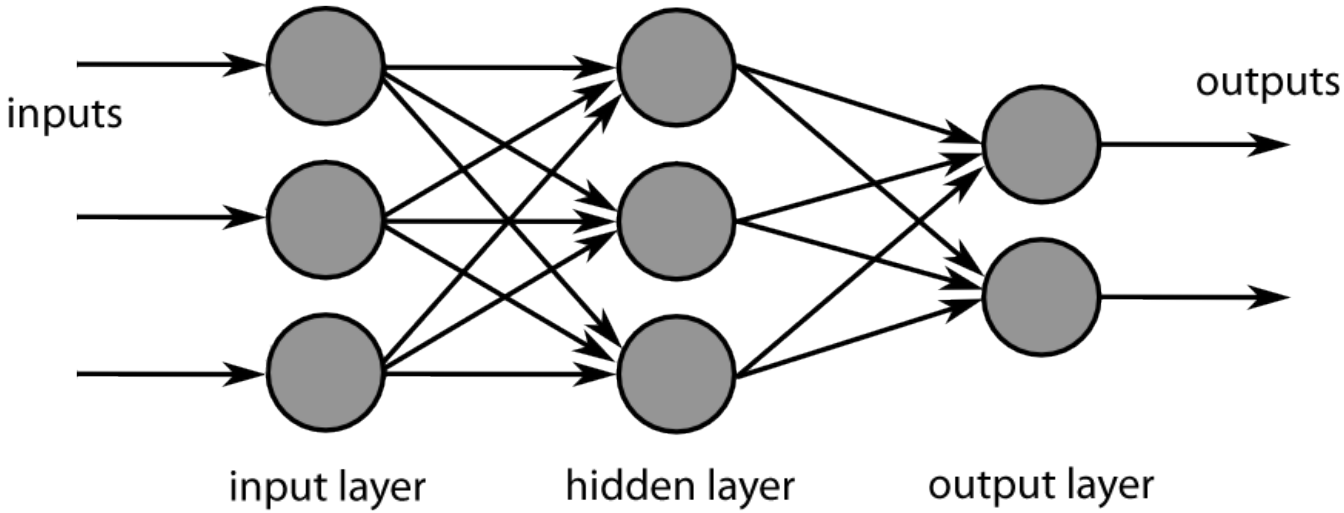
Convolutional Neural Networks.

Alvaro Soto

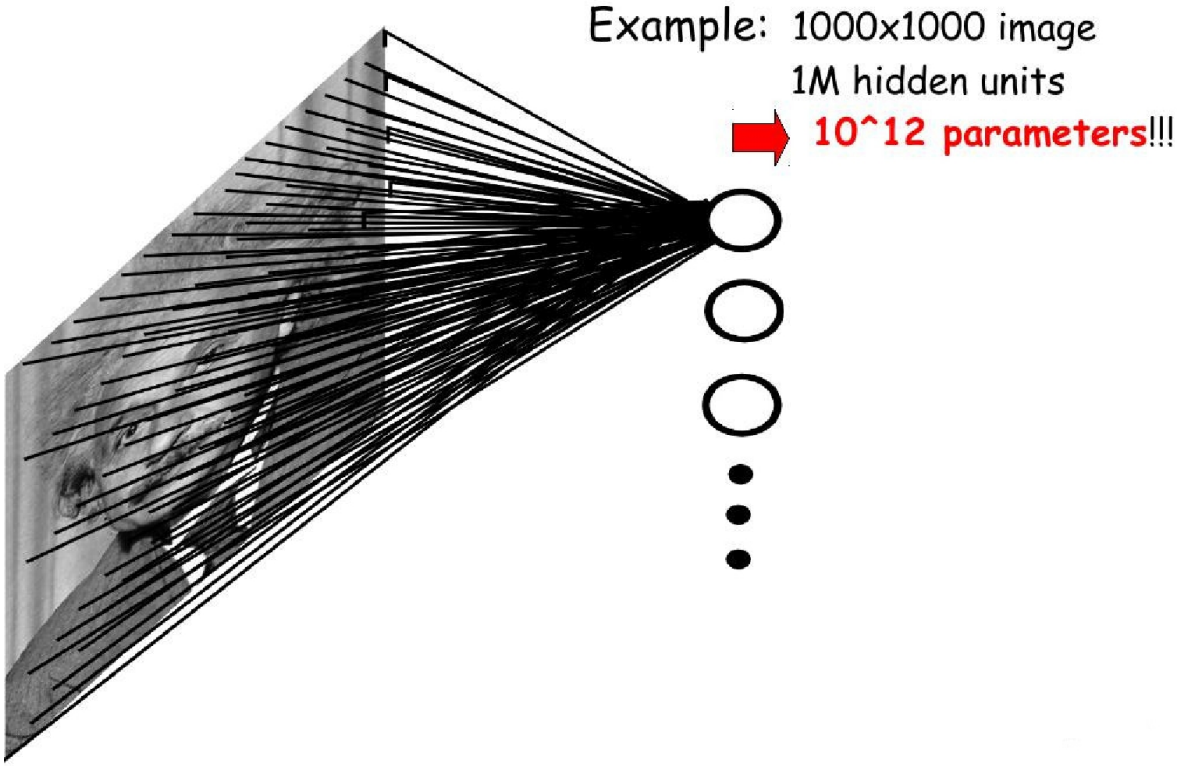
Computer Science Department, PUC

Supervised Neural Nets

Traditional approach:
Fully connected feed-forward architecture.

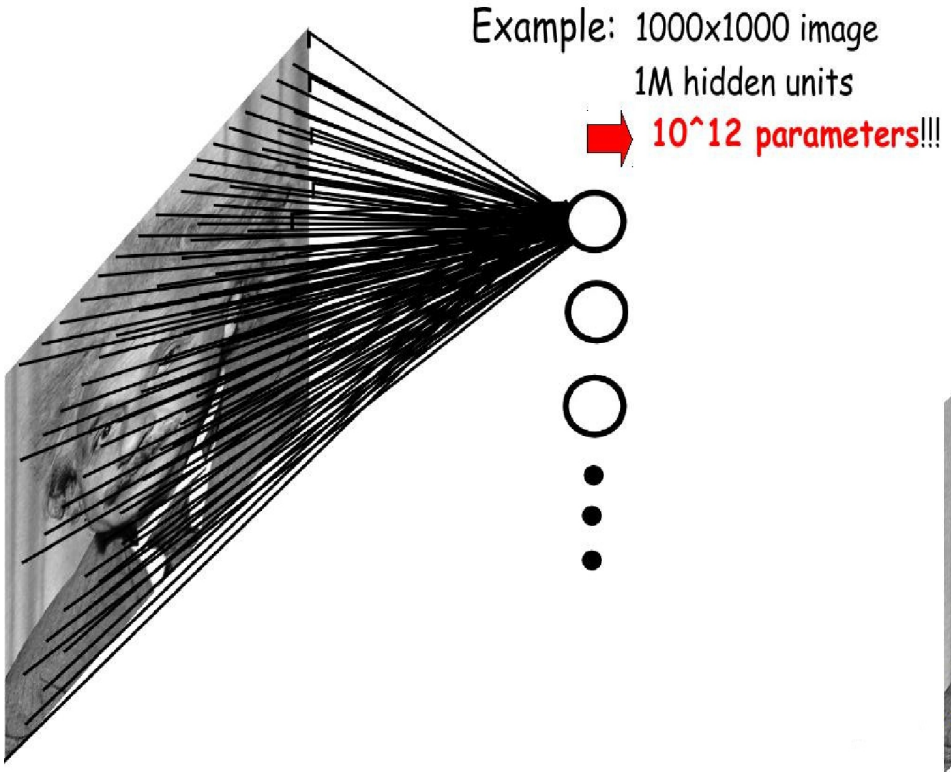


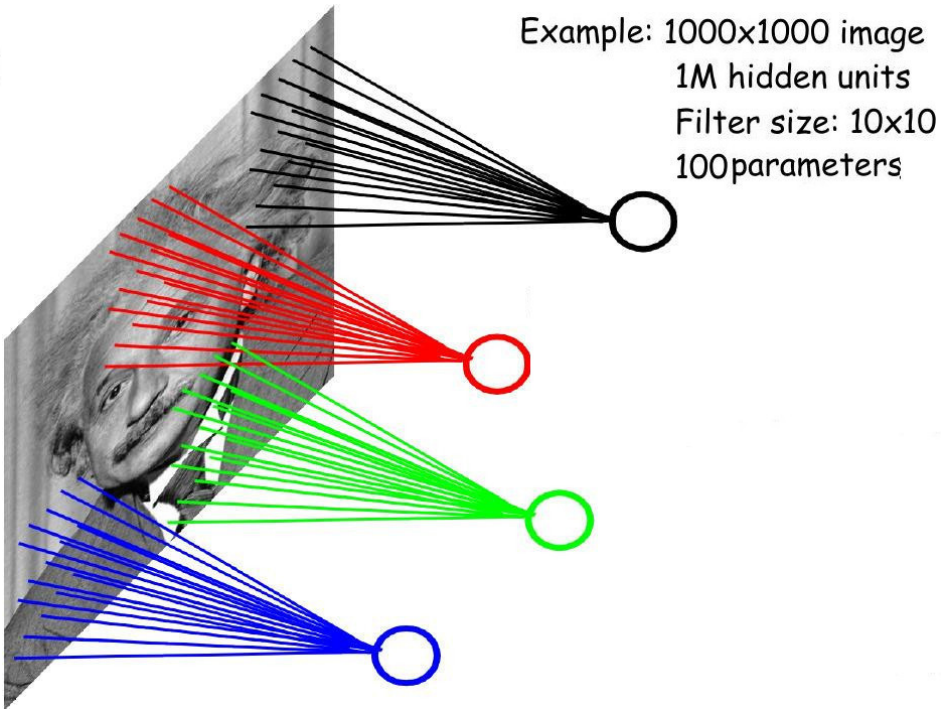
Supervised Neural Nets: Problems



- Huge number of parameters.
- It does not explicitly model local visual pattern (holistic approach).

Solution: Spatially Localized Neurons





- Spatially localized neurons provide a significant reduction in number of params. In the example, weights associated to each neuron drop from 10^6 to 10^2.
- Furthermore, these neurons allow the net to capture local image patterns.

Problem

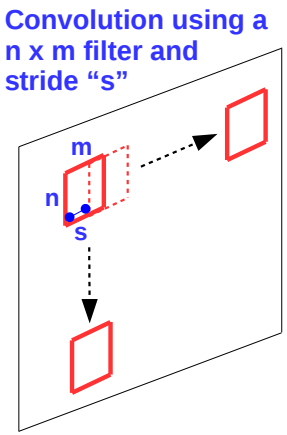
- A local pattern can potentially occur anywhere in the input, any ideas?

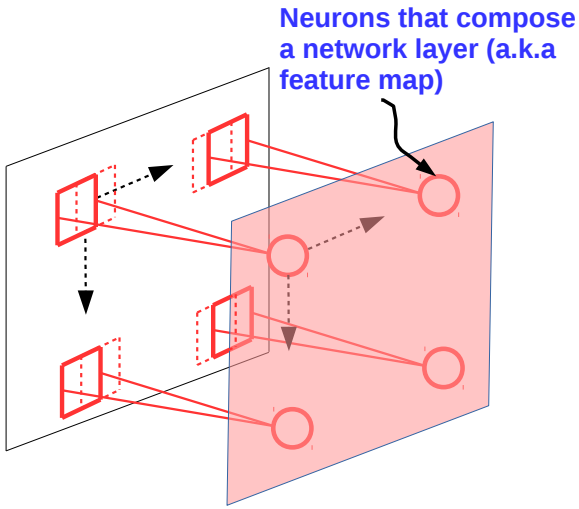
Problem

- A local pattern can potentially occur anywhere in the input, any ideas?

Solution

- Convolution provides a mechanism to search for a given pattern (feature) by sliding a set of weights (or filter) through the spatial span of the input.



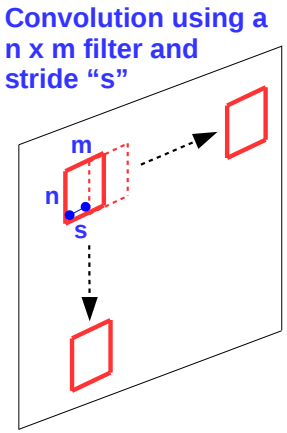


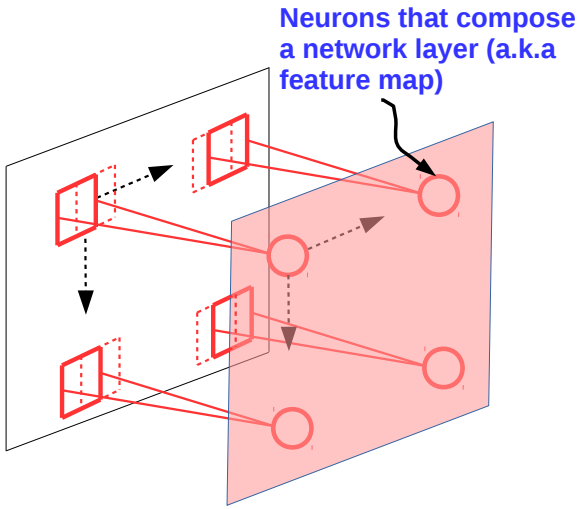
Problem

- A local pattern can potentially occur anywhere in the input, any ideas?

Solution

- Convolution provides a mechanism to search for a given pattern (feature) by sliding a set of weights (or filter) through the spatial span of the input.
- At each position, the application of the filter over the input is associated to the activation of a specific neuron. So, neurons are spatially localized.



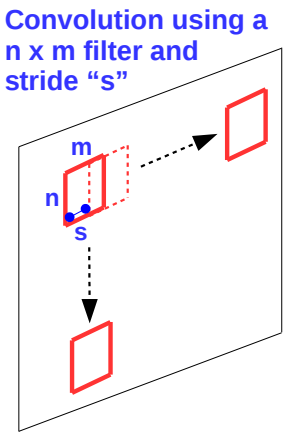


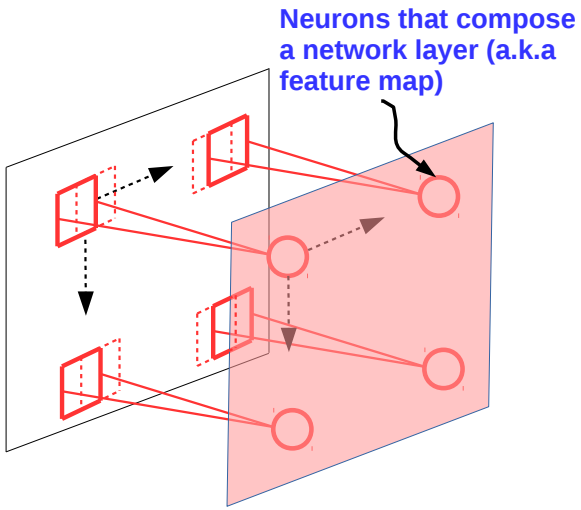
Problem

- A local pattern can potentially occur anywhere in the input, any ideas?

Solution

- Convolution provides a mechanism to search for a given pattern (feature) by sliding a set of weights (or filter) through the spatial span of the input.
- At each position, the application of the filter over the input is associated to the activation of a specific neuron. So, neurons are spatially localized.
- The set of all neurons associatiated to a given filter is known as a feature map.



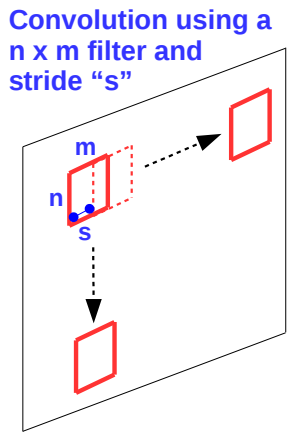


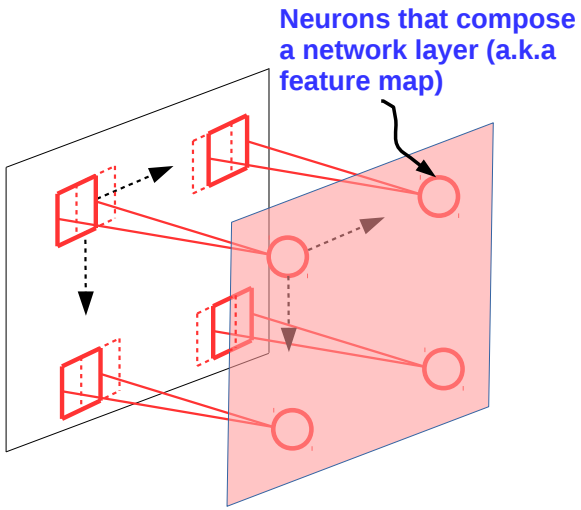
Problem

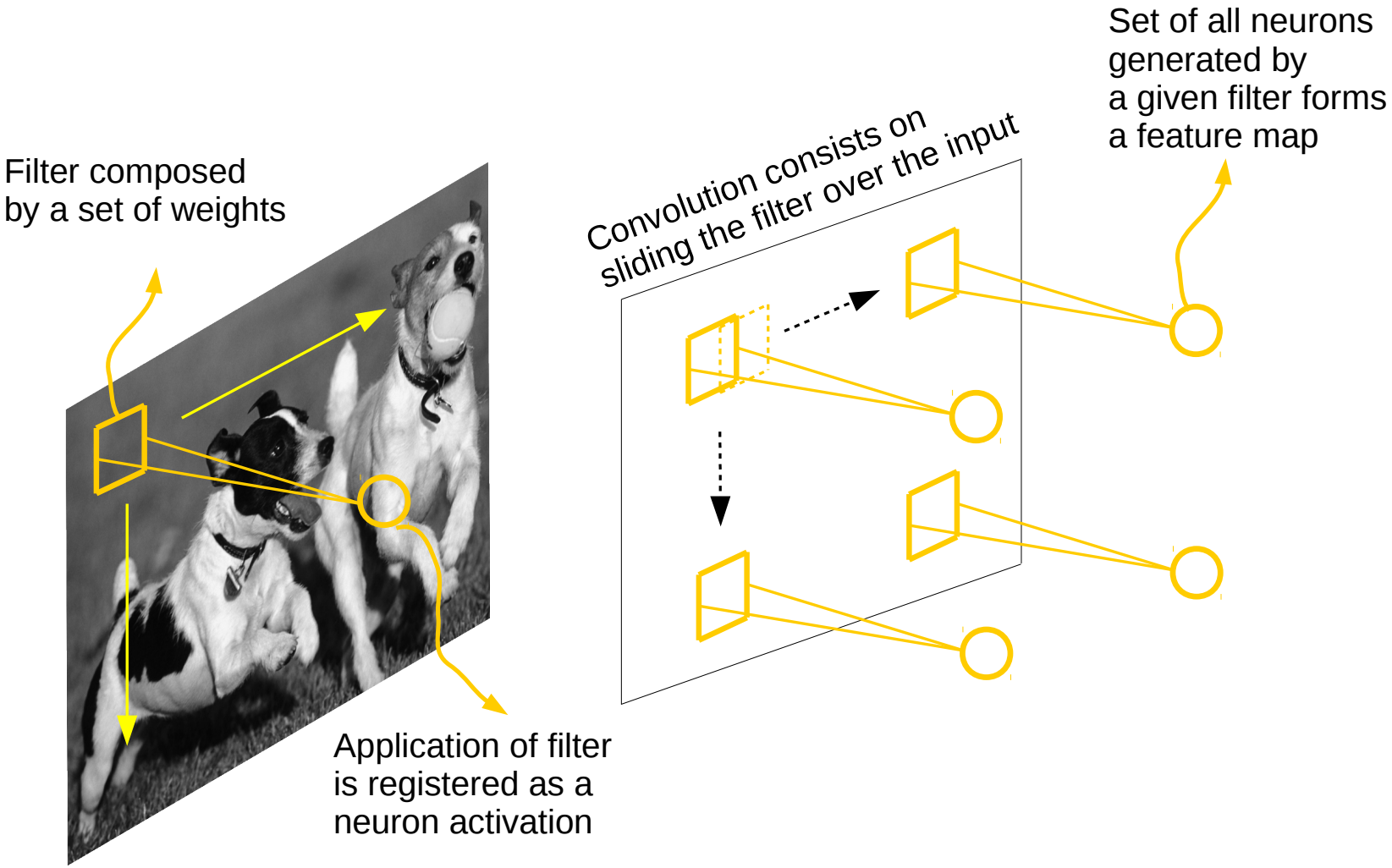
- A local pattern can potentially occur anywhere in the input, any ideas?

