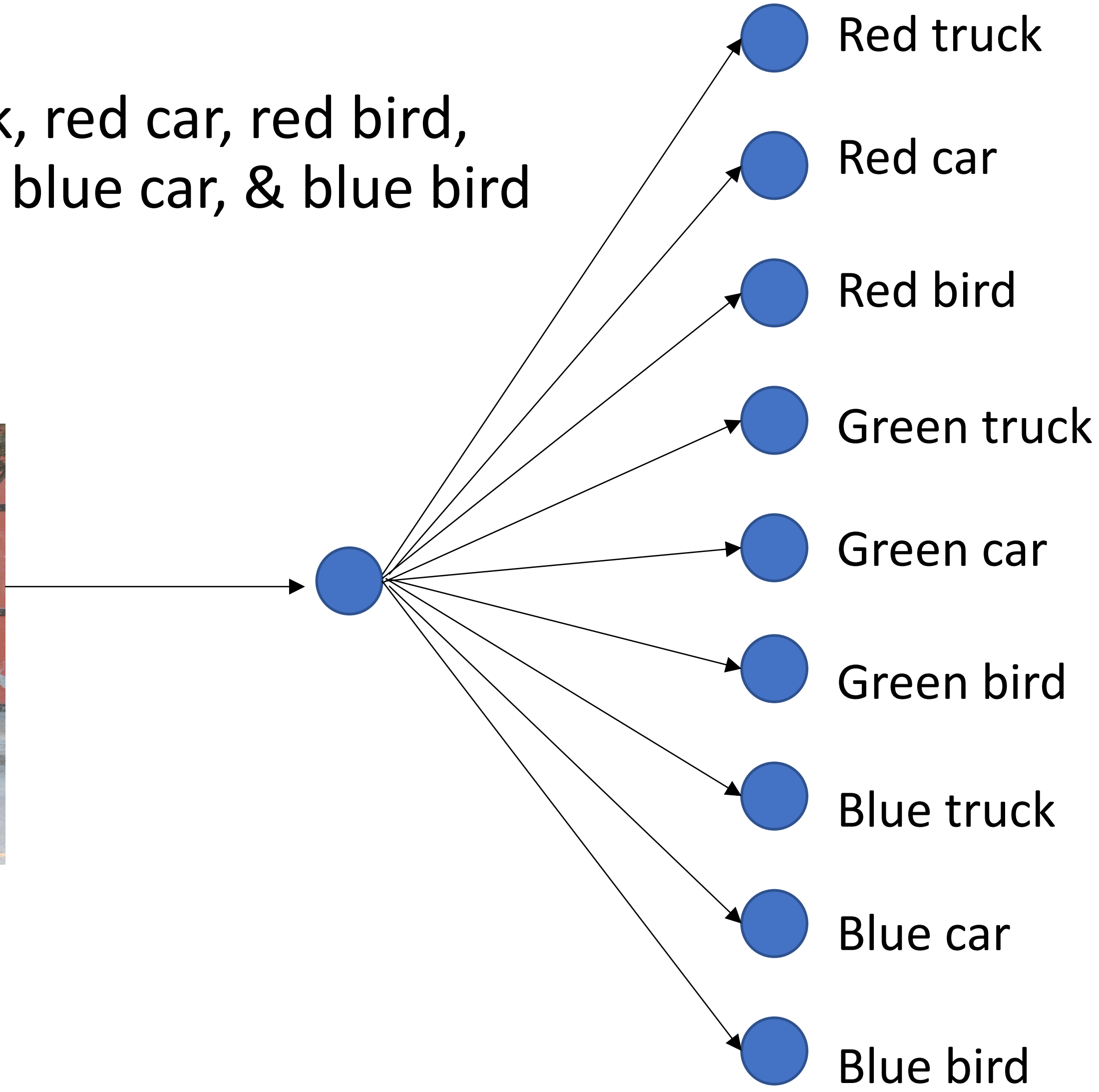


Figure Source: https://www.datacamp.com/community/tutorials/neural-network-models-r

How to Efficiently Describe/Represent Images?

e.g., predict for given image if it is a: red truck, red car, red bird, green truck, green car, green bird, blue truck, blue car, & blue bird

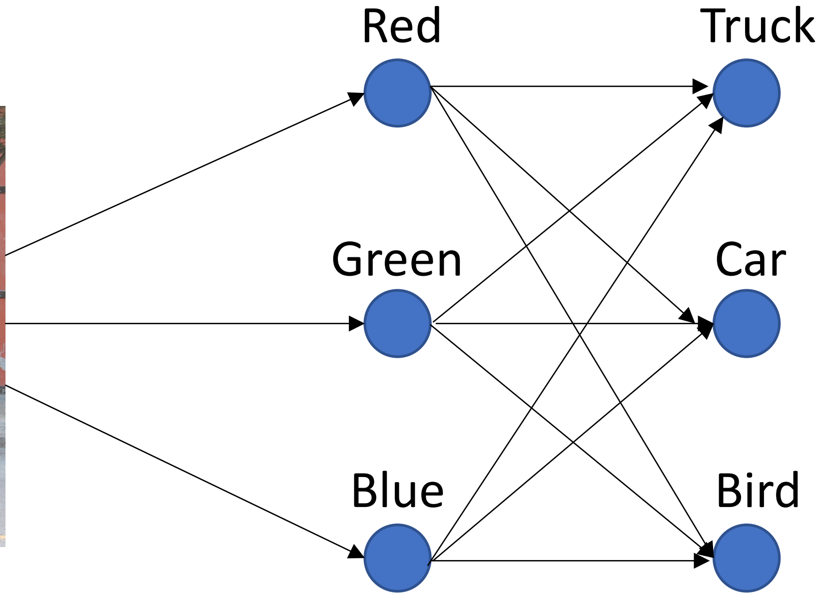
e.g., novel image



How to Efficiently Describe/Represent Images?

e.g., predict for given image if it is a: red truck, red car, red bird, green truck, green car, green bird, blue truck, blue car, & blue bird

e.g., novel image



Can design a more efficient model to first capture color and then objects (greater parameter efficiency using **hierarchical layers** of features)!

What representations are CNNs learning?

Key Tricks for Interpreting Representations

- Visualize filters and resulting activation maps
- Retrieve similar images based on feature similarity
- Analyze images that maximally activate units in a network

Key Tricks for Interpreting Representations

- Visualize filters and resulting activation maps
- Retrieve similar images based on feature similarity
- Analyze images that maximally activate units in a network

Inspecting What Was Learned: VGG16

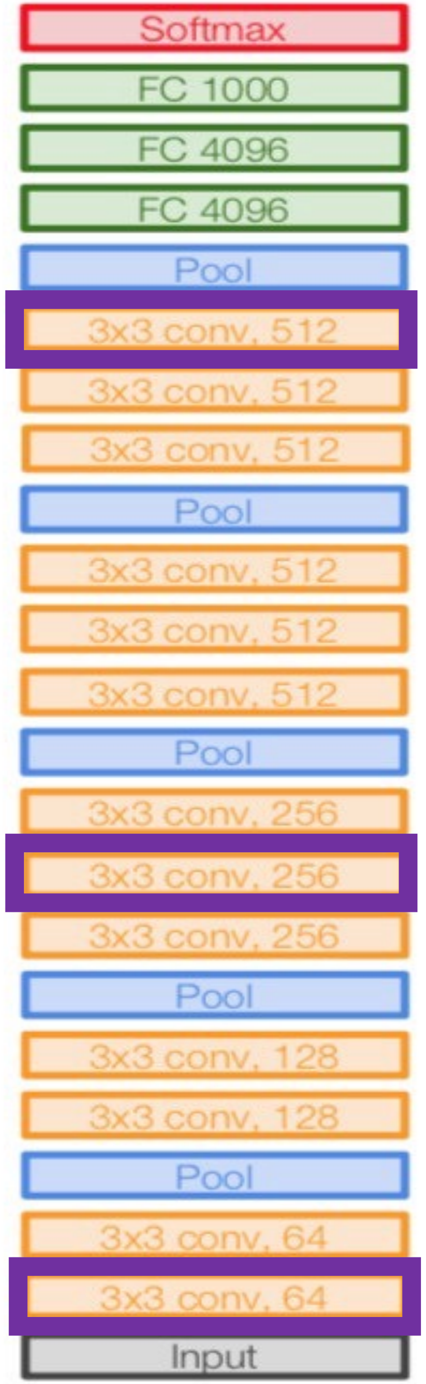
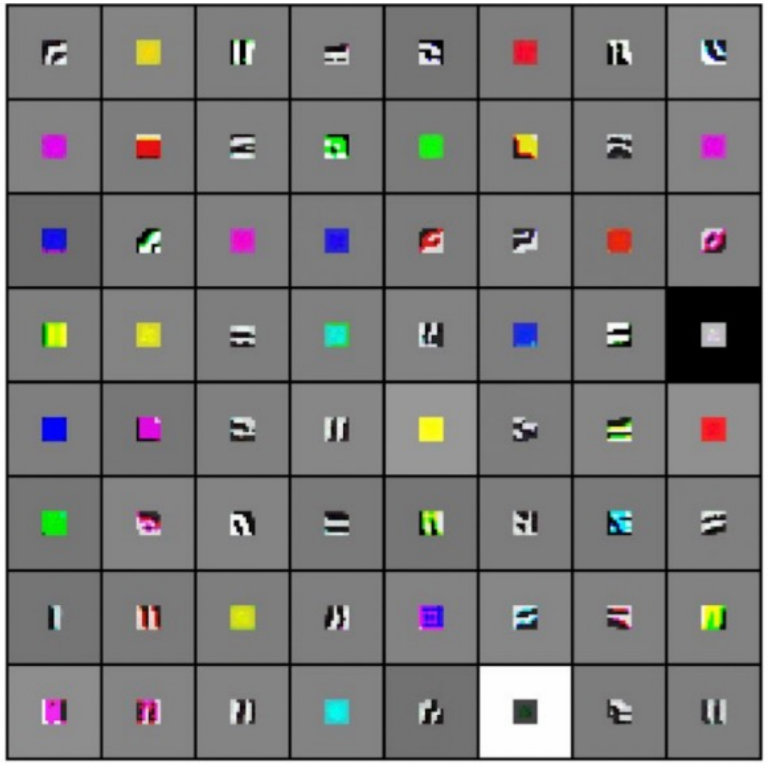
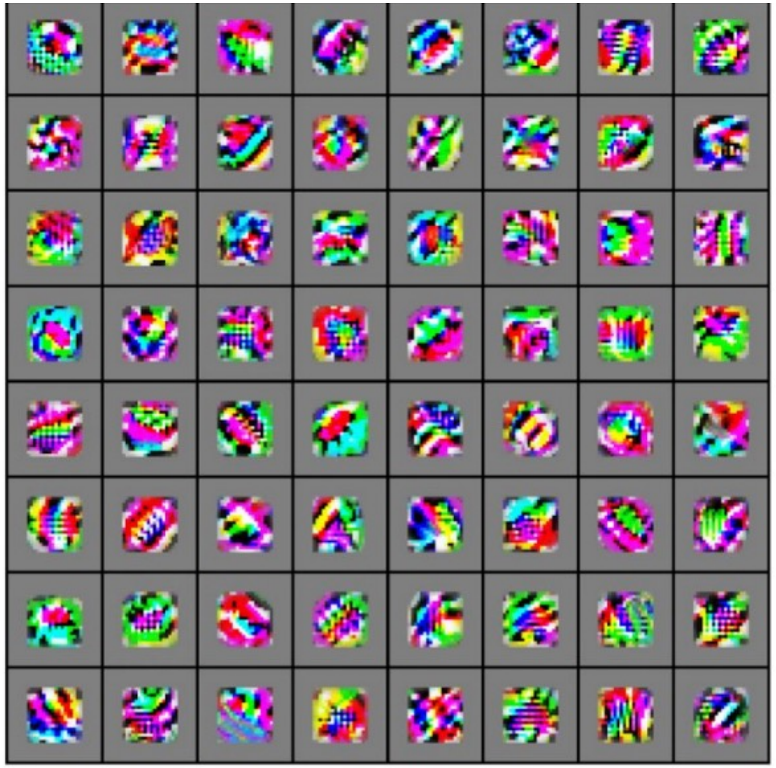


Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

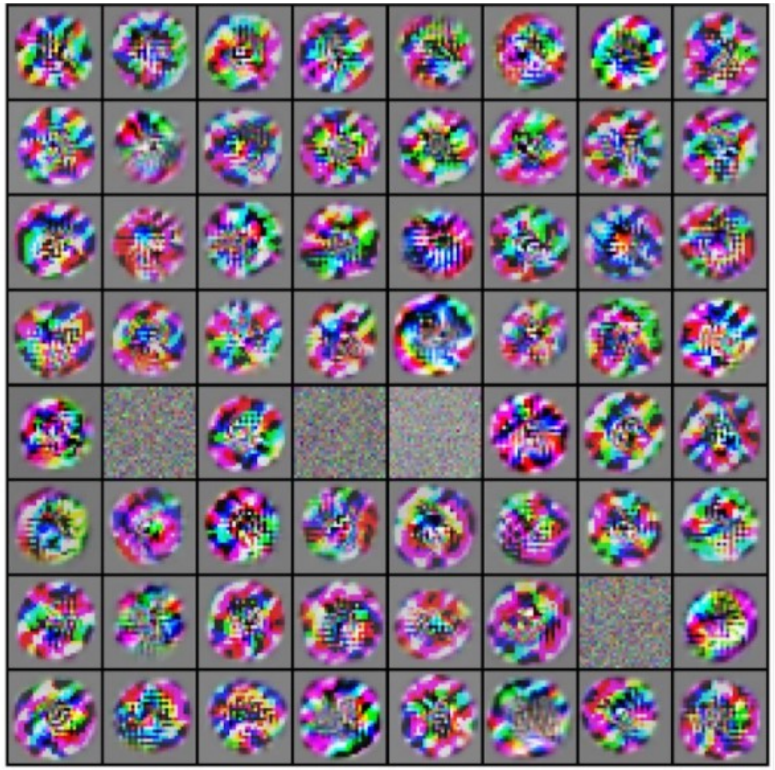
VGG16: Filters at 3 Convolutional Layers



VGG-16 Conv1_1



VGG-16 Conv3_2

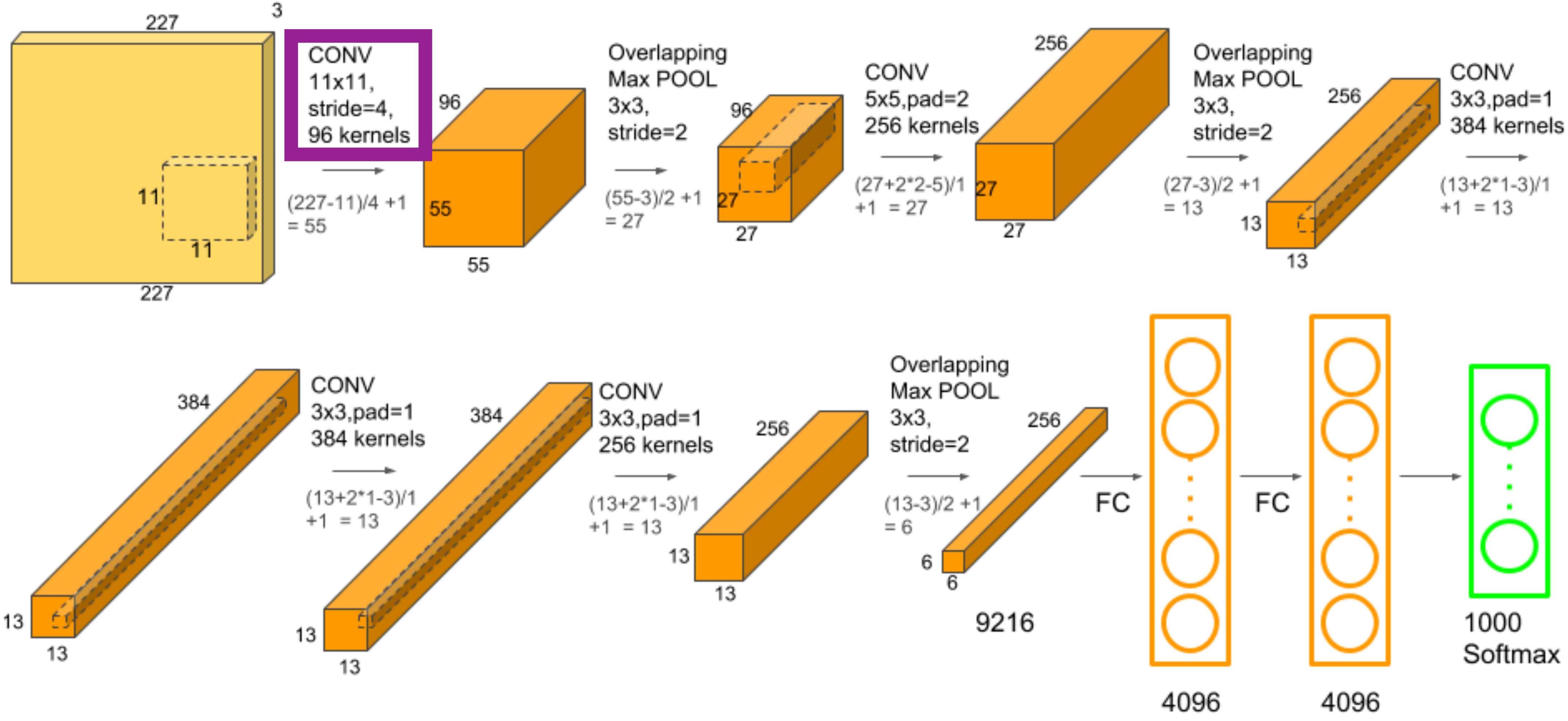


VGG-16 Conv5_3

Filters evolve from detecting simple features (e.g., edges, colors) to complex structures

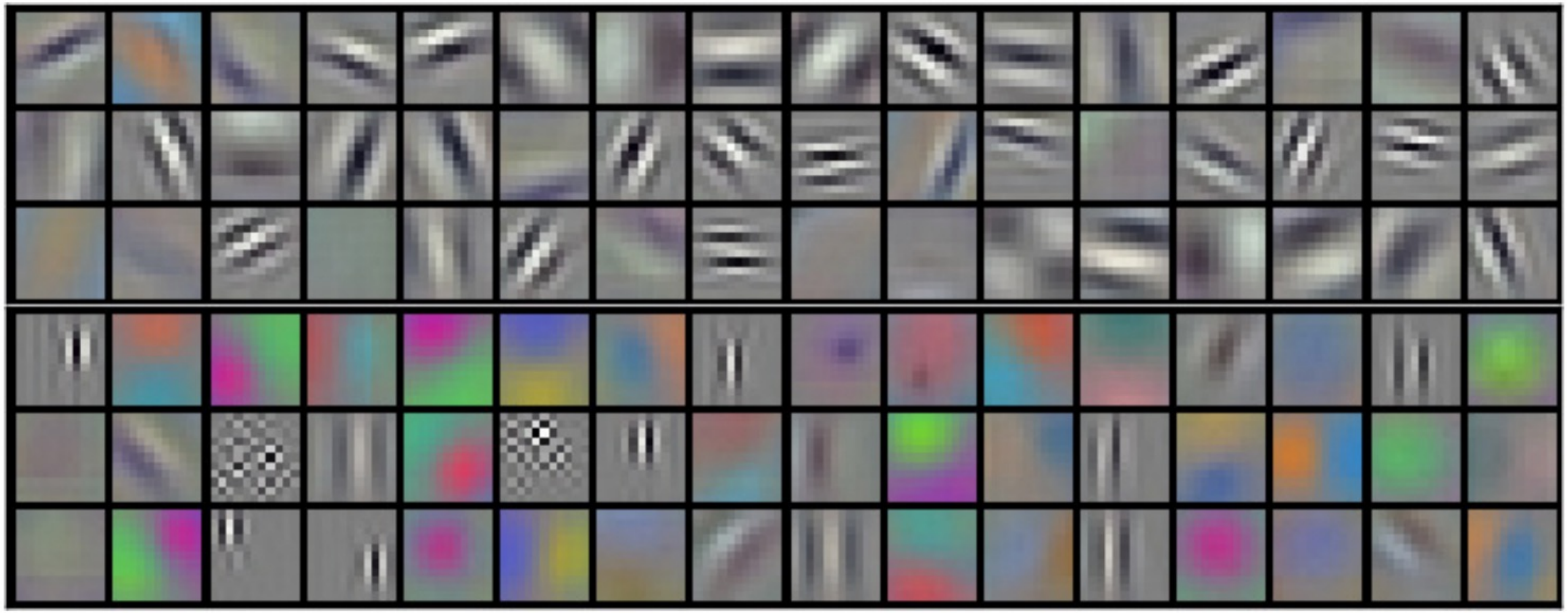
http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture05.pdf

