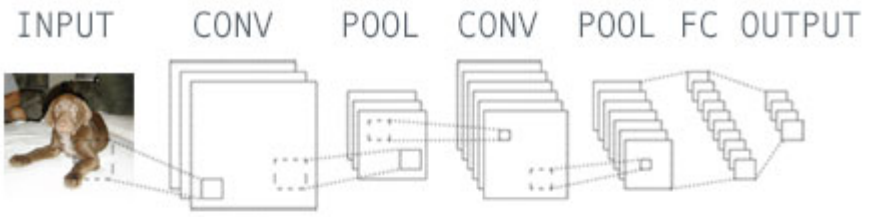
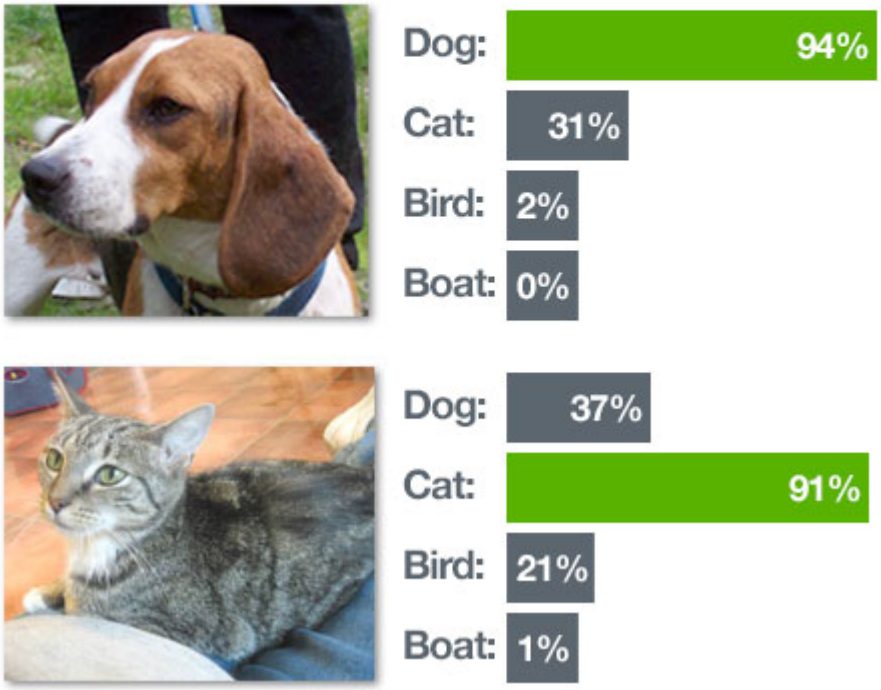
Convolutional Networks





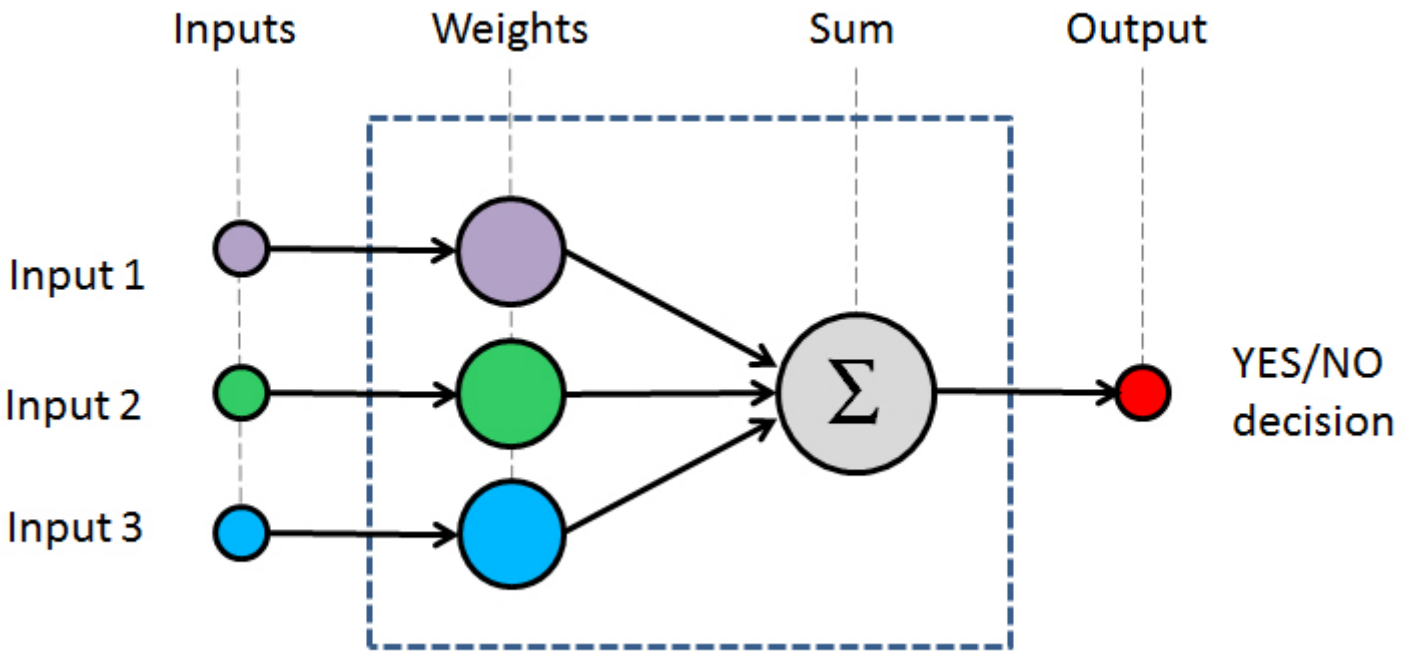
The Truth About Cats & Dogs

Tony Reina

Goals

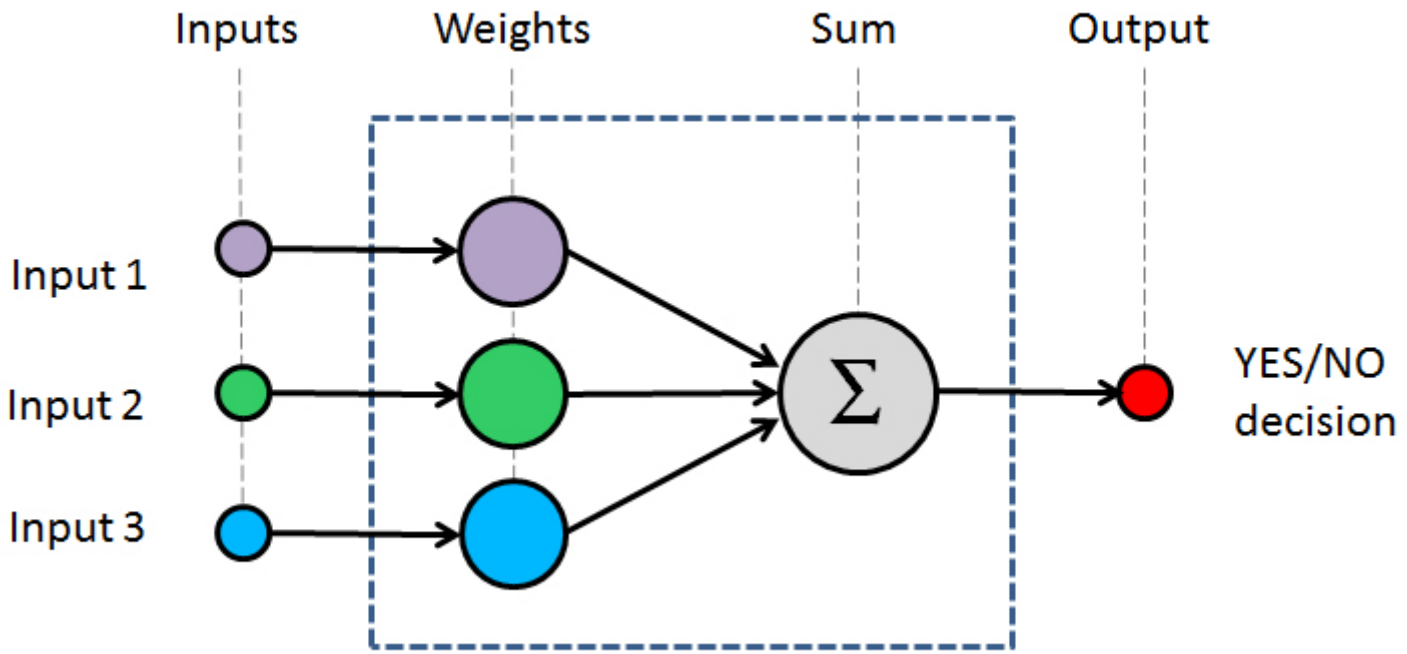
- What is convolution?
- Why should I care?
- How do I use it?

What is convolution?



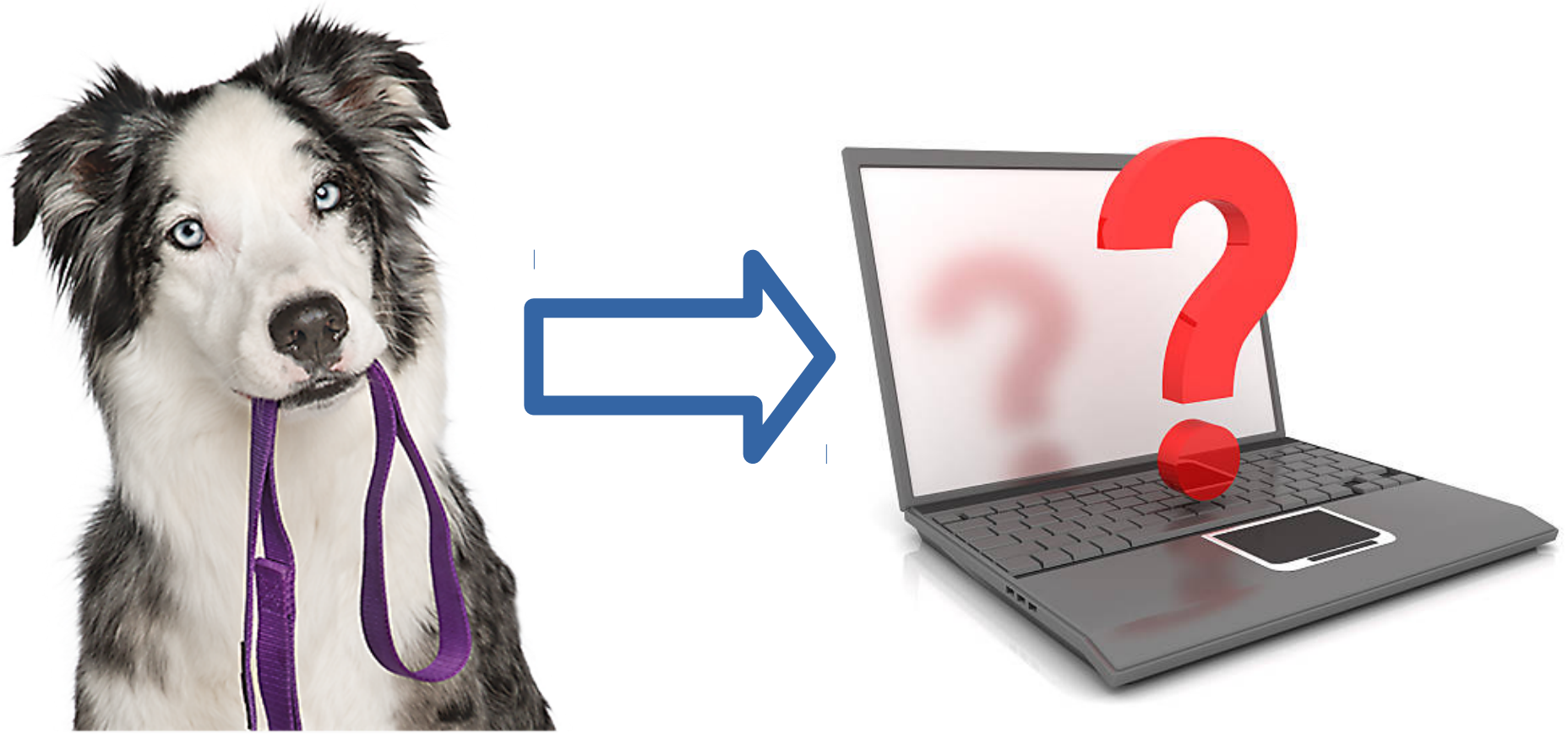
$$X \bullet W = \sum_{i=1}^{n} x_i \times w_i$$

What is convolution?

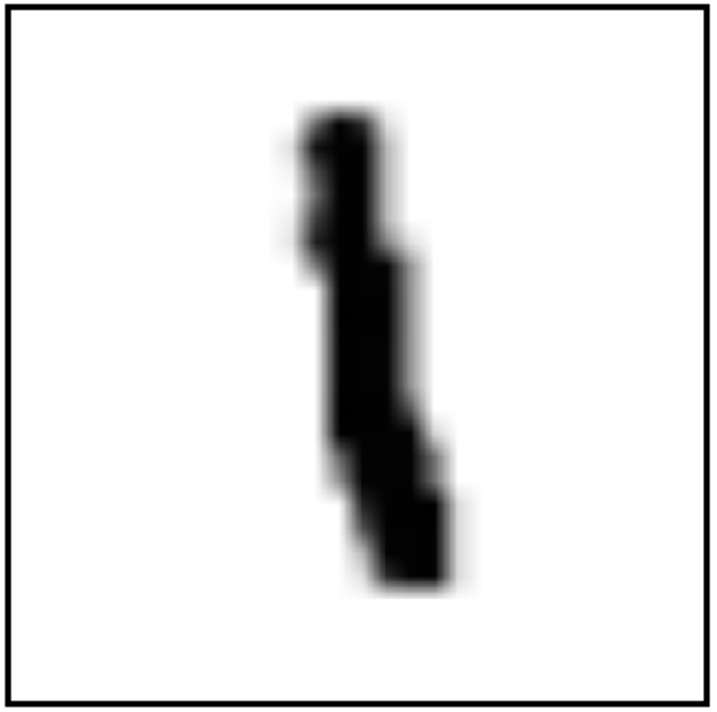



```
Out = 0
for index in number_of_inputs:
    Out = Out + (input[index] x weight[index])
```

What is convolution?