Solution

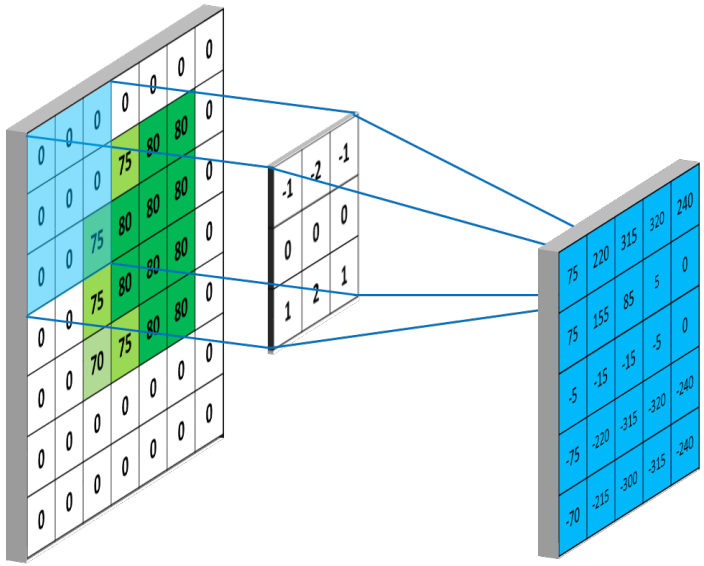
- Convolution provides a mechanism to search for a given pattern (feature) by sliding a set of weights (or filter) through the spatial span of the input.
- At each position, the application of the filter over the input is associated to the activation of a specific neuron. So, neurons are spatially localized.
- The set of all neurons associatiated to a given filter is known as a feature map.
- Therefore, the resulting feature map consists of the activation of a set of localized neurons that share a set of weights or filter.





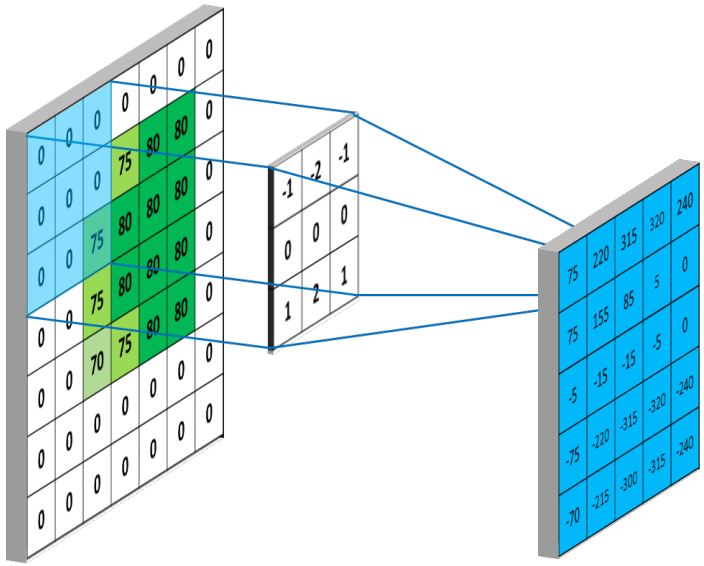

Filter composed
by a set of weights
Application of filter
is registered as a
neuron activation
Convolution consists on
sliding the filter over the input
Set of all neurons
generated by
a given filter forms
a feature map

Neuron sharing weights?



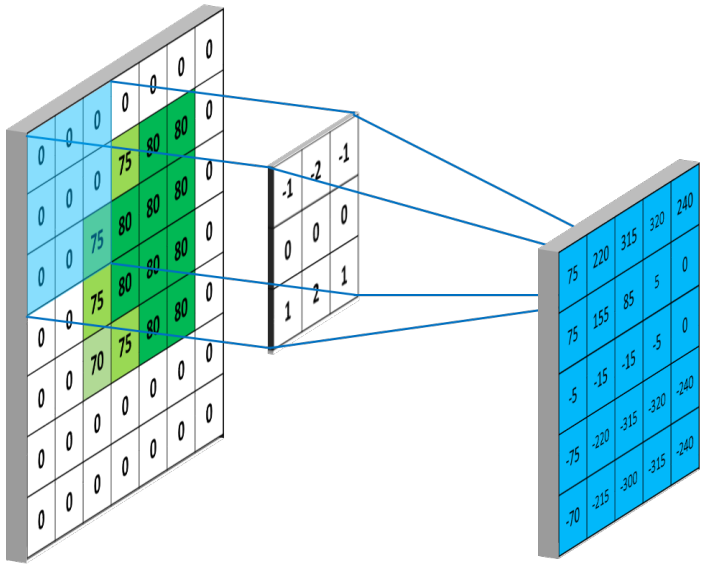
- What is the input? How many dimensions (features) in the input?

Neuron sharing weights?



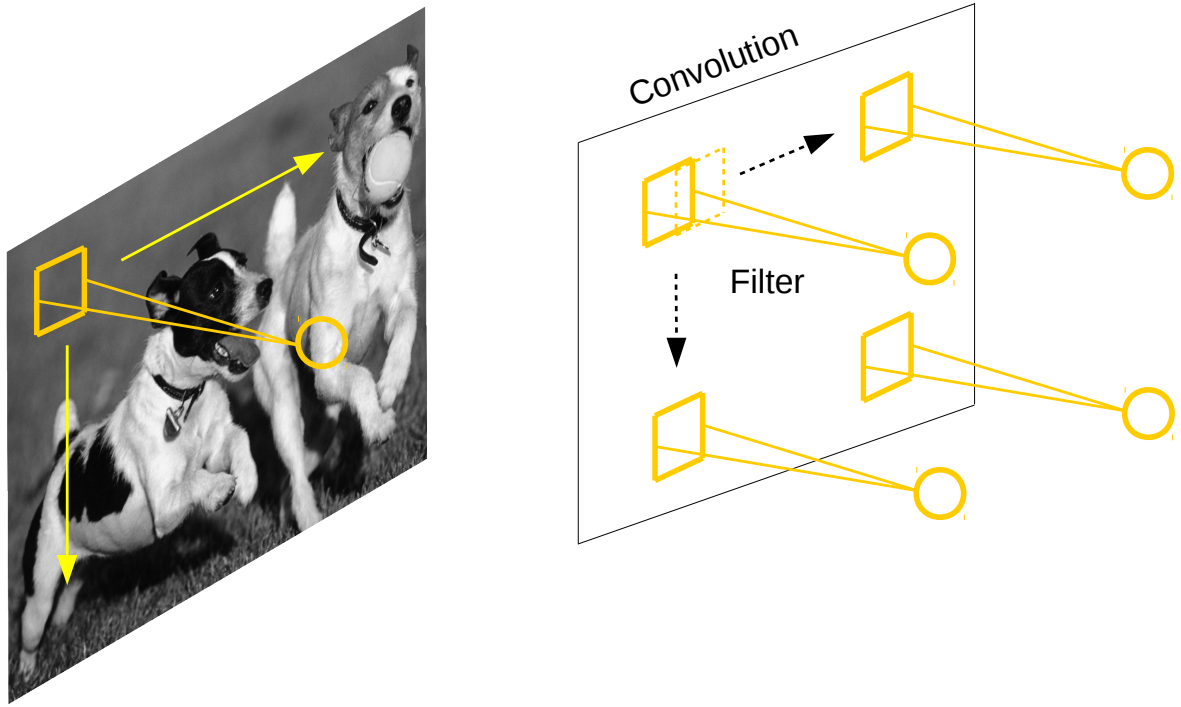
- What is the input? How many dimensions (features) in the input?
- What is the filter? How many weights in the filter?

Neuron sharing weights?



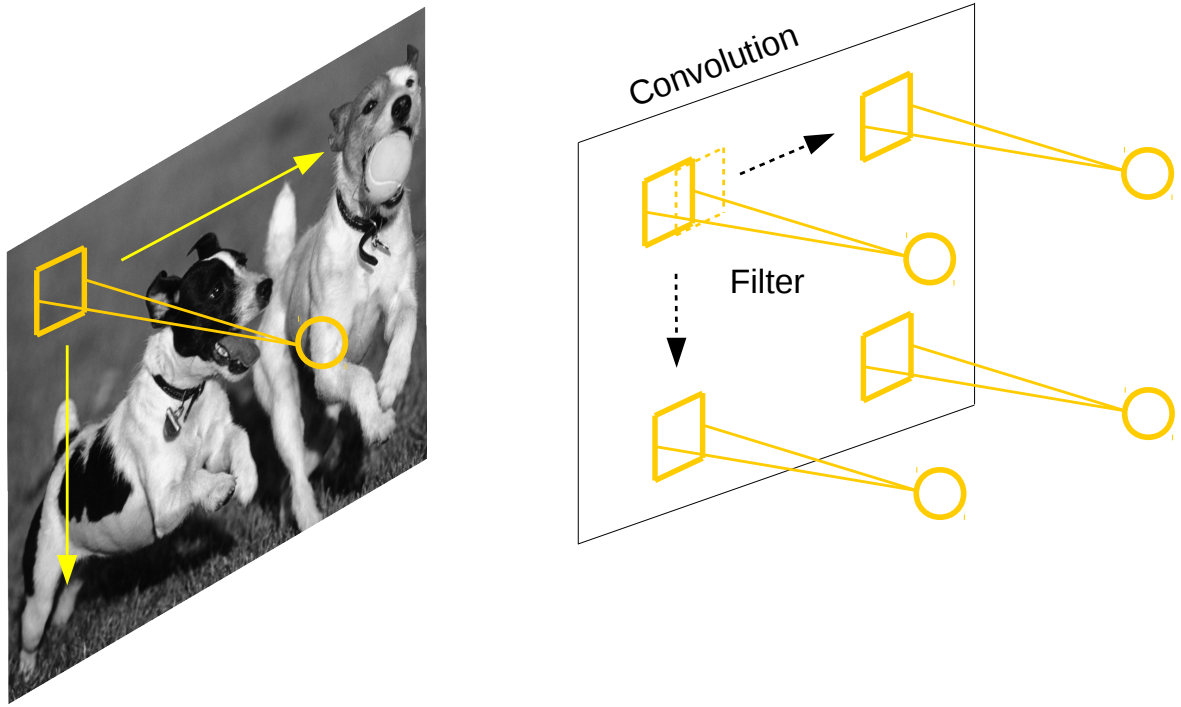
- What is the input? How many dimensions (features) in the input?
- What is the filter? How many weights in the filter?
- What is the feature map?, How many neurons in the feature map?

Convolutional Neural Networks (CNNs)



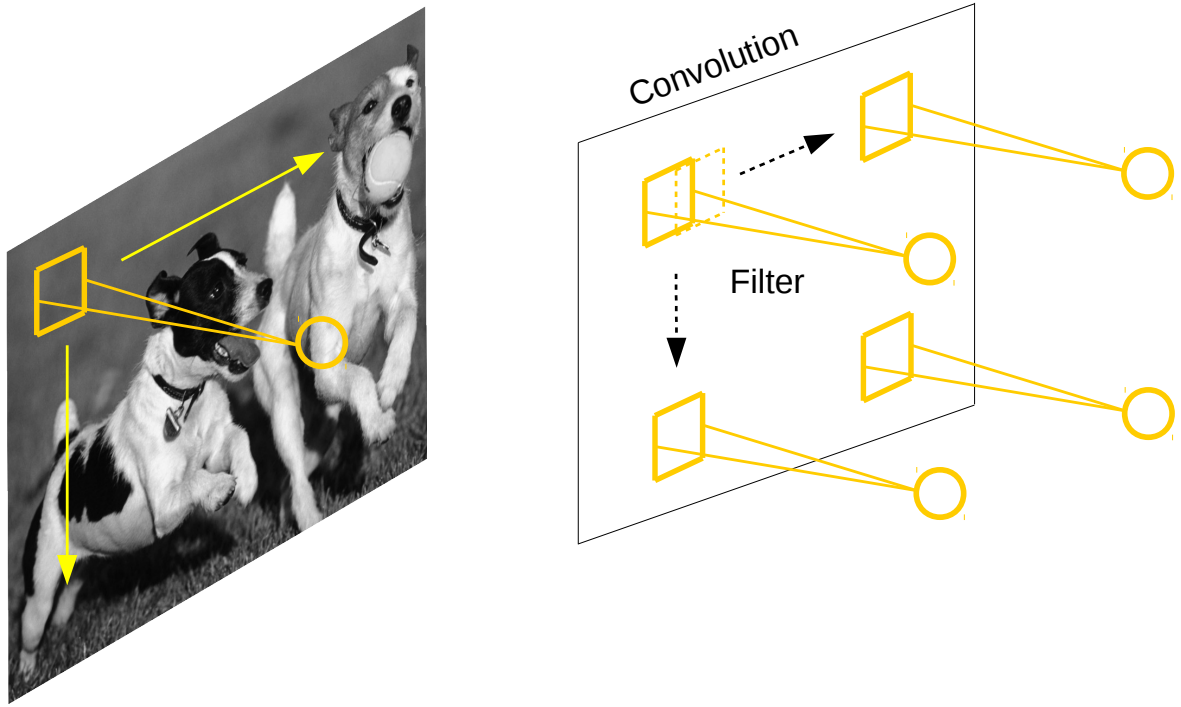
- During training the filter learns to detect a particular pattern (learn?).

Convolutional Neural Networks (CNNs)

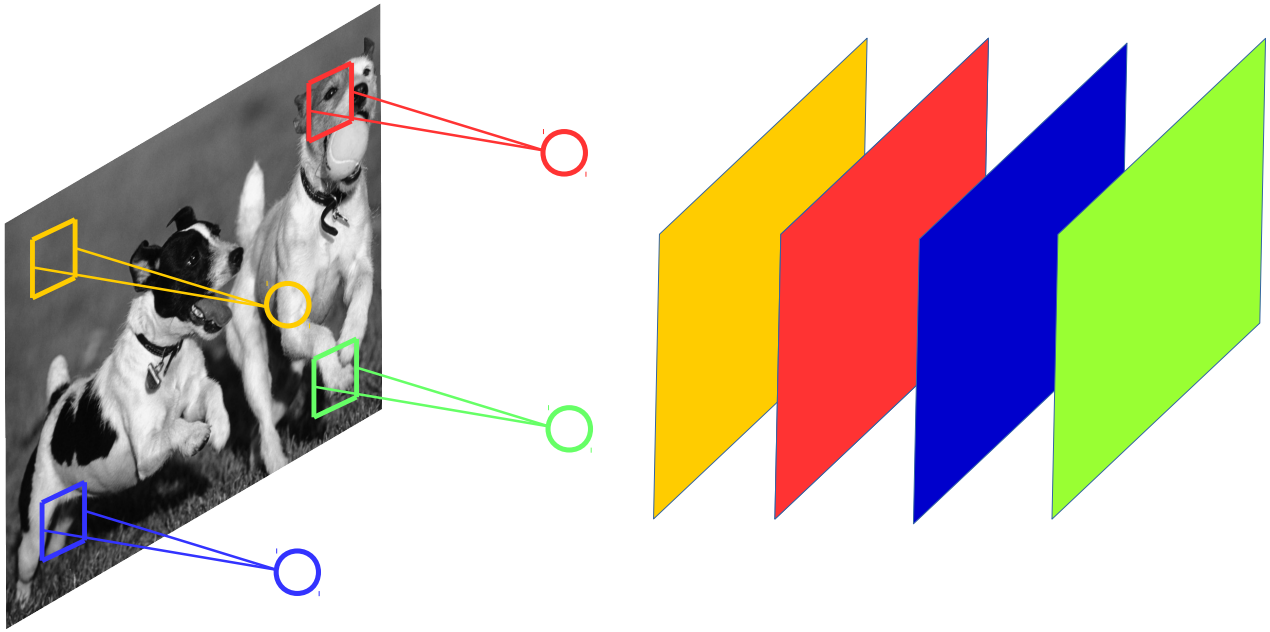


- During training the filter learns to detect a particular pattern (learn?).
- Convolution with each filter generates a particular feature map.

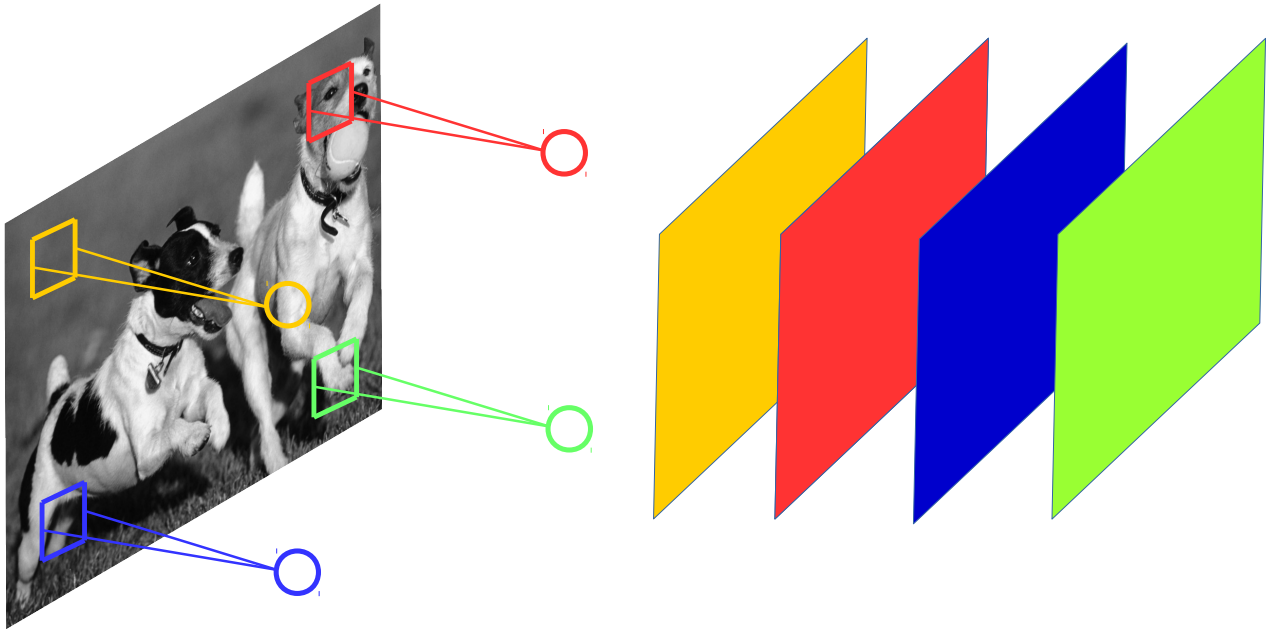
Convolutional Neural Networks (CNNs)



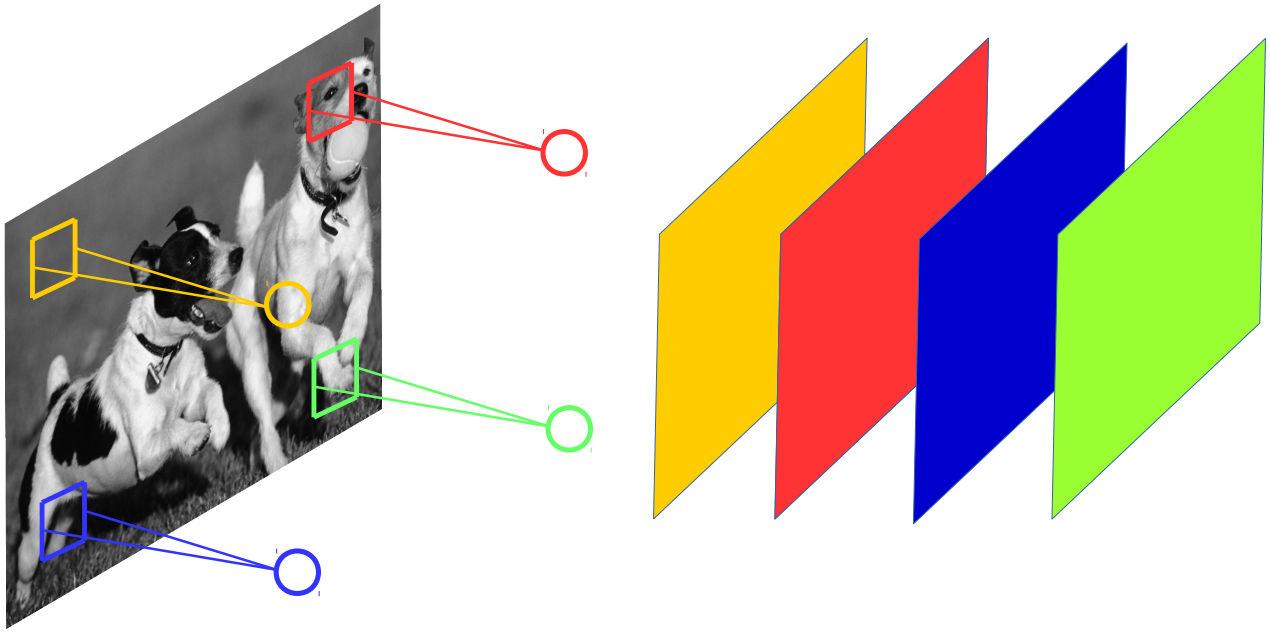
- During training the filter learns to detect a particular pattern (learn?).
- Convolution with each filter generates a particular feature map.
- What type of information contains this feature map?