Inspecting What Was Learned: AlexNet



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

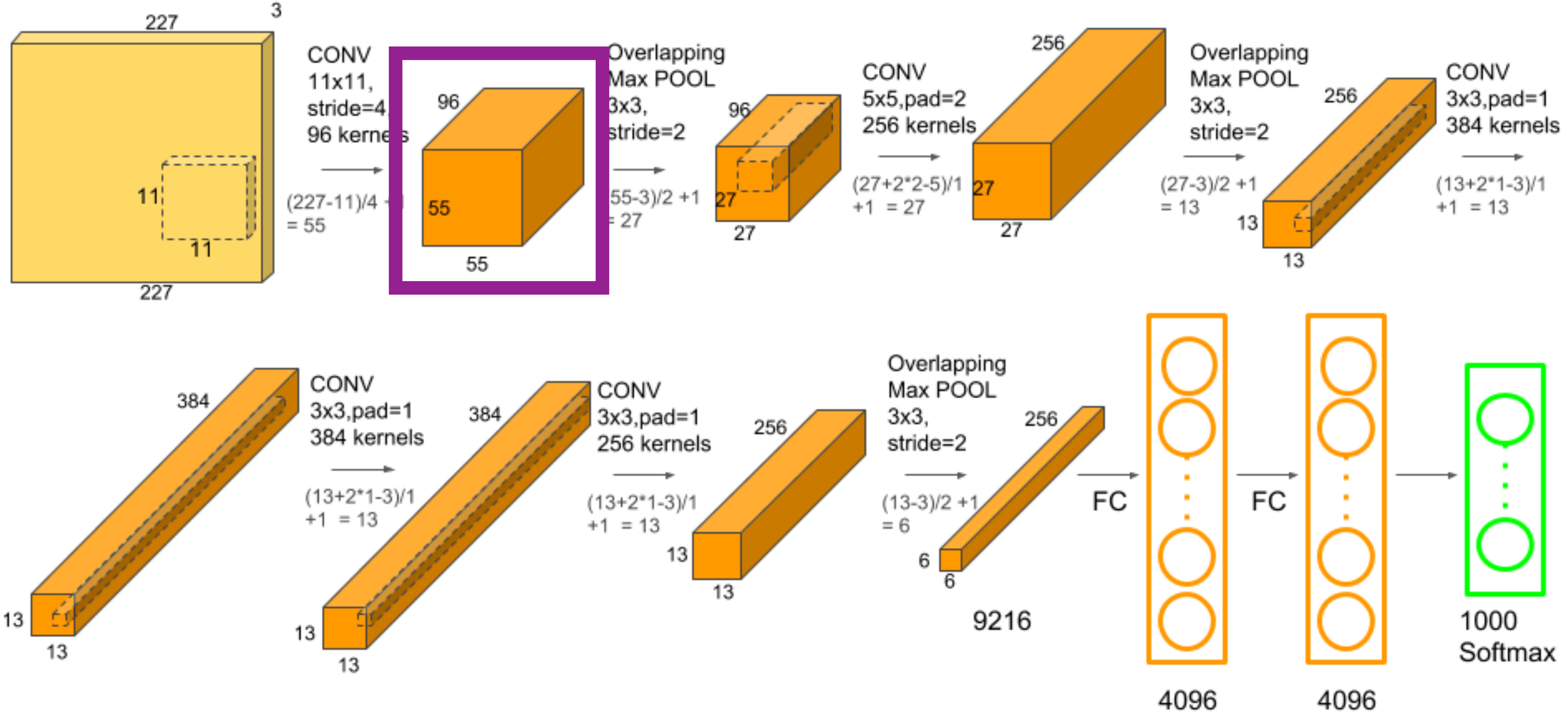
AlexNet: 96 Filters in Convolutional Layer 1



Filters for detecting different frequencies, orientations, and colors!

Krizhevsky, Sutskever, and Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NeurIPS 2012.

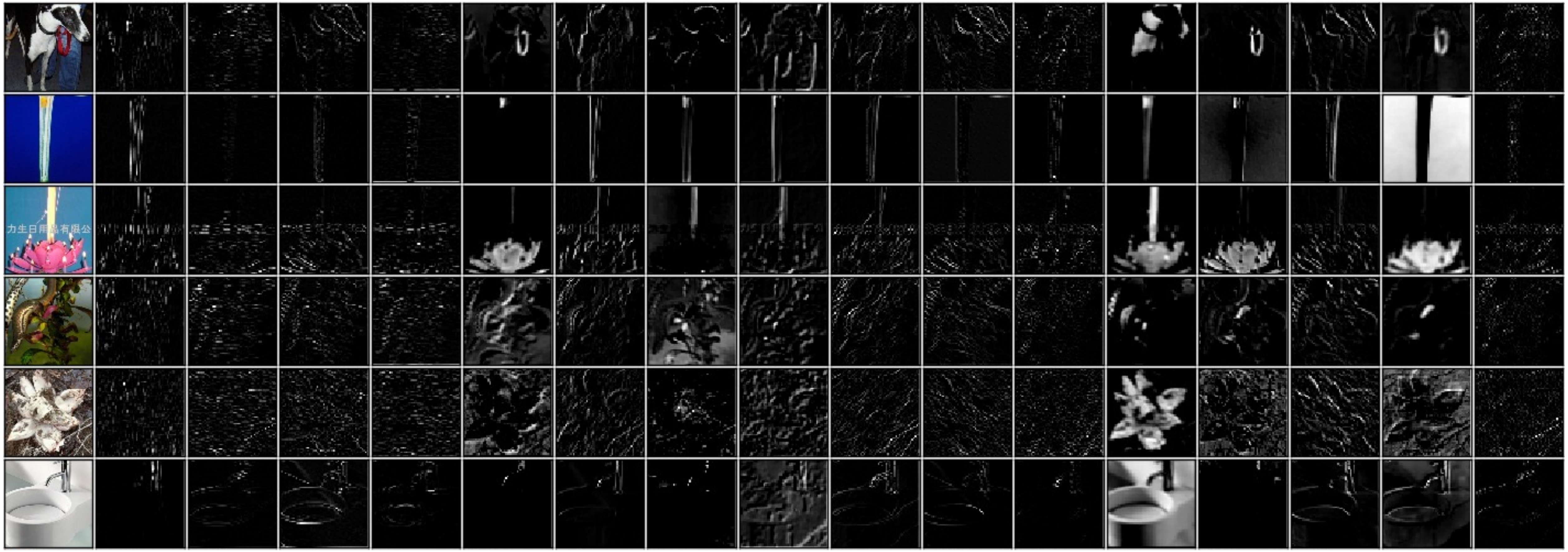
AlexNet: Example Activation Maps



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

AlexNet: Example Activation Maps (Recall Each Map Results from One Filter)

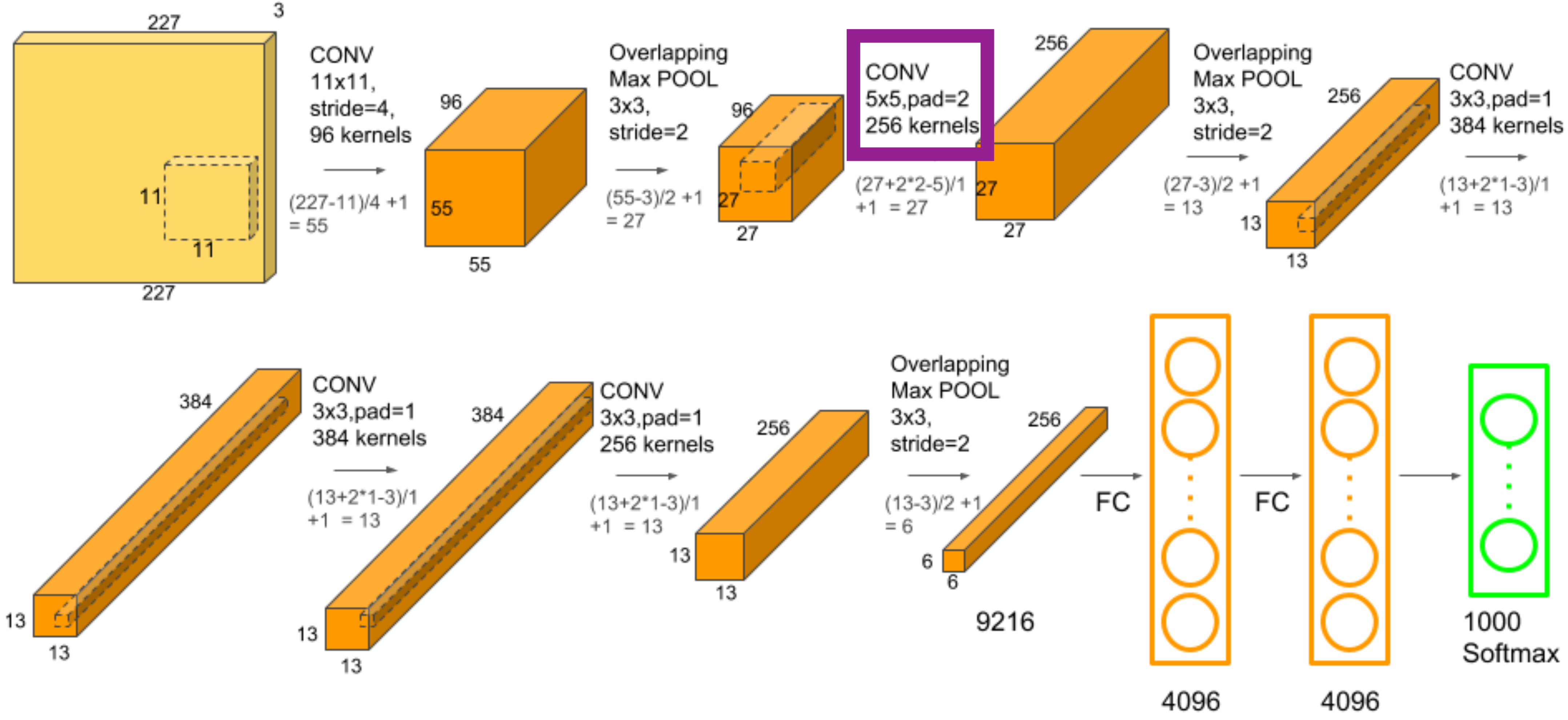
Images



Frequencies, orientations, and colors are detected

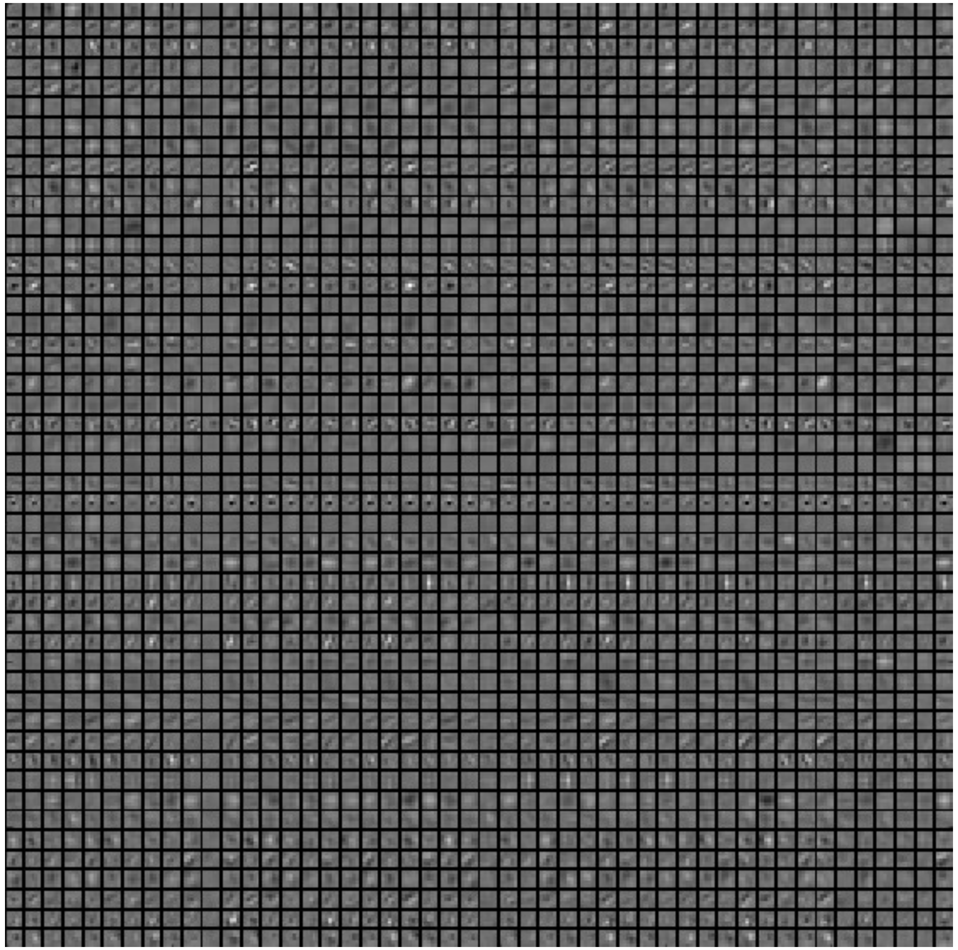
Krizhevsky, Sutskever, and Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NeurIPS 2012.

Inspecting What Was Learned: AlexNet



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

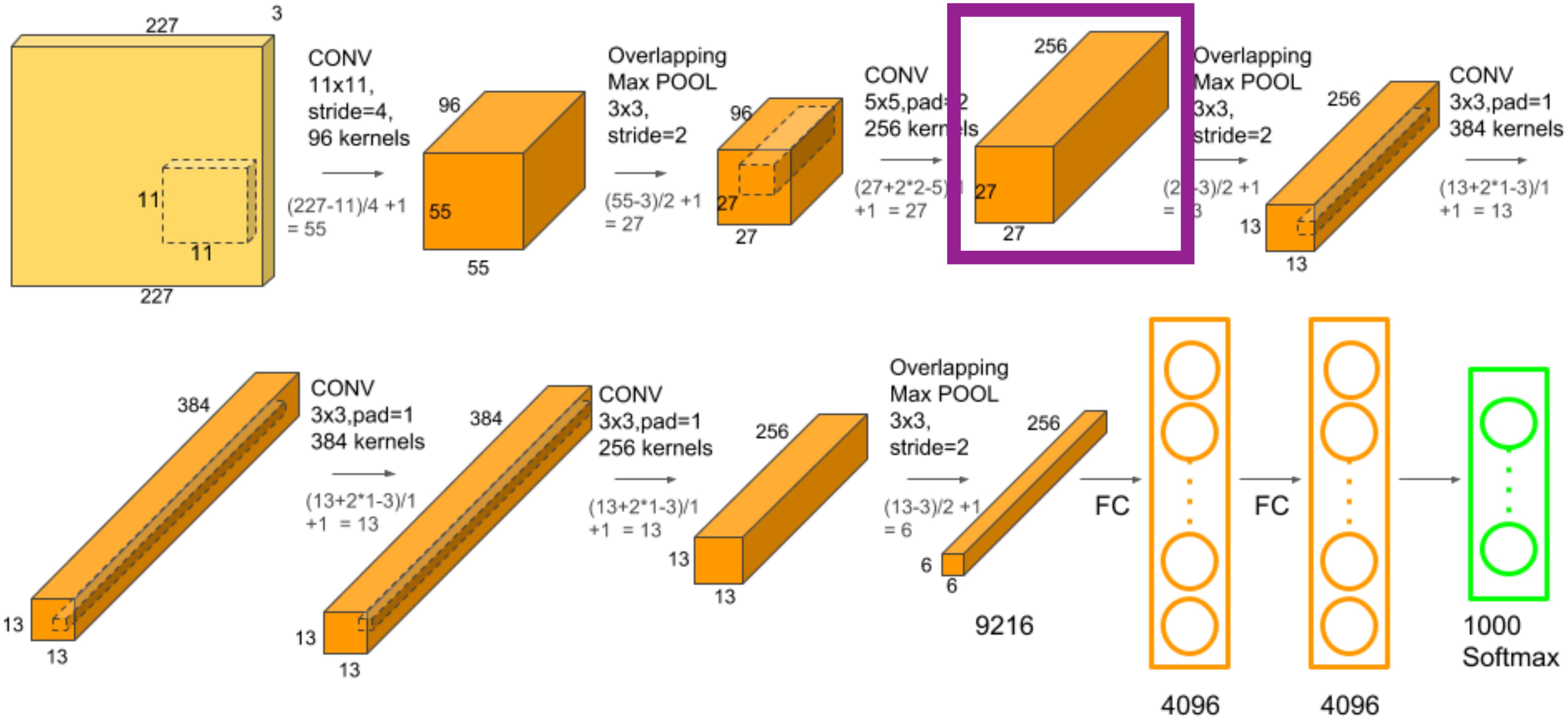
AlexNet: 256 Filters in Convolutional Layer 2



Challenging to interpret these learned filters

https://cs231n.github.io/understanding-cnn/

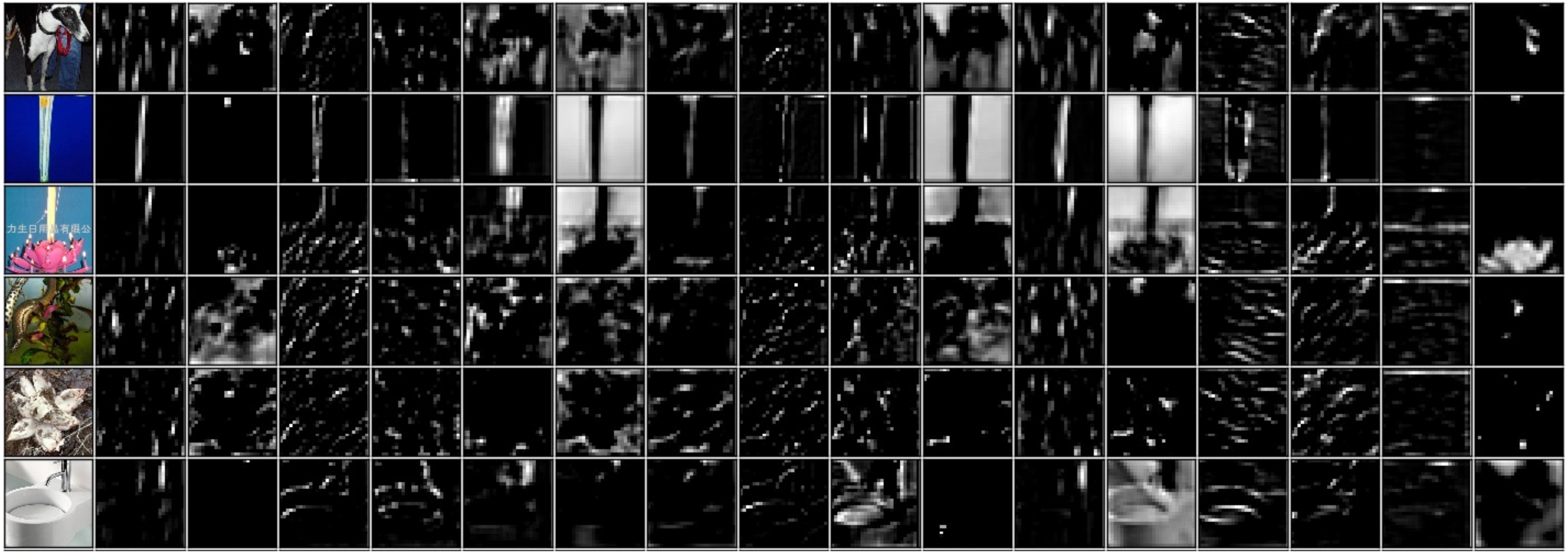
Inspecting What Was Learned: AlexNet



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

AlexNet: Sampled Activation Maps (Recall Each Map Results from One Filter)