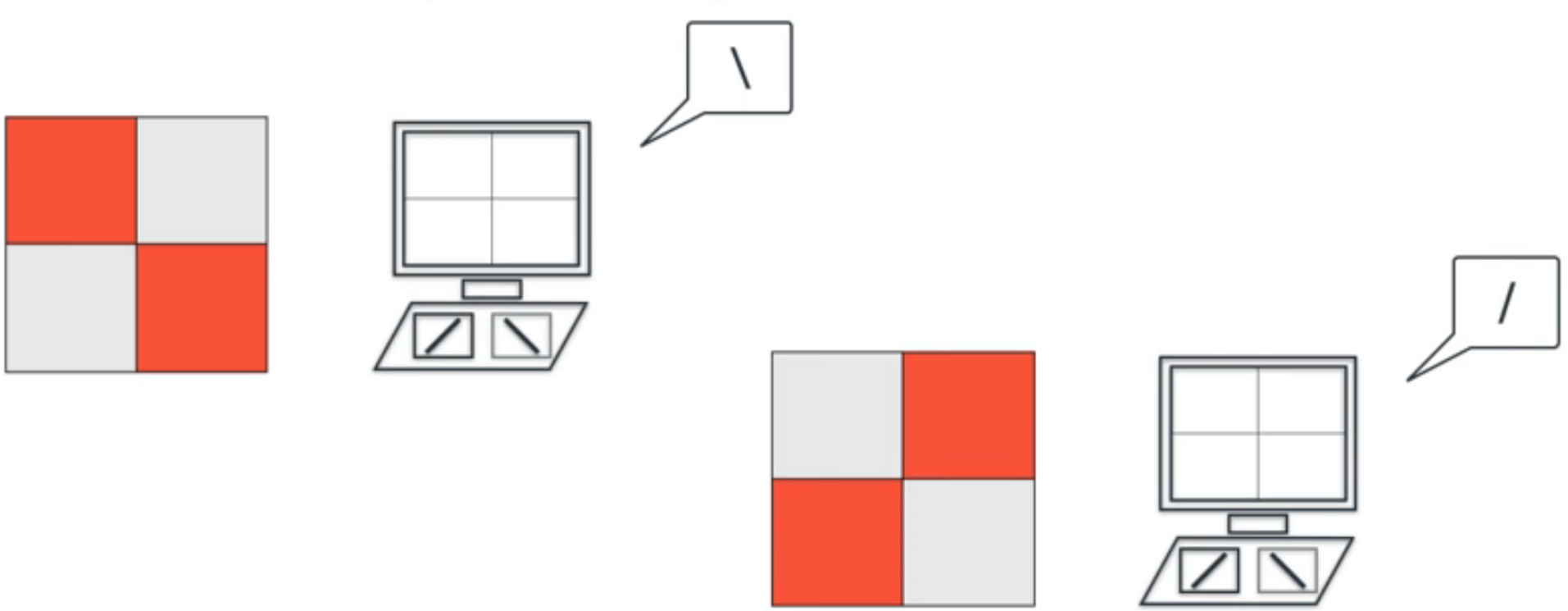
What is convolution?



$\approx$

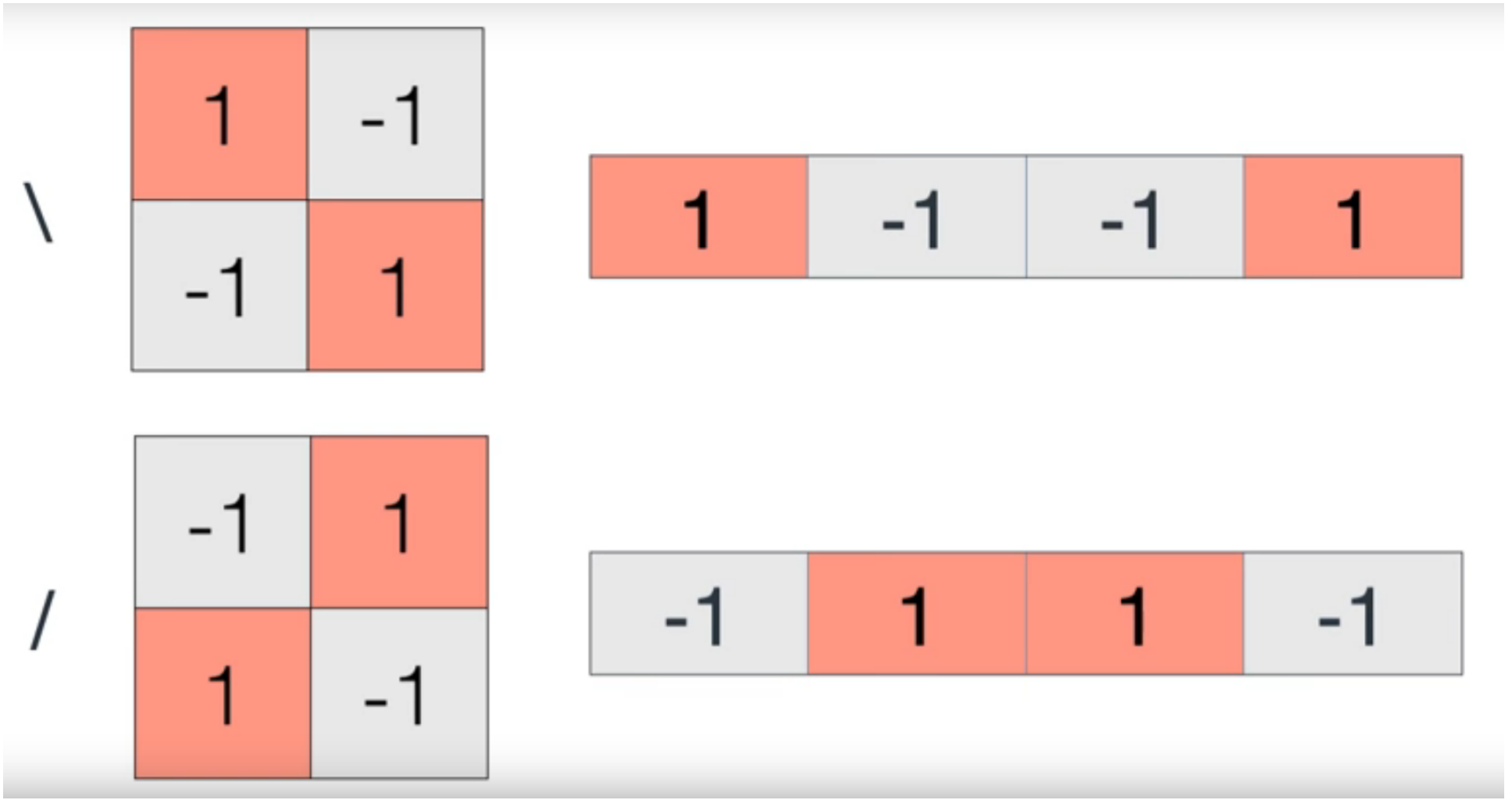
$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .6 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .7 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & .5 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & .3 & 1 & .1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

What is convolution?



SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?

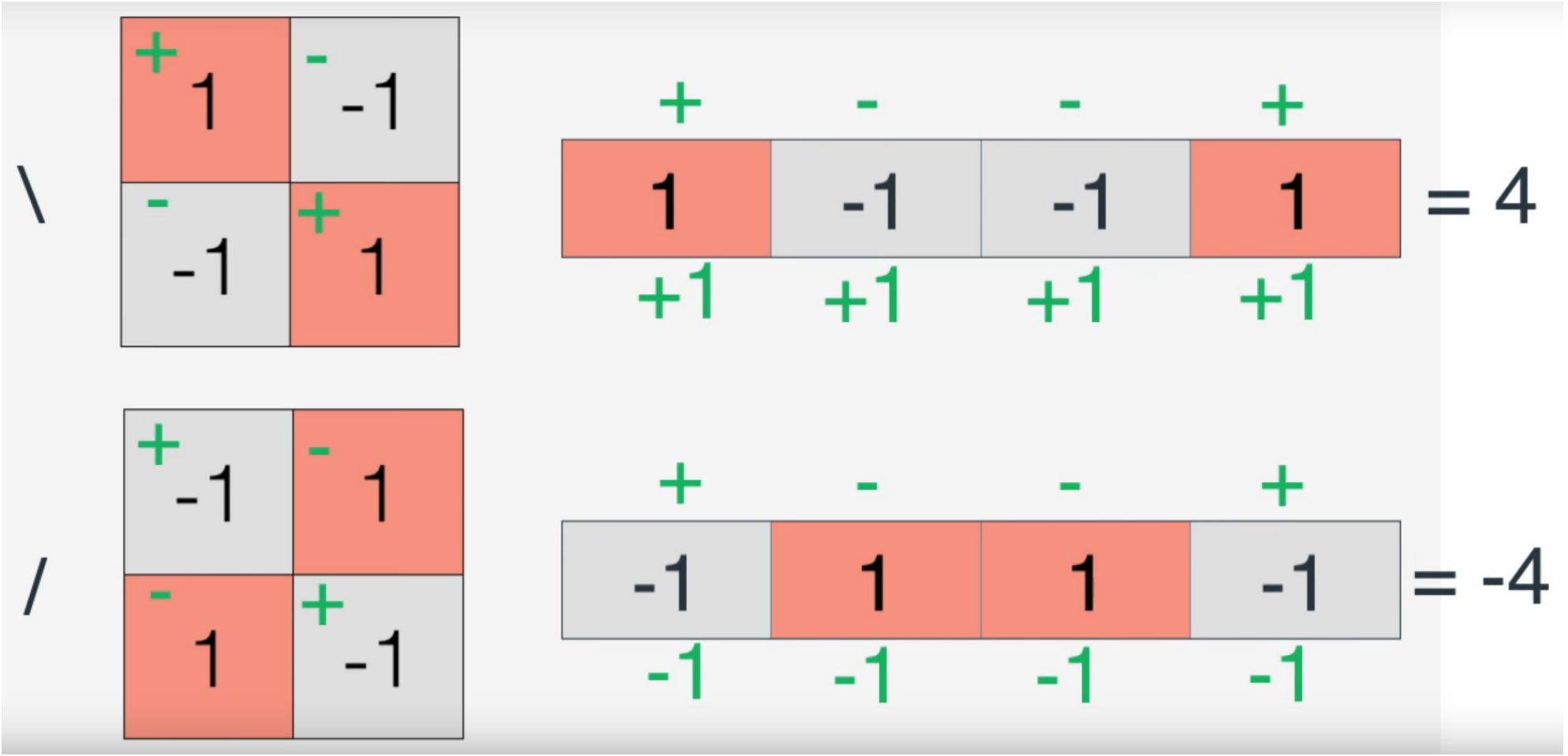


SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?

SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?



SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?

Convolutional Kernel

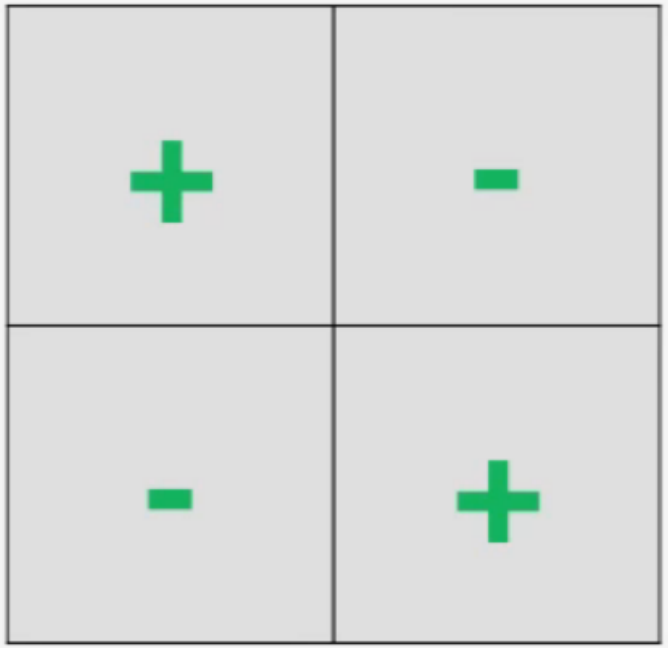


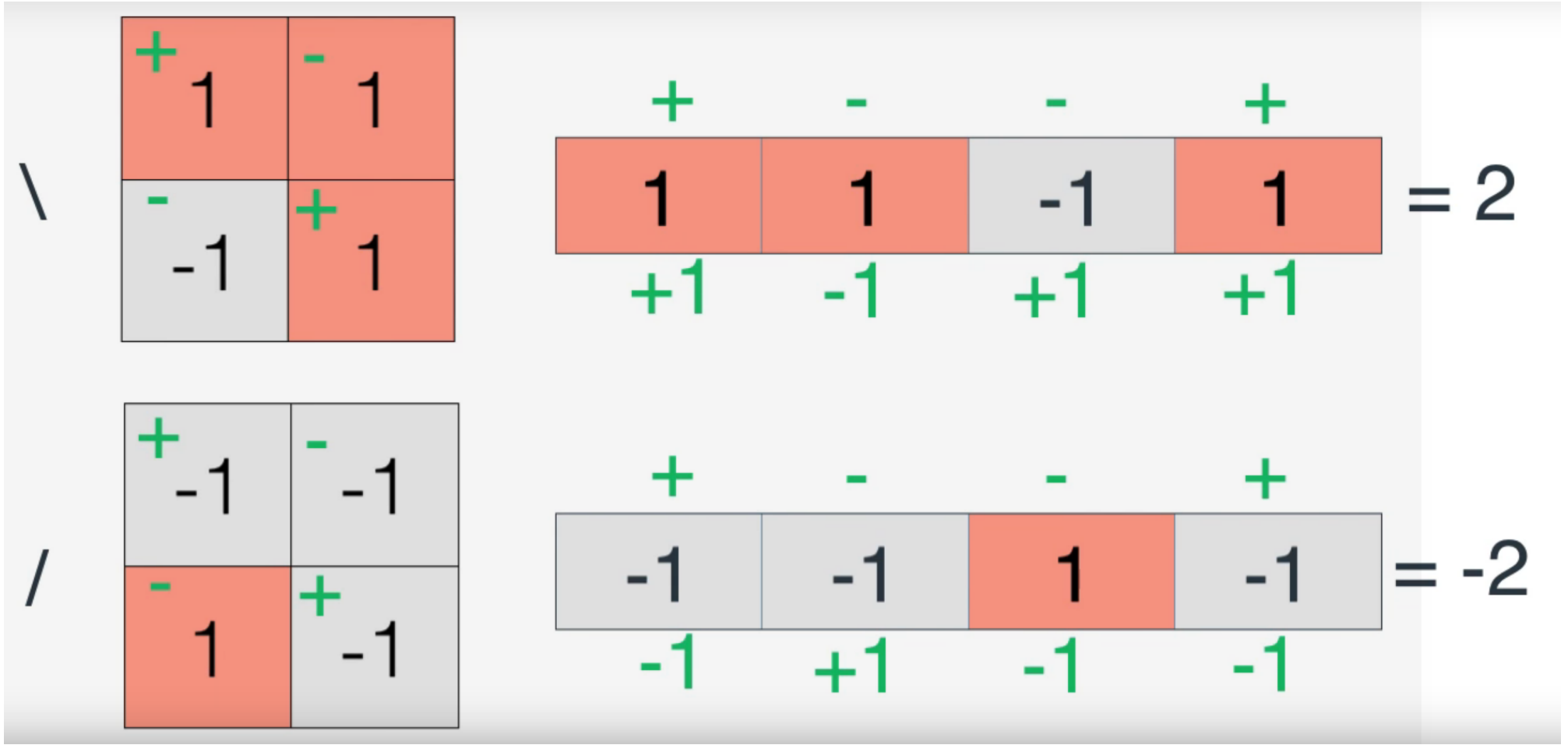
Image Classifier

If positive, “\”

If negative, “/”