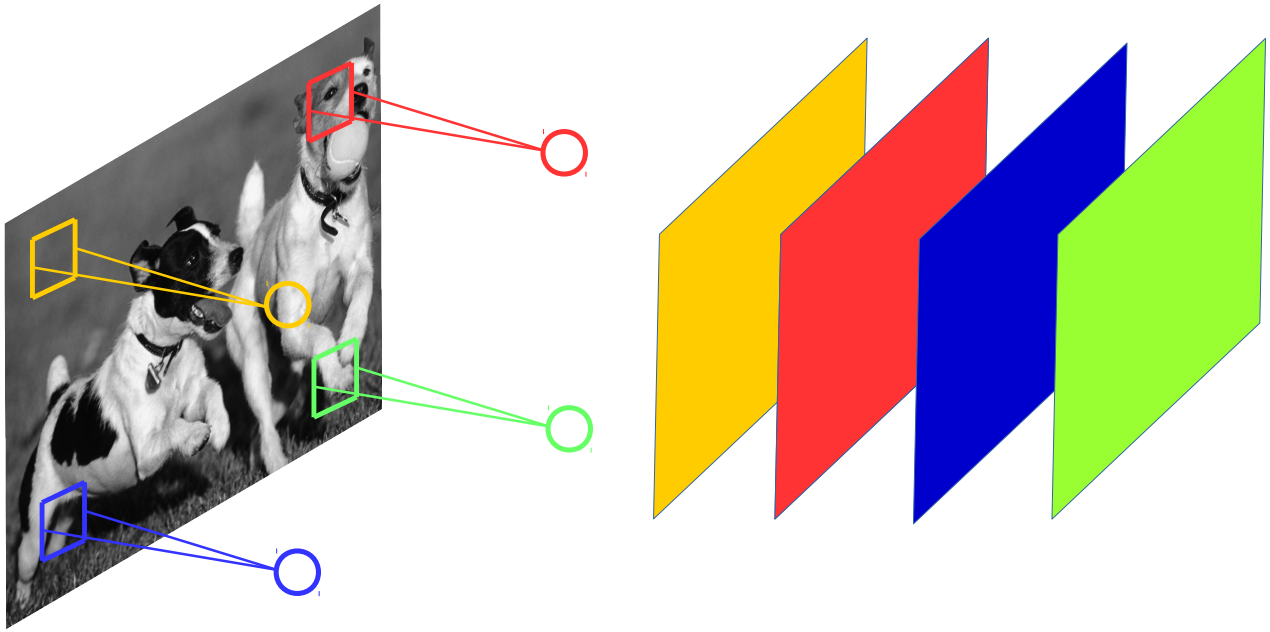
- We can use (learn) several filters.



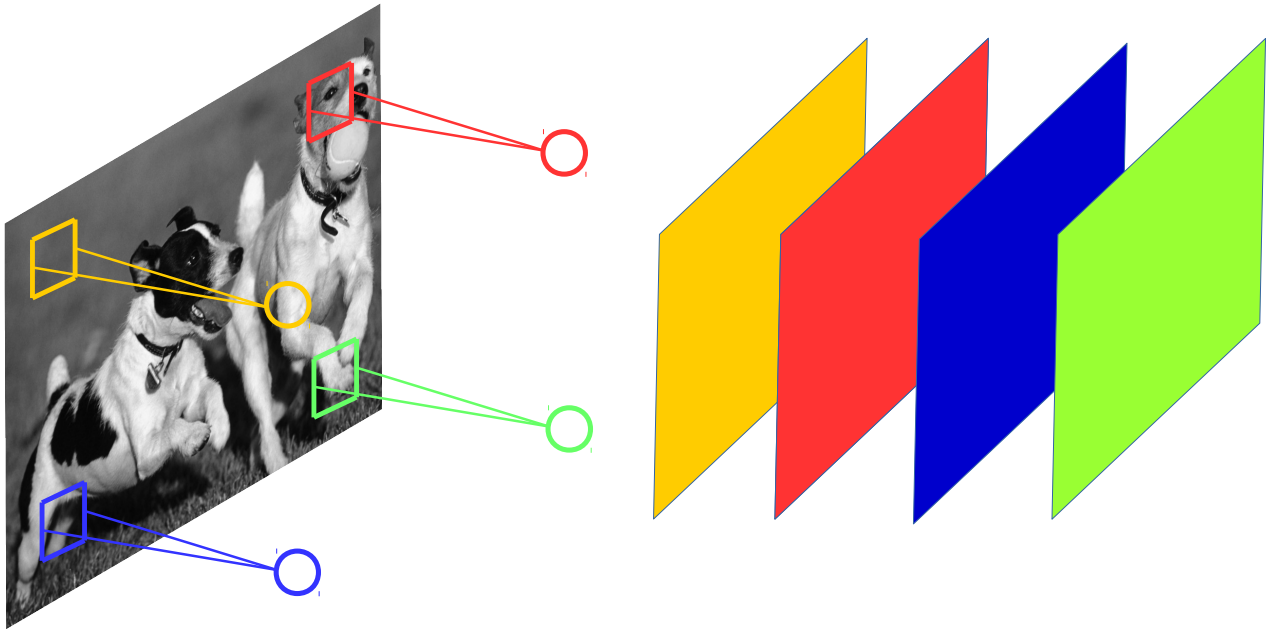
- We can use (learn) several filters.
- The convolution process generates an activation map for each filter, which is given by a set of neurons that share weights.

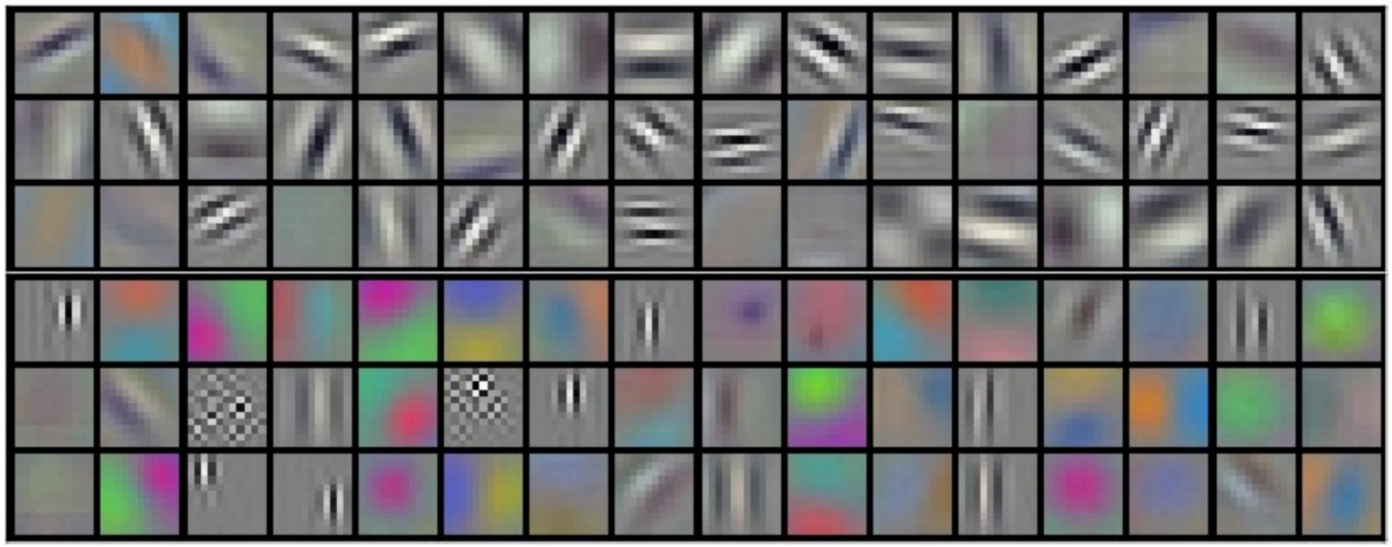


- We can use (learn) several filters.
- The convolution process generates an activation map for each filter, which is given by a set of neurons that share weights.
- By sharing weights, the model presents an improved mechanism to deal with the spatial variability of local patterns (how?).



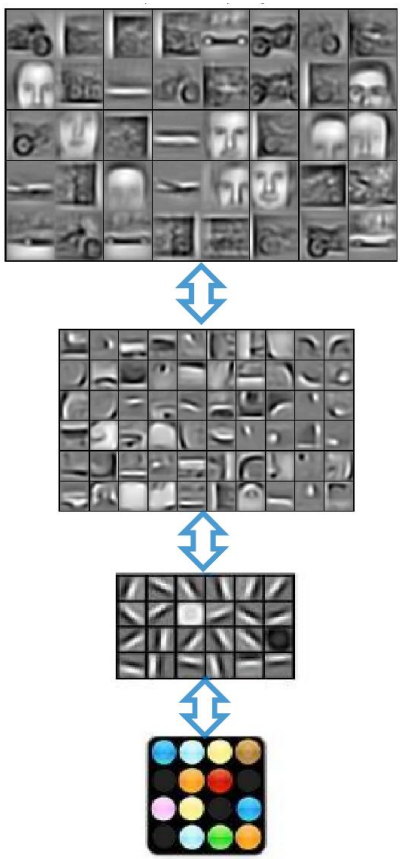
- We can use (learn) several filters.
- The convolution process generates an activation map for each filter, which is given by a set of neurons that share weights.
- By sharing weights, the model presents an improved mechanism to deal with the spatial variability of local patterns (how?).
- Each set of shared weights allows the net to learn a filter specifically tuned to detect a particular local pattern (filter banks).





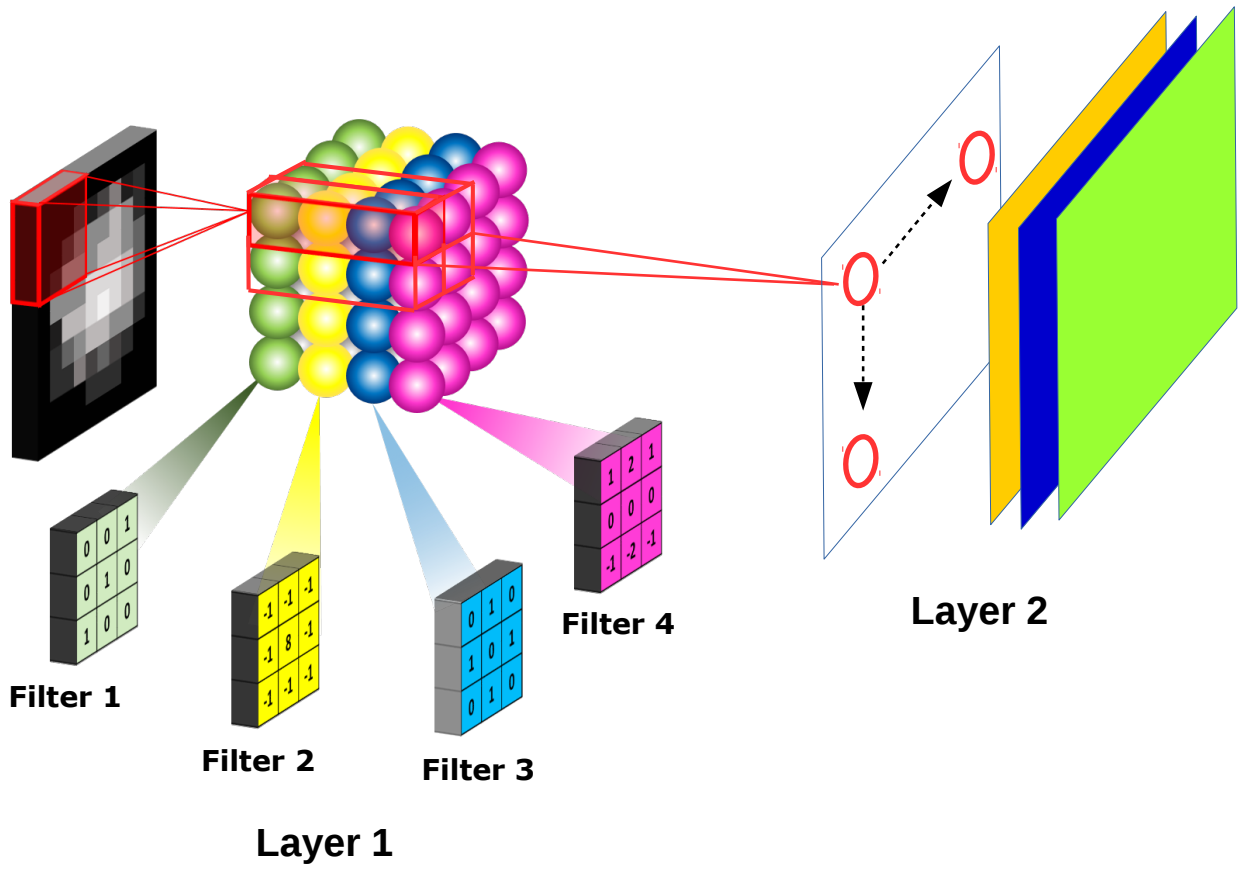
Let's now go deep: A hierarchical compositional architecture.

Hierarchical Compositional Architecture

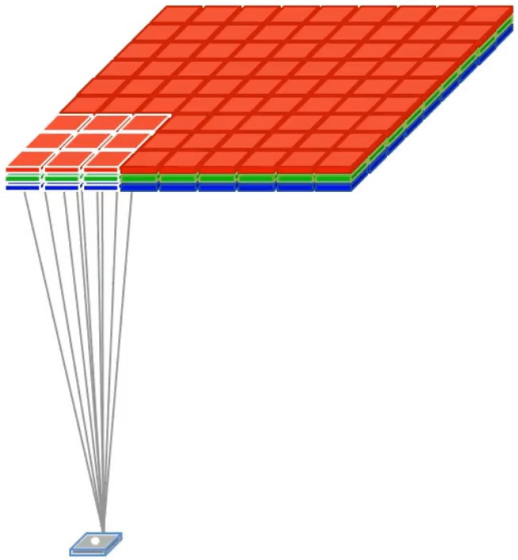


- Deep neural networks exploit the property that many natural signals form compositional hierarchies.
- Higher-level features are obtained by compositions of lower-levels.

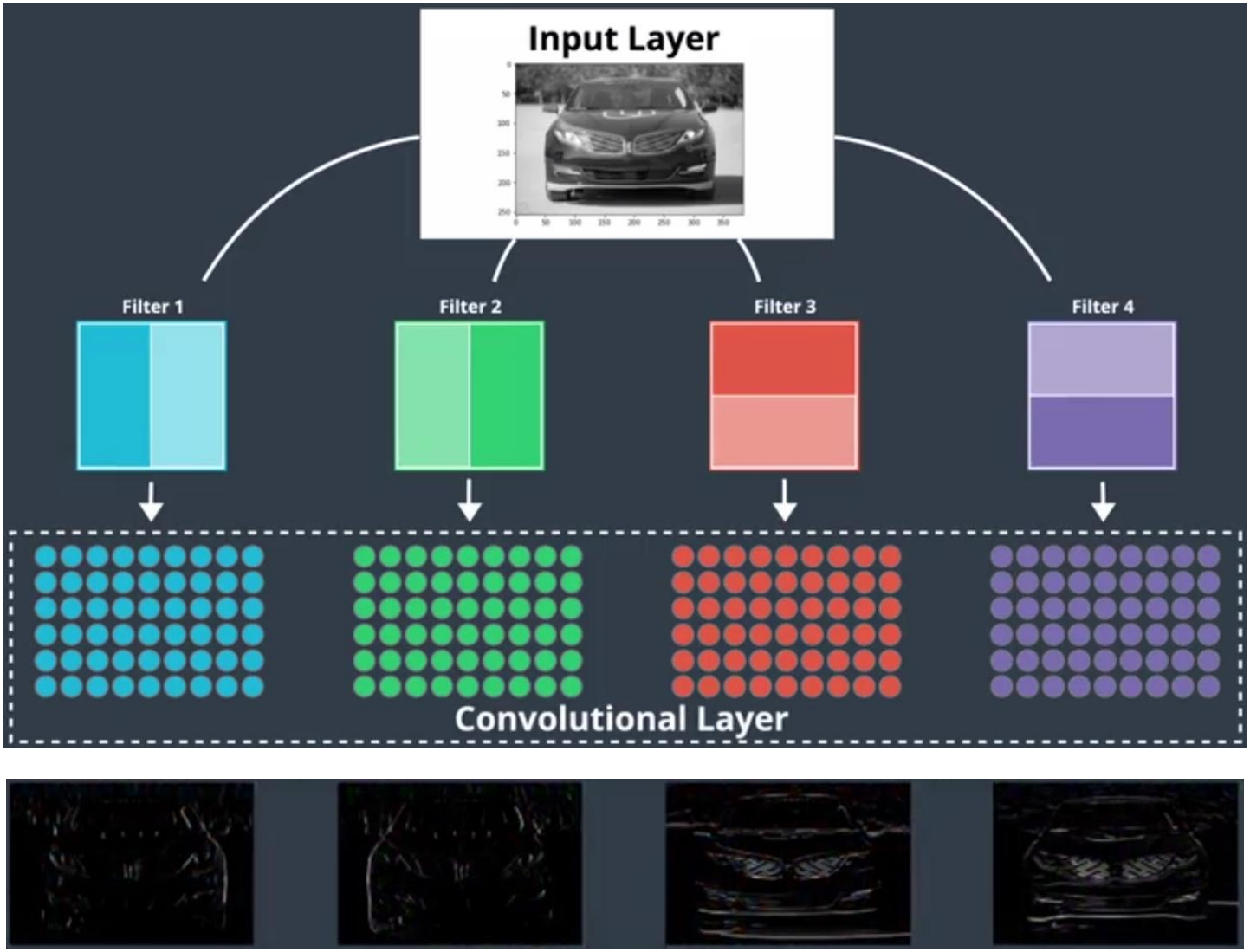
A Hierarchy of Compositional Network Layers