Images



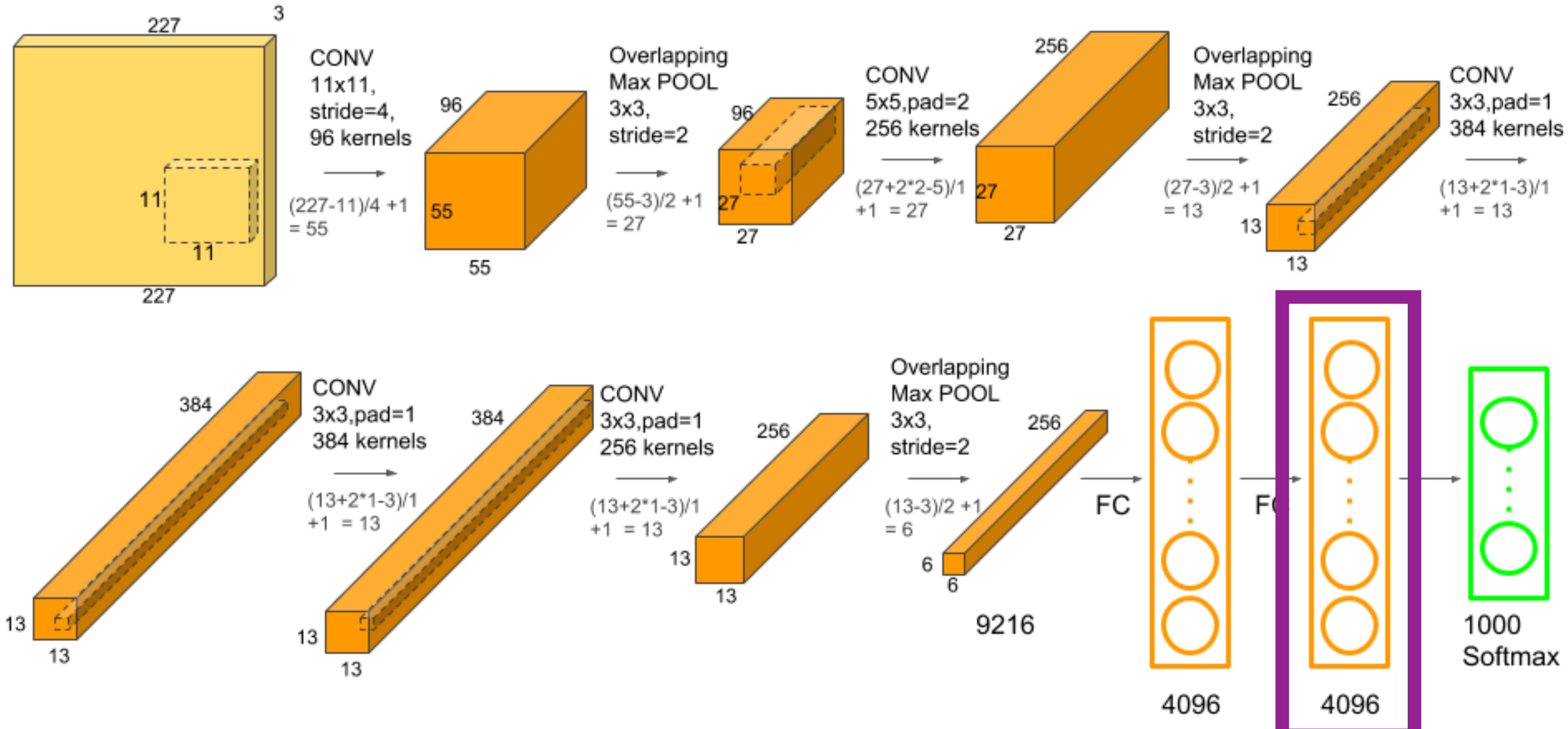
Can you infer anything about what features the filters extracted?

Krizhevsky, Sutskever, and Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NeurIPS 2012.

Key Tricks for Interpreting Representations

- Visualize filters and resulting activation maps
- Retrieve similar images based on feature similarity
- Analyze images that maximally activate units in a network

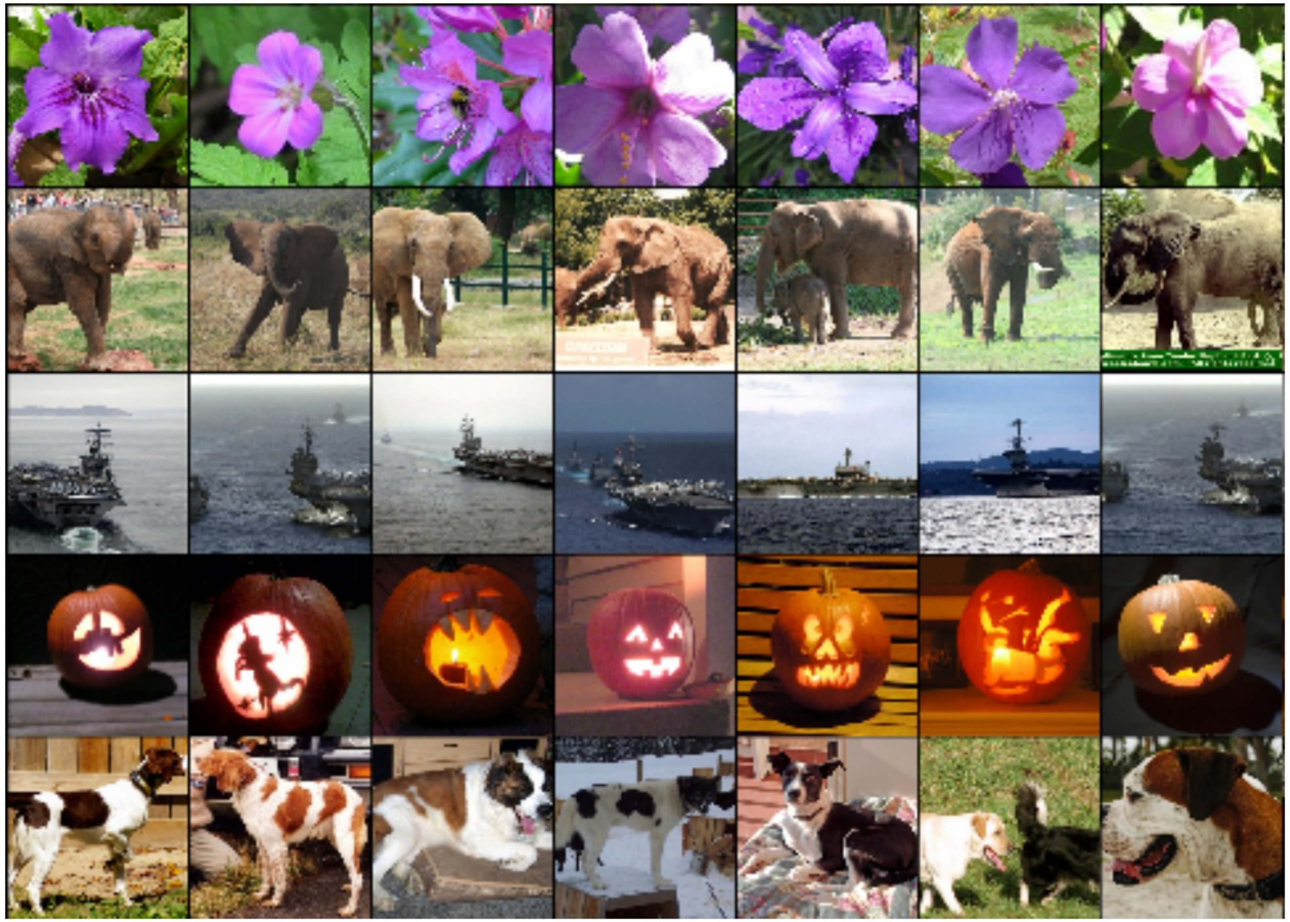
Inspecting What Was Learned: AlexNet



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

AlexNet: Retrieve Images with Similar FC7 Vectors

Test images | Training images with smallest Euclidean distance between its FC7 feature activation and that of the test image



What can you infer about what the FC7 feature represents?

- Image semantics regardless of illumination and object pose

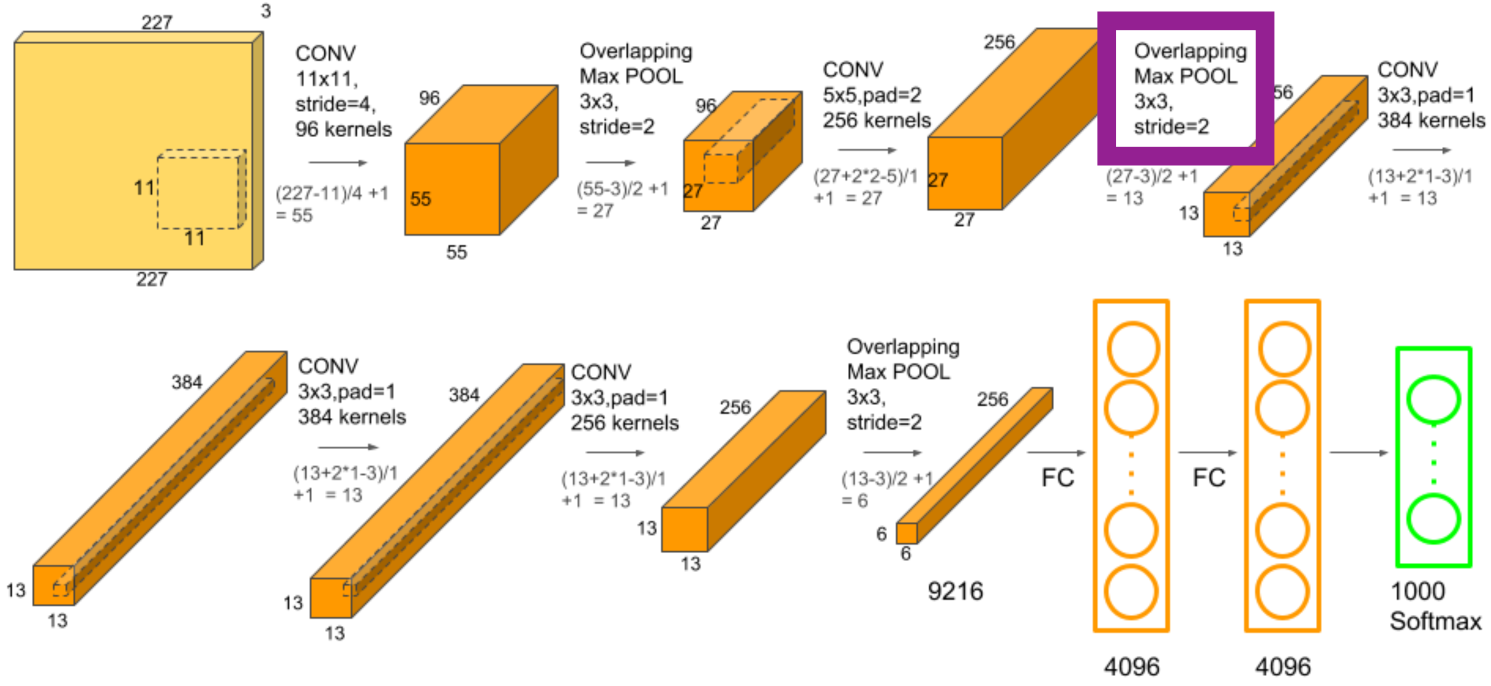
Krizhevsky, Sutskever, and Hinton. ImageNet Classification with Deep Convolutional Neural Networks. NeurIPS 2012.

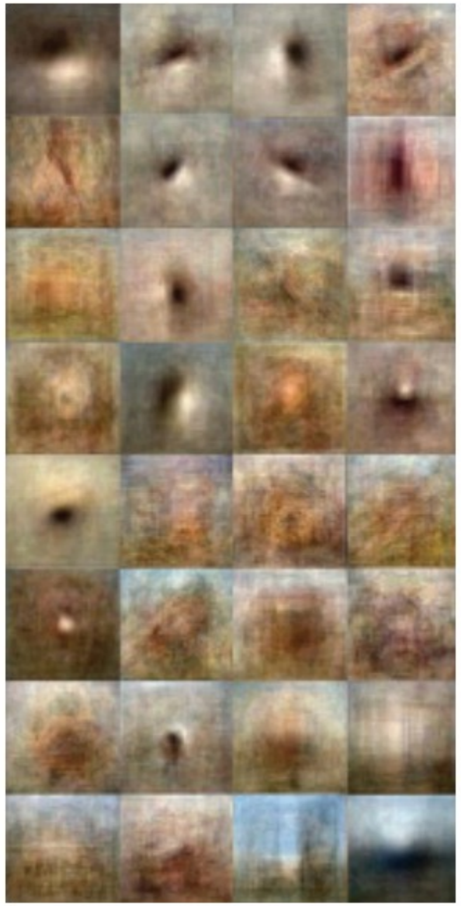
Key Tricks for Interpreting Representations

- Visualize filters and resulting activation maps
- Retrieve similar images based on feature similarity
- Analyze images that maximally activate units in a network

AlexNet: Images that Lead to Maximal Activations

Mean images from the 100 test images for each unit in each layer that fire the most (i.e., highest activation scores); e.g.,





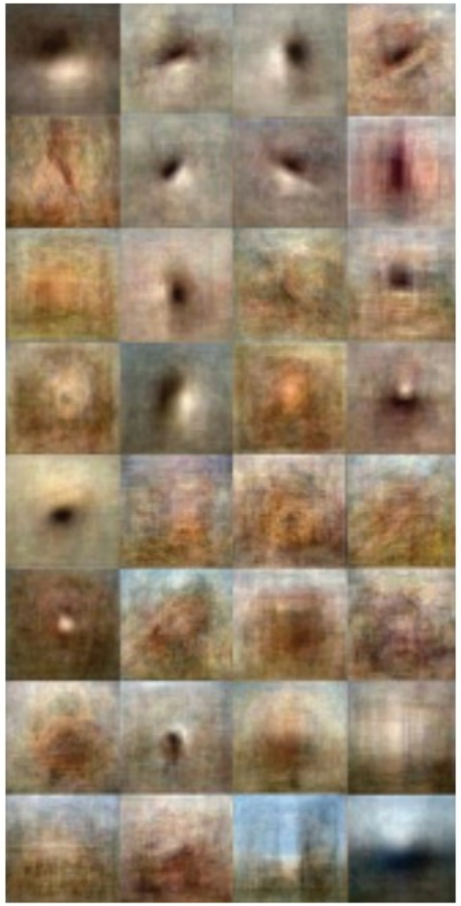
Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

AlexNet: Images that Lead to Maximal Activations

Mean images from the 100 test images for each unit in each layer that fire the most (i.e., highest activation scores); e.g.,

What type of features does the model appear to detect?

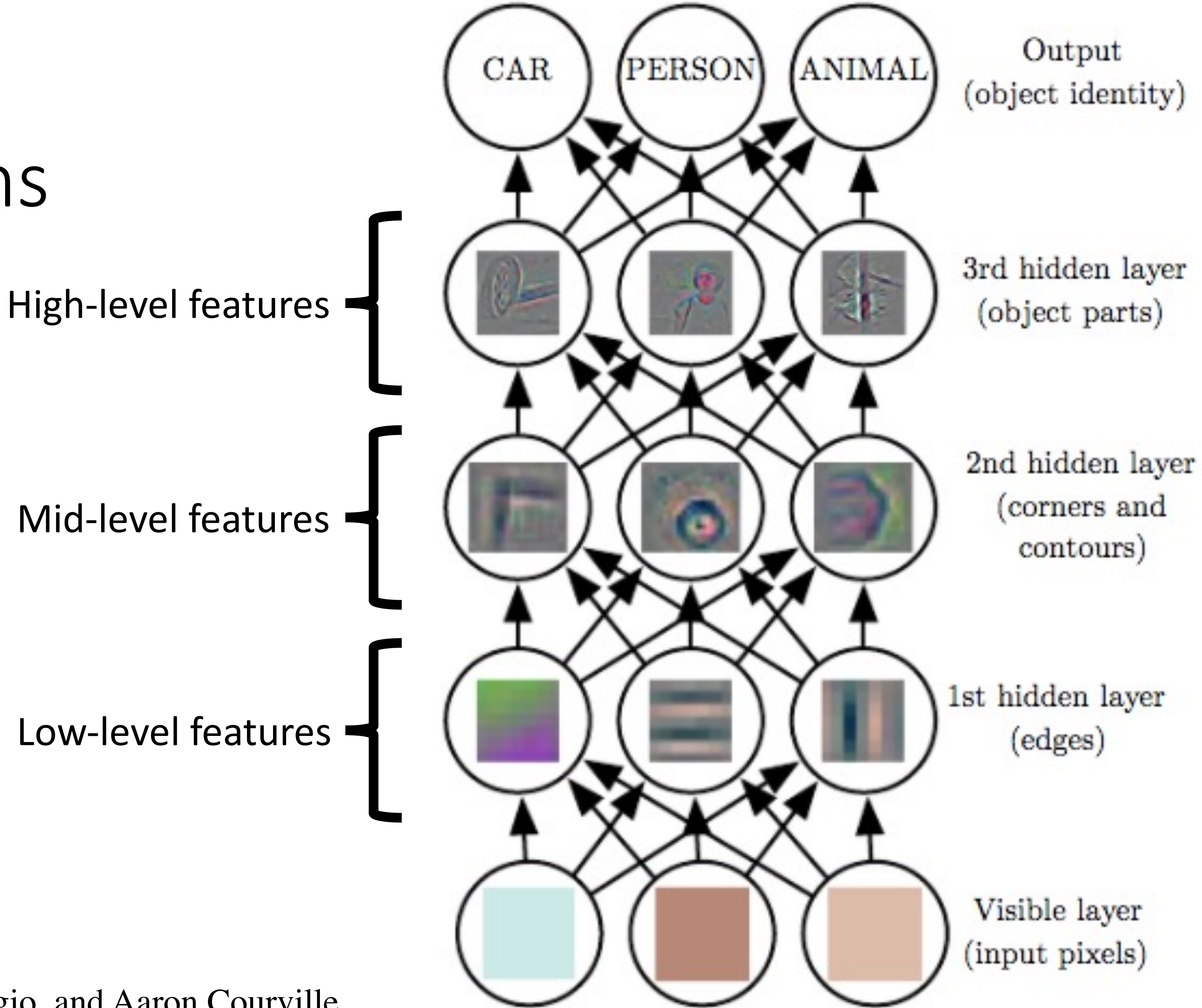


Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

Summary: Key Tricks for Interpreting Representations

- Visualize filters and resulting activation maps
- Retrieve similar images based on feature similarity
- Analyze images that maximally activate units in a network
- And many newer techniques not covered in this course…

CNN: Common Representations



Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Online Tool for Investigating CNNs

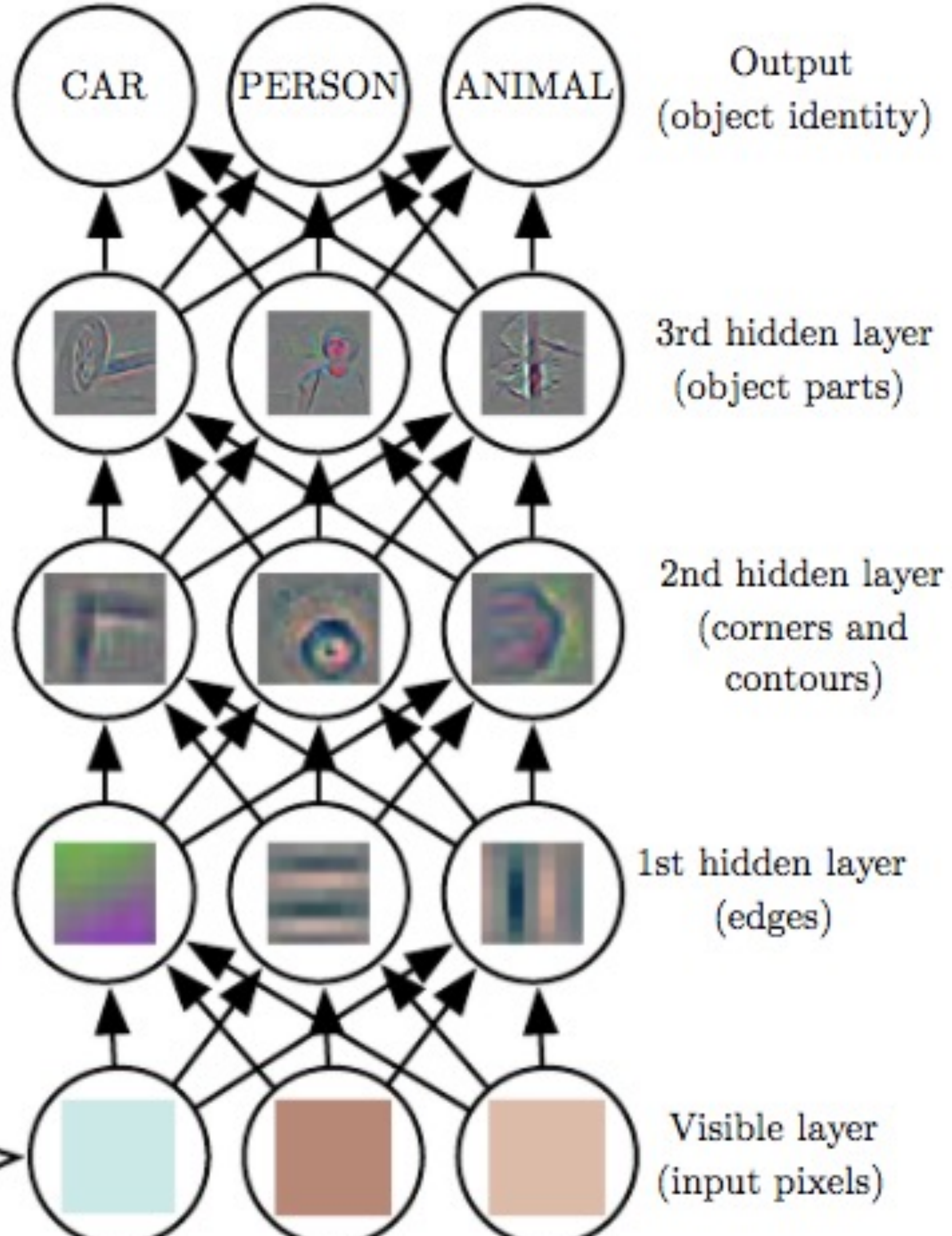
- https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