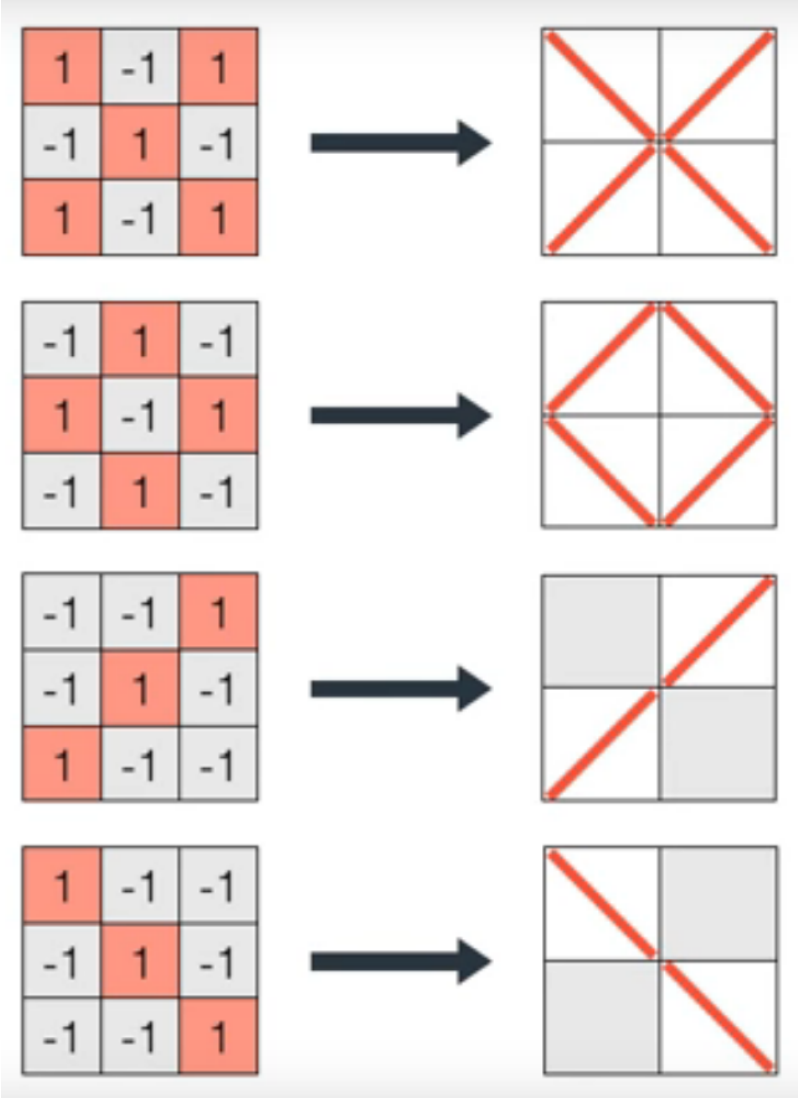
SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?



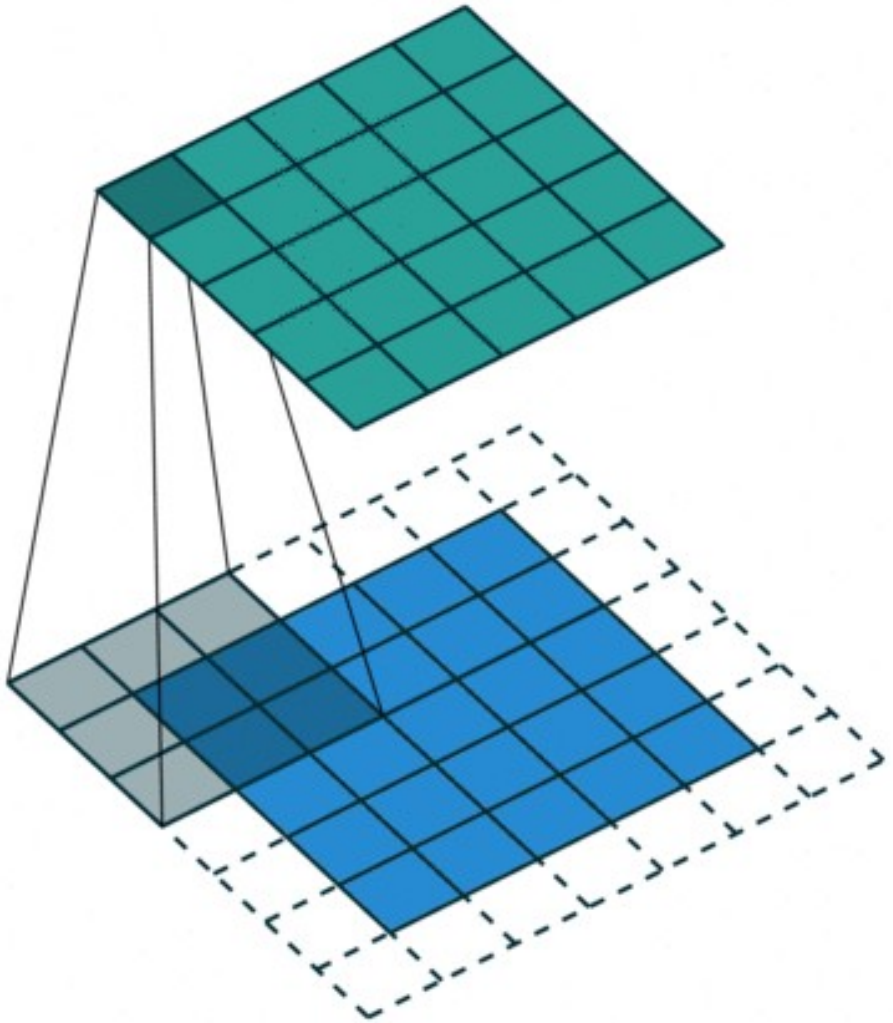
SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?



SOURCE: Luis Serrano, Udacity, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

What is convolution?



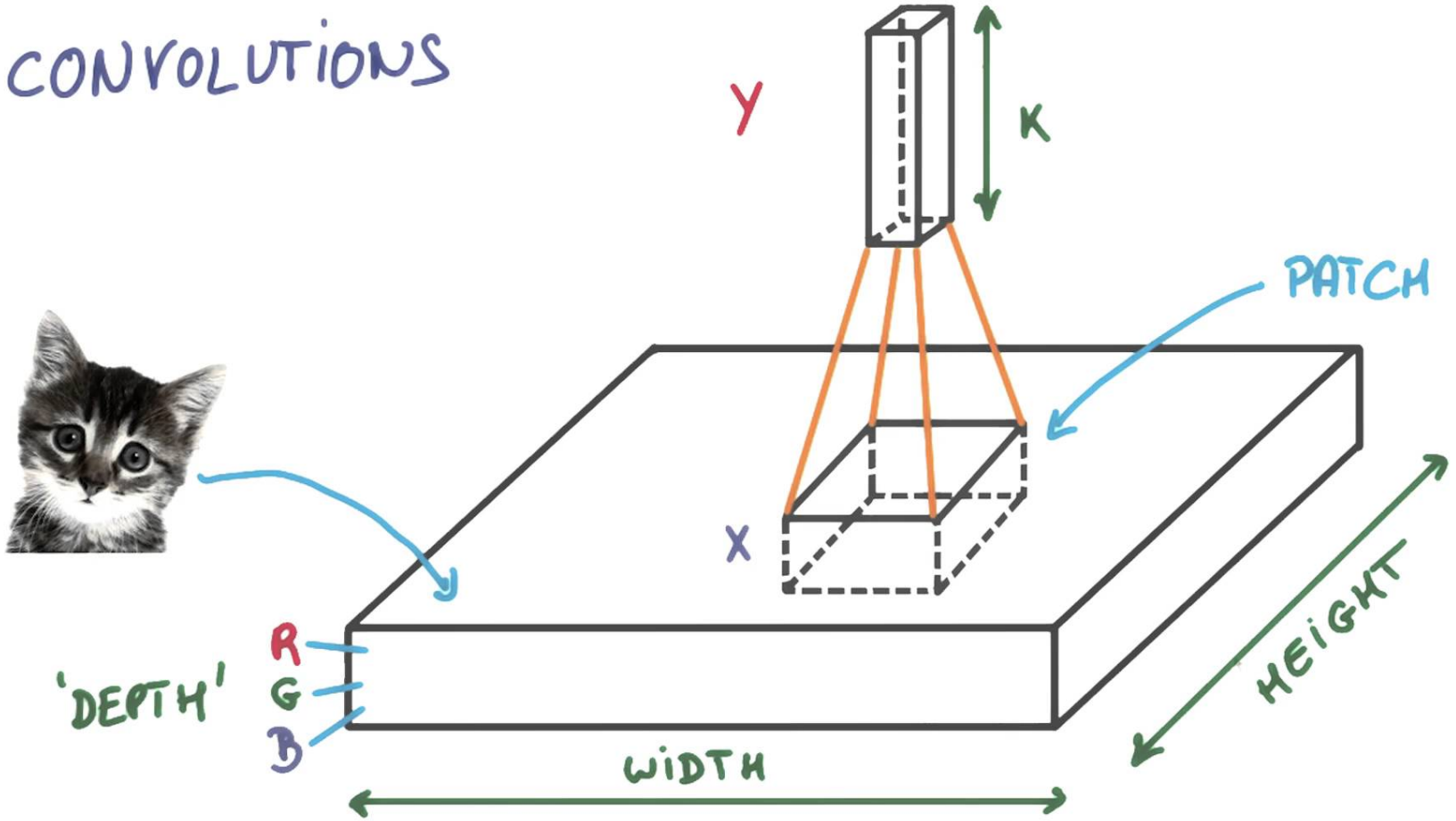
Feature Map

Kernel or Filter

Image

Source: Vincent Dumoulin, https://github.com/vdumoulin/conv_arithmetic

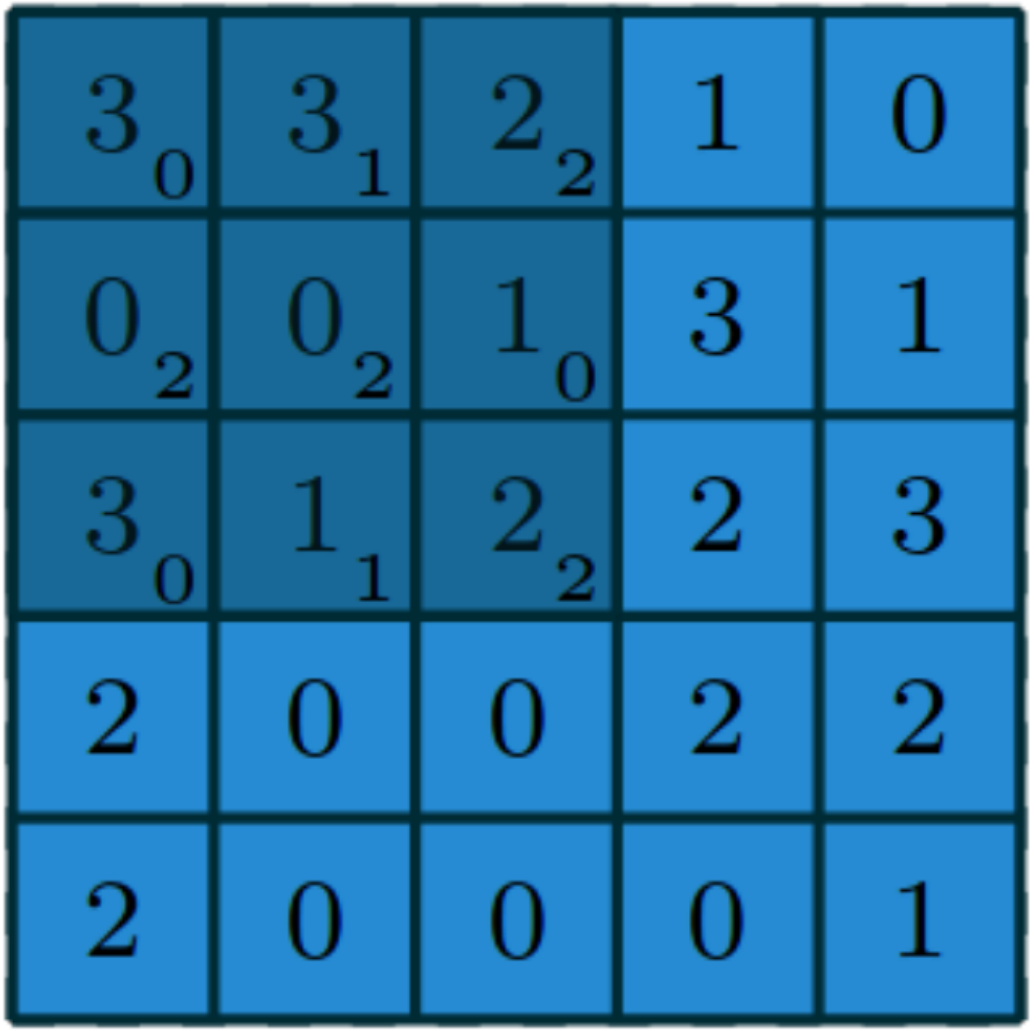
What is convolution?



Source: Vincent Vanhoucke, Udacity, https://www.youtube.com/watch?v=jajksuQW4mc

What is convolution?

Image



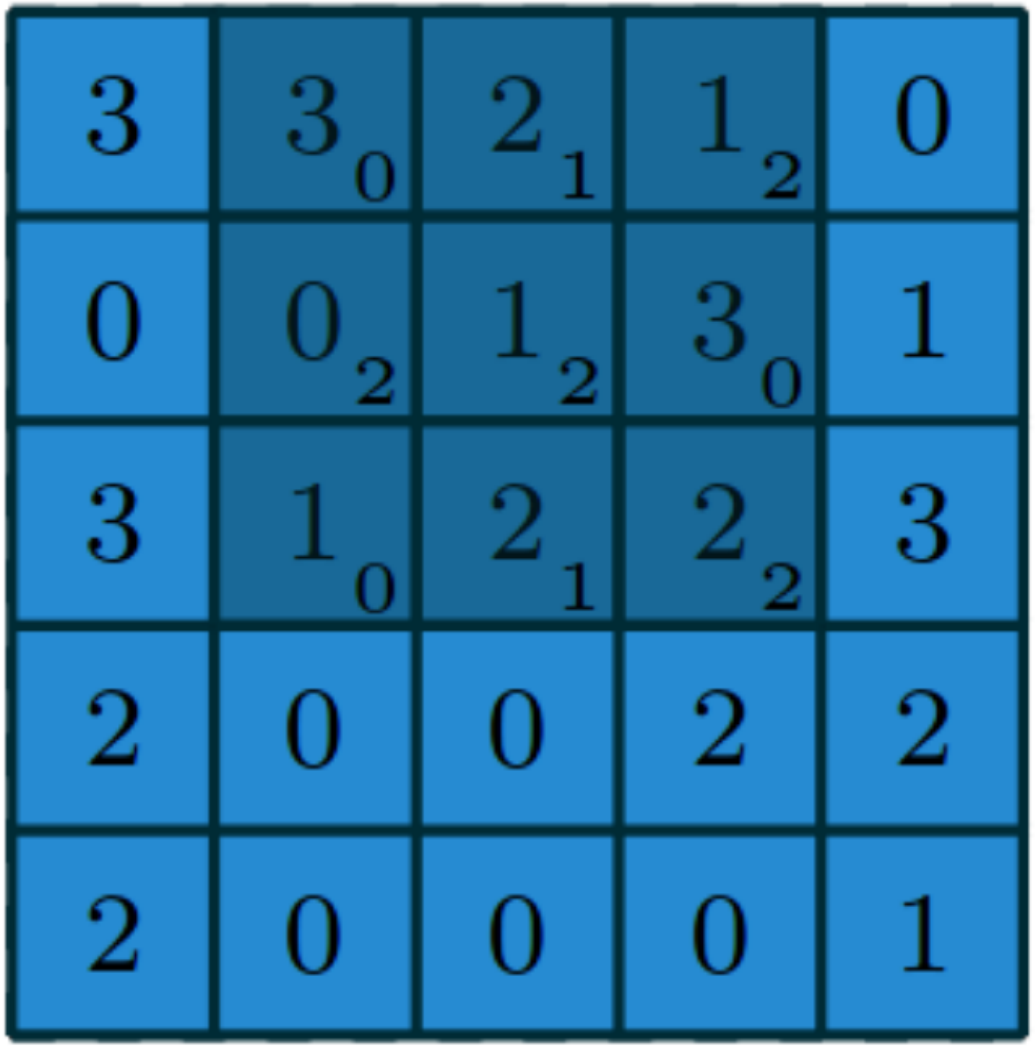
Feature Map

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Source: Vincent Dumoulin and Francesco Visin, https://arxiv.org/pdf/1603.07285.pdf

What is convolution?