A Hierarchy of Compositional Network Layers

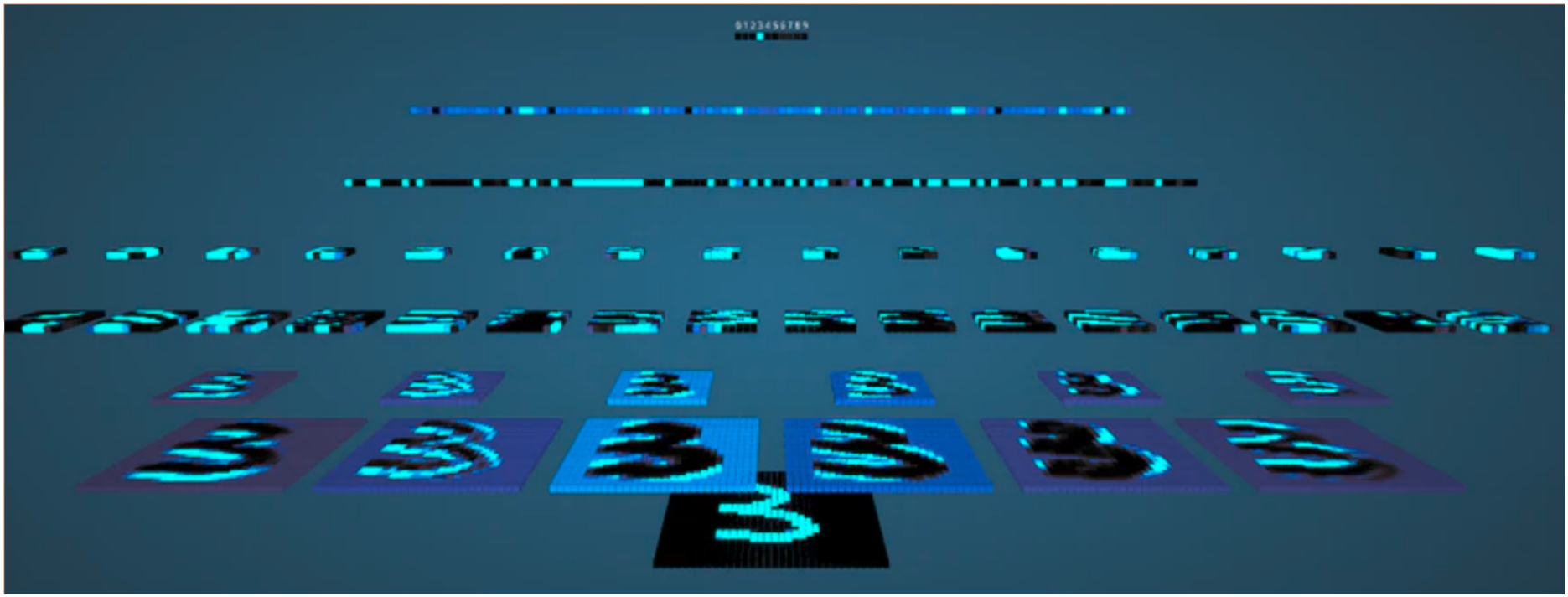


Example: Convolving Filters Over an Input Image [1]



[1] Image from audacity.com

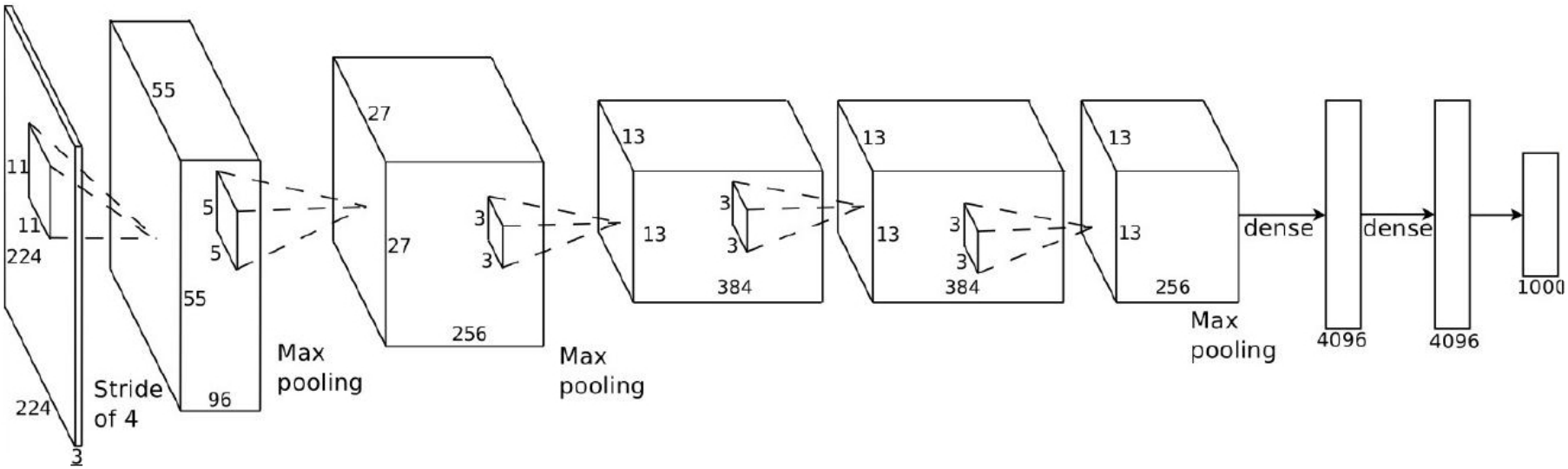
Example: Convolving Filters Over an Input Image



- Adam Harley's CNN visualization tool.
- `www.cs.cmu.edu/~aharley/vis/conv/flat.html`.

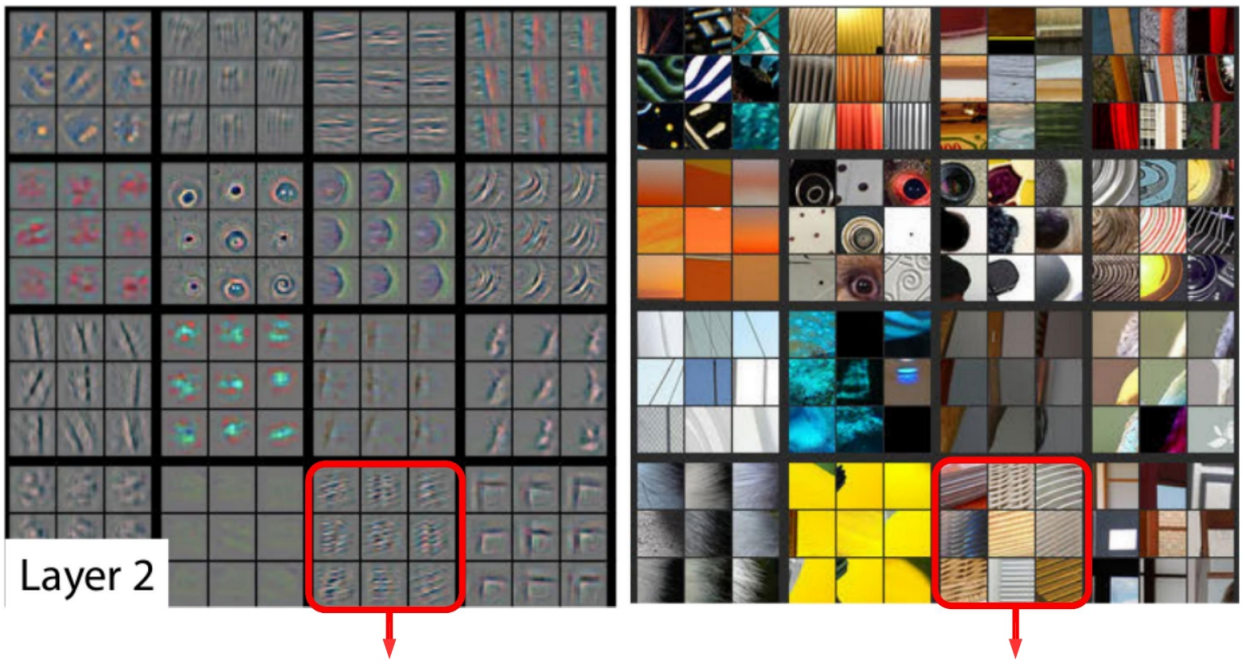
Going Deeper: Stack Several Layers of CNNs

Example: Meet Alex's Net (Krizhevsky et al., 2012)

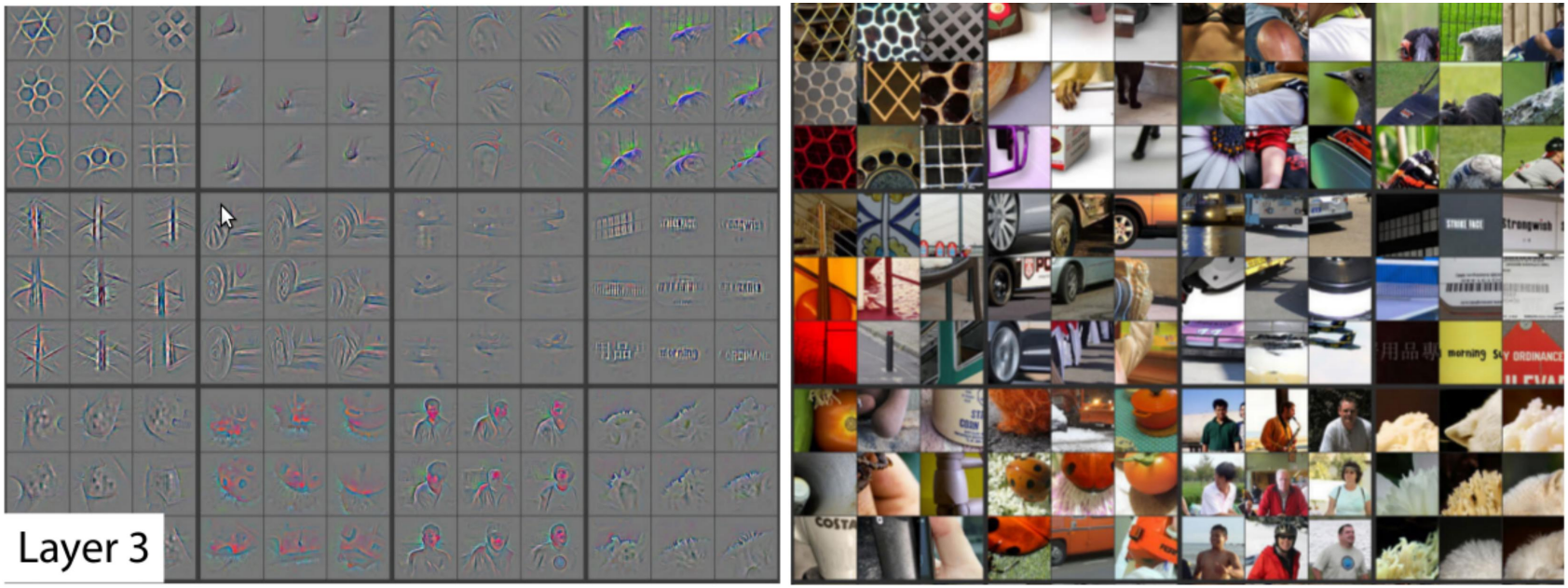


- The role of each convolutional layer is to detect local conjunctions of features from the previous layer.

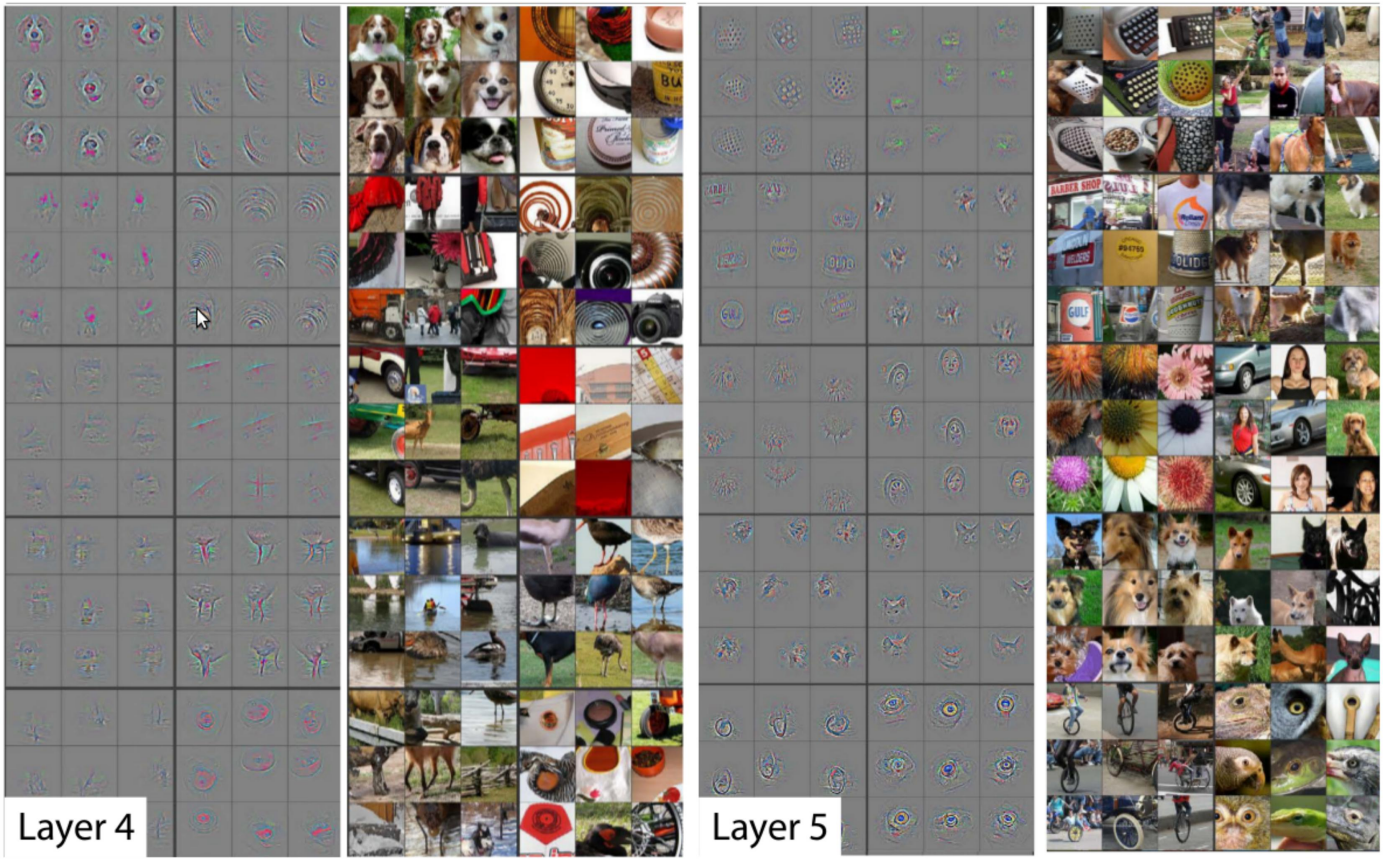
Ex. Visualization of Neuron Activations



Ex. Visualization of Neuron Activations

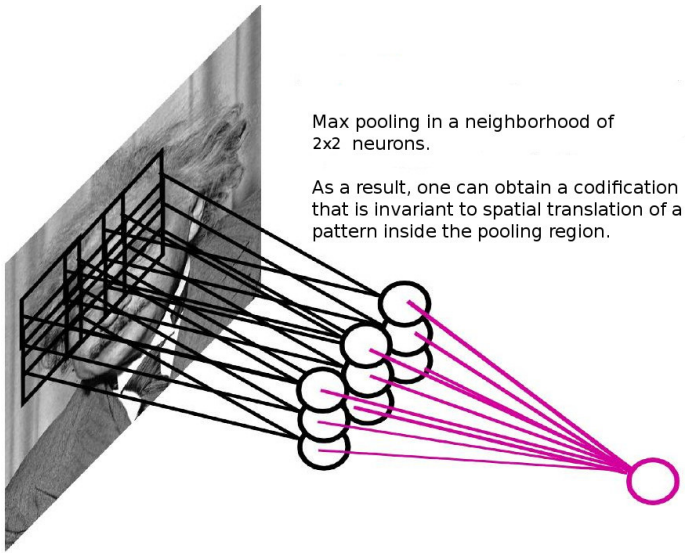


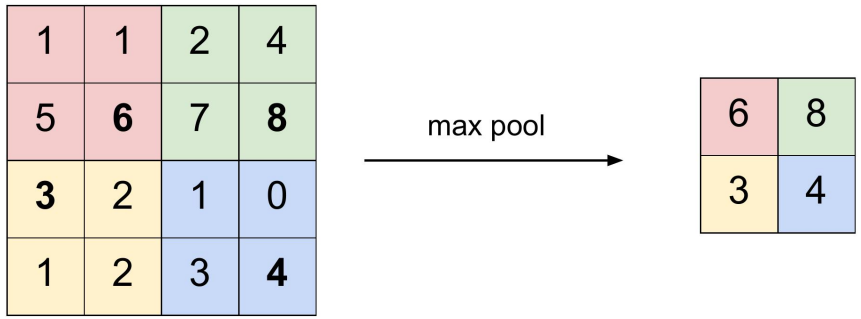
Ex. Visualization of Neuron Activations



Let's see some extra operators

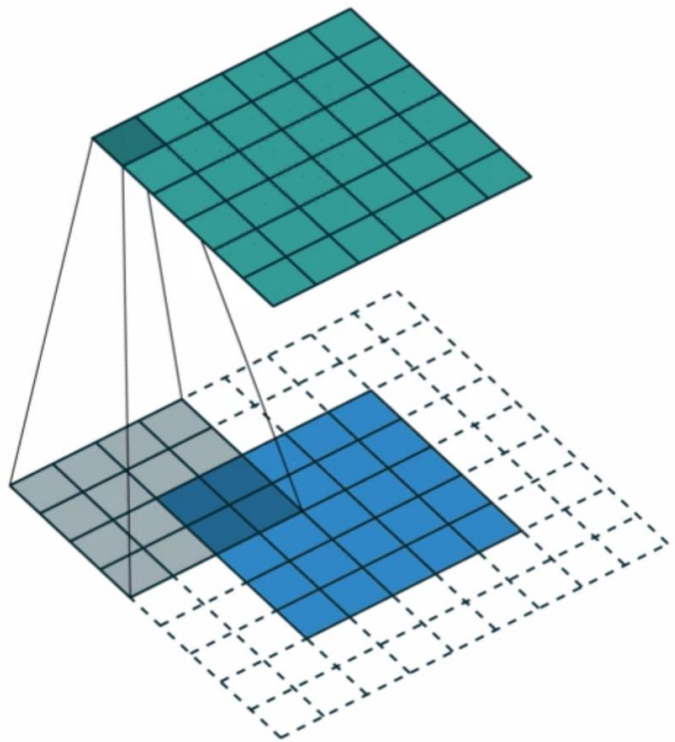
Max Pooling Operation to Avoid Spatial Variability





- Max-pooling provides a mechanism to coarse-graining the spatial position of each feature for the next layers.
- These allows the net to incrementally reduce the dimensionality of the representation and to provide certain degree of invariance to small shifts and distortions of a given pattern.
- In other words, the pooling allows representations to vary very little when elements in the previous layer present some variations in local position and appearance.

Zero Padding Operator



- By padding with zeros the borders of the images, it is possible to apply the convolution operator in these areas.