CNN: Pretrained Features

A representation of the data extracted inside a network (rather than the input or predicted output)

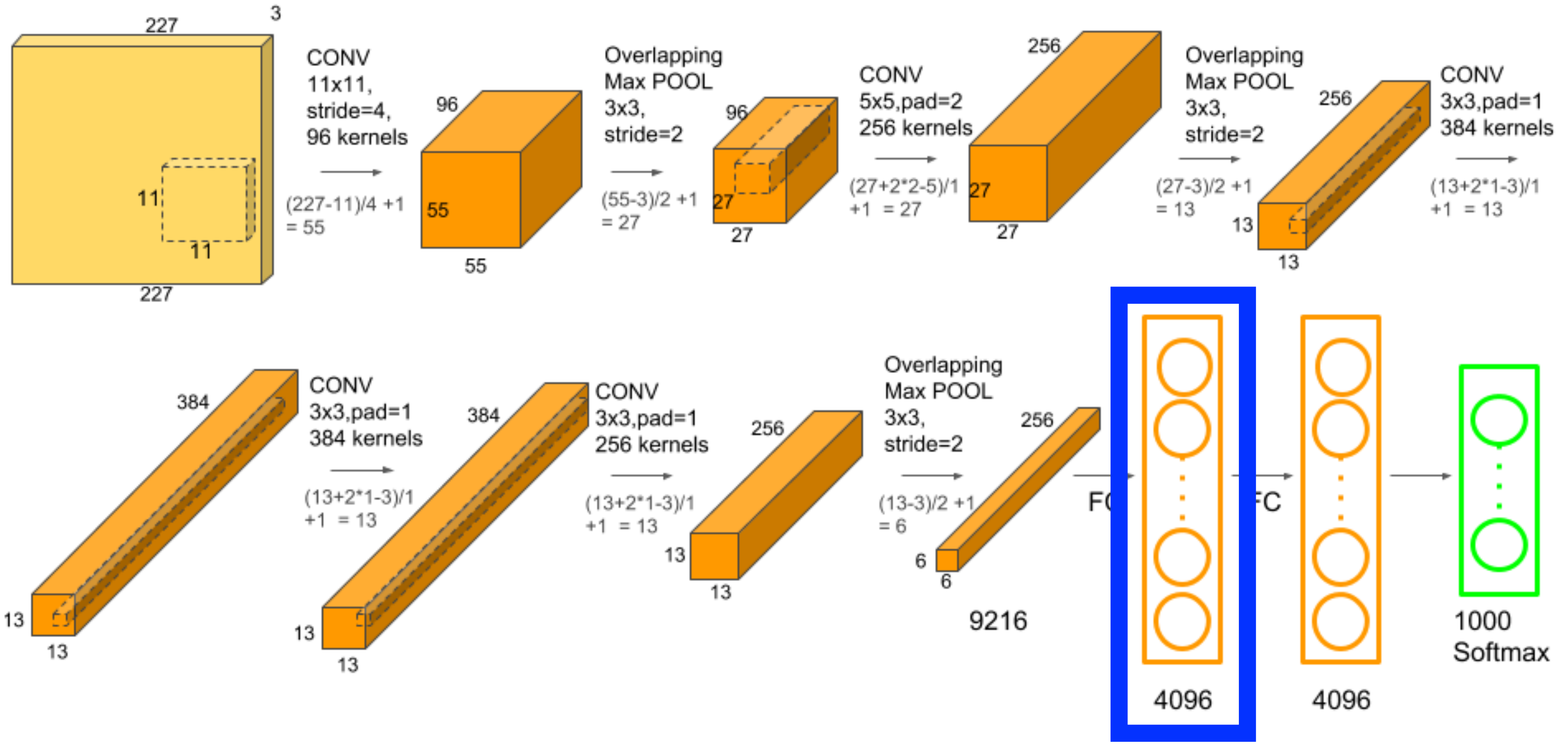


Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville

CNN: Pretrained Features (e.g., AlexNet)

What is the dimensionality of the FC6 layer?

A representation of the data extracted inside a network (rather than the input or predicted output)

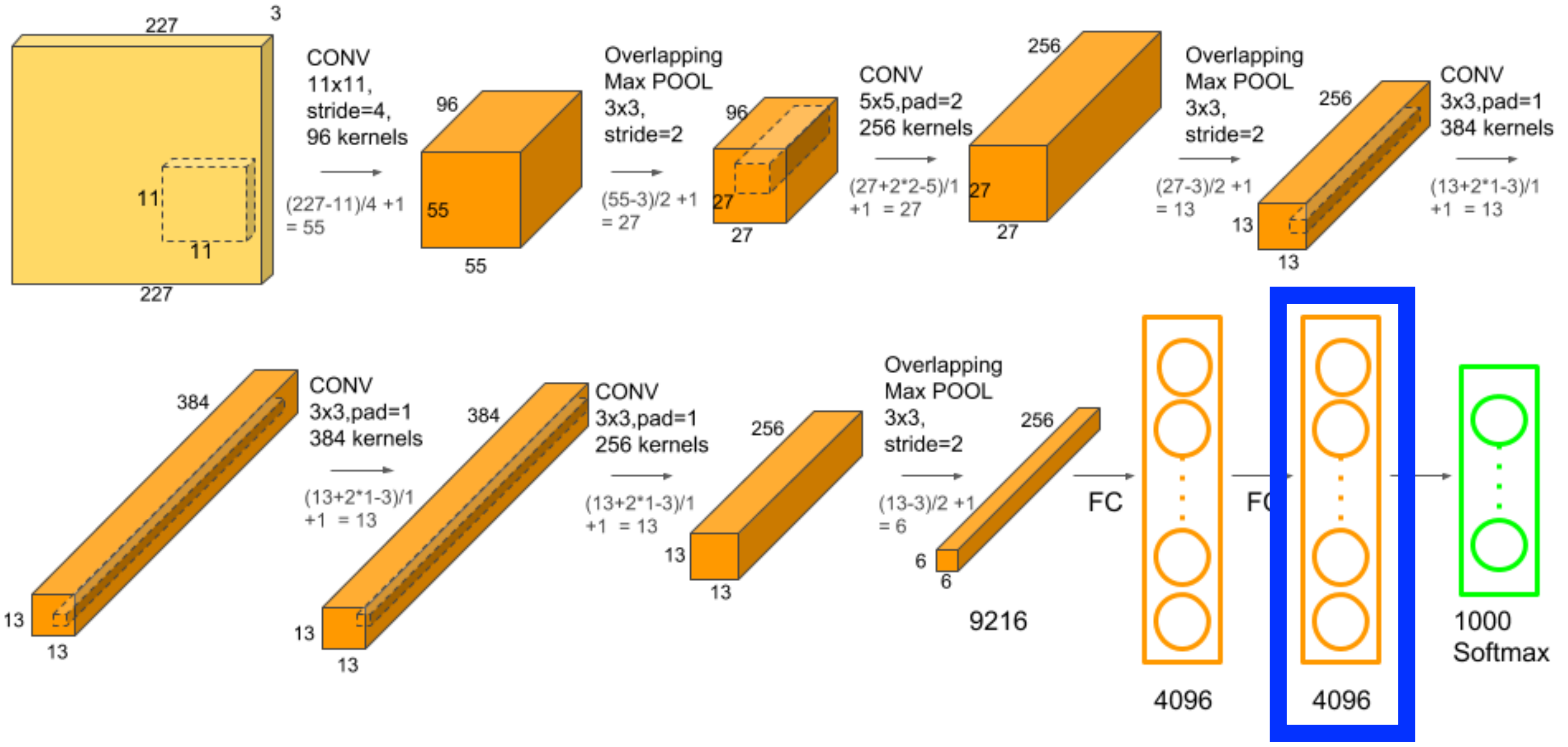


Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

CNN: Pretrained Features (e.g., AlexNet)

What is the dimensionality of the FC7 layer?

A representation of the data extracted inside a network (rather than the input or predicted output)



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

- Dataset 1: ImageNet (~1.5 million images of objects scraped from search engines)

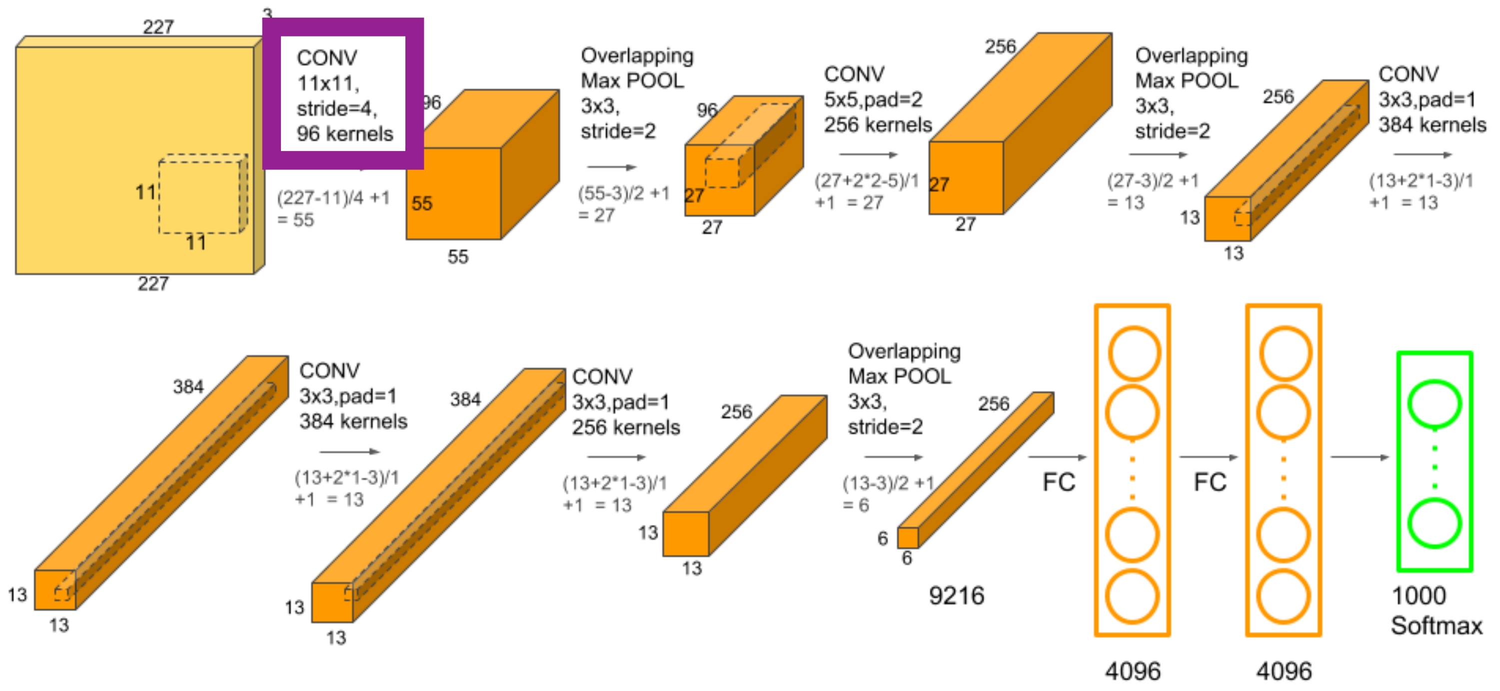
Deng et al. ImageNet: A Large-Scale Hierarchical Image Database. CVPR 2009.

- Dataset 2: Places (~2.5 million images of scenes scraped from search engines)

Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NeurIPS 2014.

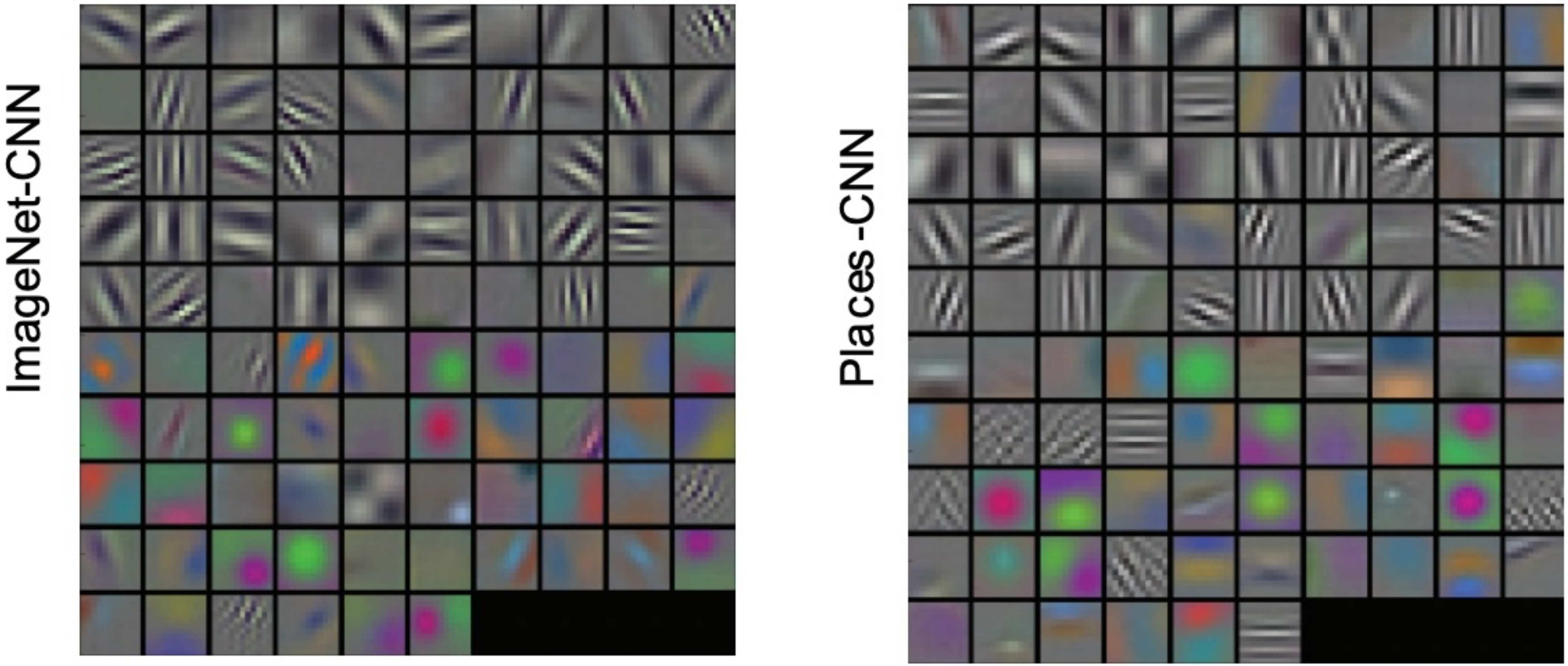
Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

- Dataset 1: ImageNet (~1.5 million images of objects scraped from search engines)
- Dataset 2: Places (~2.5 million images of scenes scraped from search engines)



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

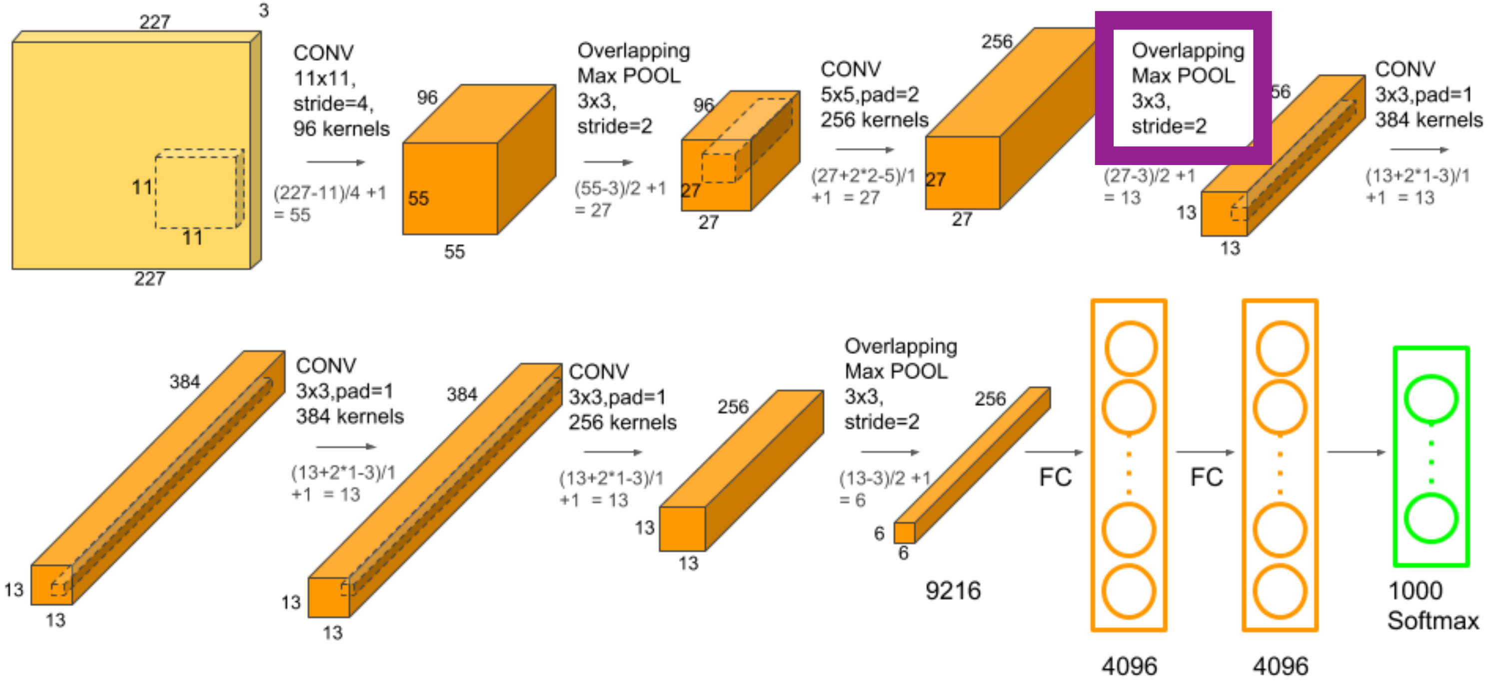


Do filters learned from the different datasets look similar or different?