Image



Feature Map

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Source: Vincent Dumoulin and Francesco Visin, https://arxiv.org/pdf/1603.07285.pdf

What is convolution?

Image

3	3	2_0	1_1	0_2
0	0	1_2	3_2	1_0
3	1	2_0	2_1	3_2
2	0	0	2	2
2	0	0	0	1

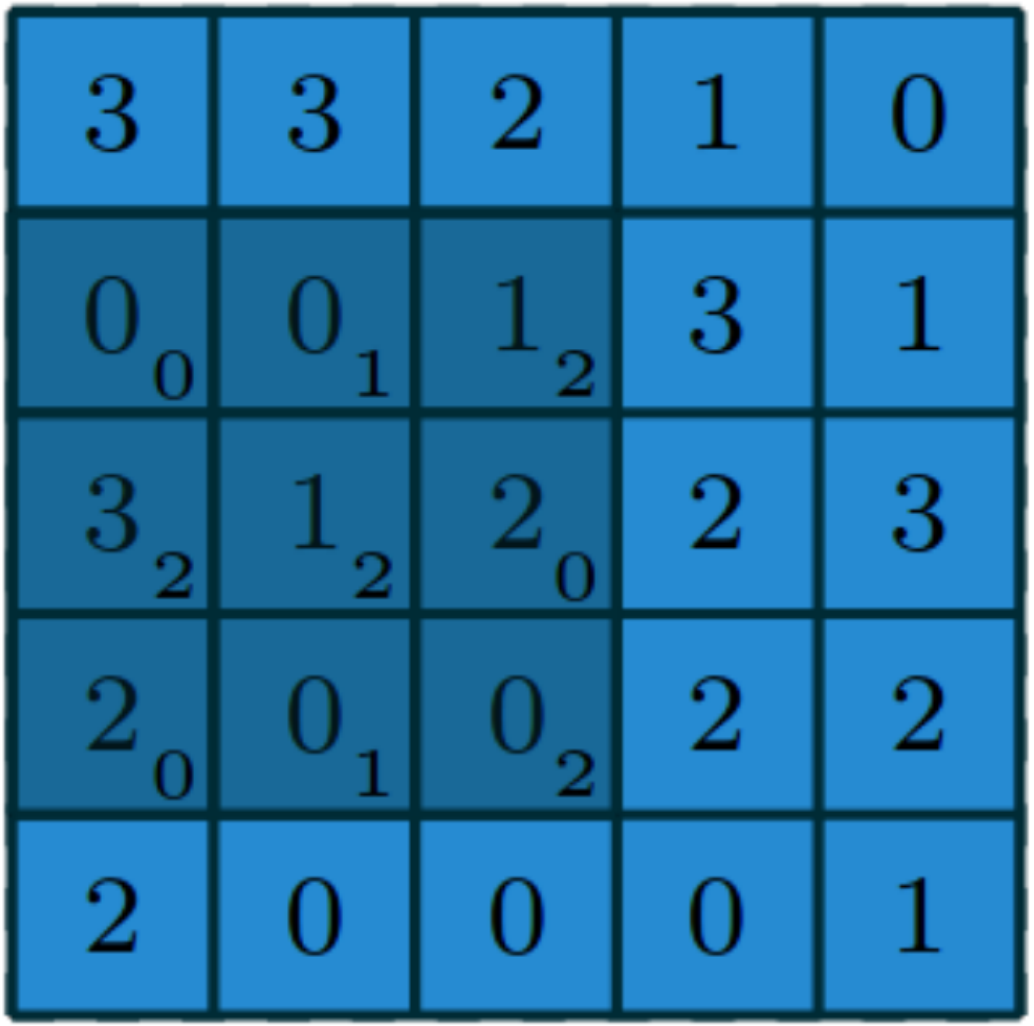
Feature Map

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Source: Vincent Dumoulin and Francesco Visin, https://arxiv.org/pdf/1603.07285.pdf

What is convolution?

Image



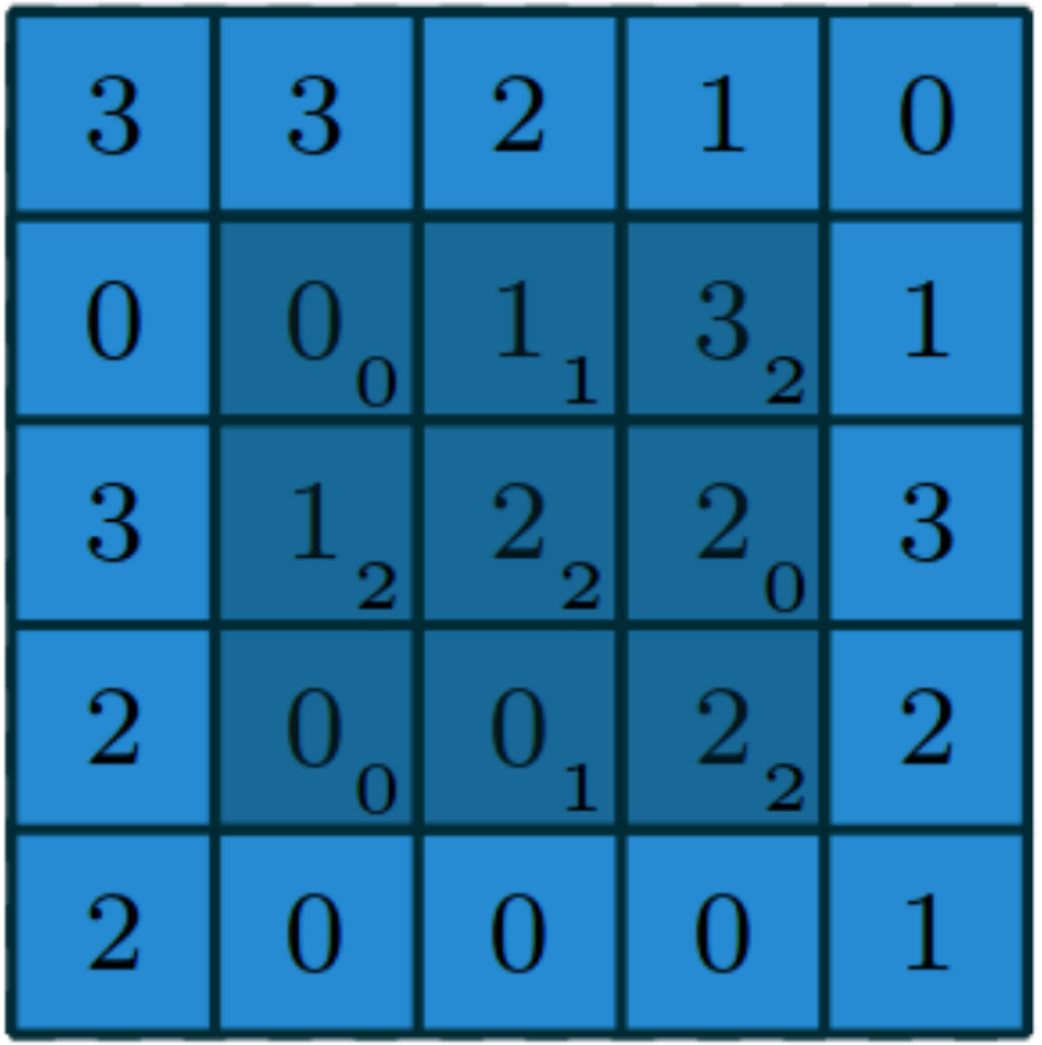
Feature Map

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Source: Vincent Dumoulin and Francesco Visin, https://arxiv.org/pdf/1603.07285.pdf

What is convolution?

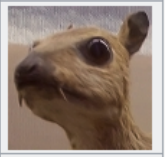
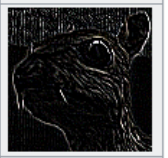
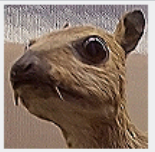
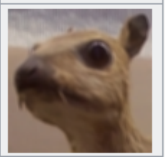
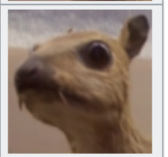
Image

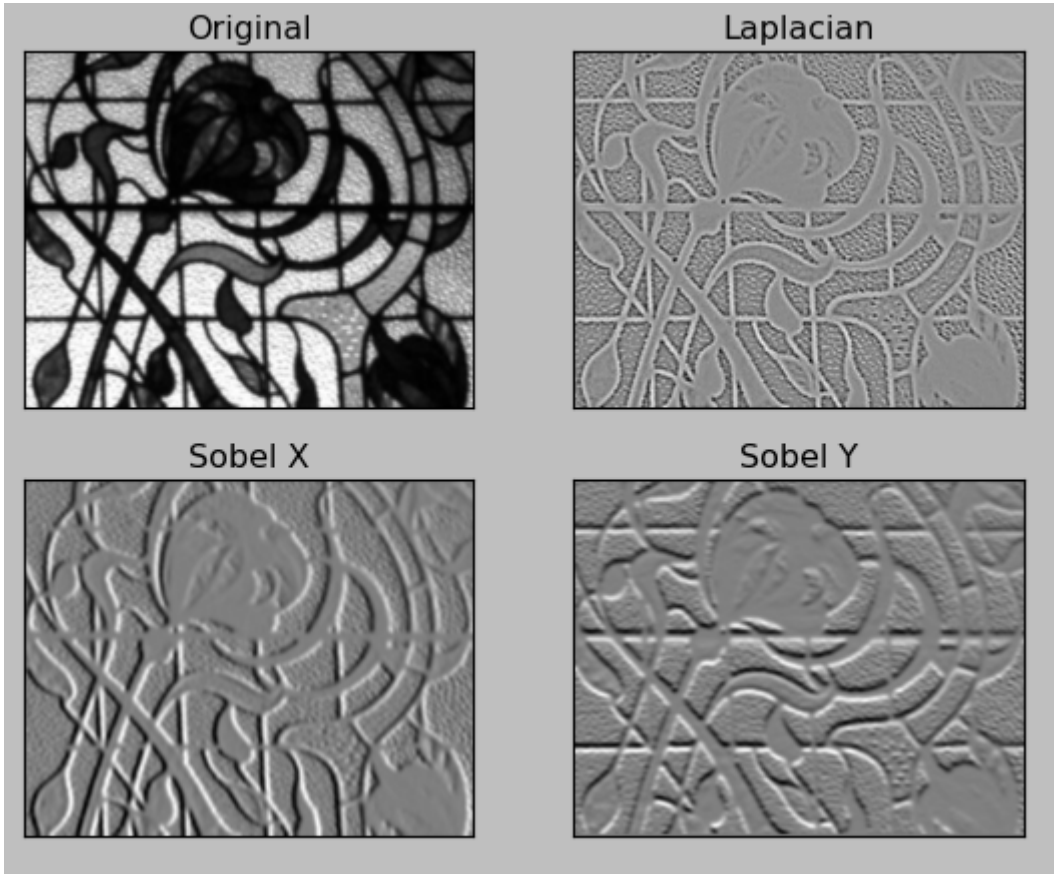


Feature Map

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Source: Vincent Dumoulin and Francesco Visin, https://arxiv.org/pdf/1603.07285.pdf

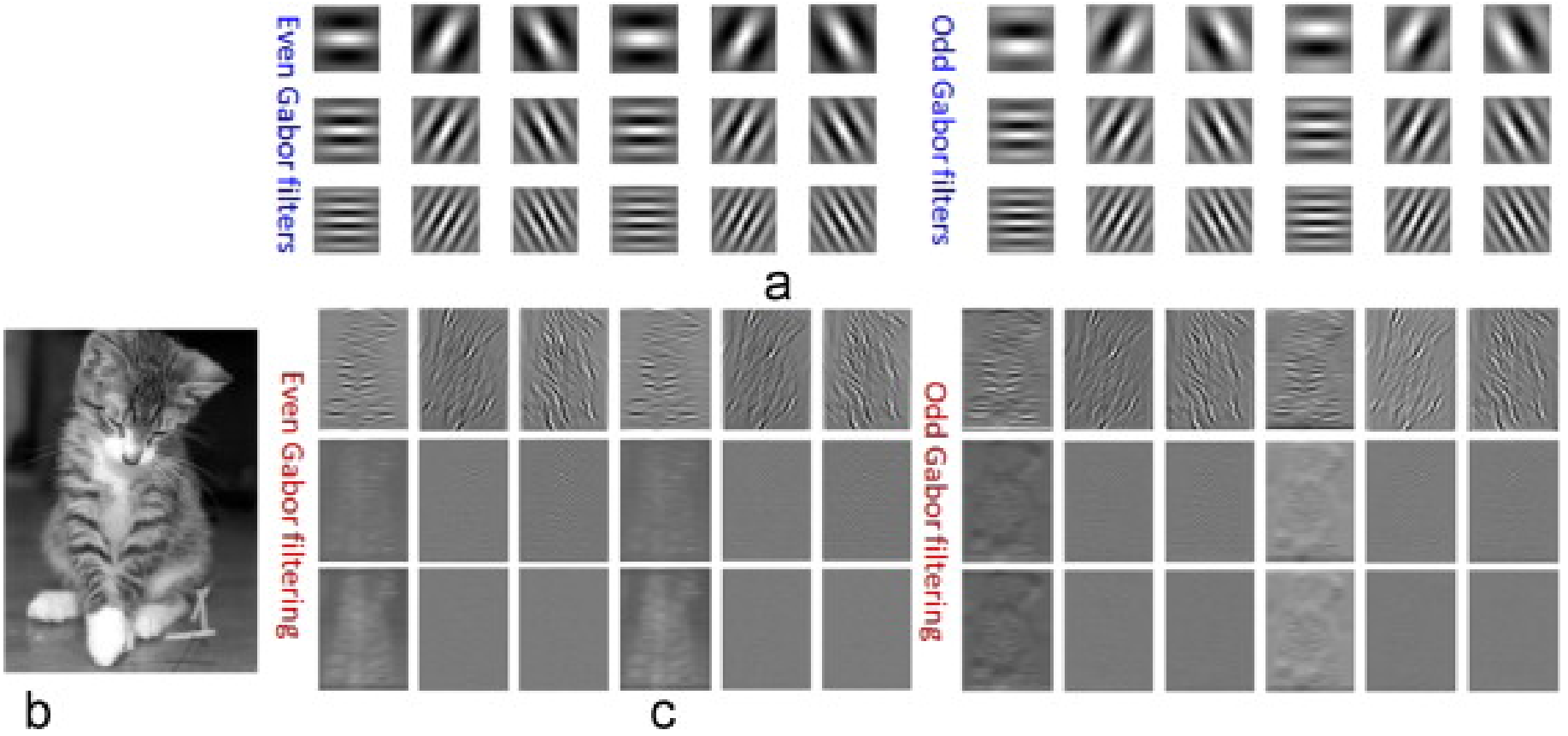
Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	