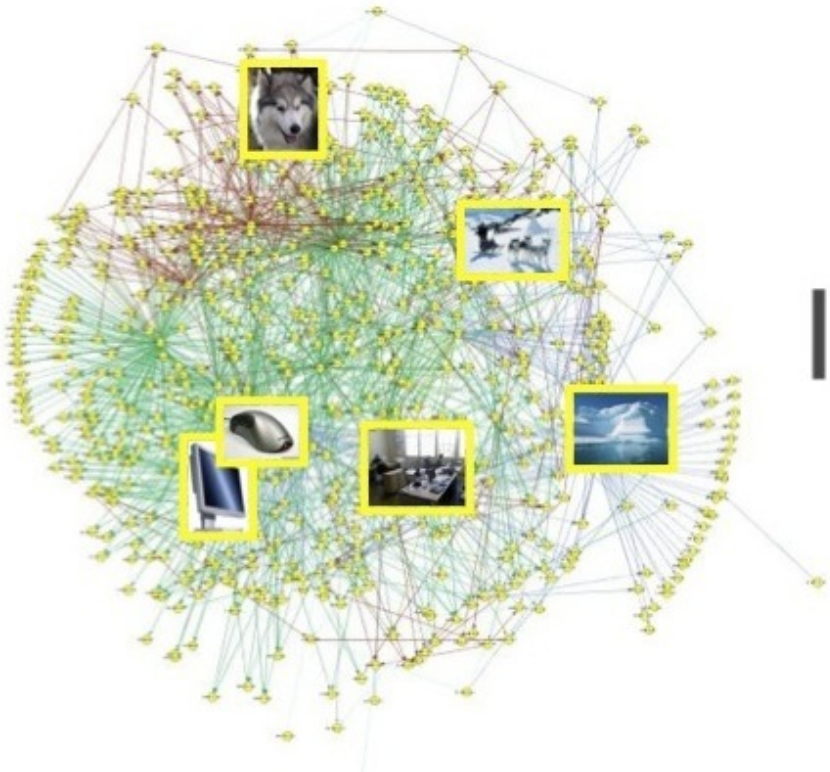
Case Example

ImageNet Classification with Deep Convolutional Neural Networks

A. Krizhevsky, I. Sutskever, and G. E. Hinton (NIPS-2012)

ImageNet

- 15M images.
- 22K categories.
- Images collected from Web.
- Human labelers (Amazon's Mechanical Turk crowd-sourcing).



IMAGENET

ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2010)

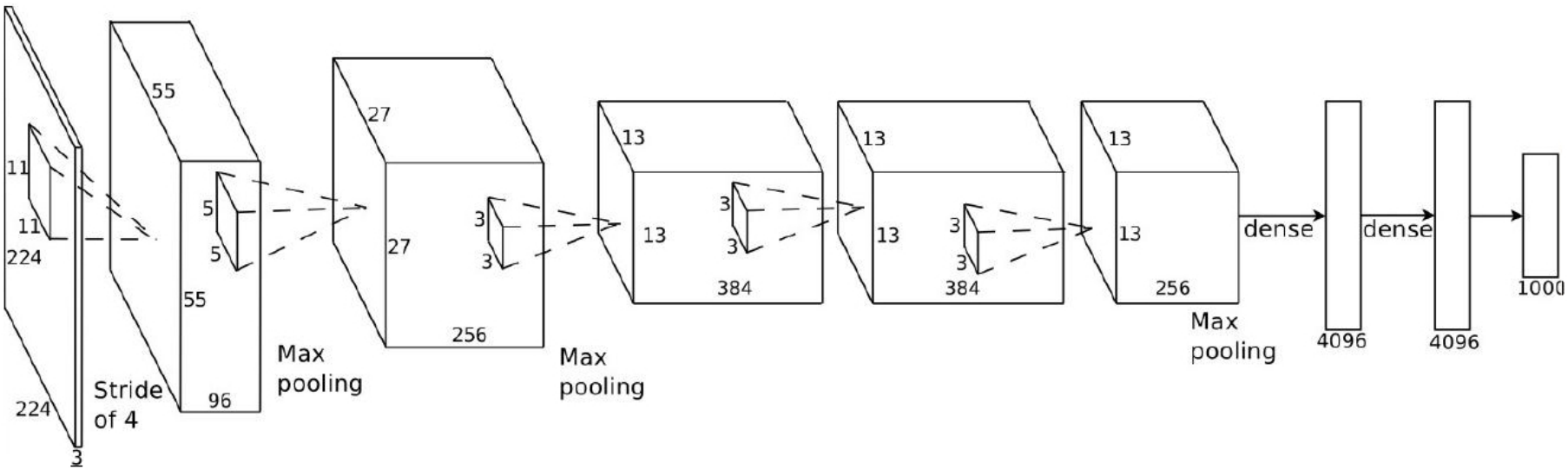
ILSVRC-2010

- 1K categories.
- 1.2M training images (1000 per category).
- 50K validation images.
- 150K testing images.
- RGB images.
- Variable-resolution, but in this application images are scaled to 256x256 pixels.

Classification goals

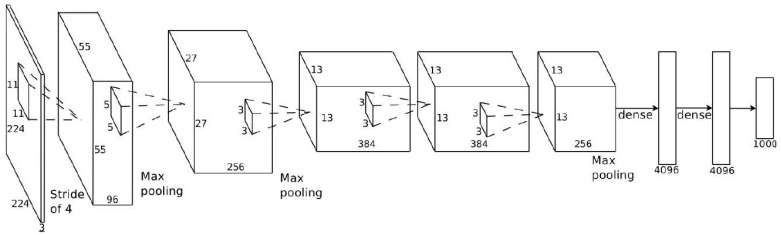
- Make 1 guess about the label (Top-1 error).
- make 5 guesses about the label (Top-5 error).

AlexNet Arquitecture:



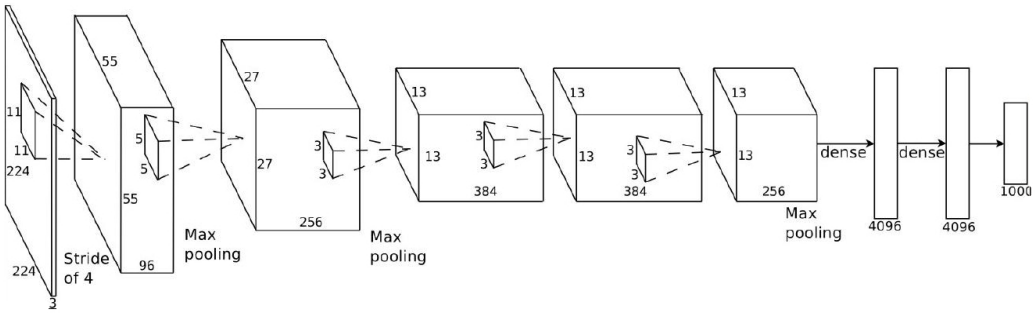
Arquitecture: Convolutional Layers

Input: 224x224x3= 150528 pixels



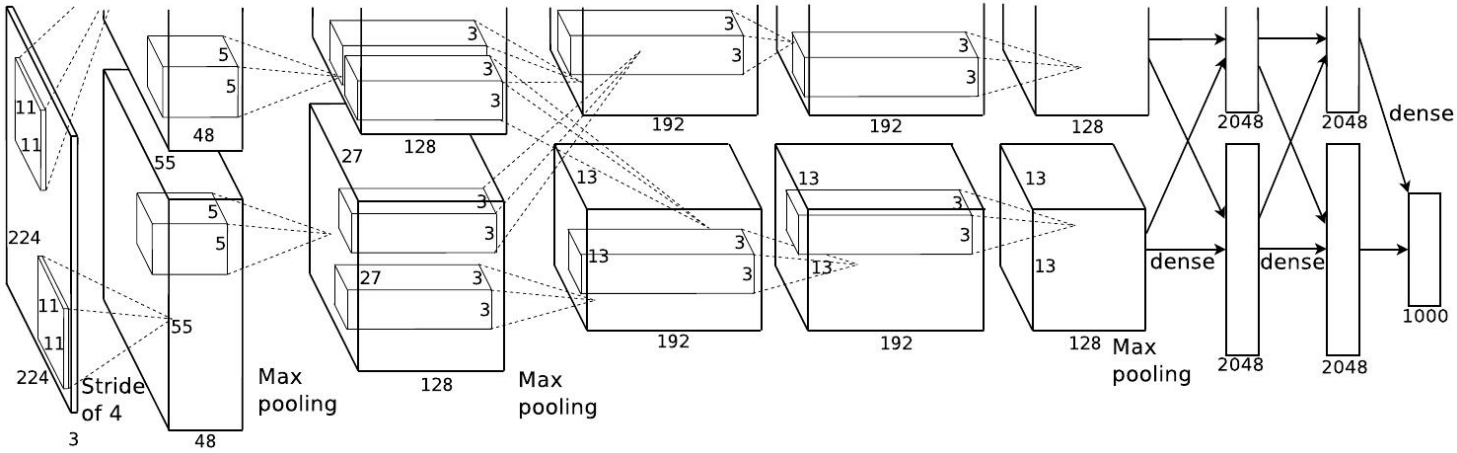
- Layer 1:
  - nFilters = 96, Size = 11*11, Depth = 3.
  - nParams (weights) = 96*11*11*3 = 34848.
  - Stride = 4, max-pooling = 3x3:1, nNeurons (outputs) = 55*55*96 = 290400.
    (obs: they use 3 zero-padding, then (227 - 11)/4 + 1 = 55)
- Layer 2:
  - nFilters=256, Size = 5*5, Depth = 96.
  - nParams = 256*5*5*96 = 614400.
  - Stride=1, max-pooling = 3x3:2, nNeurons= 27*27*256 = 186624.
- Layer 3:
  - nFilters = 384, Size = 3*3, Depth = 256.
  - nParams = 384*3*3*256 = 884736.
  - Stride=1, max-pooling = none, nNeurons = 13*13*384 = 64896.
- Layer 4:
  - nFilters= 384, Size= 3*3, Depth= 384.
  - nParams = 384*3*3*384 = 1327104.
  - Stride=1, max-pooling = none, nNeurons= 13*13*384 = 64896.
- Layer 5:
  - nFilters = 256, Size= 3*3, Depth= 384.
  - nParams = 256*3*3*384 = 884736.
  - Stride=1, max-pooling = 3x3:2, nNeurons= 13*13*256 = 43264.

Arquitecture: Fully Connected Layers



- Layer 6: Fully connected to previous layer
  - nParams = 6x6x256x4096 = 37.748.736.
  - nNeurons = 4096.
- Layer 7 : Fully connected to previous layer
  - nParams = 4096x4096=16.777.216.
  - nNeurons = 4096.
- Layer 8: Fully connected to previous layer
  - nParams = 4096x1000=4.096.000.
  - nNeurons = 1000.

Arquitecture: Trained on 2 GPUs



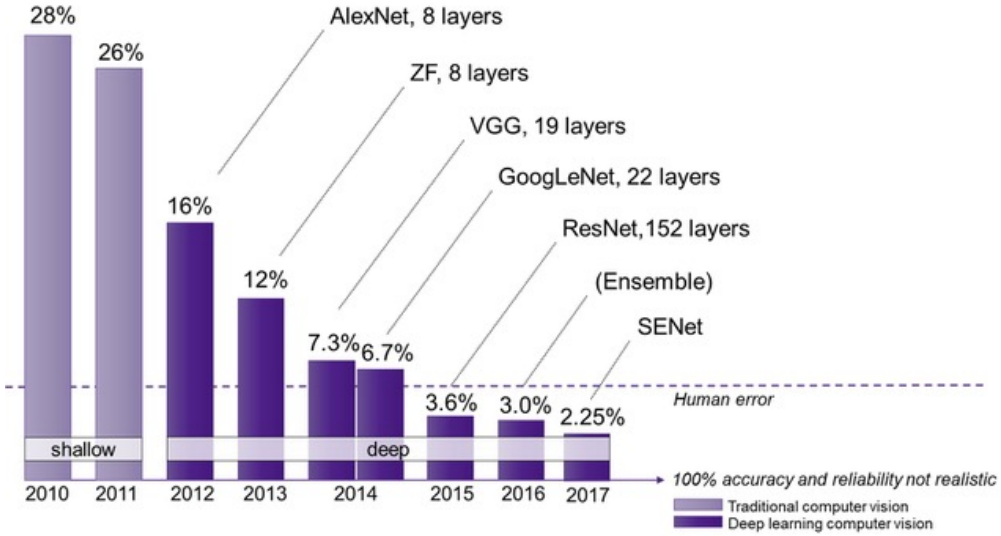
- Trained with stochastic gradient descent on two NVIDIA GTX 580 3GB GPUs for about a week.

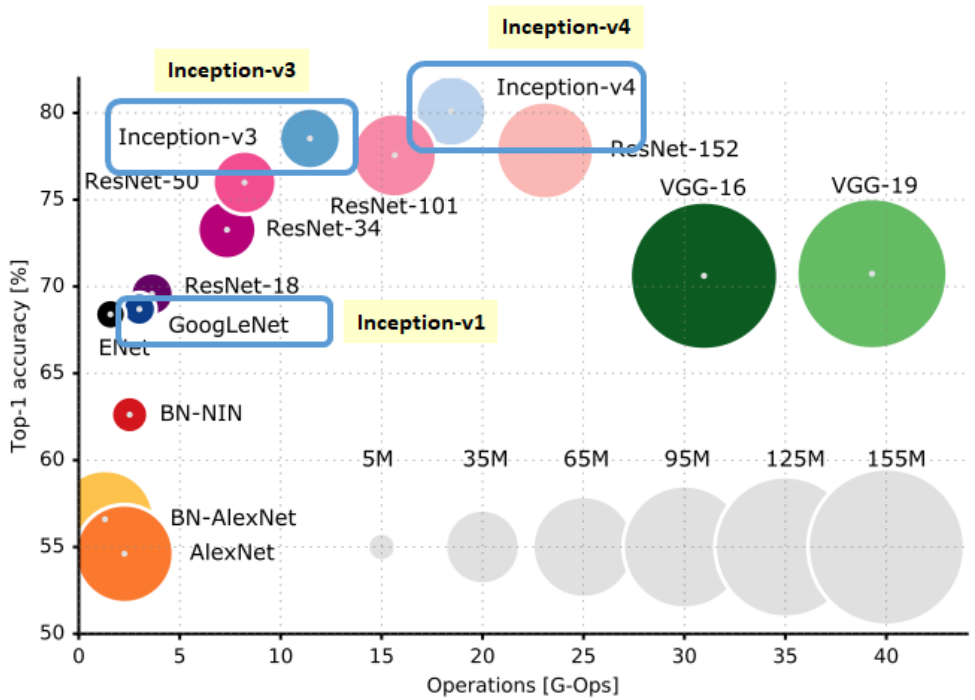
Results on ILSVRC-2012 competition

- Top-5 error rate: 15.3%.
- 2nd best team: 26.2% error.
- Human error: approx 8%.

Network size

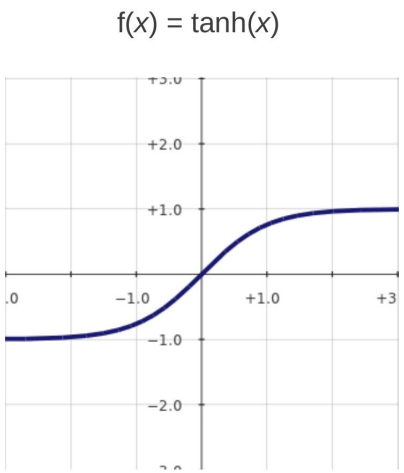
- Aprox. 650,000 neurons.
- 60,000,000 parameters.
- 630,000,000 connections.

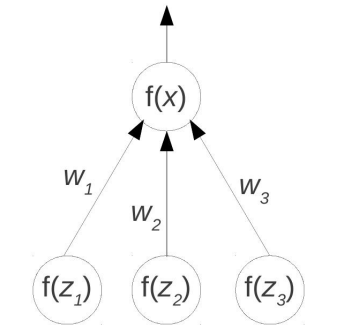

28%
26%
16%
12%
7.3%
6.7%
3.6%
3.0%
2.25%
AlexNet, 8 layers
ZF, 8 layers
VGG, 19 layers
GoogLeNet, 22 layers
ResNet,152 layers
(Ensemble)
SENet
Human error
shallow
deep
2010
2011
2012
2013
2014
2015
2016
2017
100% accuracy and reliability not realistic
Traditional computer vision
Deep learning computer vision


Inception-v4
Inception-v3
Inception-v4
Inception-v3
ResNet-152
ResNet-50
ResNet-101
VGG-16
VGG-19
ResNet-34
ResNet-18
GoogLeNet
Inception-v1
BN-NIN
BN-AlexNet
AlexNet
5M
35M
65M
95M
125M
155M
Top-1 accuracy [%]
Operations [G-Ops]
80
75
70
65
60
55
50
0
5
10
15
20
25
30
35
40

Relevant implementation Details Behind AlexNet.

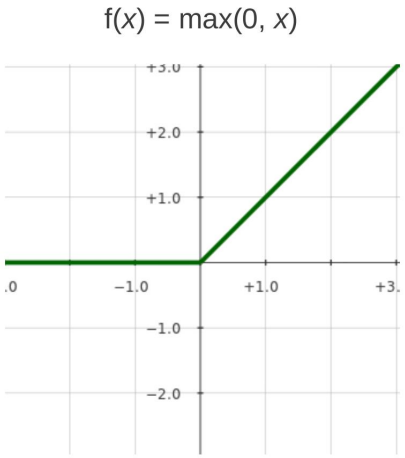
Training Tricks: Rectified Linear Units (ReLUs)





$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$

x is called the total input to the neuron, and f(x) is its output



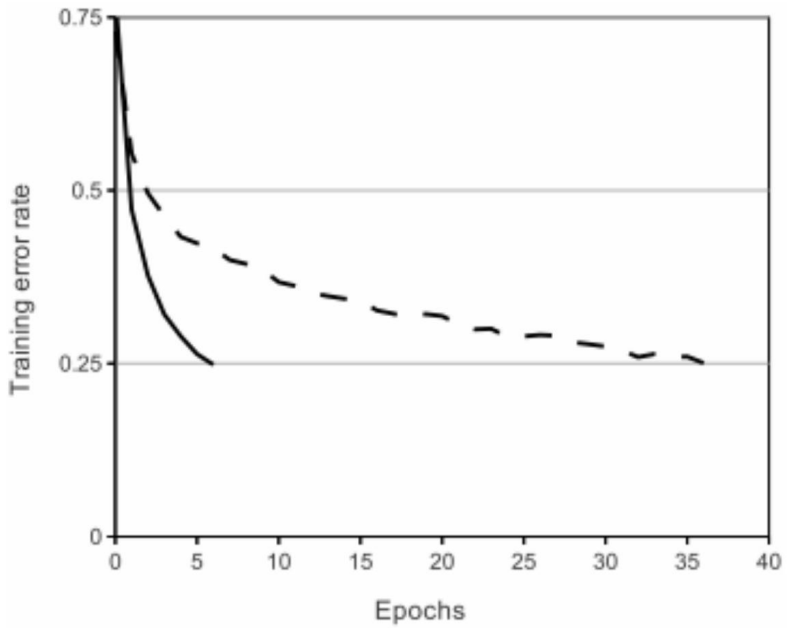
Very bad (slow to train)

Very good (quick to train)

- Usually, NNs use sigmoid activation functions, such as $tanh(x)$ or $(1 + exp^{-x})^{-1}$. Here, however, they use Rectified Linear Units (ReLU).
- Empirical observation: Deep convolutional neural networks with ReLUs train several times faster than their equivalents with sigmoid units.

Training Tricks: Rectified Linear Units (ReLUs)

- Example: A four-layer convolutional neural network with ReLUs (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with $tanh(x)$ neurons (dashed line).



Training Tricks: Data Augmentation

They employ several strategies for data augmentation:

- Image translation (5 areas of 224x224 pixels, four corner patches and center patch).
- Horizontal reflections.
- Alex's net implementation averages the softmax output of 10 patches from the test image by first resize image into sizes such that the minimal dimension is 256 while retains the aspect ratio and then center-crops into 256x256.
- The 10 patches of size 224x224 are sampled from the 256x256 crop the same way as from top left, top right, center, bottom left, bottom right of the resized test image, the second 5 patches are the horizontal mirrors of the first 5 patches.

Training Tricks: Dropout

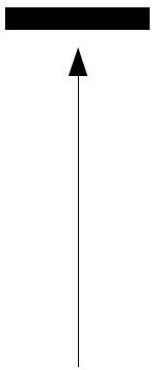
- In general, combining different models can be very useful (Mixture of experts, majority voting, boosting, etc.).
- Training many different models, however, is very time consuming.
- Here, they introduce Dropout.