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

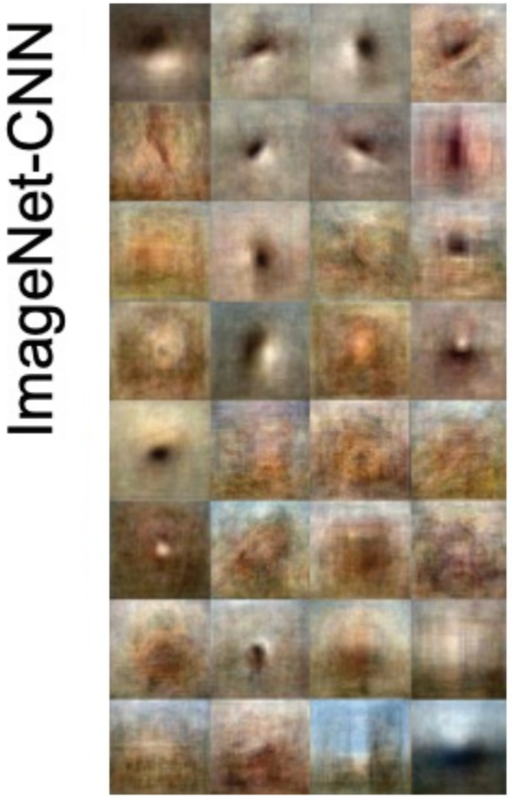
- Dataset 1: ImageNet (~1.5 million images of objects scraped from search engines)
- Dataset 2: Places (~2.5 million images of scenes scraped from search engines)

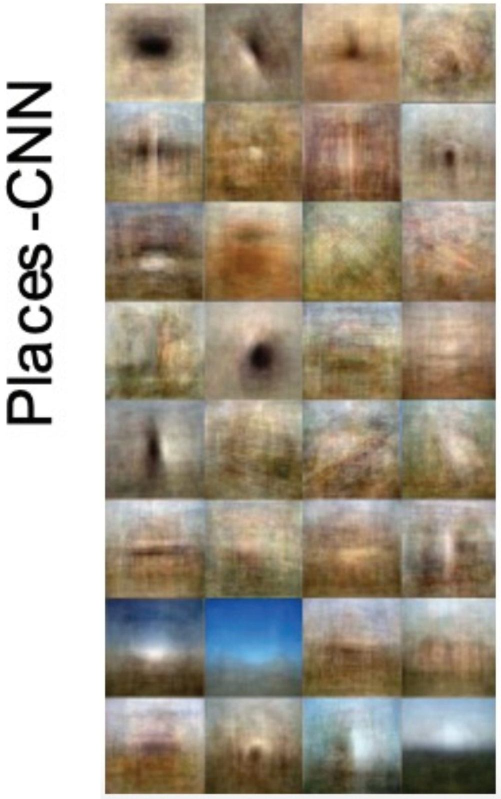


Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

Result from singling out different units in the neural networks and then generating the mean image from the 100 images which fire the most (i.e., highest activation scores)





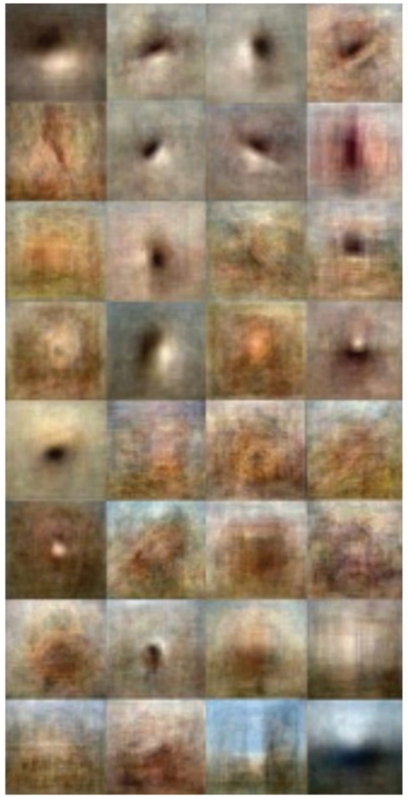
Do the filters from the different datasets appear to have learned to detect similar or different features?

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

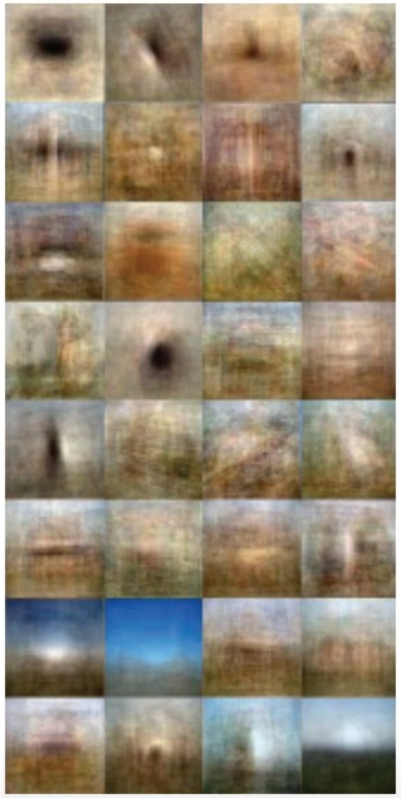
Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

Result from singling out different units in the neural networks and then generating the mean image from the 100 images which fire the most (i.e., highest activation scores)

ImageNet-CNN



Places-CNN

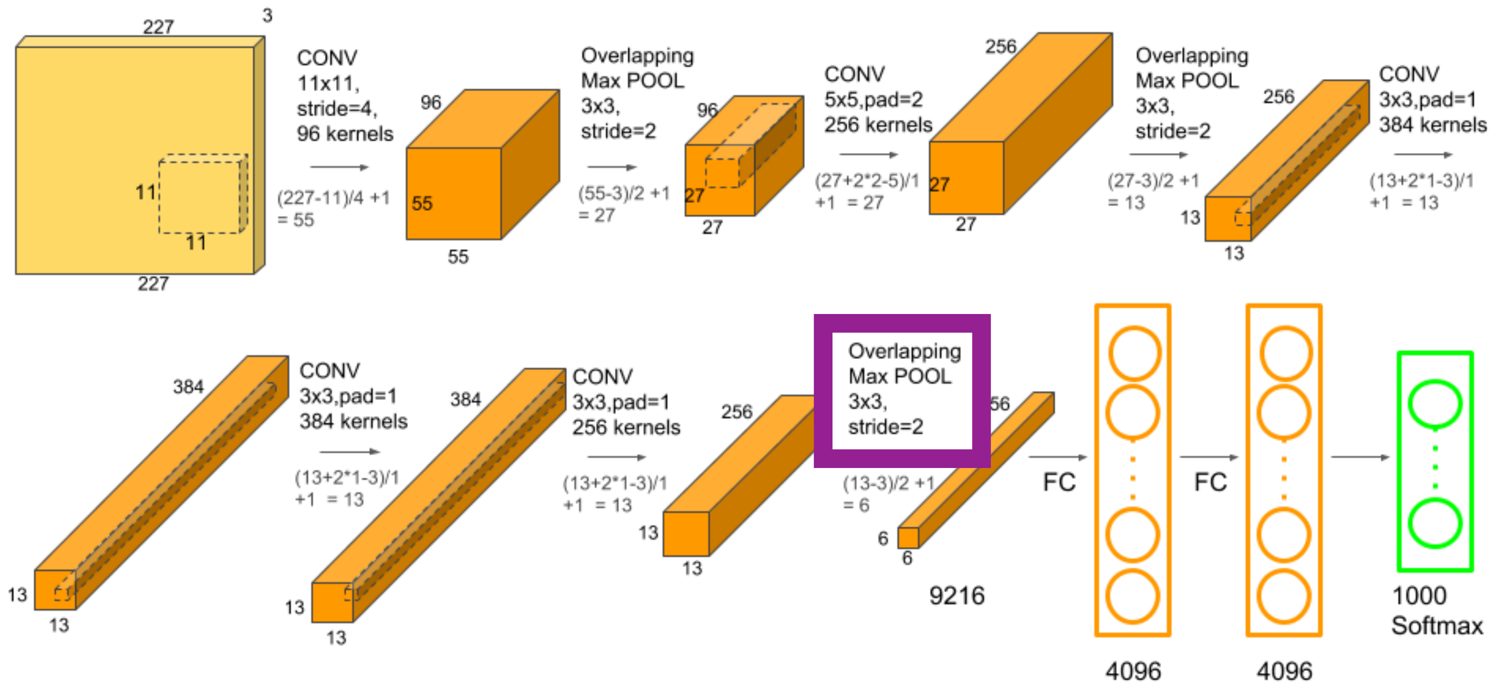


Filters from ImageNet-CNN more often fire on blob-like structures than landscape-like structures

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

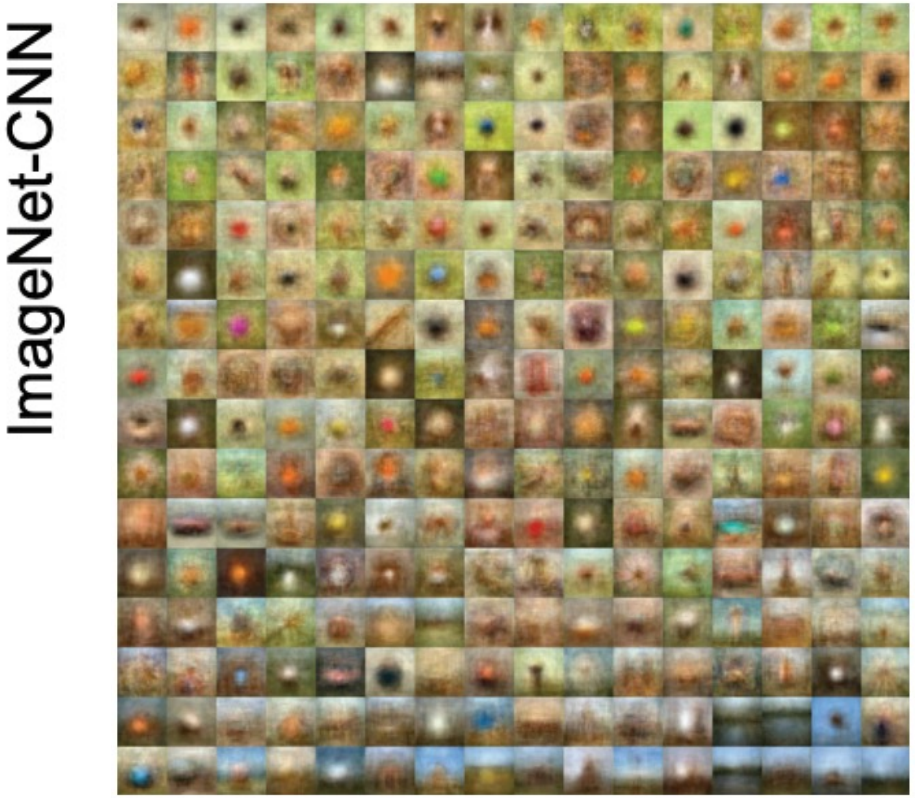
- Dataset 1: ImageNet (~1.5 million images of objects scraped from search engines)
- Dataset 2: Places (~2.5 million images of scenes scraped from search engines)

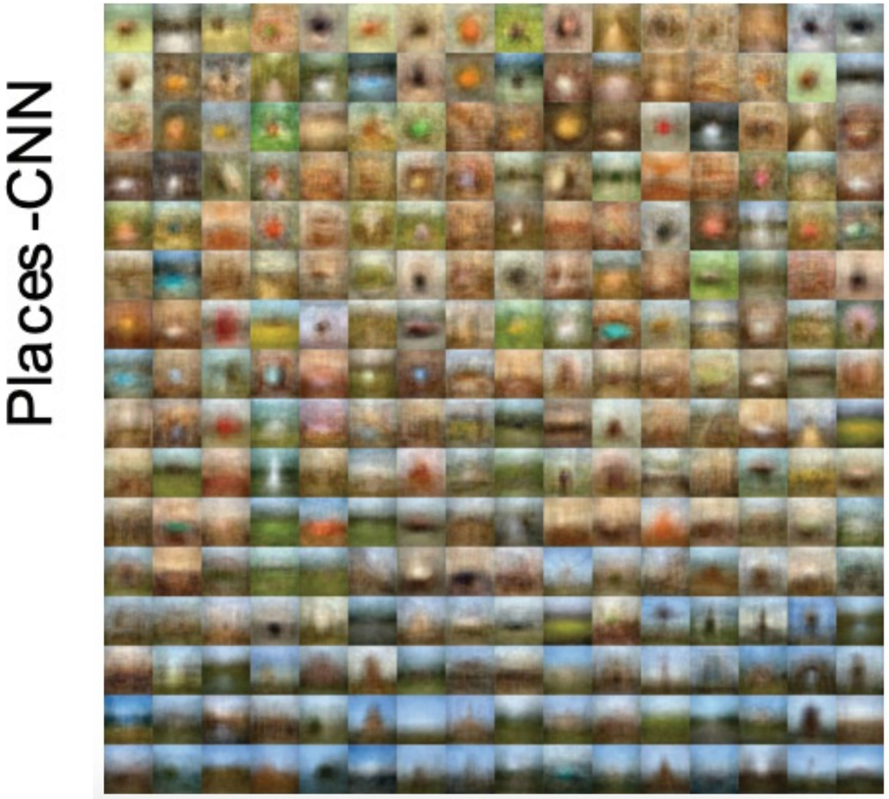


Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

Result from generating the mean image from the 100 images which fire the most for a given unit in the neural network (i.e., highest activation scores)



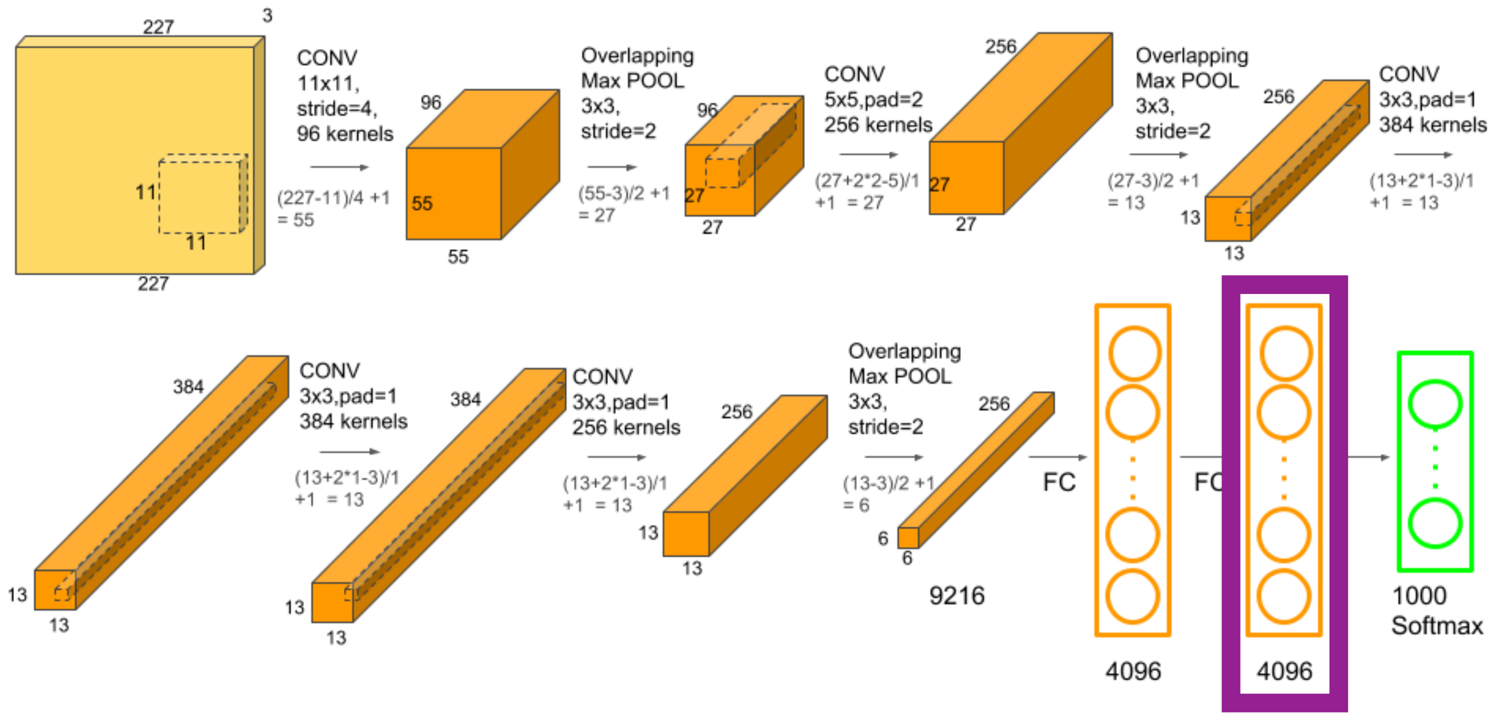


Filters from ImageNet-CNN more often fire on blob-like structures than landscape-like structures

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

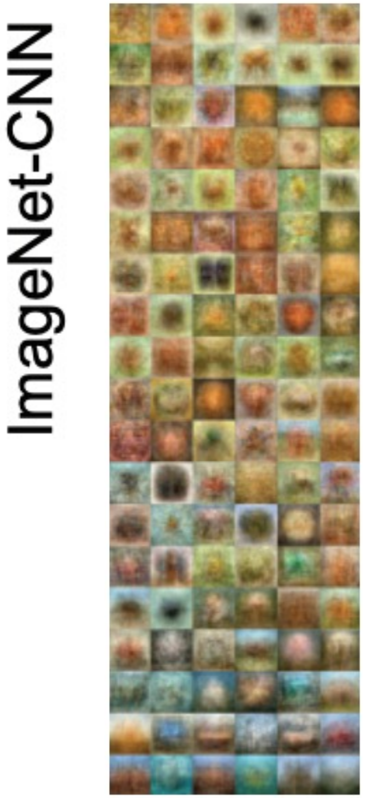
- Dataset 1: ImageNet (~1.5 million images of objects scraped from search engines)
- Dataset 2: Places (~2.5 million images of scenes scraped from search engines)



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Comparing Pretrained CNN Features Extracted by AlexNet Trained on Different Datasets

Result from generating the mean image from the 100 images which fire the most for a given unit in the neural network (i.e., highest activation scores)

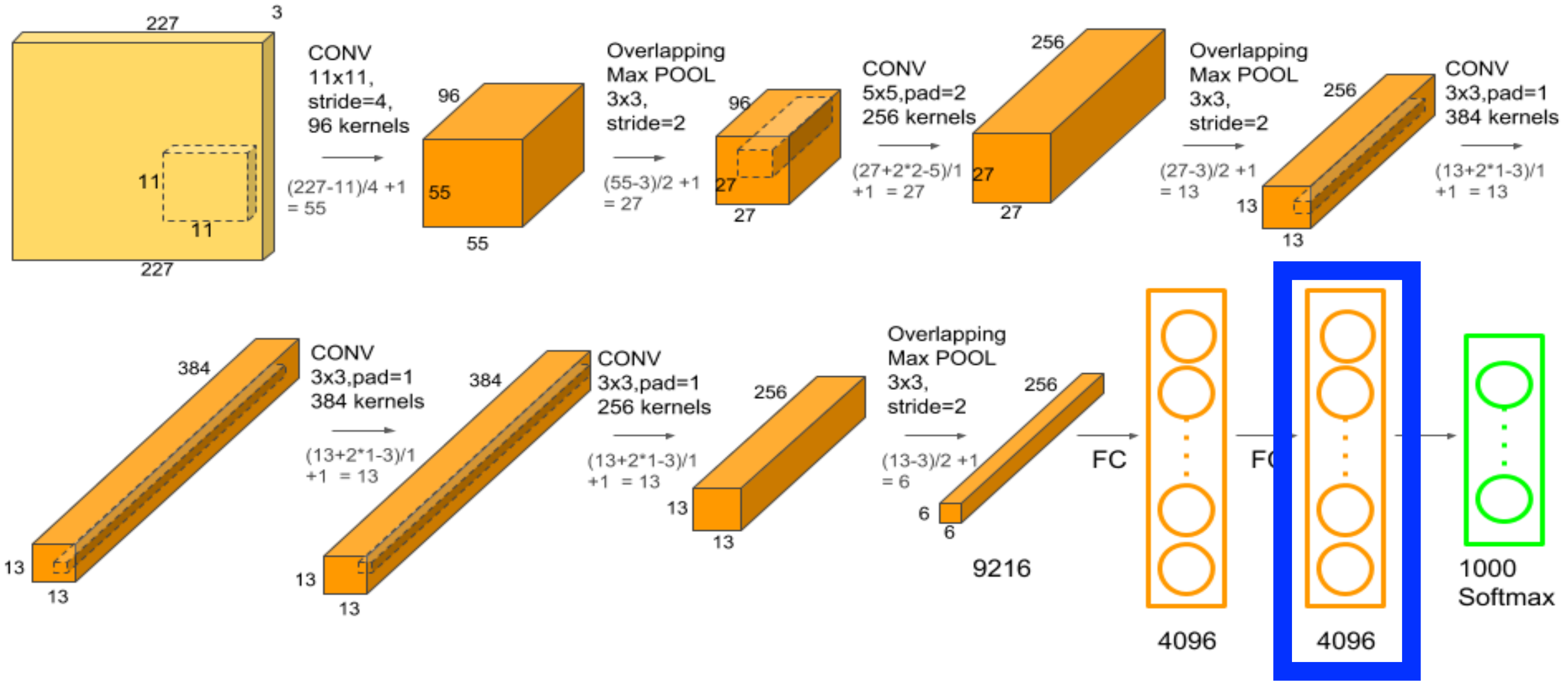


Filters from ImageNet-CNN more often fire on blob-like structures than landscape-like structures

Bolei Zhou et al. Learning Deep Features for Scene Recognition using Places Database. NIPS 2014.