Watch the video: http://setosa.io/ev/image-kernels/

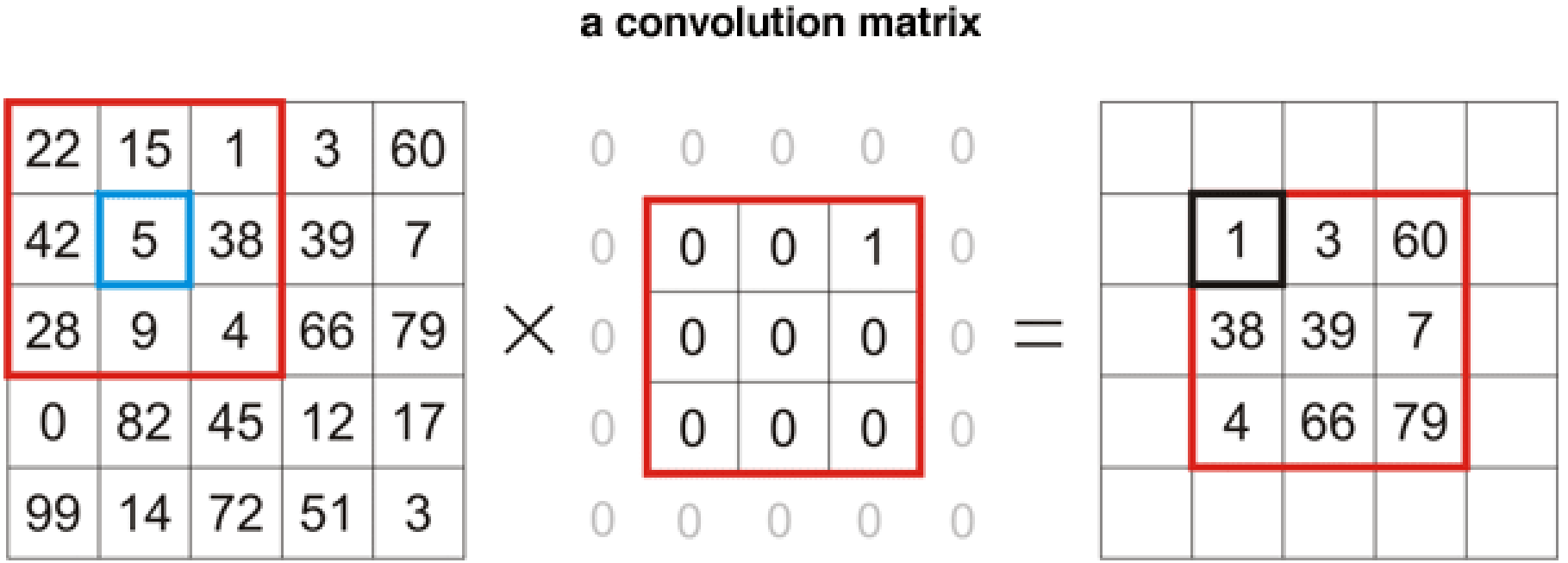
What is convolution?

Gabor Filter



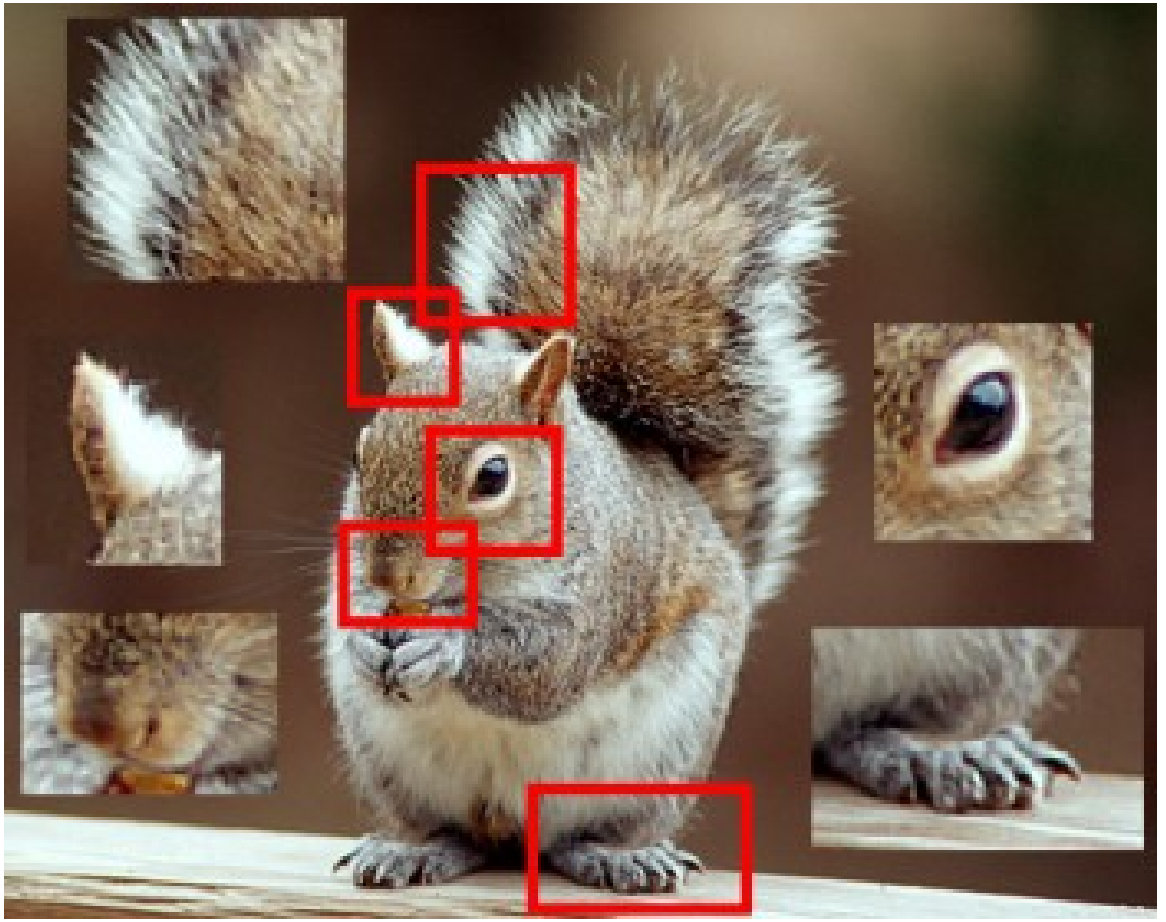
Source: http://www.sciencedirect.com/science/article/pii/S0031320315001570

What is convolution?



“Take the top right corner of the image?”

What is convolution?



Convolution creates FEATURE DETECTORS.

ImageNet

ImageNet 2010-2017

https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/

PASCAL VOC

Aeroplanes
Bicycles
Birds
Boats
Bottles
Buses
Cars
Cats
Chairs
Cows
Dining tables
Dogs
Horses
Motorbikes
People
Potted plants
Sheep
Sofas
Trains
TV/Monitors

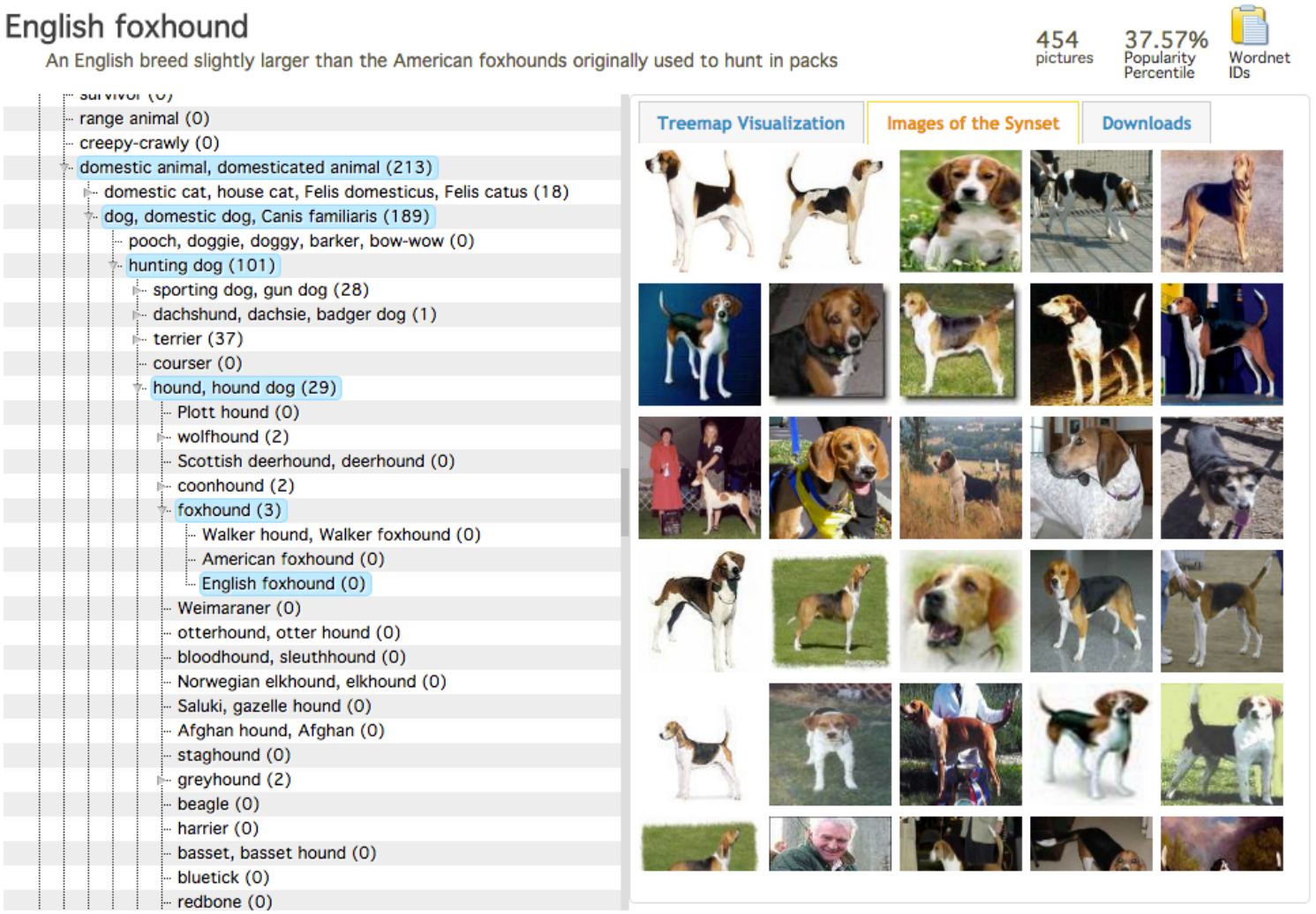
21,738

ImageNet

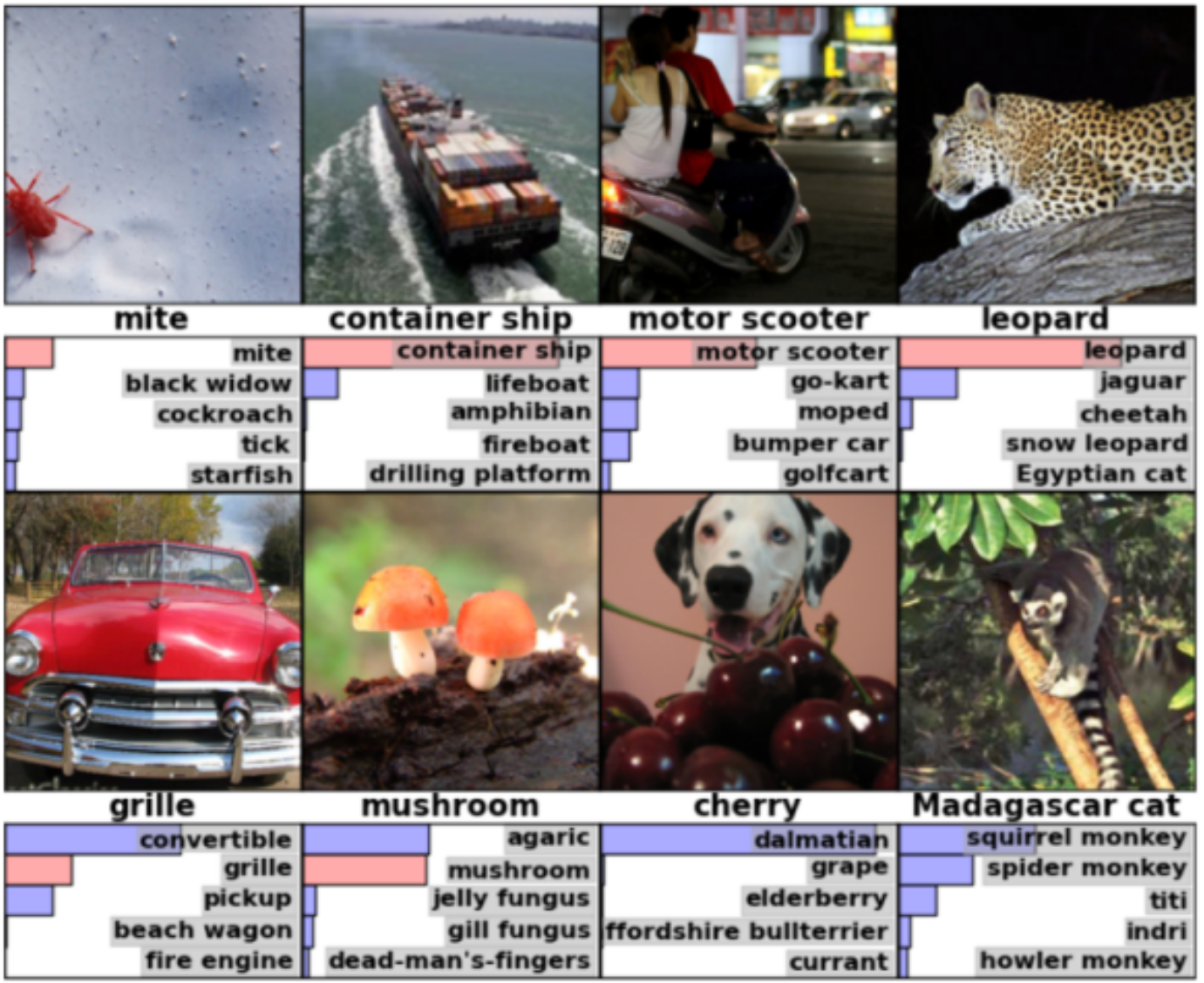
High level category	# synset (subcategories)	Avg # images per synset	Total # images
amphibian	94	591	56K
animal	3822	732	2799K
appliance	51	1164	59K
bird	856	949	812K
covering	946	819	774K
device	2385	675	1610K
fabric	262	690	181K
fish	566	494	280K
flower	462	735	339K
food	1495	670	1001K
fruit	309	607	188K
fungus	303	453	137K
furniture	187	1043	195K
geological formation	151	838	127K
invertebrate	728	573	417K
mammal	1138	821	934K
musical instrument	157	891	140K
plant	1666	600	999K
reptile	268	707	190K
sport	166	1207	200K
structure	1239	763	946K
tool	316	551	174K
tree	993	568	564K
utensil	86	912	78K
vegetable	176	764	135K
vehicle	481	778	374K
person	2035	468	952K