Training Tricks: Dropout

- Dropout: at each iteration (epoch) set the output of each hidden neuron to zero with prob=0.5.
- The neurons which are “dropped out” in this way do not contribute to the forward pass and do not participate in backpropagation.
- So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.
- This technique reduces complex co-adaptations of neurons, since each neuron cannot rely on the presence of other neurons.
- Using dropout, each neuron is forced to learn more robust features that are useful in conjunction with many different random subsets of other neurons.
- Without dropout, the network exhibits substantial overfitting.
- Downside: Dropout roughly doubles the number of iterations required to converge.

Training Tricks: Substract Mean Image

An input image (256x256)



Minus sign

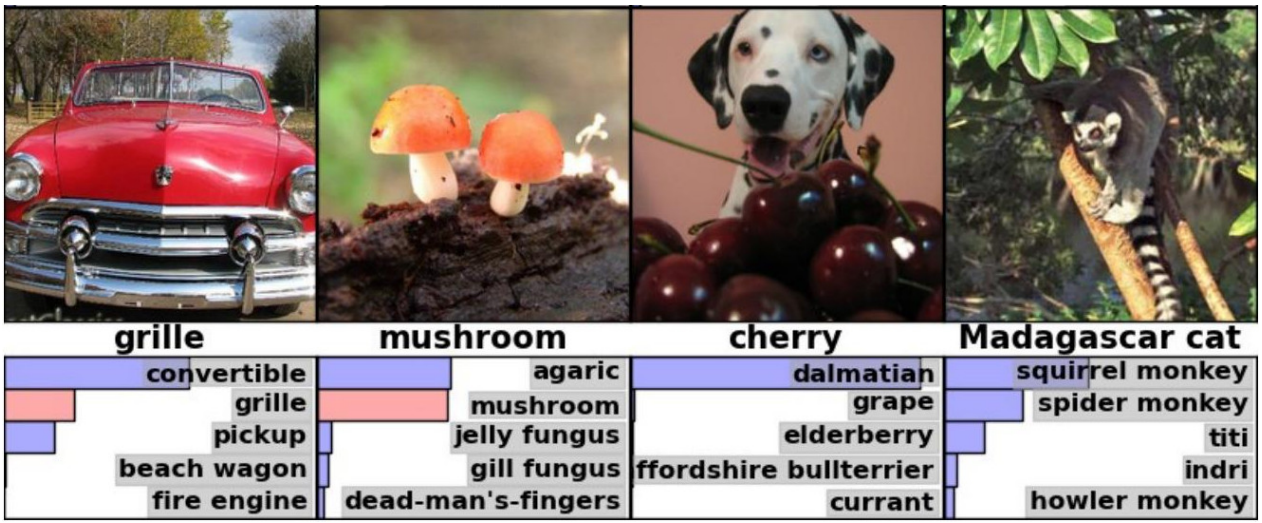
The mean input image

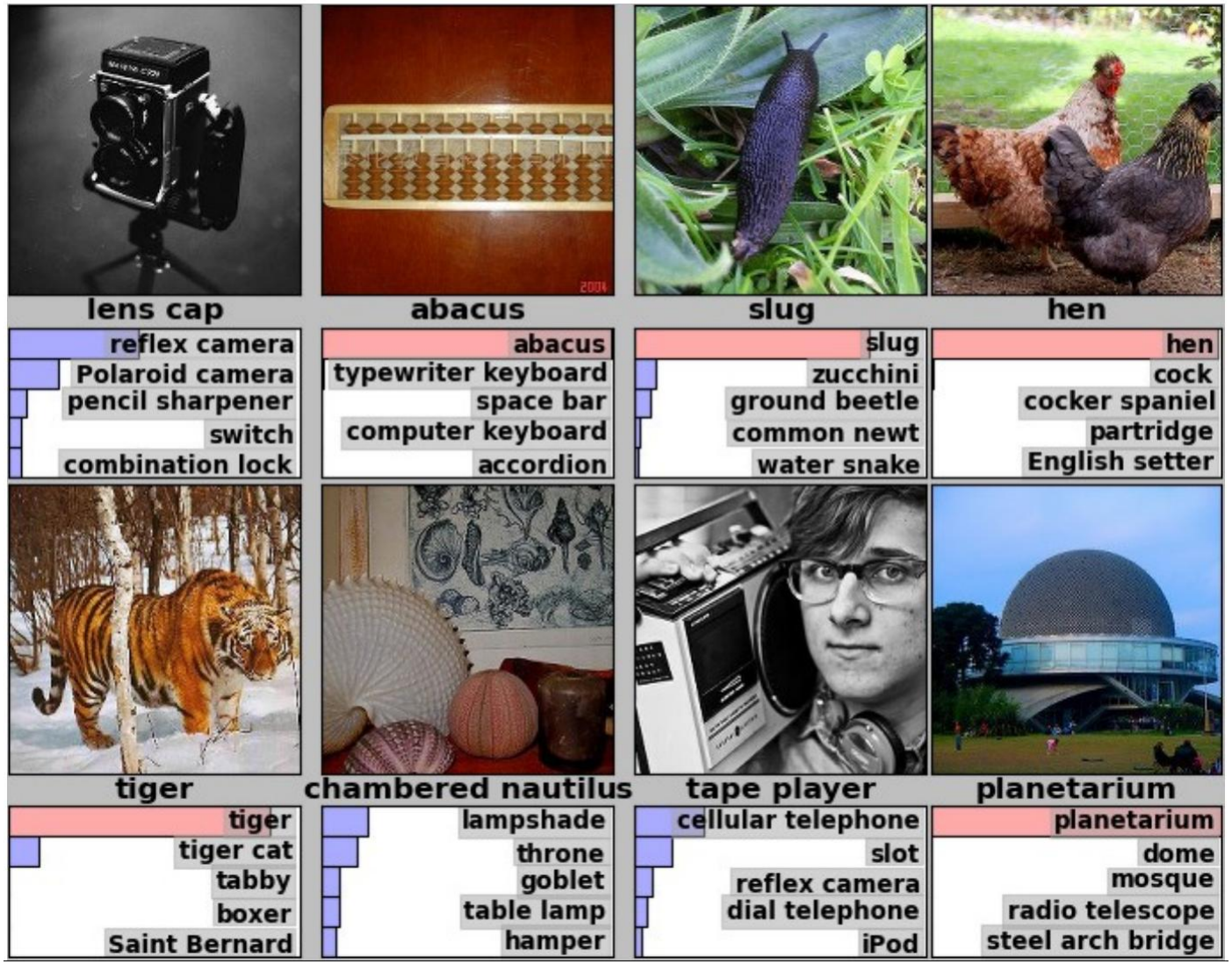
Results on ILSVRC

Results on ILSVRC-2012 competition:

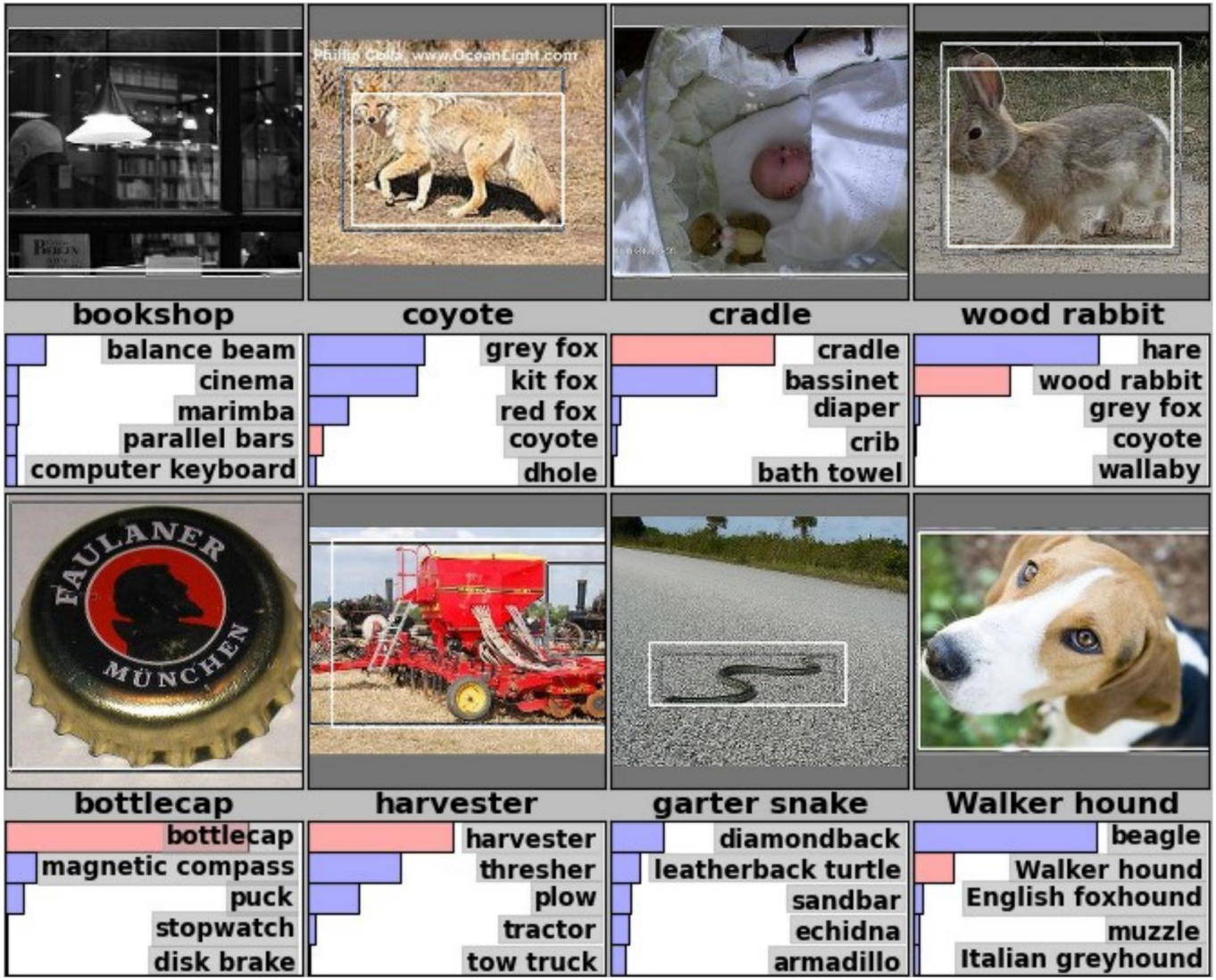
- Top-5 error rate: 15.3%.
- 2nd best team: 26.2% error.
- Human error approx. 8%.

Take these numbers with care, as an example, the following cases correspond to wrong classifications:



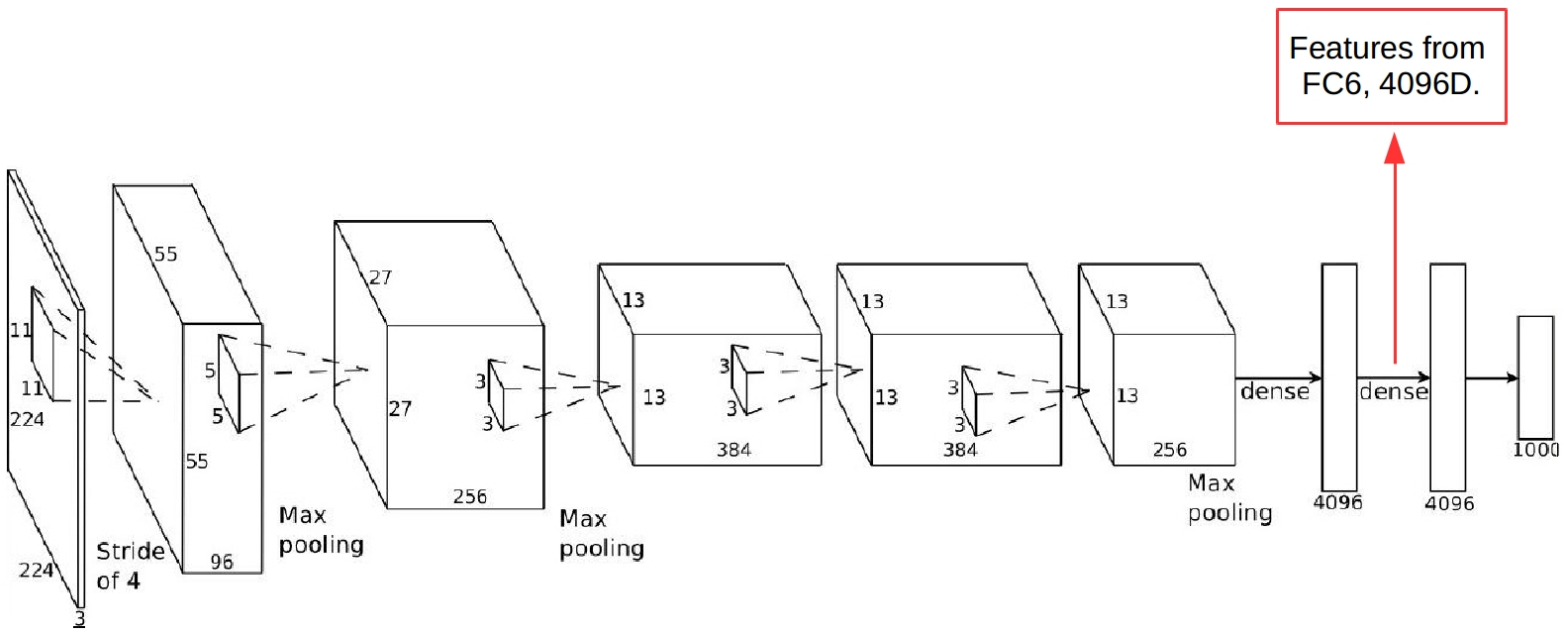

lens cap
reflex camera
Polaroid camera
pencil sharpener
switch
combination lock
abacus
abacus
typewriter keyboard
space bar
computer keyboard
accordion
slug
slug
zucchini
ground beetle
common newt
water snake
hen
hen
cock
cocker spaniel
partridge
English setter
tiger
tiger
tiger cat
tabby
boxer
Saint Bernard
chambered nautilus
lampshade
throne
goblet
table lamp
hamper
tape player
cellular telephone
slot
reflex camera
dial telephone
iPod
planetarium
planetarium
dome
mosque
radio telescope
steel arch bridge

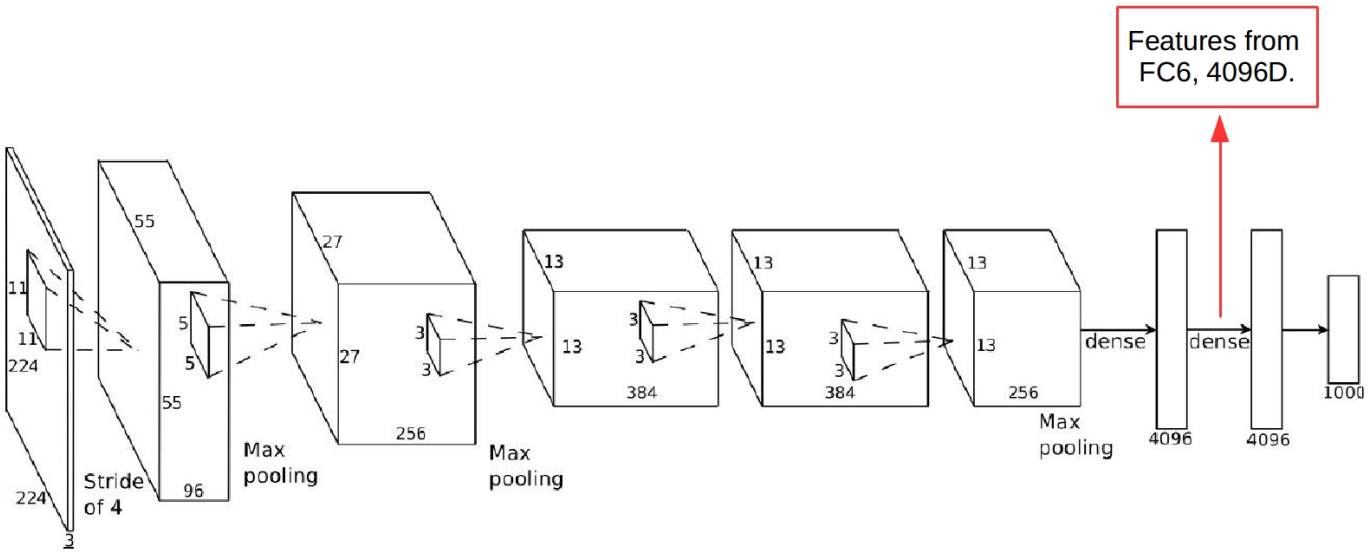

koala
wombat
Norwegian elkhound
wild boar
wallaby
koala
tiger
tiger
tiger cat
jaguar
lynx
leopard
European fire salamander
European fire salamander
spotted salamander
common newt
long-horned beetle
box turtle
loggerhead
African crocodile
Gila monster
loggerhead
mud turtle
leatherback turtle
seat belt
seat belt
ice lolly
hotdog
burrito
Band Aid
television
television
microwave
monitor
screen
car mirror
sliding door
sliding door
shoji
window shade
window screen
four-poster
wallaby
hare
wallaby
wood rabbit
Lakeland terrier
kit fox

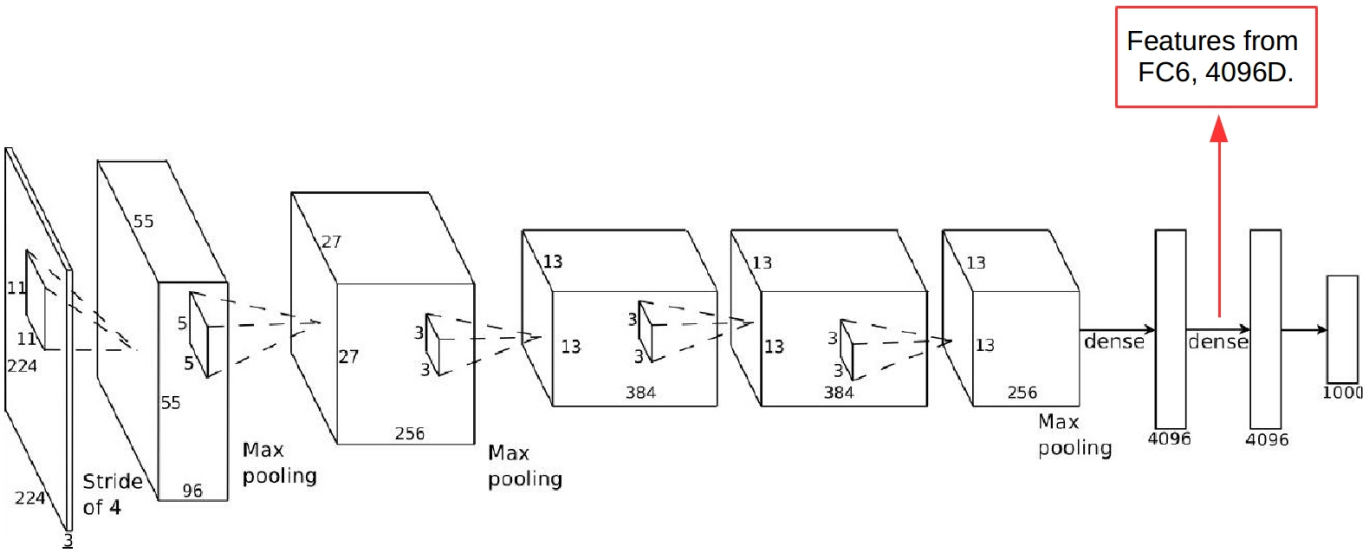

bookshop
balance beam
cinema
marimba
parallel bars
computer keyboard
coyote
grey fox
kit fox
red fox
coyote
dhole
cradle
cradle
bassinet
diaper
crib
bath towel
wood rabbit
hare
wood rabbit
grey fox
coyote
wallaby
bottlecap
bottlecap
magnetic compass
puck
stopwatch
disk brake
harvester
harvester
thresher
plow
tractor
tow truck
garter snake
diamondback
leatherback turtle
sandbar
echidna
armadillo
Walker hound
beagle
Walker hound
English foxhound
muzzle
Italian greyhound

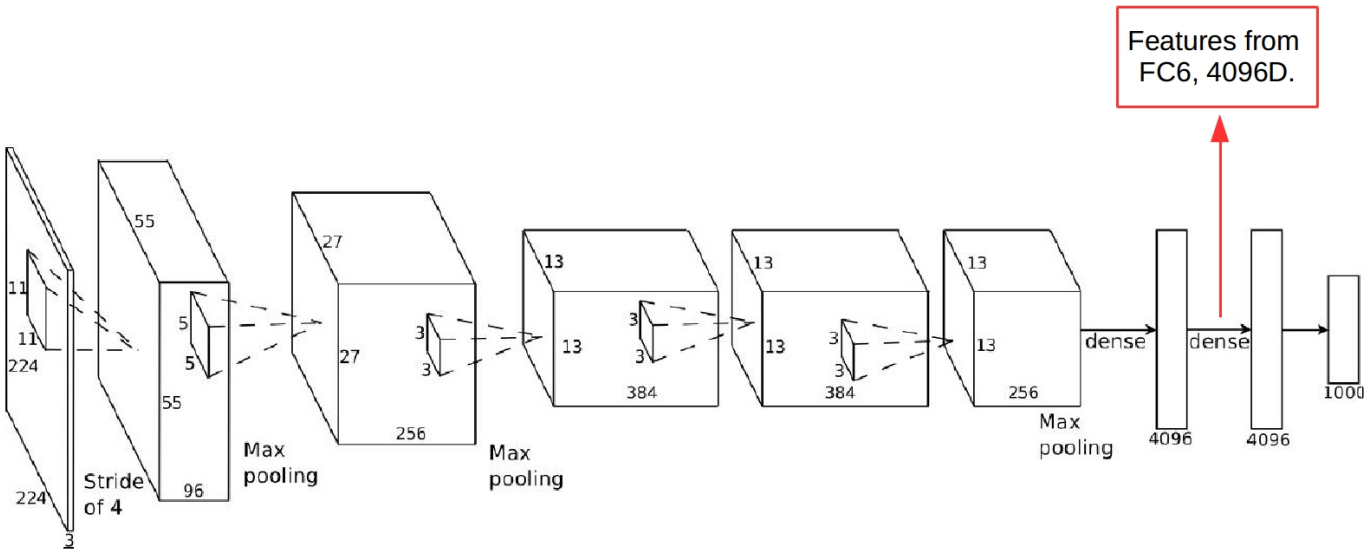
While the previous CNN was trained on ImageNet, a nice property is that the learned features can be useful to other visual applications.