Visualization of CNN Features

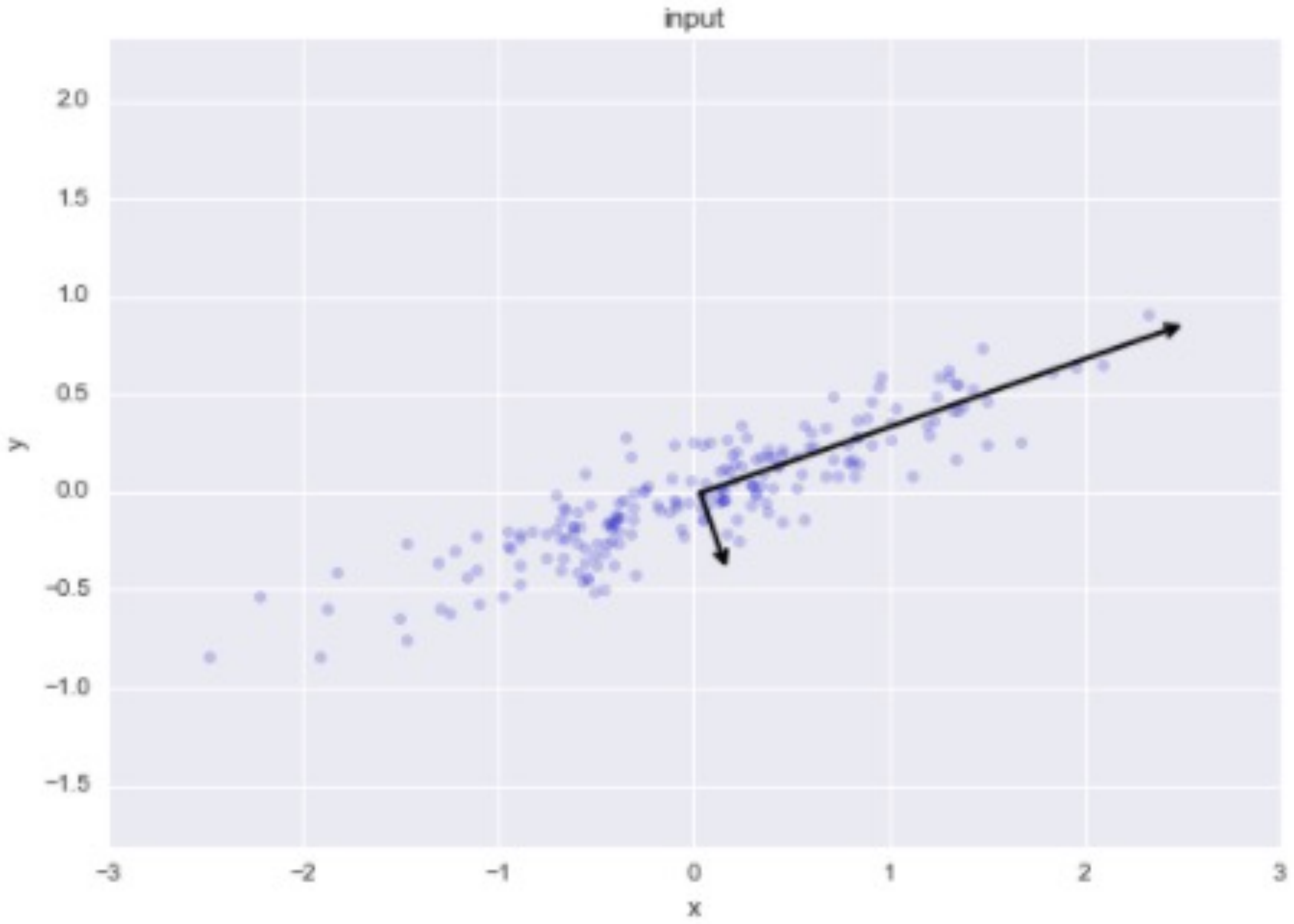
- Reduce high-dimensional data to lower dimensions for visualization; e.g., AlexNet trained on ImageNet



- Popular techniques: PCA and t-SNE

Visualization of CNN Features: PCA

- Idea: find principle axes and keep most important ones
- Vectors: *principal axes* of data
- Vector length: variance of the data described when its projected onto that axis.



https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Visualization of CNN Features: PCA

- Assumption:
  - Data is linearly separable

- Algorithm
  1. Standardize data (i.e., center data around origin)
  2. Construct covariance matrix: how random variable pairs relate to each other

$$Cov(X,Y) = \Sigma E((X-\mu)(Y-v)) / n-1$$

Random variables | Mean of X | Mean of Y | # items in dataset

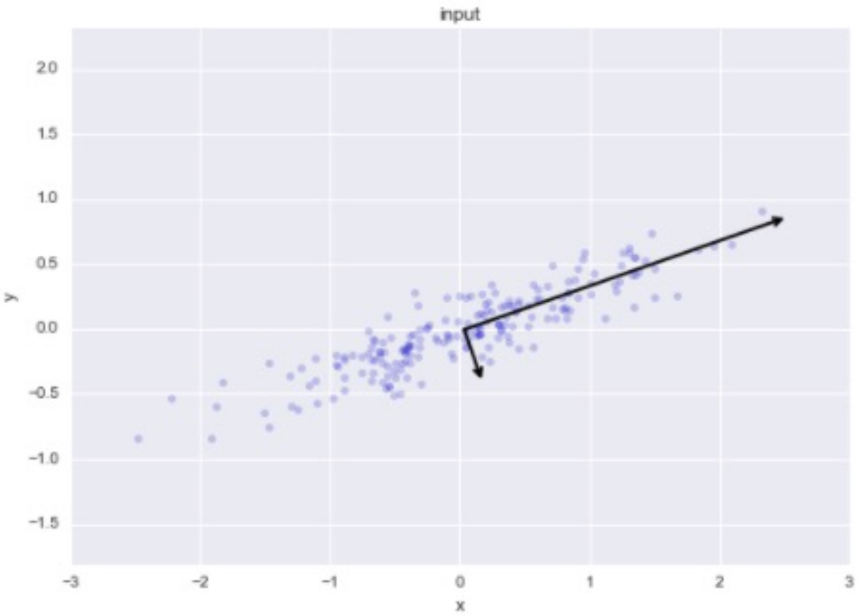
Positive when **large** values of X often occur with **large** values of Y; e.g., weight & height
Negative when **large** values of X often occur with **small** values of Y; e.g., grade and missed classes

Great tutorial: https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/covariance/

Visualization of CNN Features: PCA

- Assumption:
  - Data is linearly separable

- Algorithm
  1. Standardize data (i.e., center data around origin)
  2. Construct covariance matrix
  3. Obtain eigenvalues and eigenvectors
     - Eigenvector: represents principal components (directions of maximum variance) of the covariance matrix
     - Eigenvalues: indicates corresponding magnitude of eigenvectors with larger values indicating direction of larger variance



https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

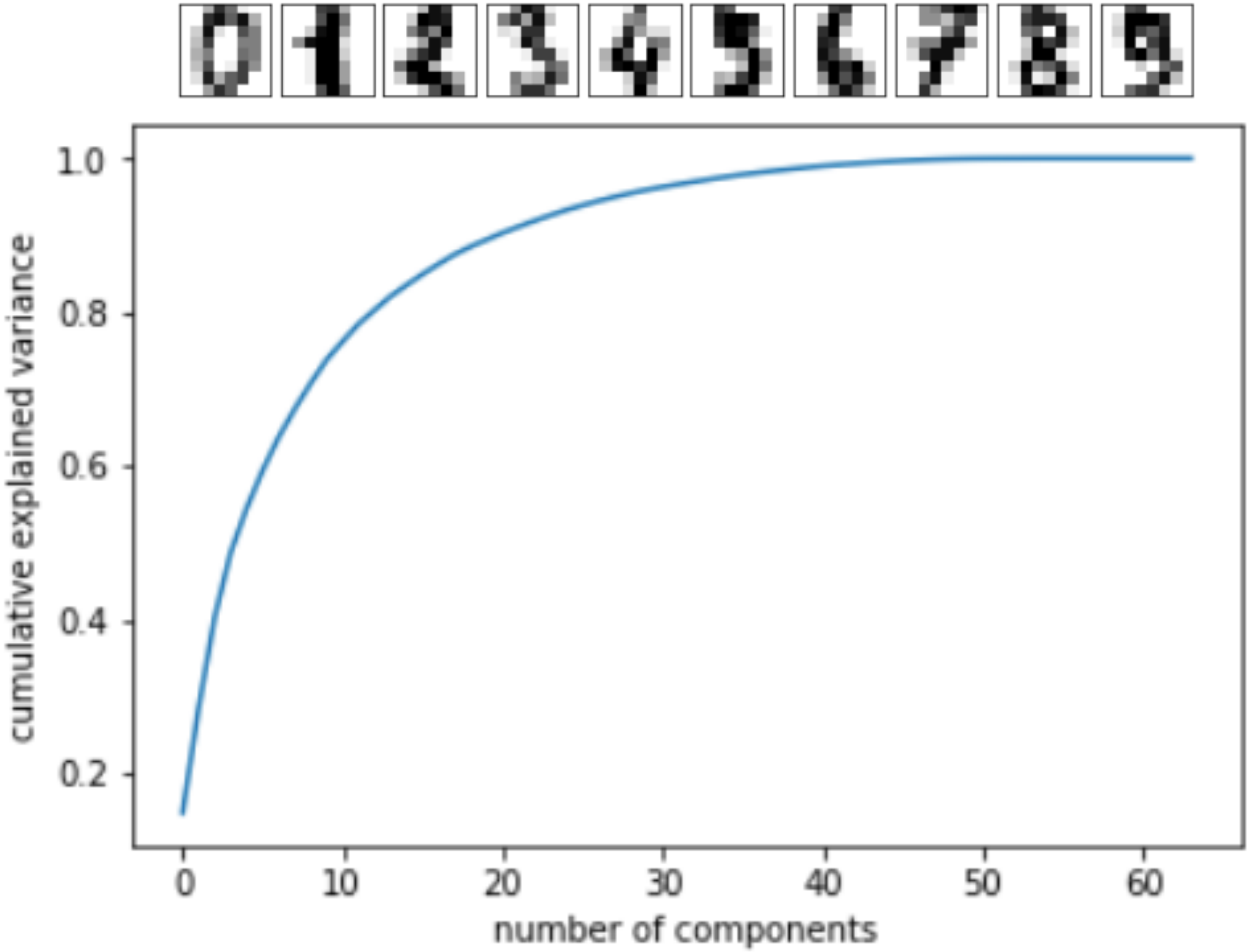
Visualization of CNN Features: PCA

- Assumption:
  - Data is linearly separable

- Algorithm
  1. Standardize data (i.e., center data around origin)
  2. Construct covariance matrix
  3. Obtain eigenvalues and eigenvectors
  4. Sort eigenvalues by decreasing order to rank eigenvectors

Visualization of CNN Features: PCA

e.g., data with 64 initial values

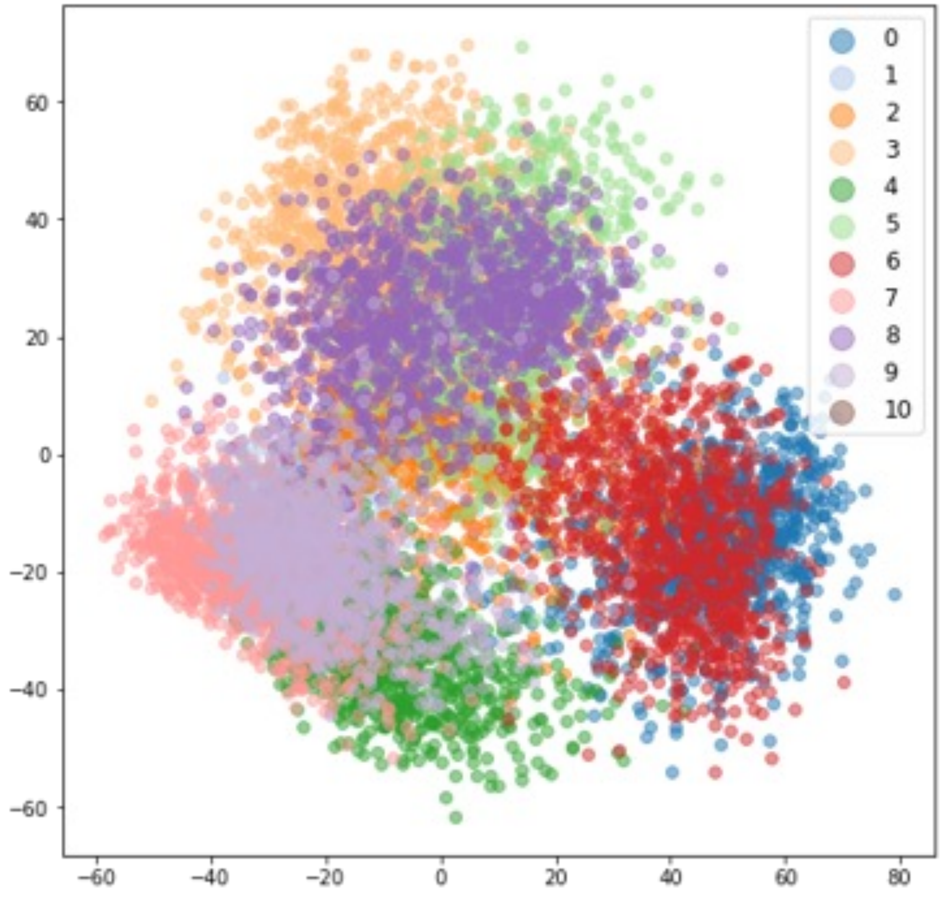


How many principal components are needed to preserve the information in the original data?

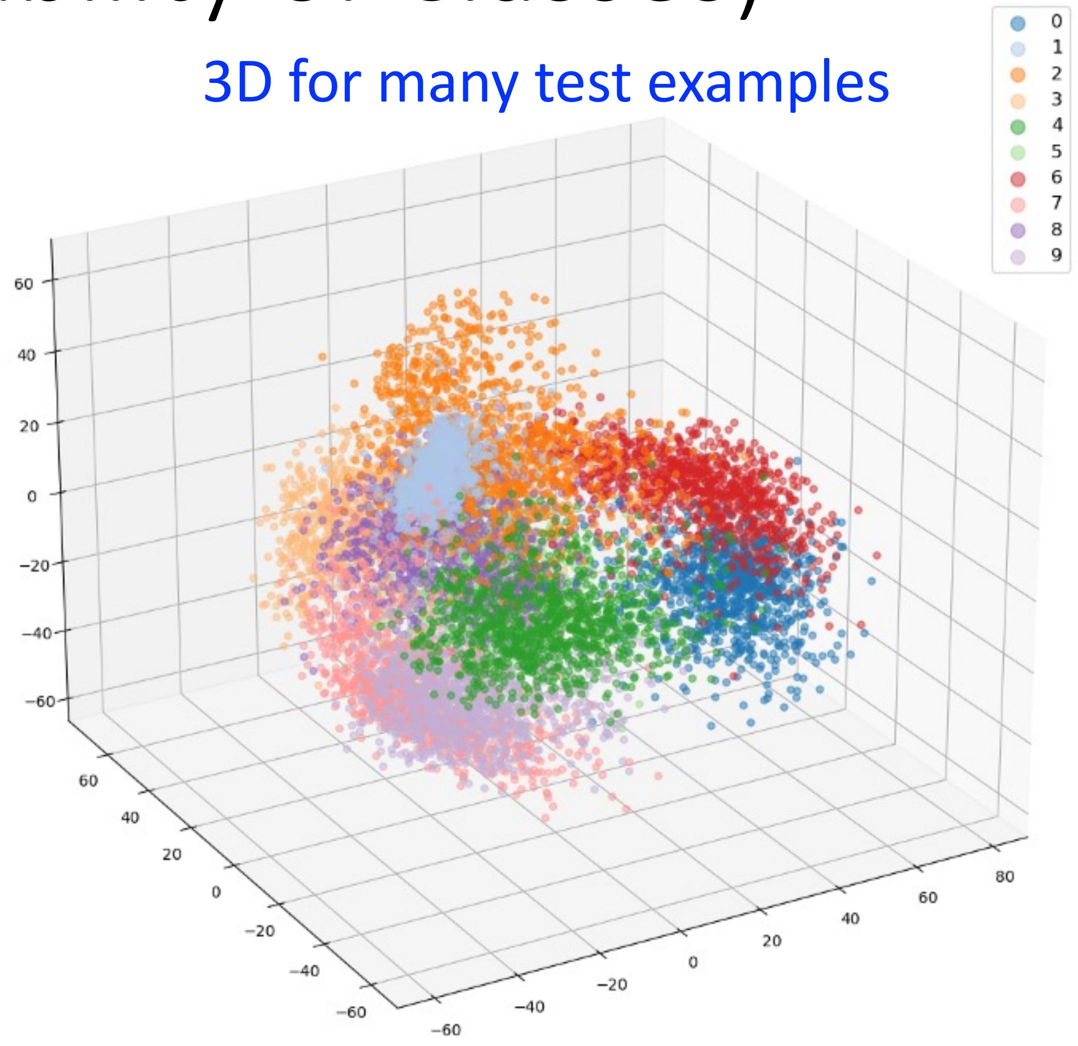
https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

Visualization of CNN Features: PCA (e.g., Visualizing Separability of Classes)

2D for many test examples



3D for many test examples



https://towardsdatascience.com/visualizing-feature-vectors-embeddings-using-pca-and-t-sne-ef157cea3a42

Summary

- Feature representations are determined by many factors including:
  1. The layer used to extract the feature
  2. The type of data used to train the model

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

What Neural Networks Learn

A pretrained network can be "fine-tuned" for a different dataset and/or task

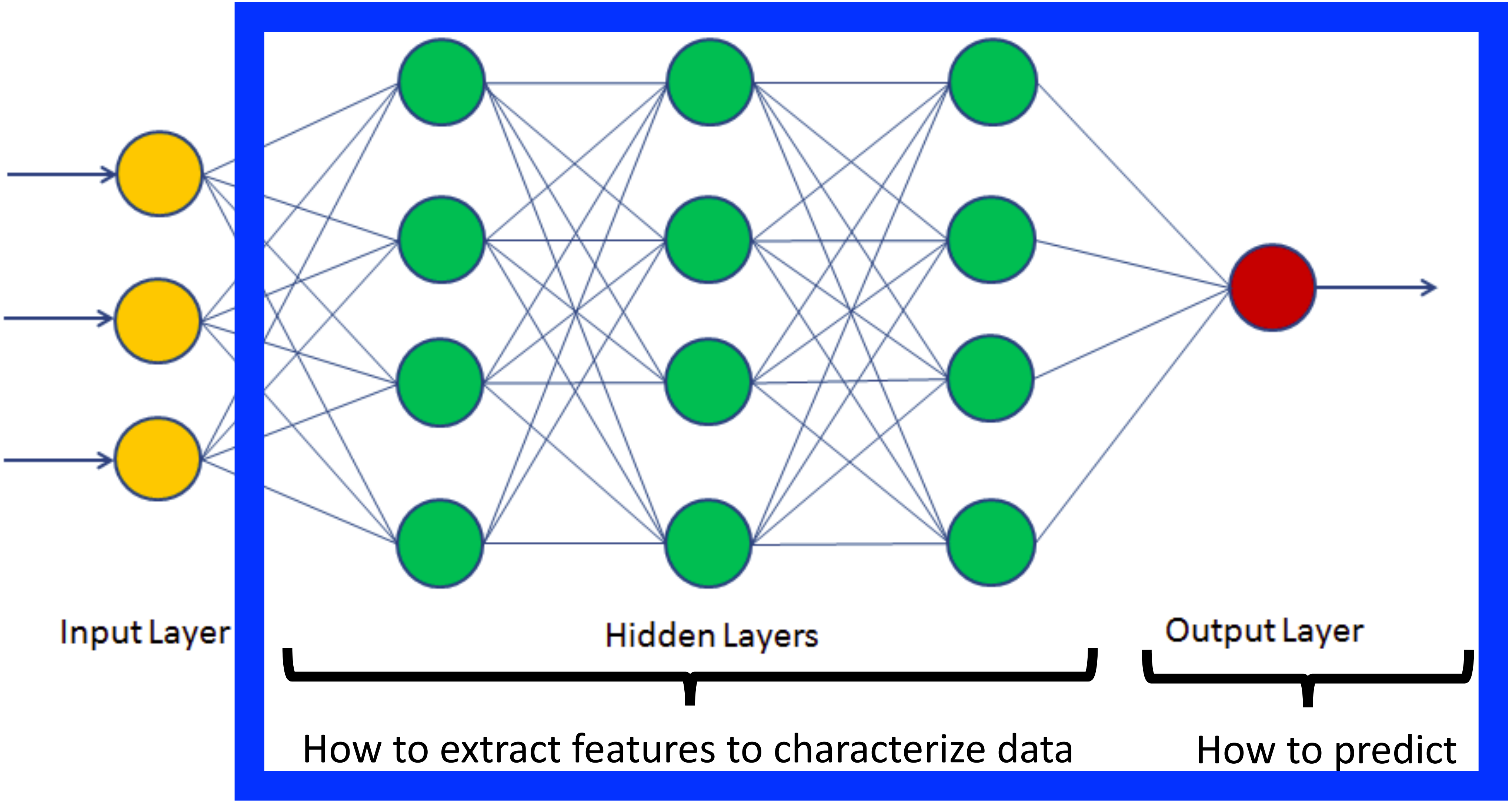


Figure Source: https://www.datacamp.com/community/tutorials/neural-network-models-r