14,197,122

ImageNet

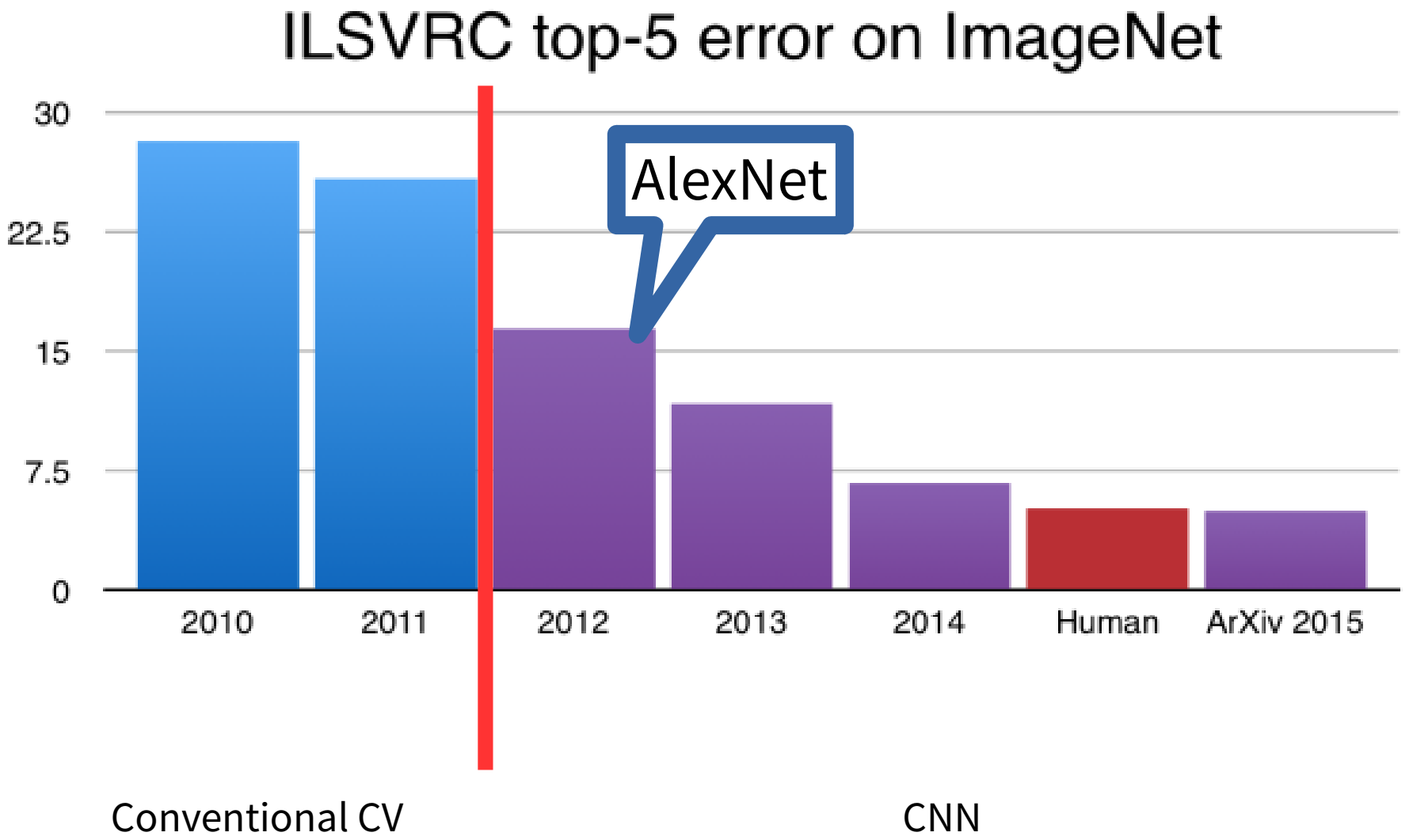

English foxhound
An English breed slightly larger than the American foxhounds originally used to hunt in packs
454 pictures
37.57% Popularity Percentile
Wordnet IDs
range animal (0)
creepy-crawly (0)
domestic animal, domesticated animal (213)
domestic cat, house cat, Felis domesticus, Felis catus (18)
dog, domestic dog, Canis familiaris (189)
pooch, doggie, doggy, barker, bow-wow (0)
hunting dog (101)
sporting dog, gun dog (28)
dachshund, dachsie, badger dog (1)
terrier (37)
courser (0)
hound, hound dog (29)
Plott hound (0)
wolfhound (2)
Scottish deerhound, deerhound (0)
coonhound (2)
foxhound (3)
Walker hound, Walker foxhound (0)
American foxhound (0)
English foxhound (0)
Weimaraner (0)
otterhound, otter hound (0)
bloodhound, sleuthhound (0)
Norwegian elkhound, elkhound (0)
Saluki, gazelle hound (0)
Afghan hound, Afghan (0)
staghound (0)
greyhound (2)
beagle (0)
harrier (0)
basset, basset hound (0)
bluetick (0)
redbone (0)
Treemap Visualization
Images of the Synset
Downloads

ImageNet



Source: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

ImageNet

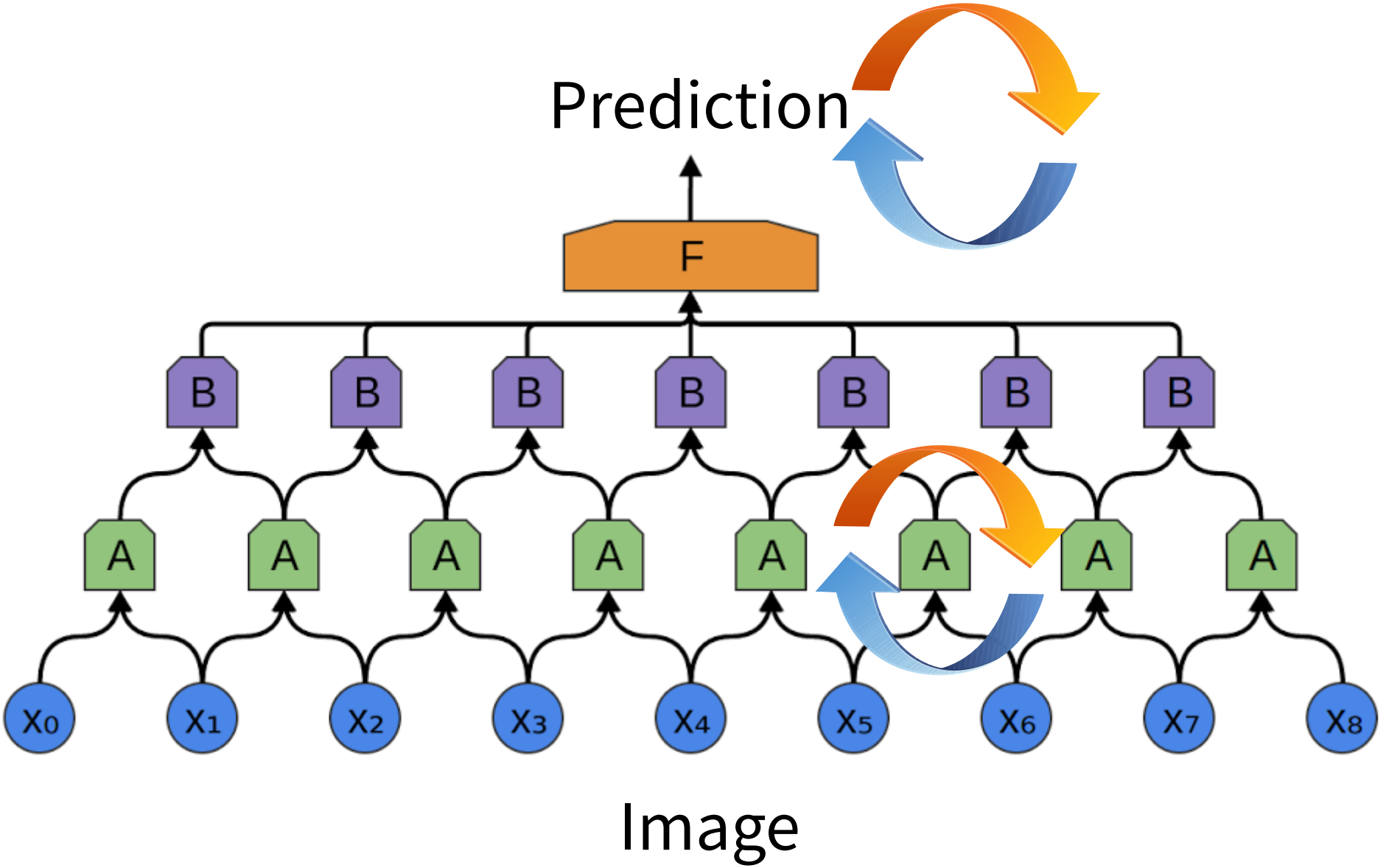

ILSVRC top-5 error on ImageNet
30
22.5
15
7.5
0
AlexNet
2010
2011
2012
2013
2014
Human
ArXiv 2015
Conventional CV
CNN

Convolutional Networks

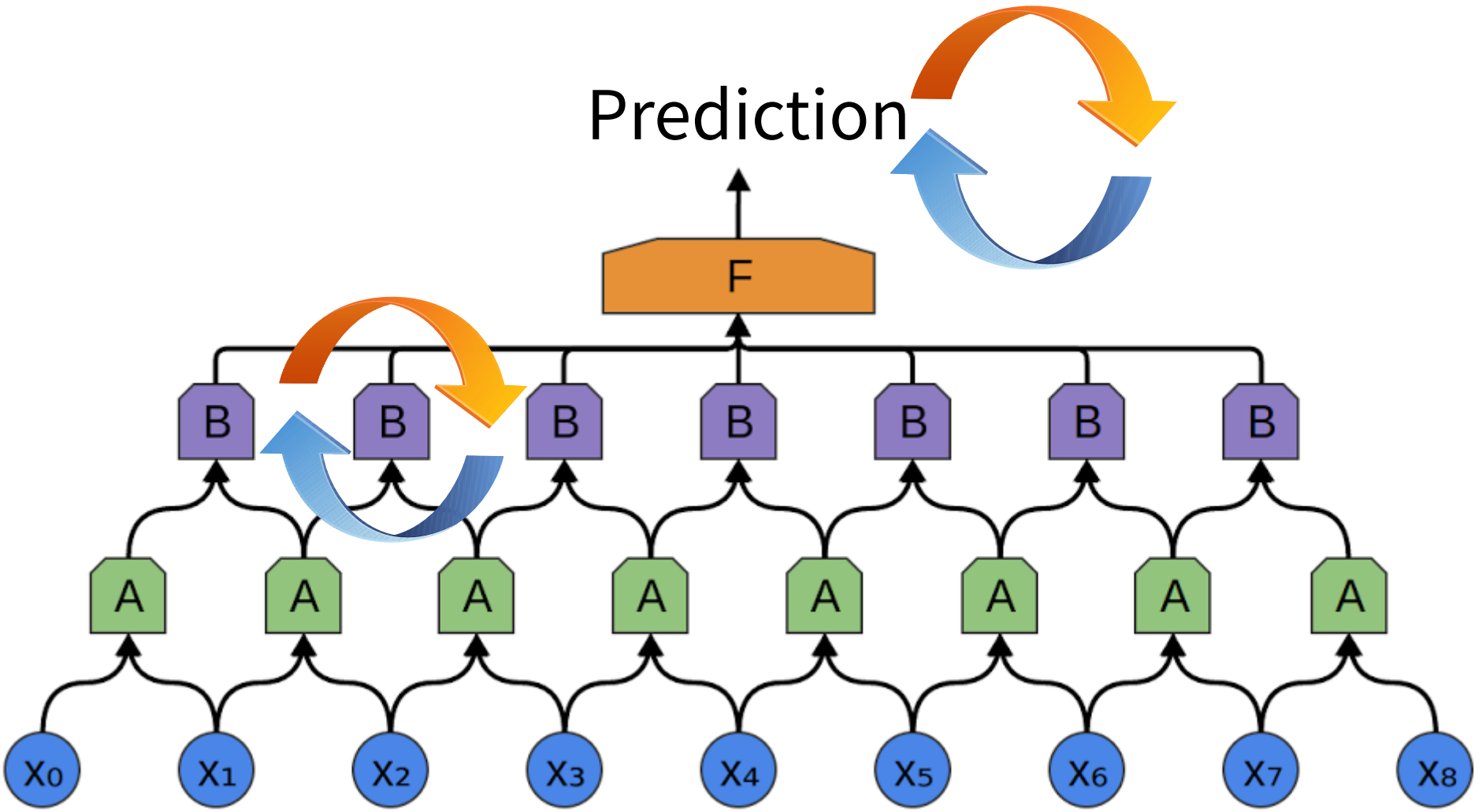
AlexNet

- Convolutional Network
- Dropout
- Rectified Linear Unit (ReLu)
- Backpropagation
- Stochastic Gradient Descent