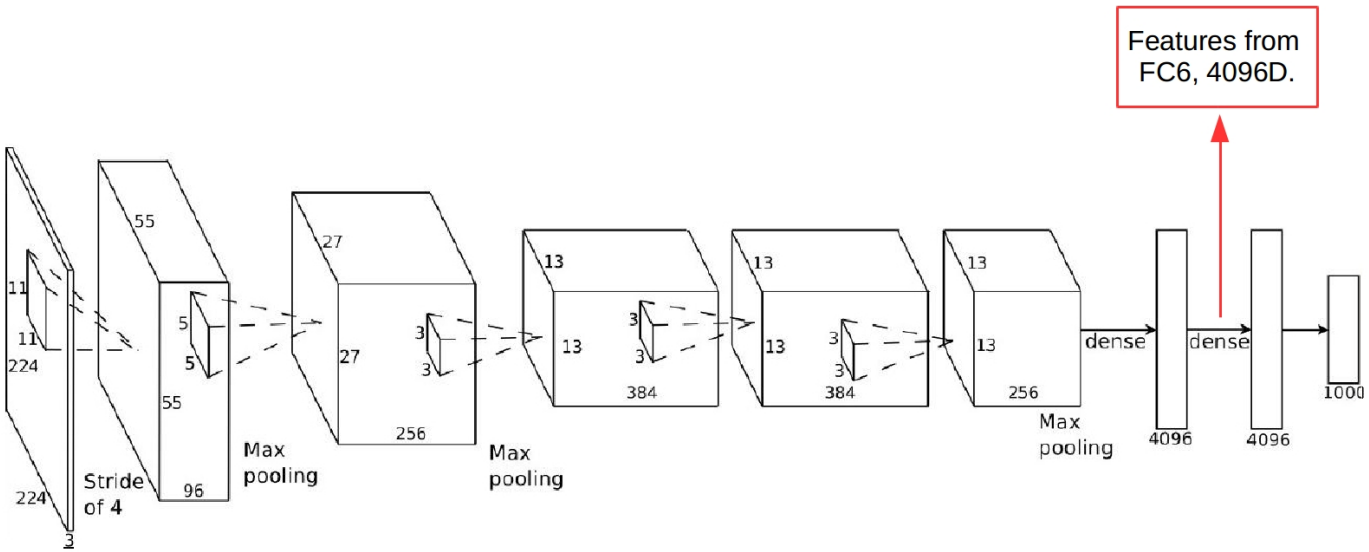
- As an example, you can use features from layer 6, also known as fully connected layer 6 (FC6) to feed classifiers for visual recognition tasks.

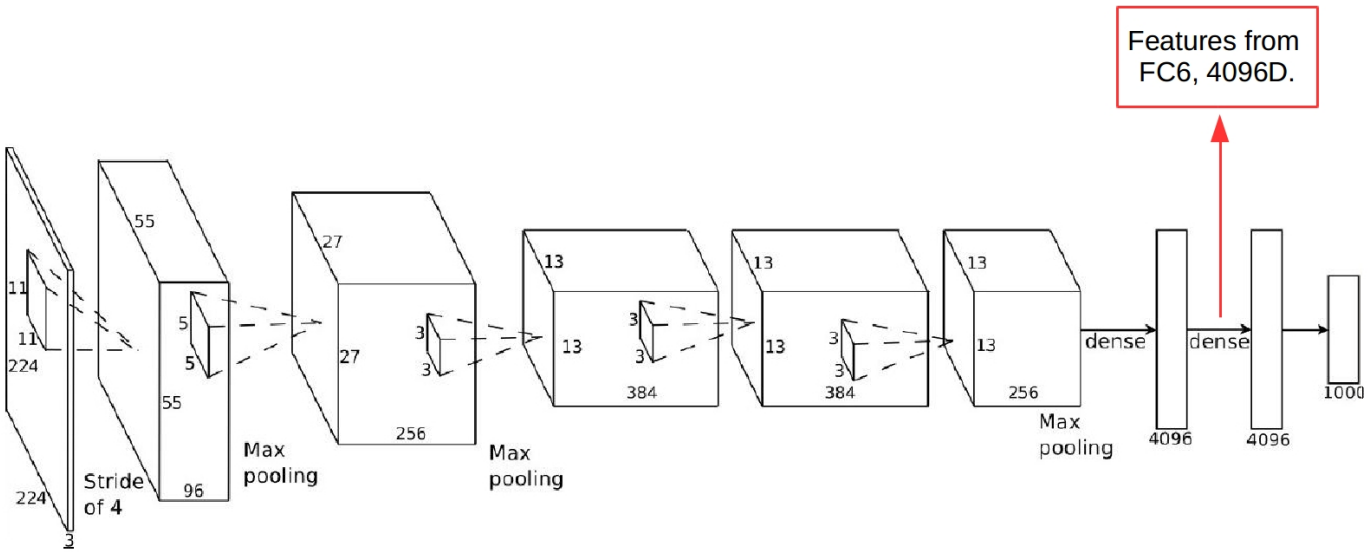


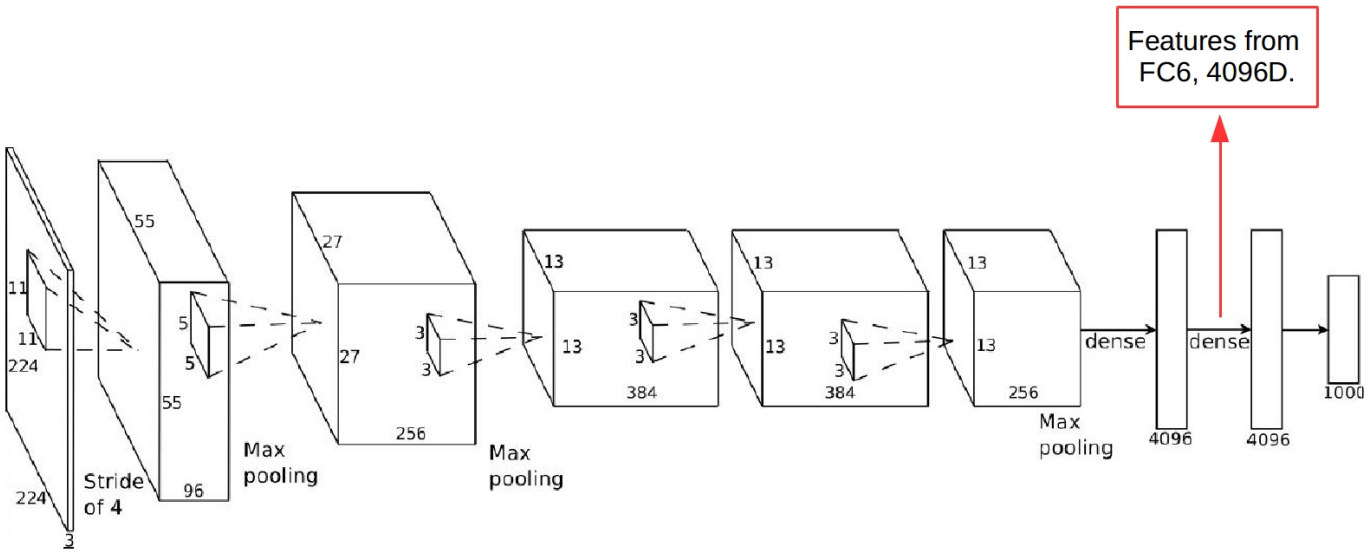

Features from
FC6, 4096D.
11
11
224
224
3
55
55
96
5
5
Stride
of 4
27
27
256
3
3
Max
pooling
13
13
384
3
3
Max
pooling
13
13
384
3
3
13
13
256
Max
pooling
dense
4096
dense
4096
1000

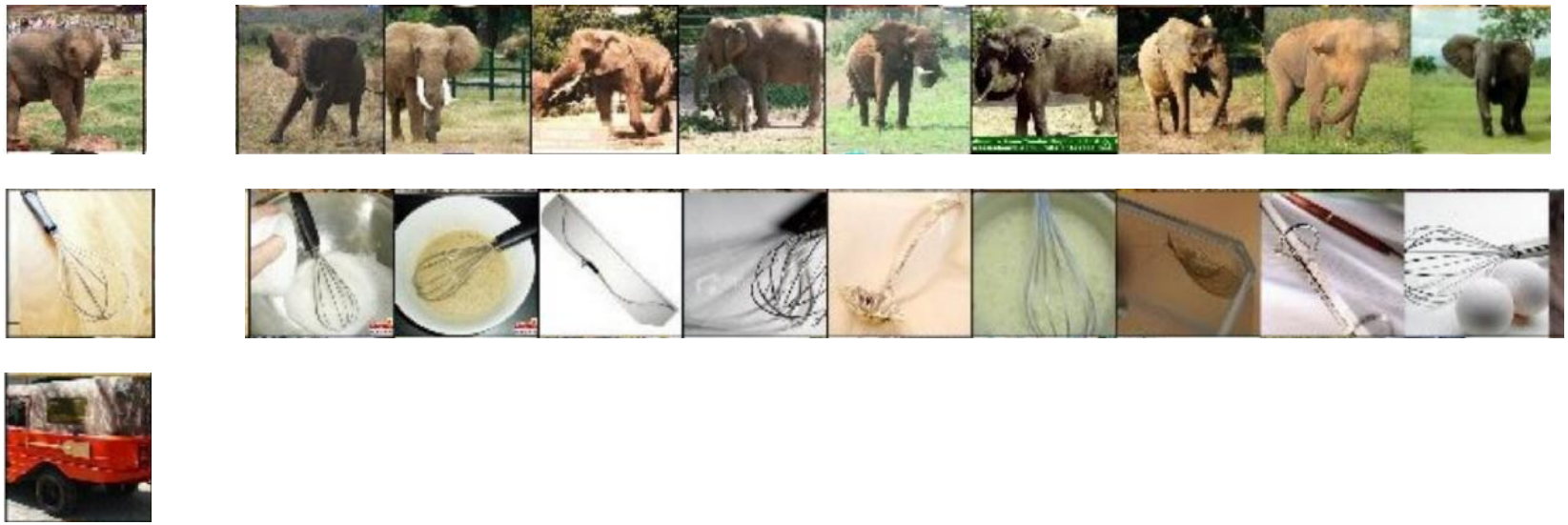

Features from
FC6, 4096D.
11
11
224
224
3
55
55
96
5
5
Stride
of 4
27
27
256
3
3
Max
pooling
13
13
384
3
3
Max
pooling
13
13
384
3
3
13
13
256
Max
pooling
dense
4096
dense
4096
1000


Features from FC6, 4096D.
11
11
224
224
3
55
55
96
5
5
Stride of 4
27
27
256
3
3
Max pooling
13
13
384
3
3
Max pooling
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000


Features from FC6, 4096D.
11
11
224
224
3
55
55
5
5
96
Stride of 4
27
27
3
3
256
Max pooling
13
13
3
3
384
Max pooling
13
13
3
3
384
13
13
256
Max pooling
dense
4096
dense
4096
1000


Features from FC6, 4096D.
11
11
224
224
3
55
55
96
5
5
Stride of 4
27
27
256
3
3
Max pooling
13
13
384
3
3
Max pooling
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000


Features from FC6, 4096D.
11
11
224
224
3
55
55
96
5
5
Stride of 4
27
27
256
3
3
Max pooling
13
13
384
3
3
Max pooling
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000



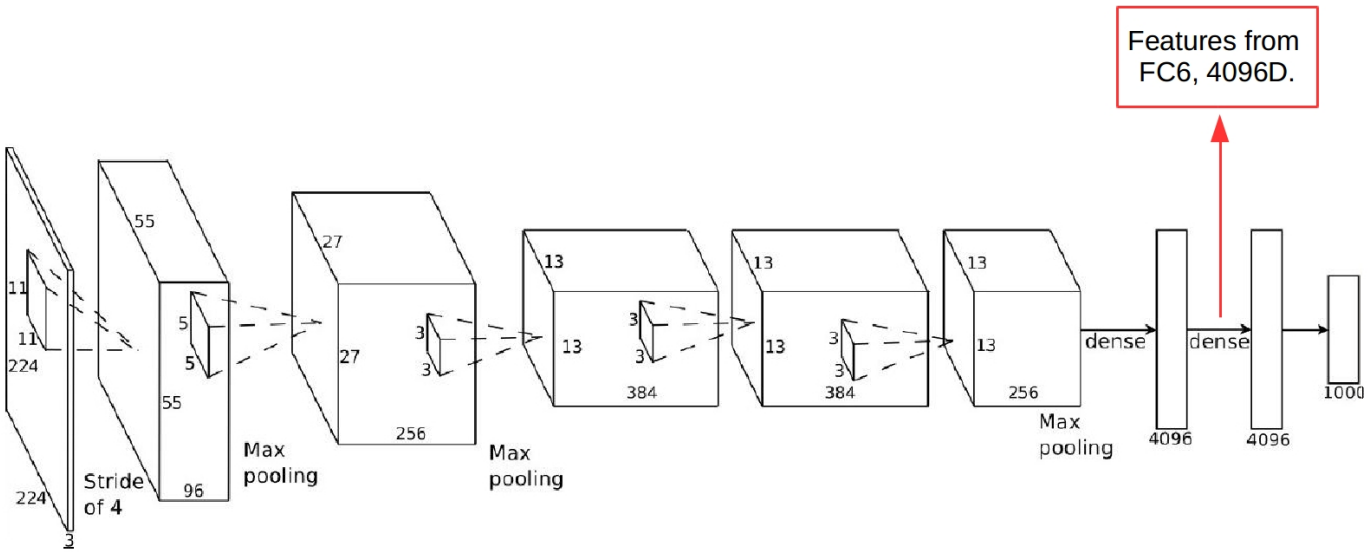

Features from FC6, 4096D.
11
11
224
224
3
55
55
5
5
96
Stride of 4
27
27
256
3
3
Max pooling
13
13
384
3
3
Max pooling
13
13
384
3
3
13
13
256
Max pooling
dense
4096
dense
4096
1000

Image Retrieval

First column contains query images from ILSVRC-2010 test set, remaining columns contain retrieved images from training set.