Fine-Tuning (aka, Transfer Learning)

Use pretrained network as a starting point to train for a different dataset and/or task; e.g.,

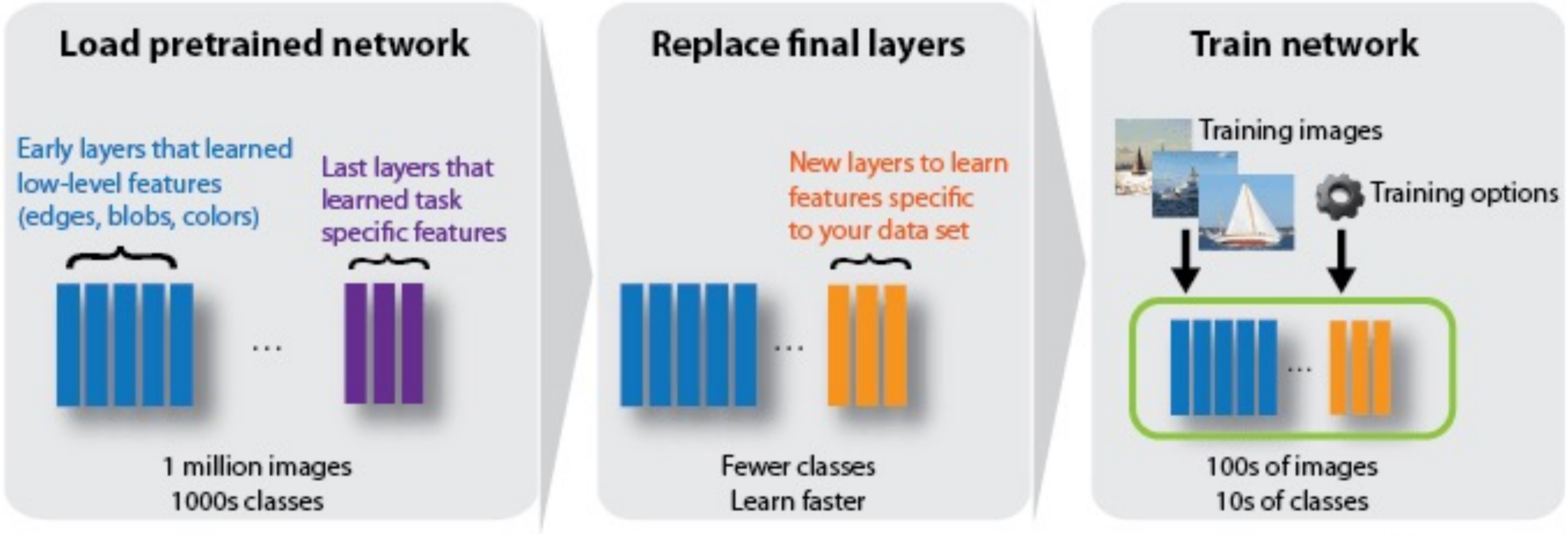


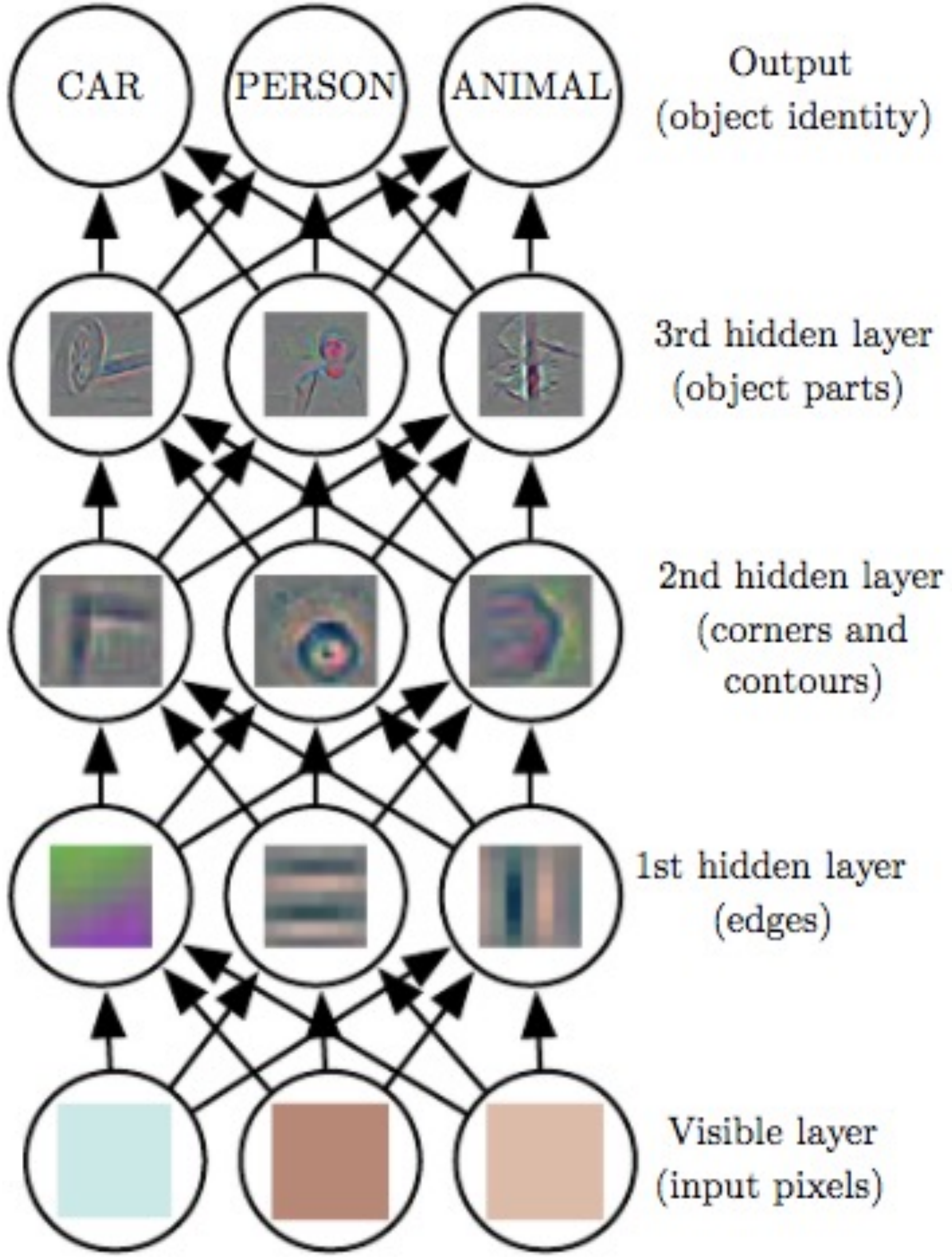
Image Source: https://www.mathworks.com/help/deeplearning/ug/transfer-learning-using-alexnet.html

Key Choice: Freeze vs Fine-Tune Layers?

High-level features

Mid-level features

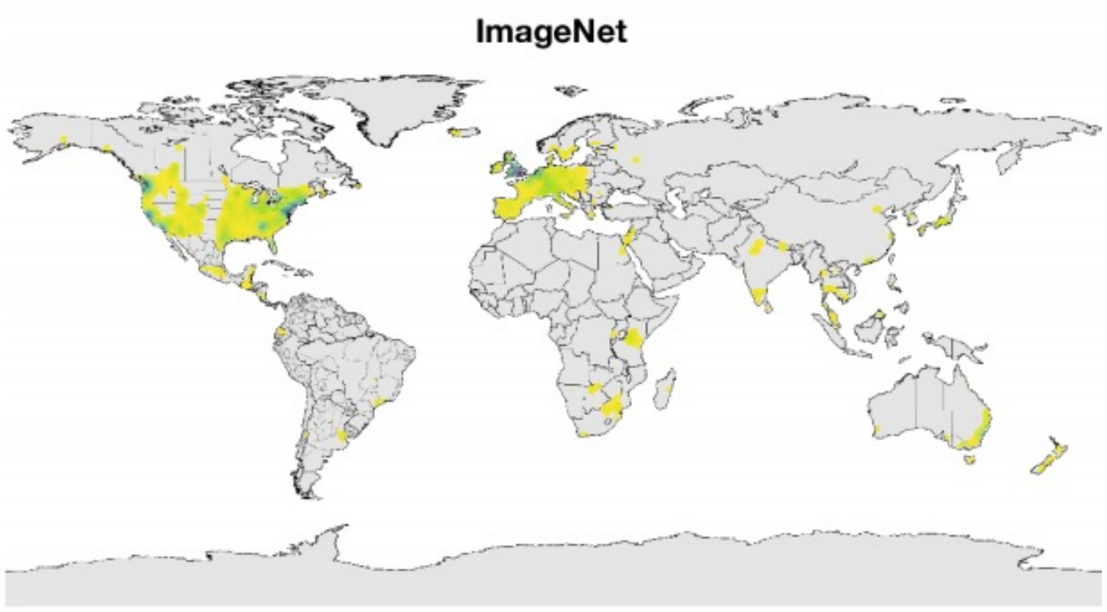
Low-level features



Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville

Class Discussion

- Assume you need to develop a classifier that recognizes common items in countries with low house incomes
  - If you fine-tuned AlexNet pretrained on ImageNet, which layers would you remove and/or freeze? Why?

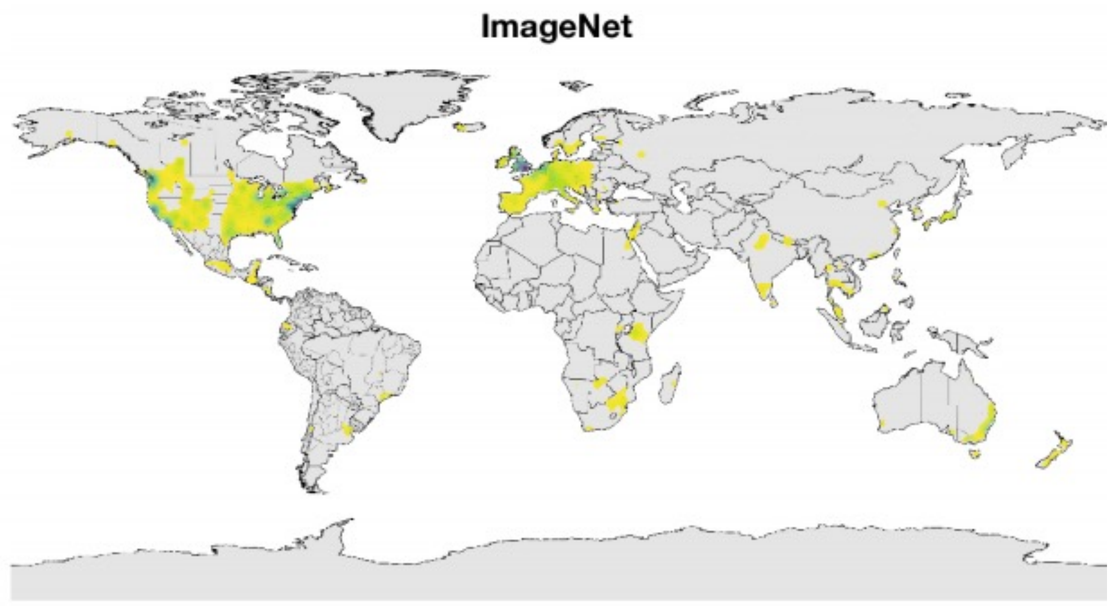


Zhao et al. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. 2017.

DeVries et al. Does object recognition work for everyone? CVPR workshops, 2019.

Class Discussion

- Assume you need to develop a classifier that recognizes common items in countries with low house incomes
  - If a large-scale dataset of low household income items was available, would you train AlexNet from scratch or fine-tune an ImageNet pretrained model? Why?



Zhao et al. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. 2017.

DeVries et al. Does object recognition work for everyone? CVPR workshops, 2019.

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

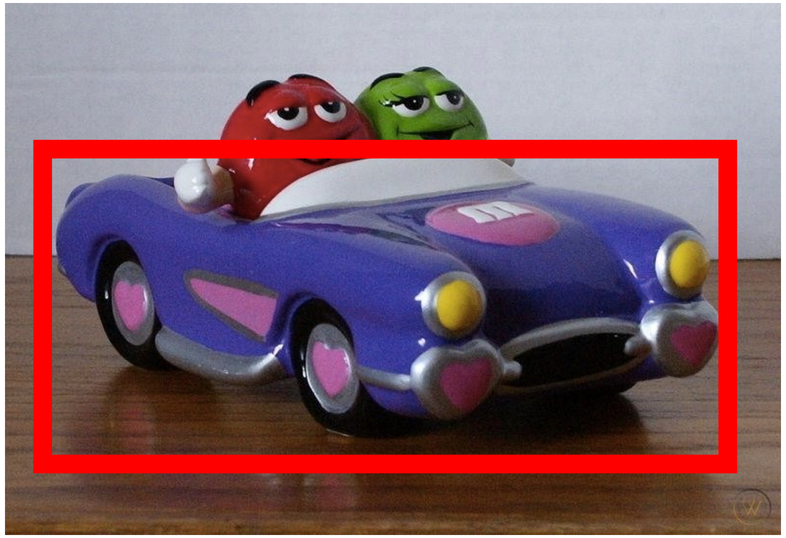
Recall: Key Ingredients for Success With Neural Networks

An **algorithm** learns from **data**
on a **processor** the patterns that
will be used to make a prediction

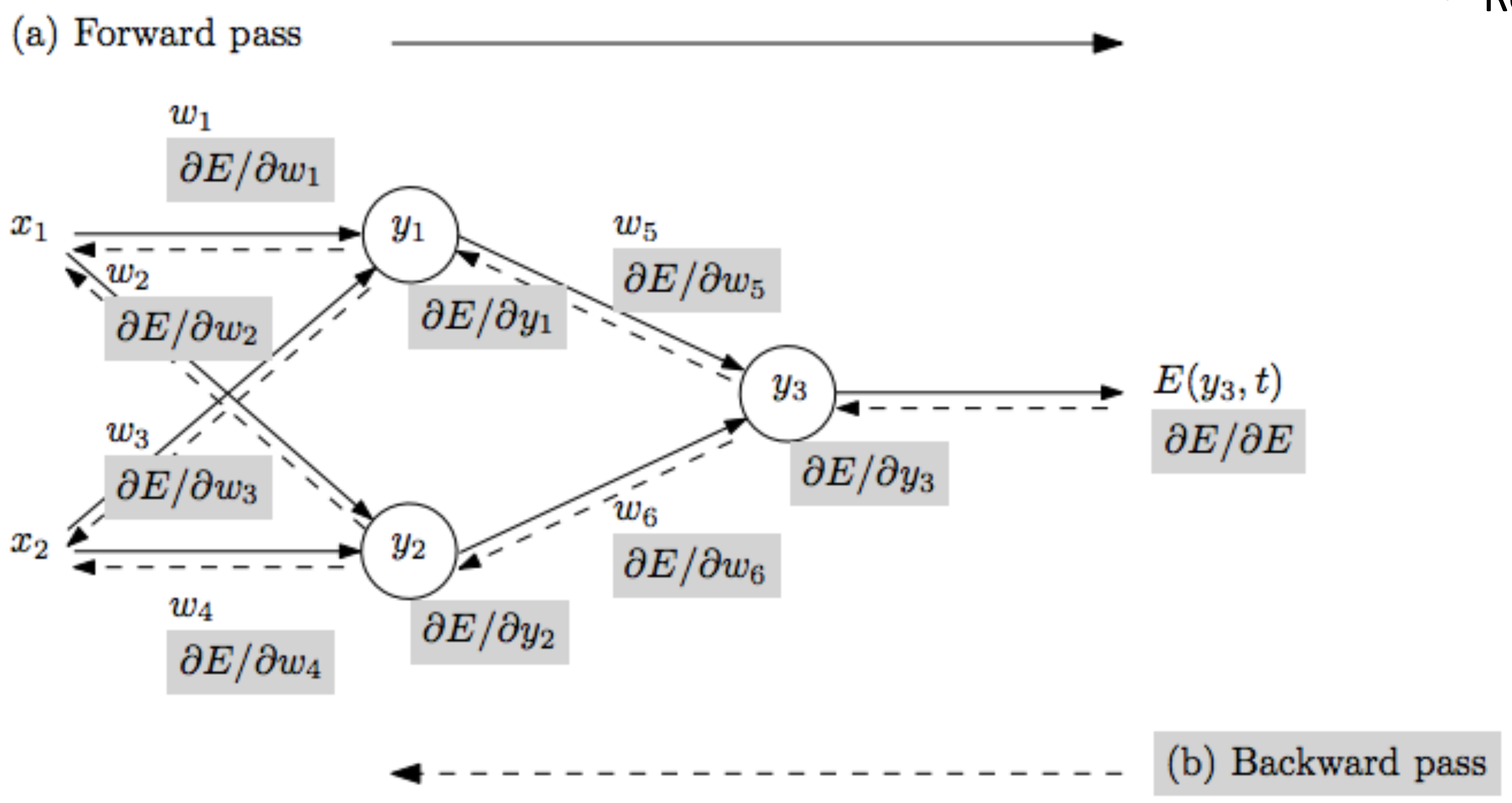
Analogous to a Love Story of Partnering Up and Road Tripping Somewhere

Recall: Key Ingredients for Success With Neural Networks

Key Issue: How Long Will It Take to Get There?