Convolutional Networks

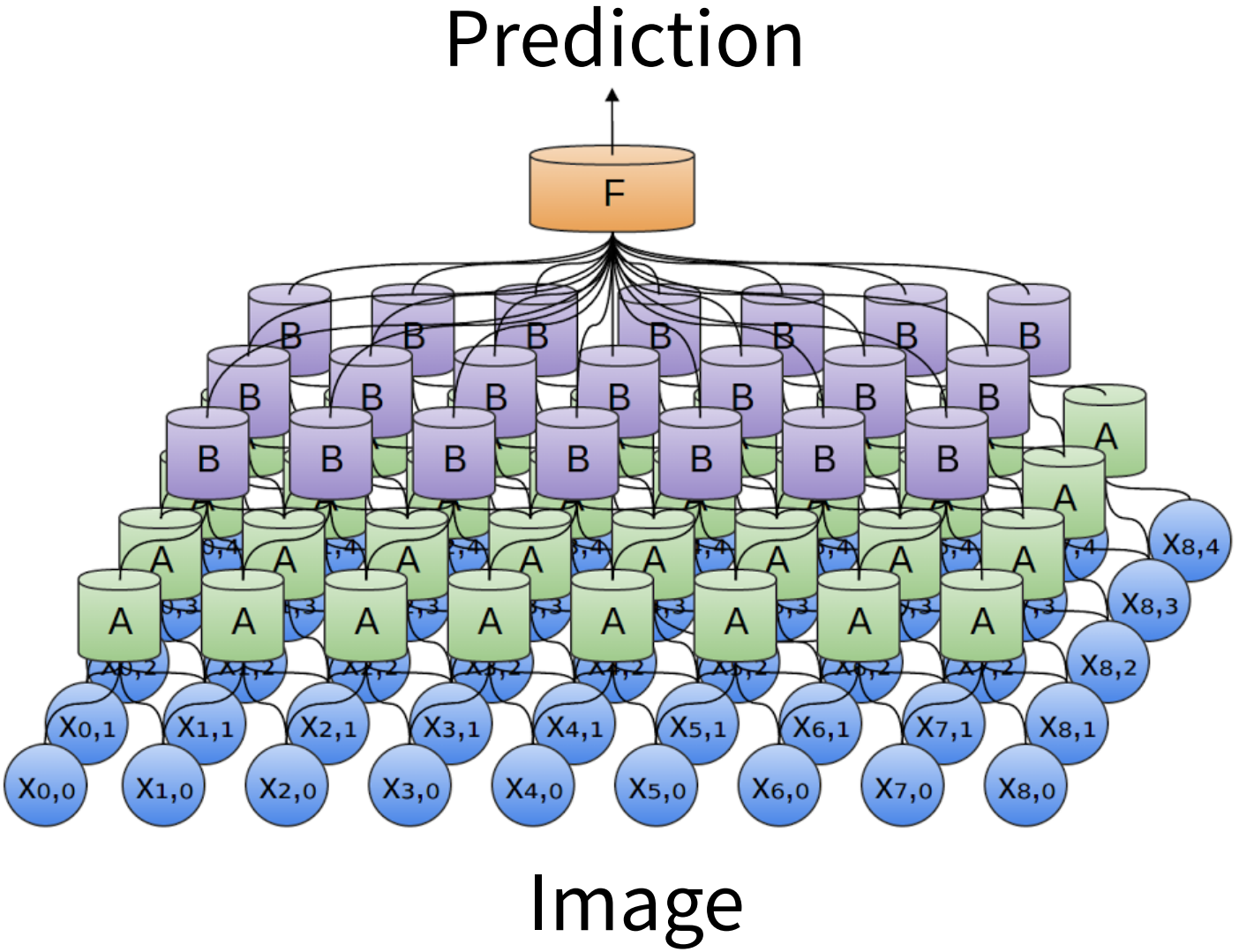


Convolutional Networks

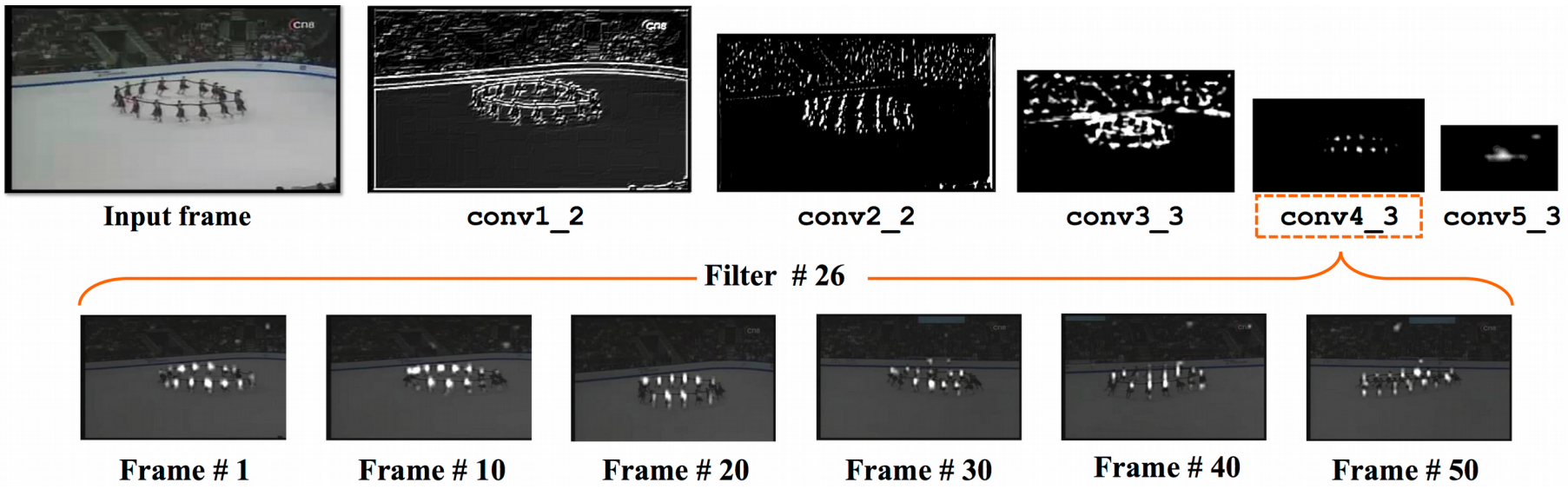


Backpropagation: http://colah.github.io/posts/2015-08-Backprop/

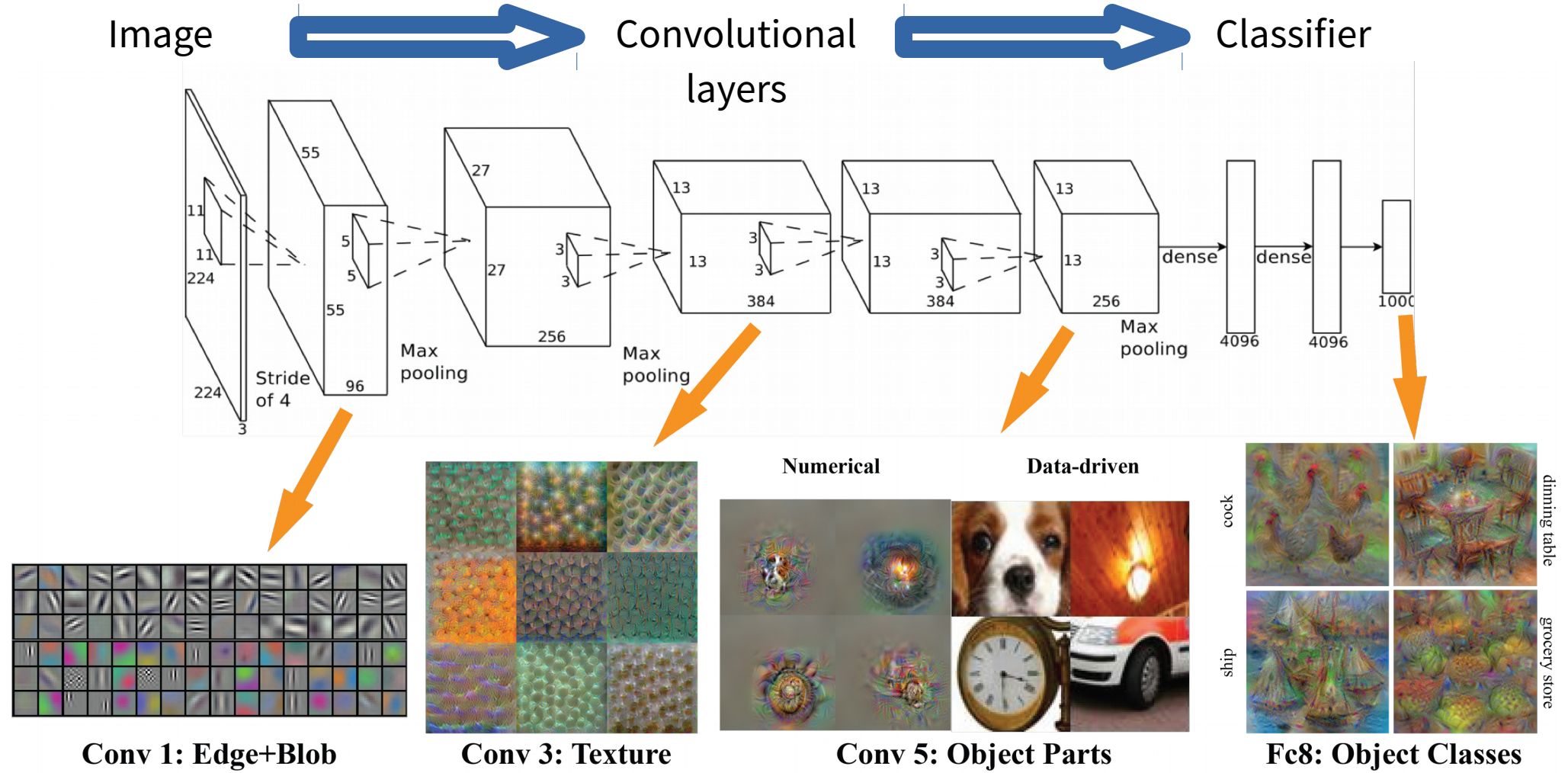
Convolutional Networks



Convolutional Networks



Convolutional Networks



Source: http://vision03.csail.mit.edu/cnn_art/data/single_layer.png

Convolutional Networks

Deep Visualization Toolbox

yosinski.com/deepvis

#deepvis


Jason Yosinski


Jeff Clune


Anh Nguyen


Thomas Fuchs

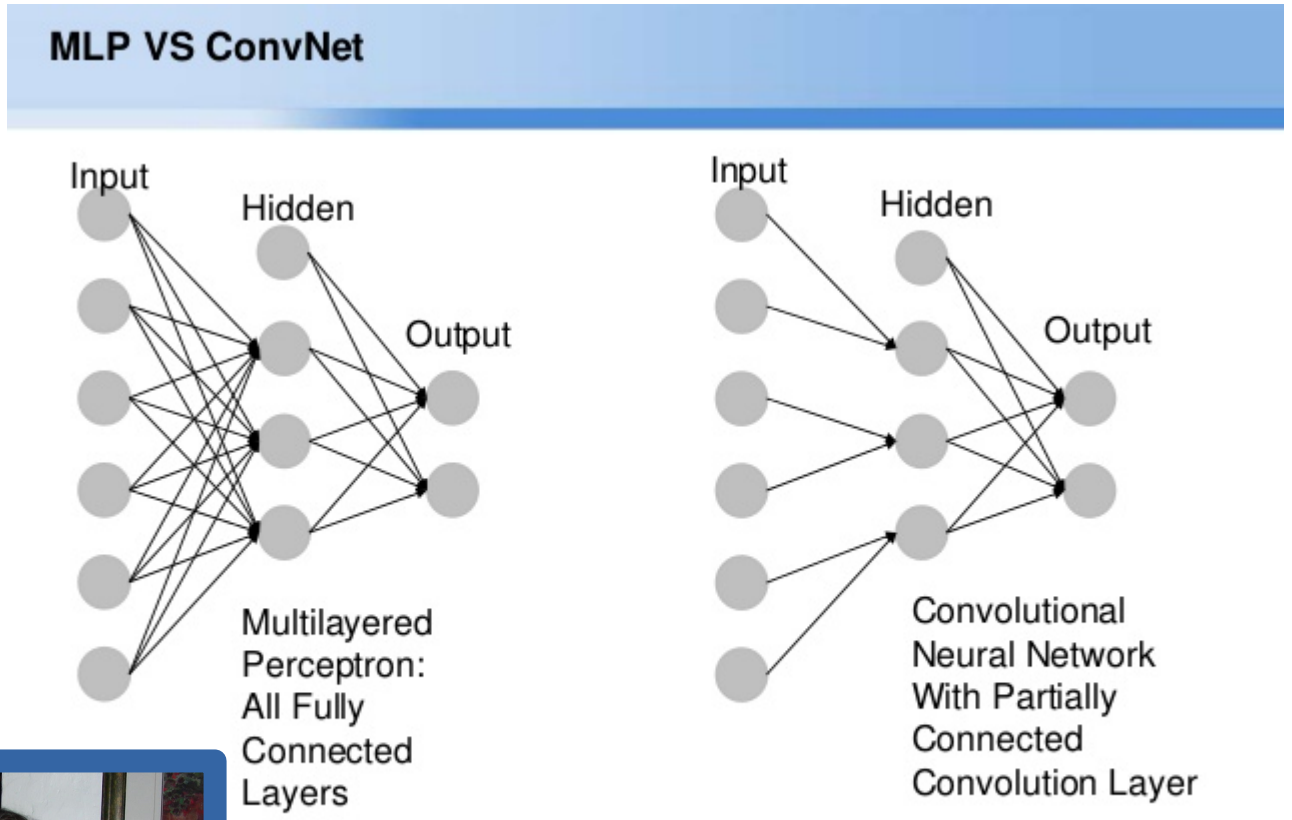

Hod Lipson

Cornell University

University of Wyoming

NASA Jet Propulsion Laboratory
California Institute of Technology

Convolutional Networks



Convolutional Networks

MLP

98.2%

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 784)               615440
_________________________________________________________________
dense_2 (Dense)              (None, 10)                7850
=================================================================
Total params: 623,290
Trainable params: 623,290
Non-trainable params: 0
```

CNN

99.3%

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 30, 24, 24)        780
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 30, 12, 12)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 15, 10, 10)        4065
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 5, 5)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 15, 5, 5)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 375)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               48128
_________________________________________________________________
dense_2 (Dense)              (None, 50)                6450
_________________________________________________________________
dense_3 (Dense)              (None, 10)                510
=================================================================
Total params: 59,933
Trainable params: 59,933
Non-trainable params: 0
```

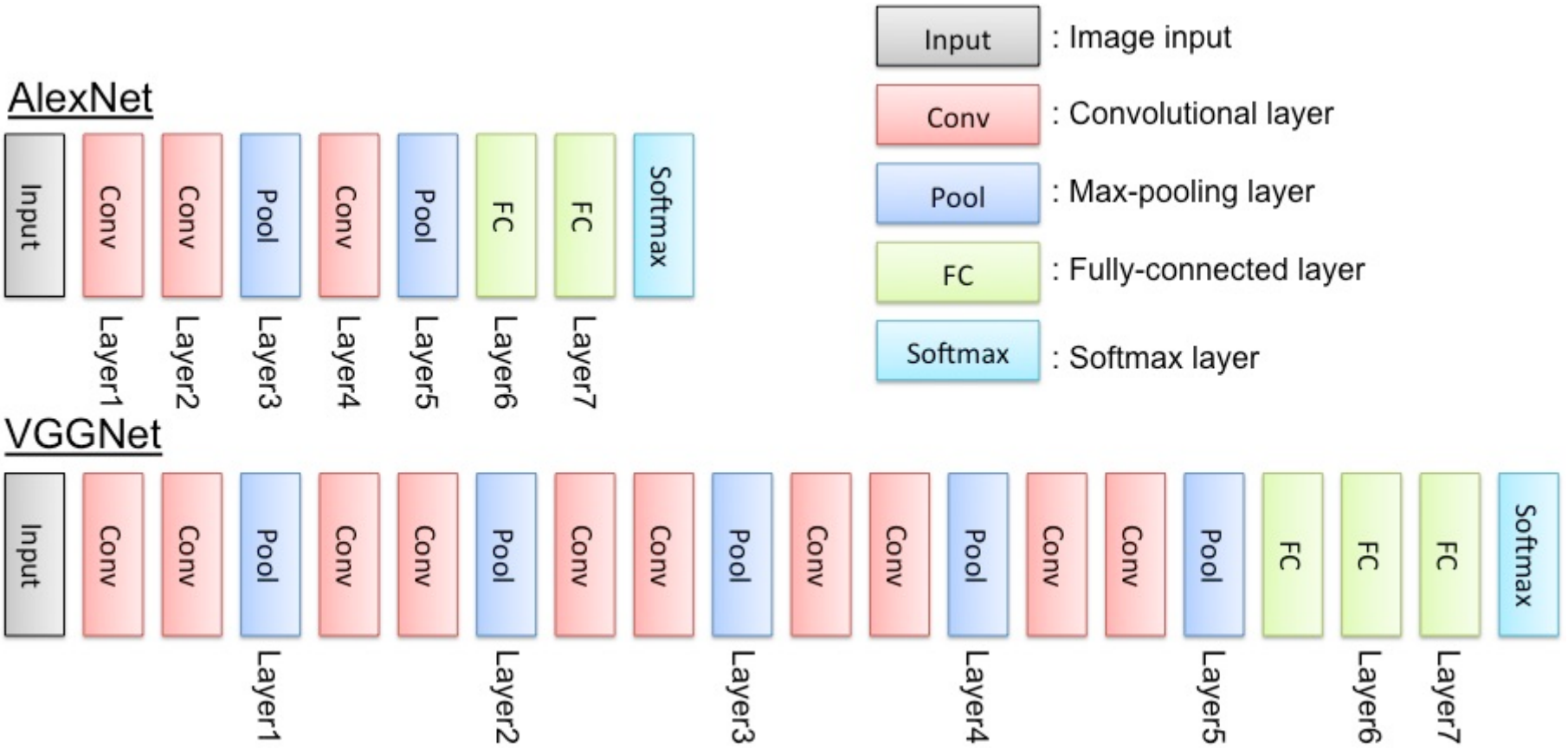
https://github.com/mas-dse-greina/dse999/blob/master/chapter_19/MNIST%20CNN%20Large%20Notebook.ipynb

Code Example in Keras

```
# define the convolutional model
def conv_model():

    model = Sequential()

    # input_shape tells Keras/Tensorflow to
    #      expect a tensor input of 1 x 28 x 28
    model.add(Conv2D(30, (5, 5), input_shape=(1, 28, 28), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Conv2D(15, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))

    # Compile model
    model.compile(loss='categorical_crossentropy',
                  optimizer='adam', metrics=['accuracy'])
    return model
```

https://github.com/mas-dse-greina/dse999/blob/master/chapter_19/MNIST%20CNN%20Large%20Notebook.ipynb

AlexNet v. VGG



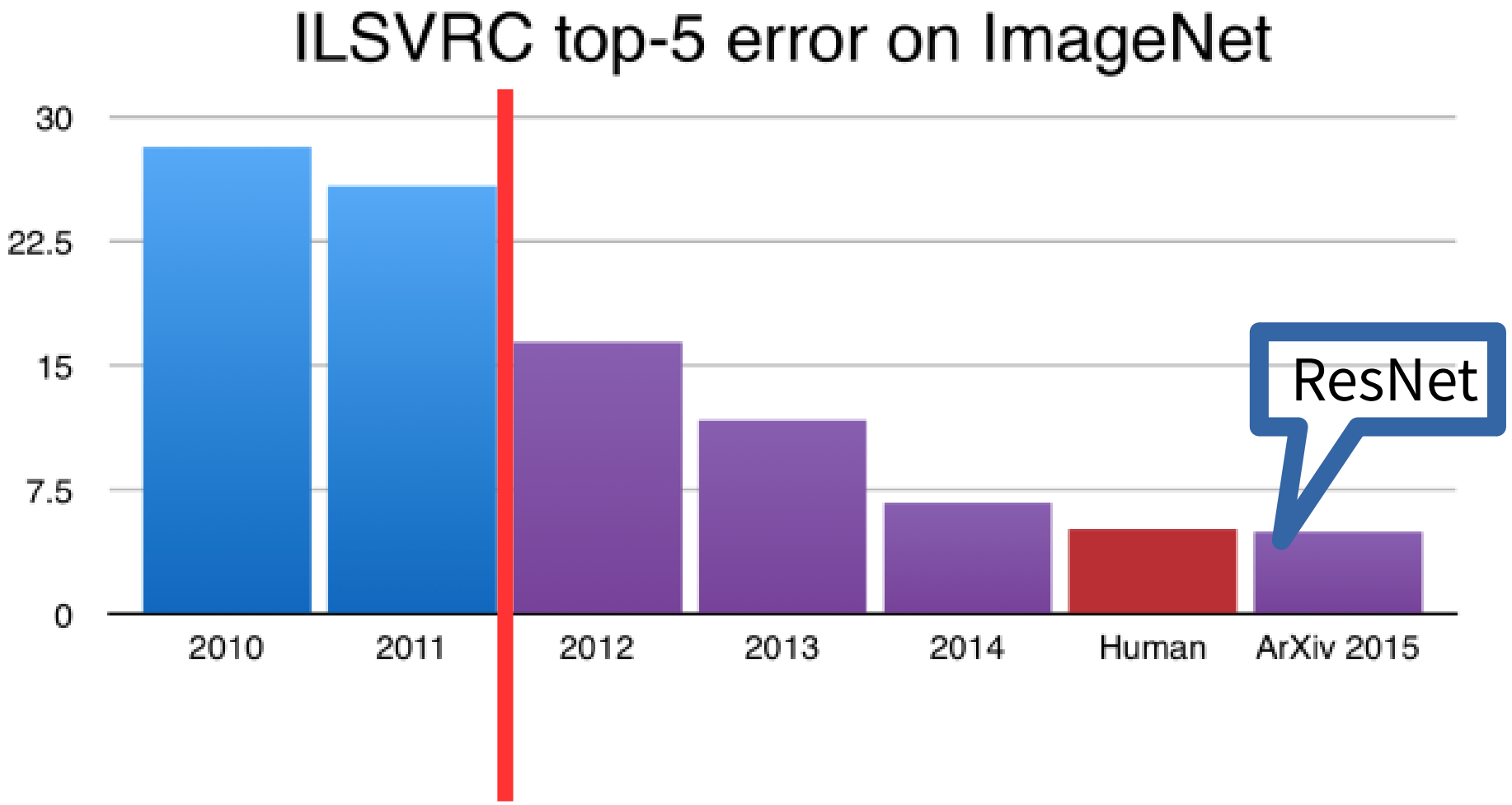
https://github.com/mas-dse-greina/neon/blob/master/luna16/old_code/LUNA16_VGG.py

Convolutional Networks

```
# Set up the model layers
vgg_layers = []
# set up 3x3 conv stacks with different number of filters
vgg_layers.append(Conv((3, 3, 64), **conv_params))
vgg_layers.append(Conv((3, 3, 64), **conv_params))
vgg_layers.append(Pooling(2, strides=2))
vgg_layers.append(Conv((3, 3, 128), **conv_params))
vgg_layers.append(Conv((3, 3, 128), **conv_params))
vgg_layers.append(Pooling(2, strides=2))
vgg_layers.append(Conv((3, 3, 256), **conv_params))
vgg_layers.append(Conv((3, 3, 256), **conv_params))
vgg_layers.append(Pooling(2, strides=2))
vgg_layers.append(Conv((3, 3, 512), **conv_params))
vgg_layers.append(Conv((3, 3, 512), **conv_params))
vgg_layers.append(Pooling(2, strides=2))
vgg_layers.append(Conv((3, 3, 512), **conv_params))
vgg_layers.append(Conv((3, 3, 512), **conv_params))
vgg_layers.append(Pooling(2, strides=2))
vgg_layers.append(Affine(nout=4096, init=GlorotUniform(), bias=Constant(0), activation=relu))
vgg_layers.append(Dropout(keep=0.5))
vgg_layers.append(Affine(nout=4096, init=GlorotUniform(), bias=Constant(0), activation=relu))
vgg_layers.append(Dropout(keep=0.5))
vgg_layers.append(Affine(nout=1, init=GlorotUniform(), bias=Constant(0), activation=Logistic(), name="class_layer"))
```

https://github.com/mas-dse-greina/neon/blob/master/luna16/old_code/LUNA16_VGG.py

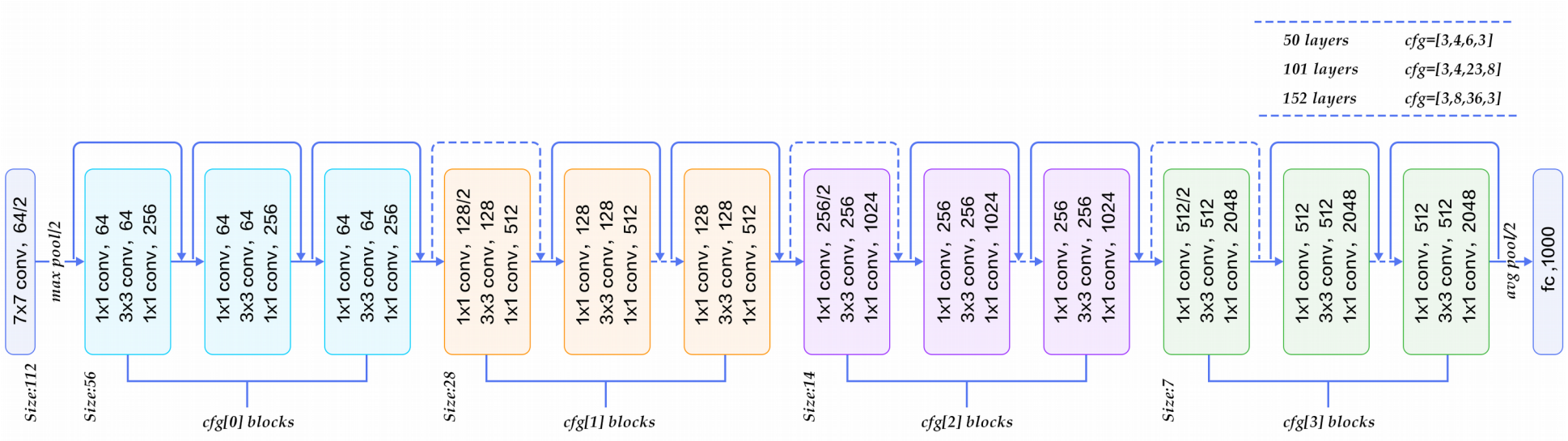
ImageNet



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

ResNet

The Residual Network

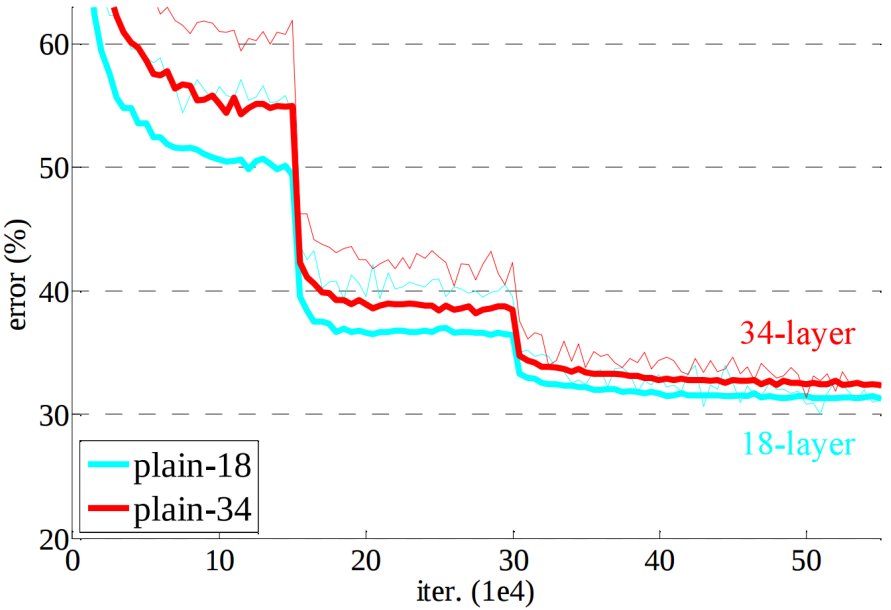


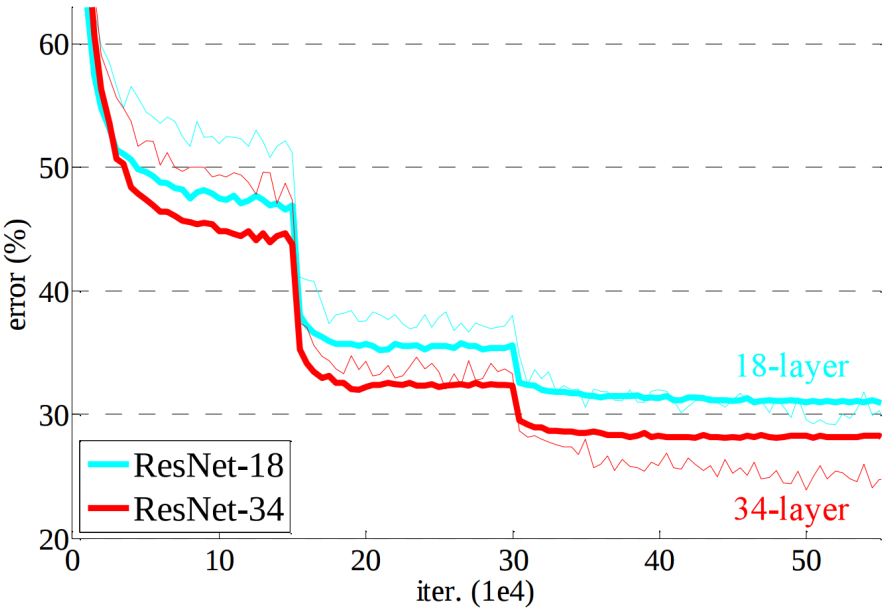
Kaiming He, Microsoft Research Asia (now at Facebook with Yann LeCun)
https://arxiv.org/abs/1512.03385

https://github.com/mas-dse-greina/neon/blob/master/luna16/LUNA16_resnet_HDF5.py

ResNet

The Residual Network

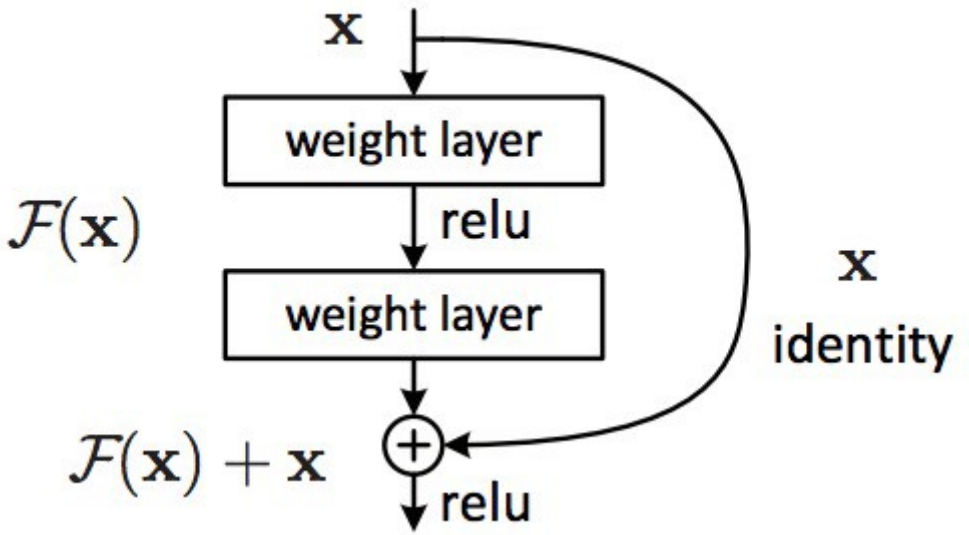




https://github.com/mas-dse-greina/neon/blob/master/luna16/LUNA16_resnet_HDF5.py

ResNet

The Residual Network



Let the network "decide" how deep it needs to be.

https://github.com/mas-dse-greina/neon/blob/master/luna16/LUNA16_resnet_HDF5.py

For a much better lecture...

https://www.youtube.com/watch?v=LxfUGhug-iQ&list=PLkt2uSq6rBVctENoVBg1TpCC7OQi31AlC&index=7

Andrej Karpathy's blog:
http://karpathy.github.io/