Challenge: Training Neural Network **Requires Many Computations** (e.g., millions of model parameters)



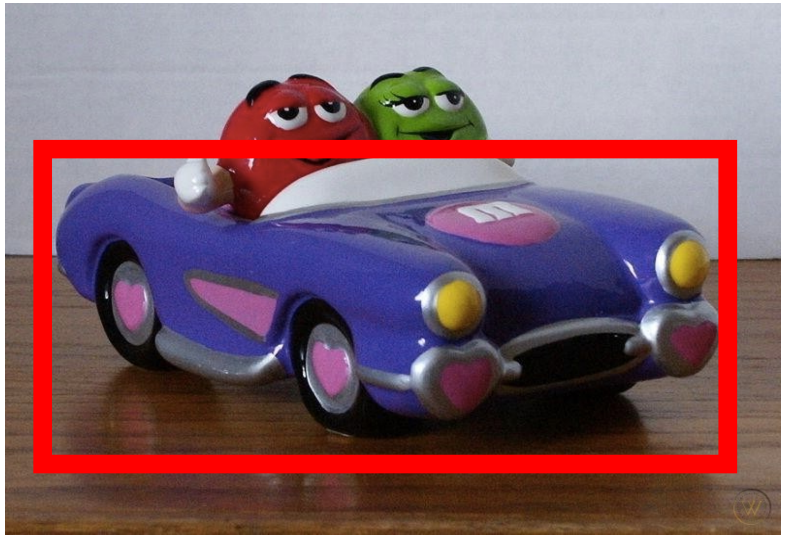
- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make prediction
  2. Quantify the dissatisfaction with a model's results on the training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

Figure from: Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, Jeffrey Mark Siskind; Automatic Differentiation in Machine Learning: a Survey; 2018

Idea: Better Hardware

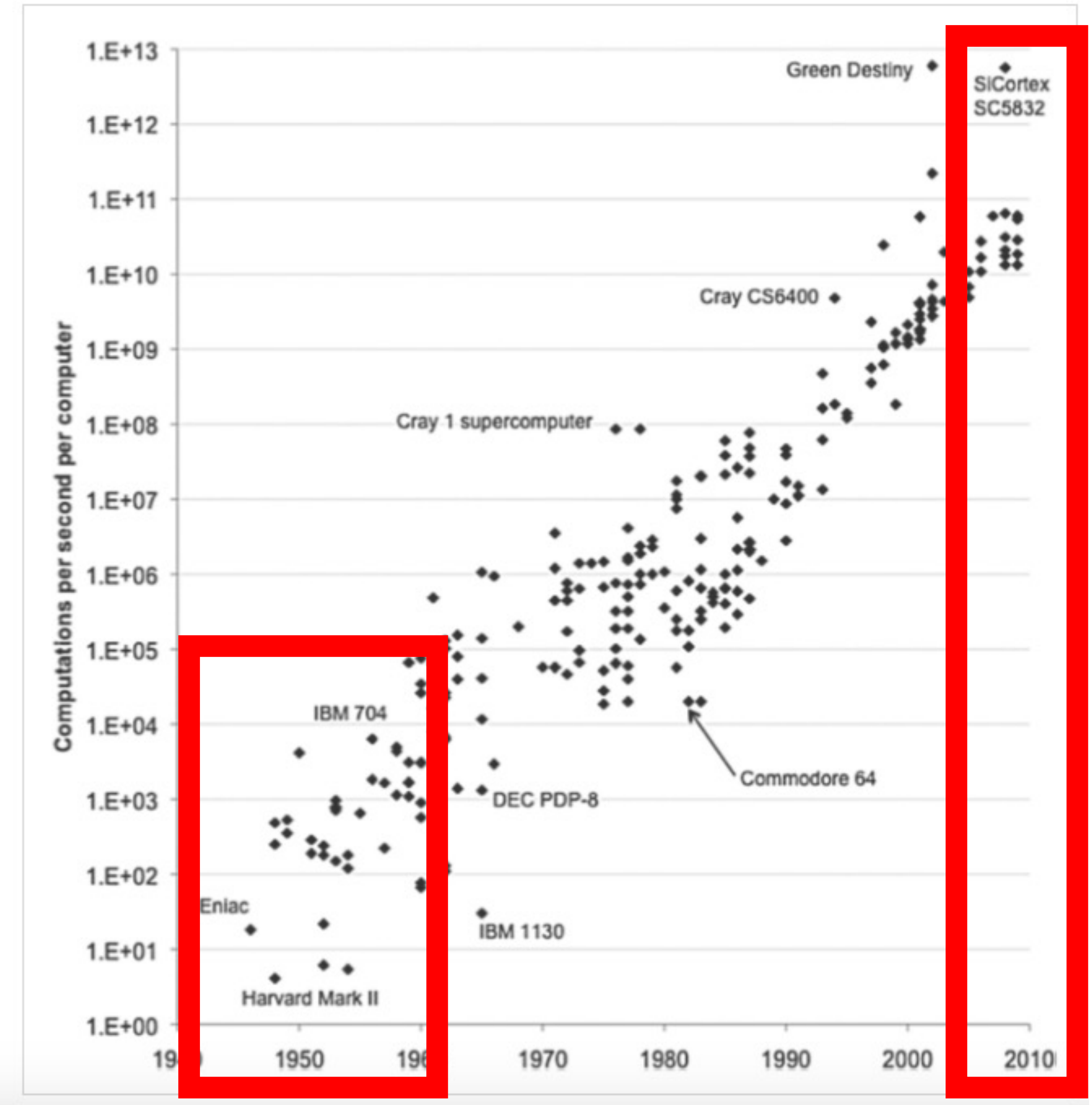
Idea: Train Algorithms Using

GPUs (think Porsche) Instead of CPUs (think Golf Cart)



Historical Perspective

- Better Hardware
  - e.g., faster processing -- GPUs



https://ourworldindata.org/technological-progress

Historical Perspective

- Better Hardware
  - e.g., faster processing -- GPUs
  - e.g., more data storage

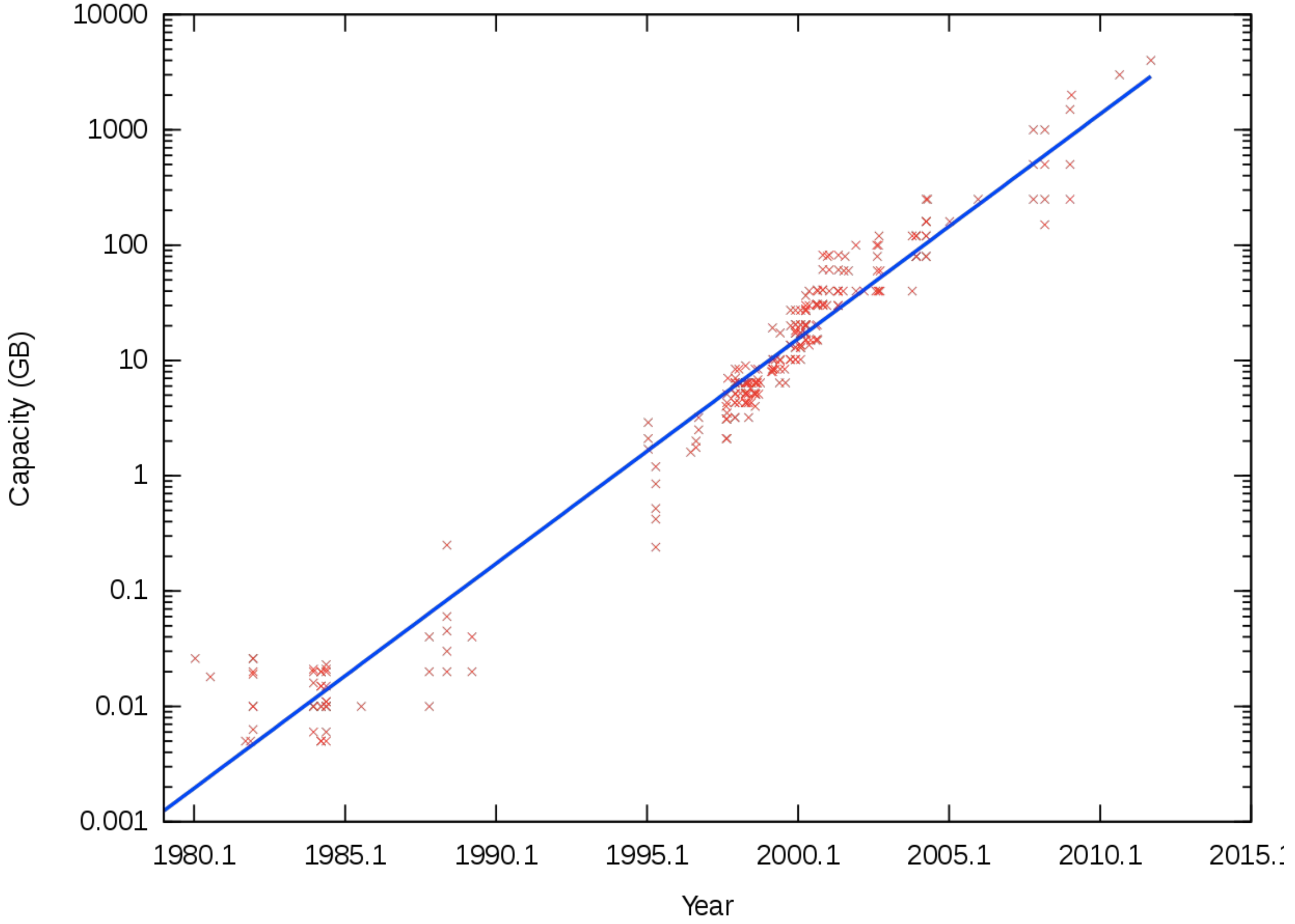
The IBM Model 350 disk file with a storage space of 5MB fro 1956 nd a Micro SD Card

https://ourworldindata.org/technological-progress

Historical Perspective

- Better Hardware
  - e.g., faster processing -- GPUs
  - e.g., more data storage

Hard Drive Capacity

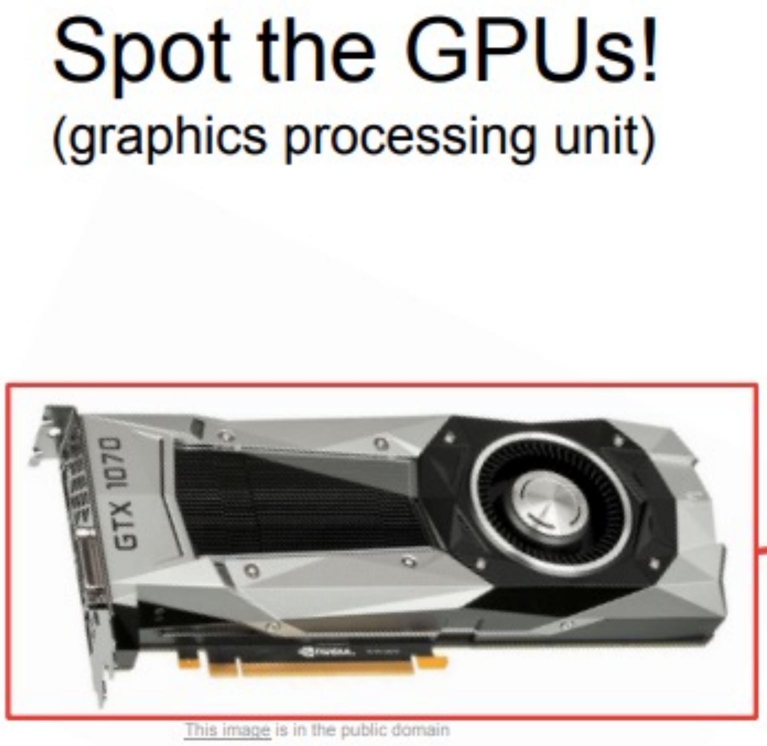


https://ourworldindata.org/technological-progress

Hardware: CPU versus GPU

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture08.pdf

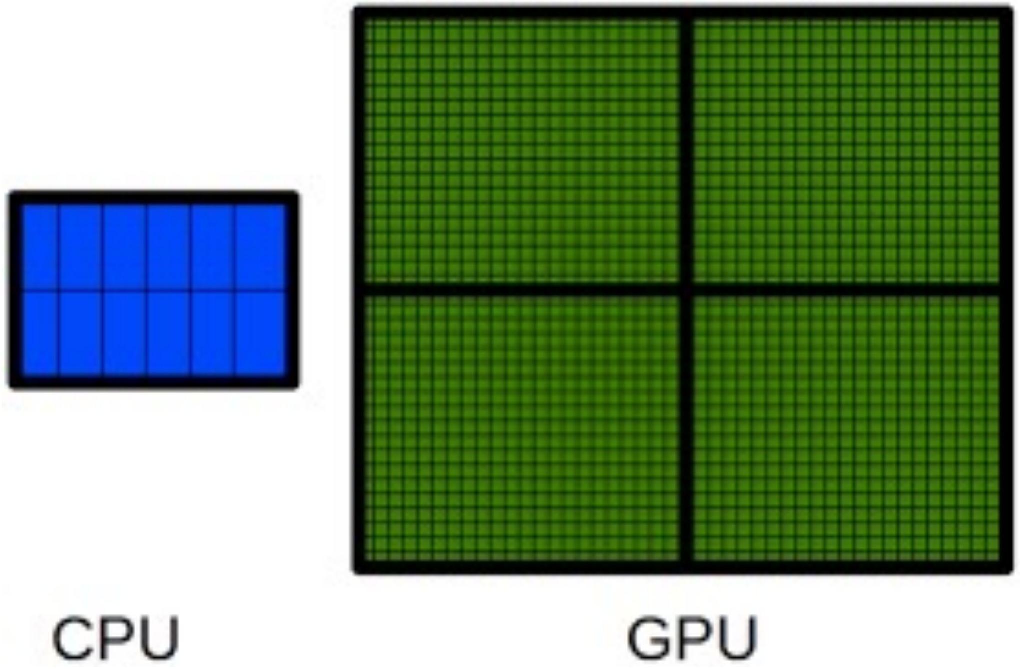
Hardware: CPU versus GPU



http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture08.pdf

Hardware: CPU versus GPU

- Graphical Processing Units: accelerates computational workloads due to MANY more processing cores



https://www.researchgate.net/figure/The-main-difference-between-CPUs-and-GPUs-is-related-to-the-number-of-available-cores-A_fig7_273383346

Hardware: Training Models with GPUs

Model is here

Data is here

If you aren't careful, training can bottleneck on reading data and transferring to GPU!

Solutions:

- Read all data into RAM
- Use SSD instead of HDD
- Use multiple CPU threads to prefetch data

http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture08.pdf

GPU Machines: Rent Versus Buy?

Rent from Cloud

(e.g., Microsoft Azure):

Instance	Core(s)	RAM	Temporary storage	GPU	Pay as you go with AHB
ND96asr A100 v4	96	900 GiB	6,500 GiB	8x A100 (NVlink)	$27.197/hour

Buy:

Lambda Bare Metal

- 4-8x NVIDIA A100 SXM4 GPUs
- Install in your Datacenter or Lambda Colo
- Customize CPU, RAM, Storage & Network
- Delivered in 2-4 weeks

Starting at

$ **89,283.00**

Rise of “Deep Learning” Open Source Platforms

Motivation:

Can run on GPUs:

OpenMP support	OpenCL support	CUDA support	Automatic differentiation[1]

Simplifies using popular neural network architectures:

Has pretrained models	Recurrent nets	Convolutional nets	RBM/DBNs	Parallel execution (multi node)

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

Rise of "Deep Learning" Open Source Platforms

torch (Collobert et al.) — 2011

theano (Bastien et al.) — 2012

Caffe (Jia et al.) — 2013

2014

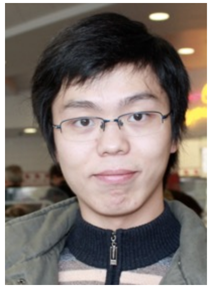
Keras

TensorFlow™ (Abadi et al.) — 2015

2016

PYTORCH (Paszke et al.) — 2017

Université de Montréal



idiap RESEARCH INSTITUTE

Berkeley UNIVERSITY OF CALIFORNIA

Google

Popular Options Today

Rise of "Deep Learning" Open Source Platforms

Software	Creator	Software license[a]	Open source	Platform	Written in	Interface	OpenMP support	OpenCL support	CUDA support	Automatic differentiation[1]	Has pretrained models	Recurrent nets	Convolutional nets	RBM/DBNs	Parallel execution (multi node)	Actively Developed
roNNie.ai	Kevin Lok	MIT license	Yes	Linux, macOS, Windows	Python	Python			Yes		Yes	Yes	Yes			
BigDL	Jason Dai	Apache 2.0	Yes	Apache Spark	Scala	Scala, Python			No		Yes	Yes	Yes			
Caffe	Berkeley Vision and Learning Center	BSD	Yes	Linux, macOS, Windows[2]	C++	Python, MATLAB, C++	Yes	Under development[3]	Yes	Yes	Yes[4]	Yes	Yes	No	?	
Deeplearning4j	Skymind engineering team; Deeplearning4j community; originally Adam Gibson	Apache 2.0	Yes	Linux, macOS, Windows, Android (Cross-platform)	C++, Java	Java, Scala, Clojure, Python (Keras), Kotlin	Yes	On roadmap[5]	Yes[6][7]	Computational Graph	Yes[8]	Yes	Yes	Yes	Yes[9]	
Chainer	Preferred Networks	MIT license	Yes	Linux, macOS, Windows		Python	No	No[10][11]	Yes	Yes	Yes	Yes	Yes			
Darknet	Joseph Redmon	Public Domain	Yes	Cross-Platform	C	C, Python	Yes	No[12]	Yes	Yes						
Dlib	Davis King	Boost Software License	Yes	Cross-Platform	C++	C++	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	
DataMelt (DMelt)	S.Chekanov	Freemium	Yes	Cross-Platform	Java	Java	No	No	No	No	No	No	No	No	No	
DyNet	Carnegie Mellon University	Apache 2.0	Yes	Linux, macOS, Windows		C++, Python		No[13]	Yes	Yes	Yes					
Intel Data Analytics Acceleration Library	Intel	Apache License 2.0	Yes	Linux, macOS, Windows on Intel CPU[14]	C++, Python, Java	C++, Python, Java[14]	Yes	No	No	Yes	No		Yes		Yes	
Intel Math Kernel Library	Intel	Proprietary	No	Linux, macOS, Windows on Intel CPU[15]		C[16]	Yes[17]	No	No	Yes	No	Yes[18]	Yes[18]		No	
Keras	François Chollet	MIT license	Yes	Linux, macOS, Windows	Python	Python, R	Only if using Theano as backend	Can use Theano or Tensorflow as backends	Yes	Yes	Yes[19]	Yes	Yes	Yes	Yes[20]	
MATLAB + Neural Network Toolbox	MathWorks	Proprietary	No	Linux, macOS, Windows	C, C++, Java, MATLAB	MATLAB	No	No	Train with Parallel Computing Toolbox and generate CUDA code with GPU Coder[21]	No	Yes[22][23]	Yes[22]	Yes[22]	No	With Parallel Computing Toolbox[24]	
Microsoft Cognitive Toolkit	Microsoft Research	MIT license[25]	Yes	Windows, Linux[26] (macOS via Docker on roadmap)	C++	Python (Keras), C++, Command line,[27] BrainScript[28] (.NET on roadmap[29])	Yes[30]	No	Yes	Yes	Yes[31]	Yes[32]	Yes[32]	No[33]	Yes[34]	
Apache MXNet	Apache Software Foundation	Apache 2.0	Yes	Linux, macOS, Windows,[35][36] AWS, Android,[37] iOS, JavaScript[38]	Small C++ core library	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl	Yes	On roadmap[39]	Yes	Yes[40]	Yes[41]	Yes	Yes	Yes	Yes[42]	
Neural Designer	Artelnics	Proprietary	No	Linux, macOS, Windows	C++	Graphical user interface	Yes	No	No	?	?	No	No	No	?	
OpenNN	Artelnics	GNU LGPL	Yes	Cross-platform	C++	C++	Yes	No	Yes	?	?	No	No	No	?	
PaddlePaddle	Baidu	Apache License	Yes	Linux, macOS, Windows	C++, Python	Python	No	Yes	Yes	Yes	Yes	Yes	Yes	?	Yes	
PlaidML	Vertex.AI	AGPL3	Yes	Linux, macOS, Windows	C++, Python	Keras, Python, C++, C	No	Yes	Yes	Yes		Yes	Yes	?	Yes	
PyTorch	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	BSD	Yes	Linux, macOS, Windows	Python, C, CUDA	Python	Yes	Via separately maintained package[43][44][45]	Yes	Yes	Yes	Yes	Yes		Yes	
Apache SINGA	Apache Incubator	Apache 2.0	Yes	Linux, macOS, Windows	C++	Python, C++, Java	No	No	Yes	?	Yes	Yes	Yes	Yes	Yes	
TensorFlow	Google Brain team	Apache 2.0	Yes	Linux, macOS, Windows,[46] Android	C++, Python, CUDA	Python (Keras), C/C++, Java, Go, R[47], Julia, Swift	No	On roadmap[48] but already with SYCL[49] support	Yes	Yes[50]	Yes[51]	Yes	Yes	Yes	Yes	
TensorLayer	Hao Dong	Apache 2.0	Yes	Linux, macOS, Windows,[52] Android	C++, Python,	Python	No	On roadmap[48] but already with SYCL[49] support	Yes	Yes[53]	Yes[54]	Yes	Yes	Yes	Yes	
Theano	Université de Montréal	BSD	Yes	Cross-platform	Python	Python (Keras)	Yes	Under development[55]	Yes	Yes[56][57]	Through Lasagne's model zoo[58]	Yes	Yes	Yes	Yes[59]	No
Torch	Ronan Collobert, Koray Kavukcuoglu, Clement Farabet	BSD	Yes	Linux, macOS, Windows,[60] Android,[61] iOS	C, Lua	Lua, LuaJIT,[62] C, utility library for C++/OpenCL[63]	Yes	Third party implementations[64][65]	Yes[66][67]	Through Twitter's Autograd[68]	Yes[69]	Yes	Yes	Yes	Yes[70]	
Wolfram Mathematica	Wolfram Research	Proprietary	No	Windows, macOS, Linux, Cloud computing	C++, Wolfram Language, CUDA	Wolfram Language	Yes	No	Yes	Yes	Yes[71]	Yes	Yes	Yes	Under Development	
VerAI	VerAI	Proprietary	No	Linux, Web-based	C++,Python, Go, Angular	Graphical user interface, cli	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	

Excellent comparison: https://skymind.ai/wiki/comparison-frameworks-dl4j-tensorflow-pytorch

Excellent comparison: https://arxiv.org/pdf/1511.06435.pdf

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial

Today’s Topics

- Representation learning
- Pretrained features
- Fine-tuning
- Training neural networks: hardware & software
- Programming tutorial


The End