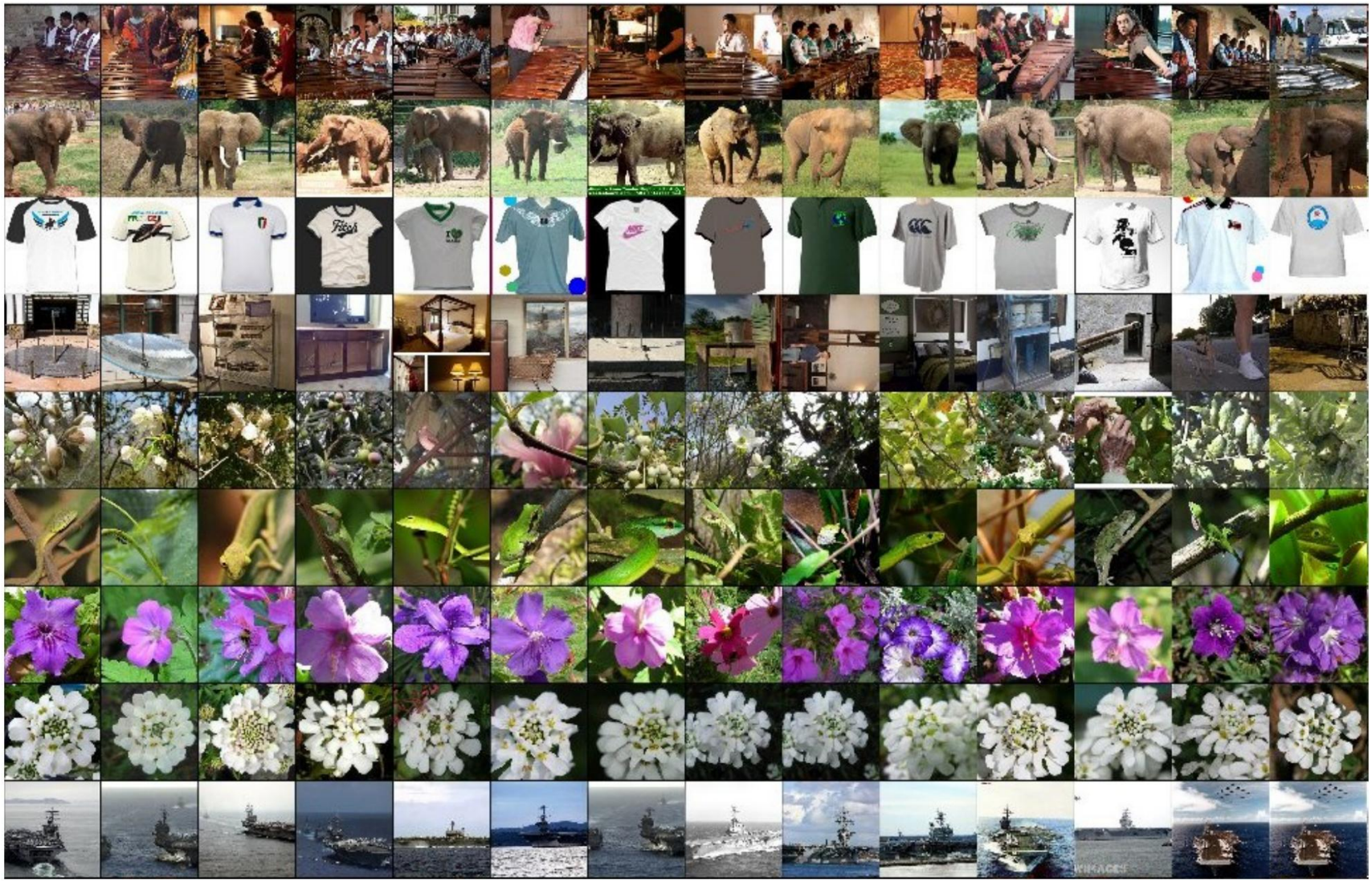
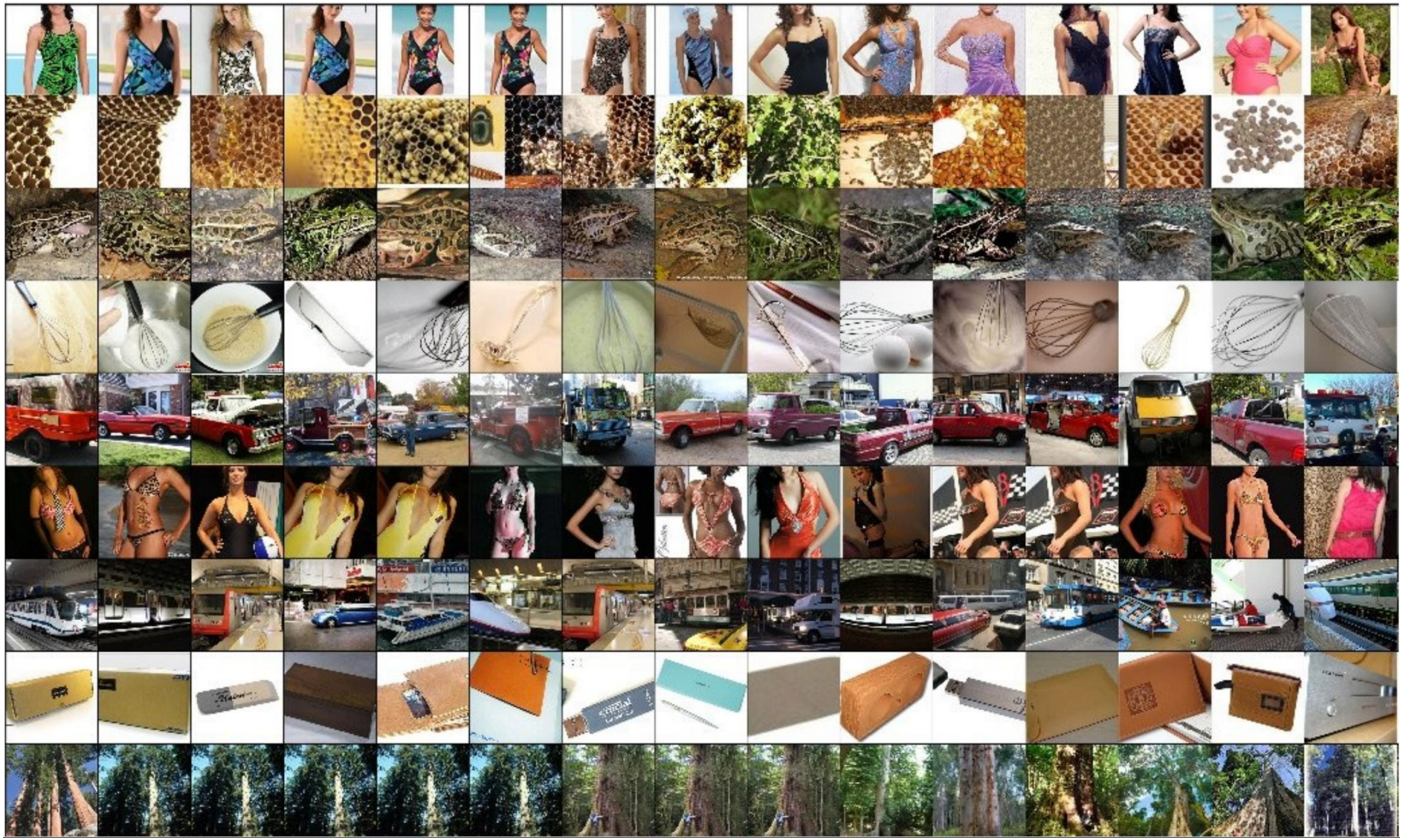


Image Retrieval



Some Classes in ImageNet Are Very Specific and Similar

(a) Siberian husky

(b) Eskimo dog

Today, several closely related but deeper CNN architectures, have reduced the previous error rate. As an example, the work by K. Simonyan and A. Zisserman achieves a Top-5 error rate of 6.8% (deep: 16-19 weight layers).[2]

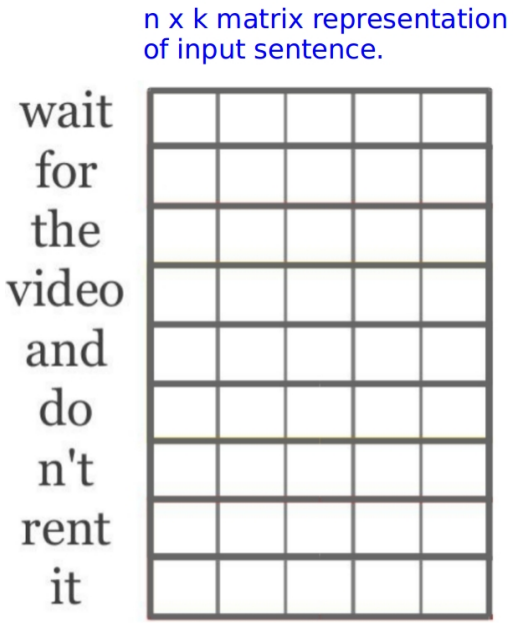
[2] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. International Conference on Learning Representations, 2015.

Case Example

Text Mining and Convolutional Neural Networks

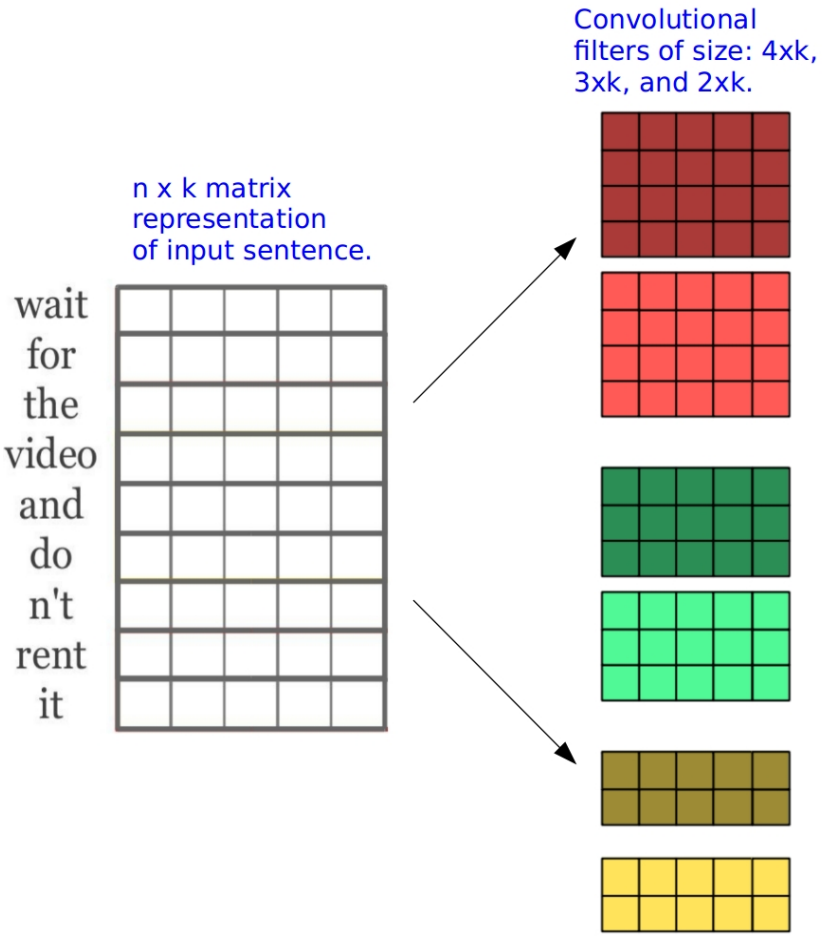
Kim et al., 2014.

A Simple CNN Model for Text Mining (Kim et al., 2014)



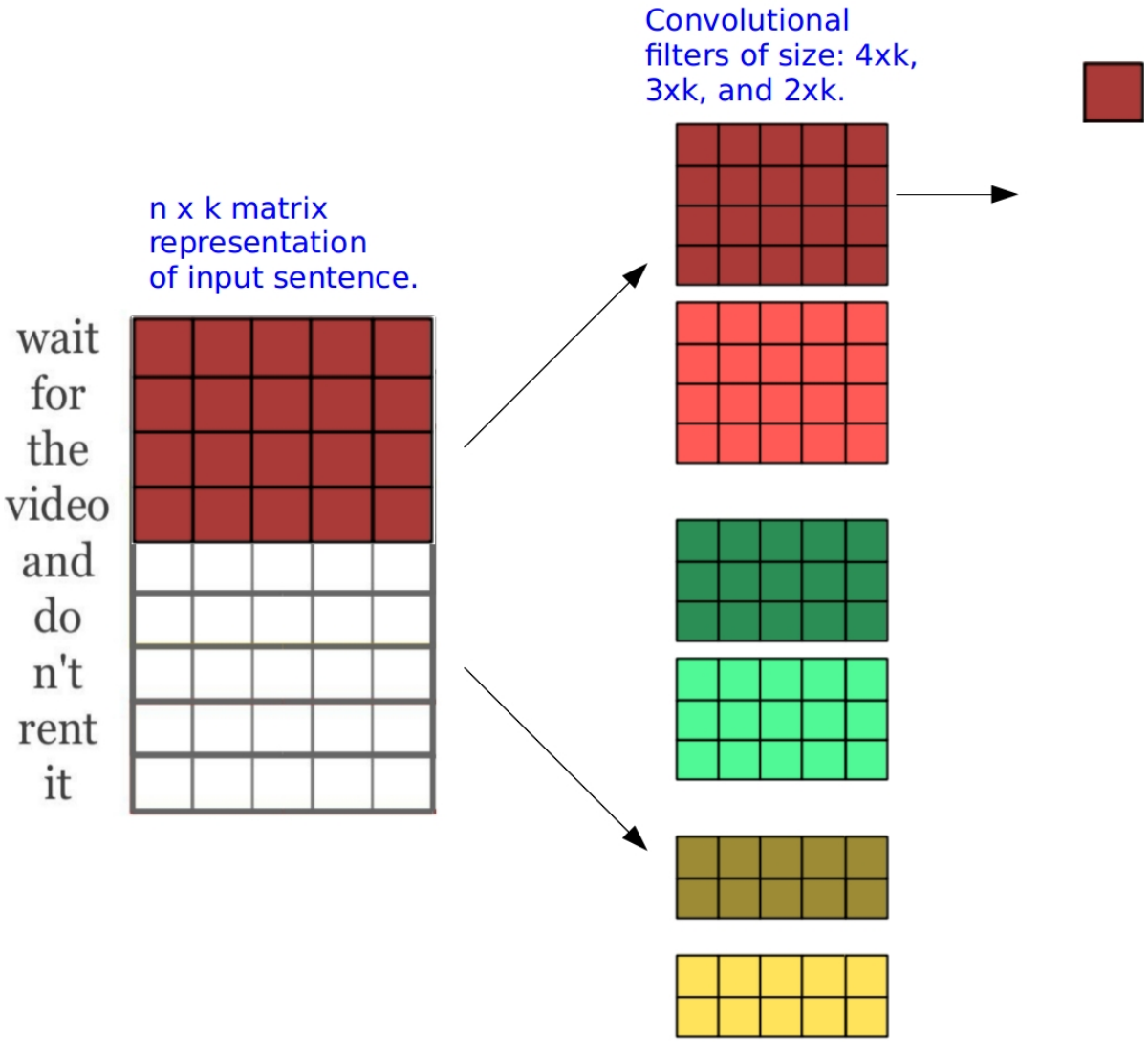
- Input is a sentence matrix, where each row corresponds to the coding of a sentence word using Word2Vec.
- Therefore, for a sentence of *n* words and a coding using *k* dimensions, the input matrix has size *nxk*.
- Kim et al. use a 300D Word2Vec model trained on 100 billion words of Google News (k=300).

A Simple CNN Model for Text Mining (Kim et al., 2014)

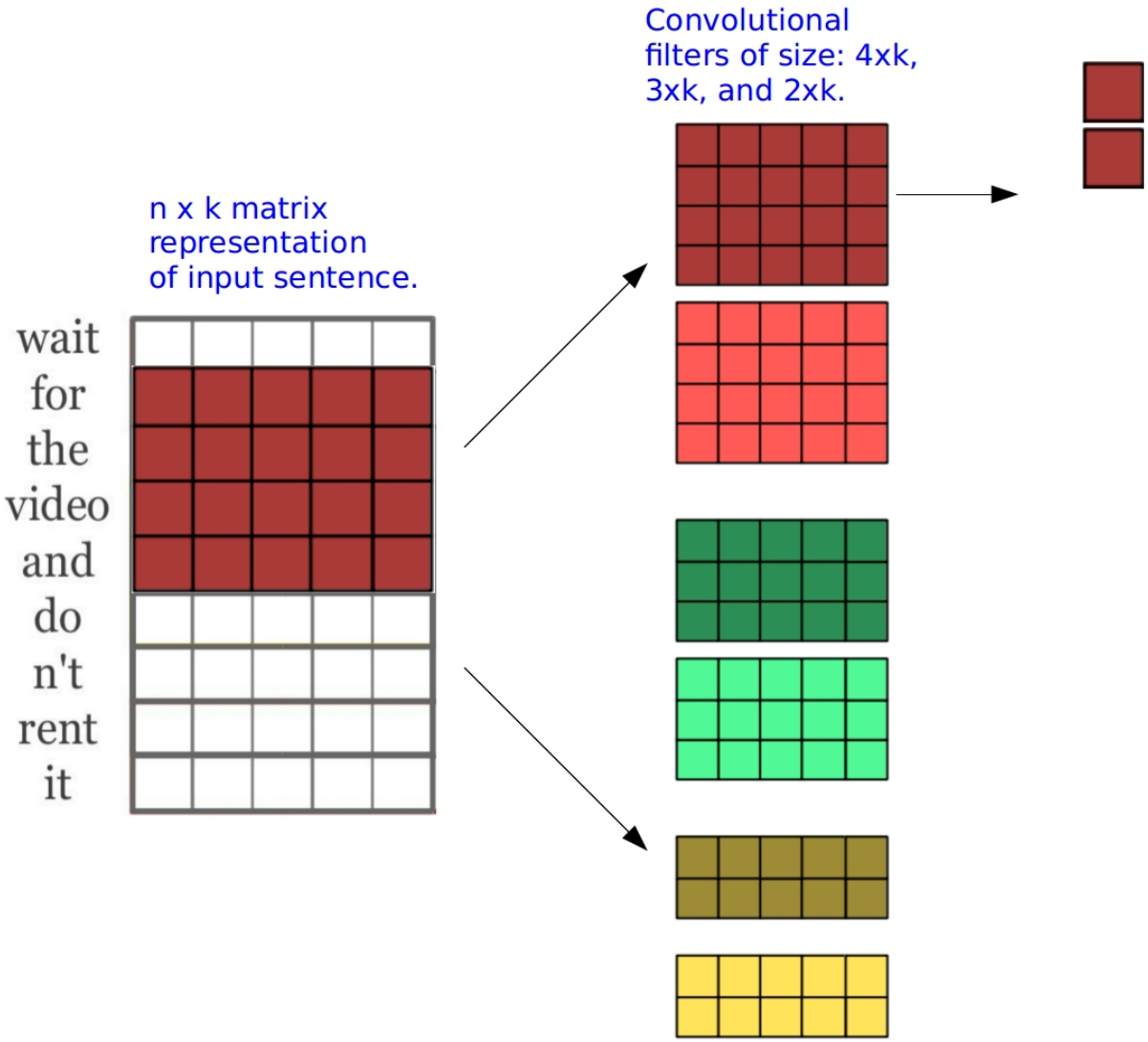


- As in Alex's Net, a key step of CNN's training is learning weights of convolutional filters.

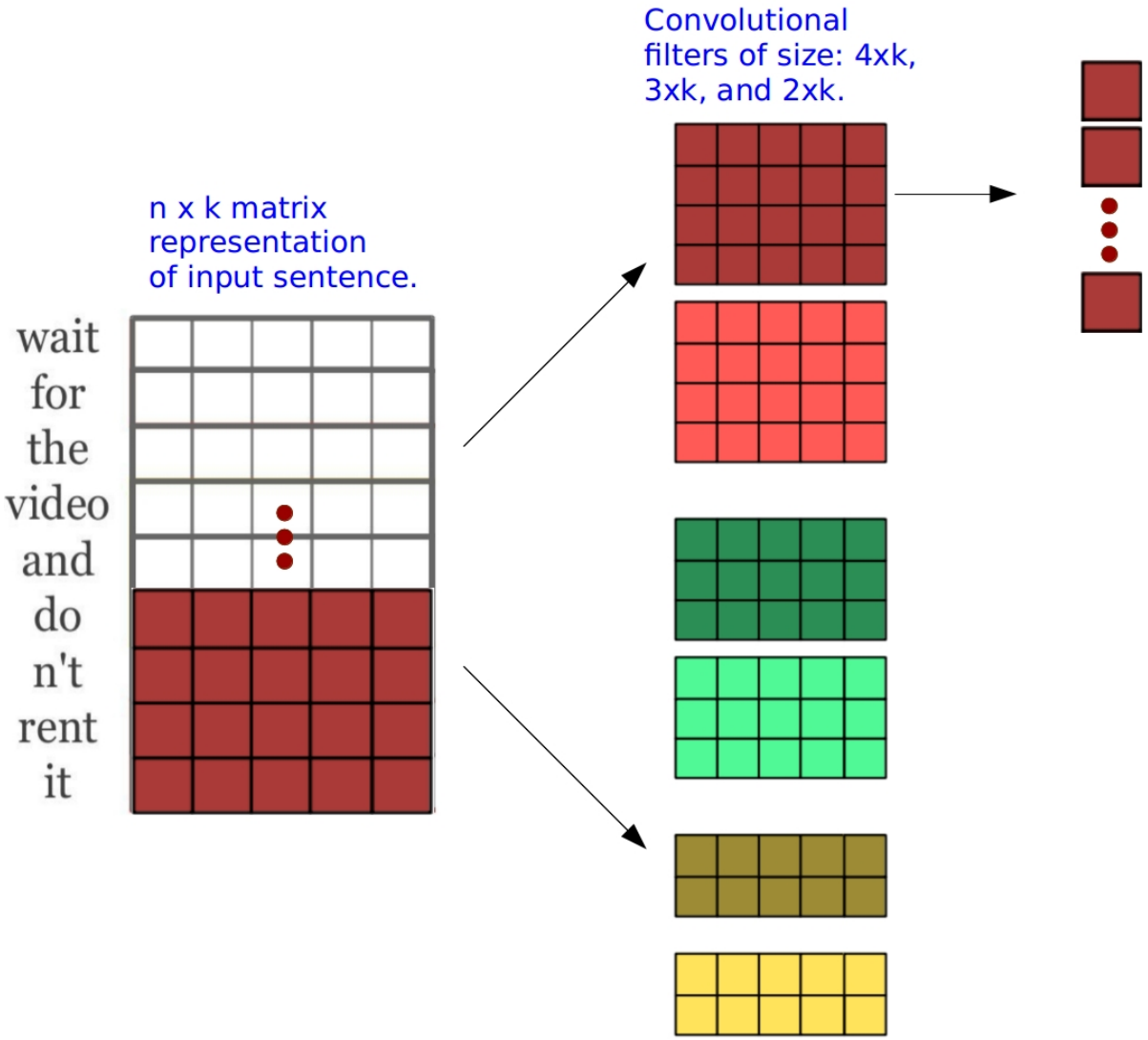
Convolution



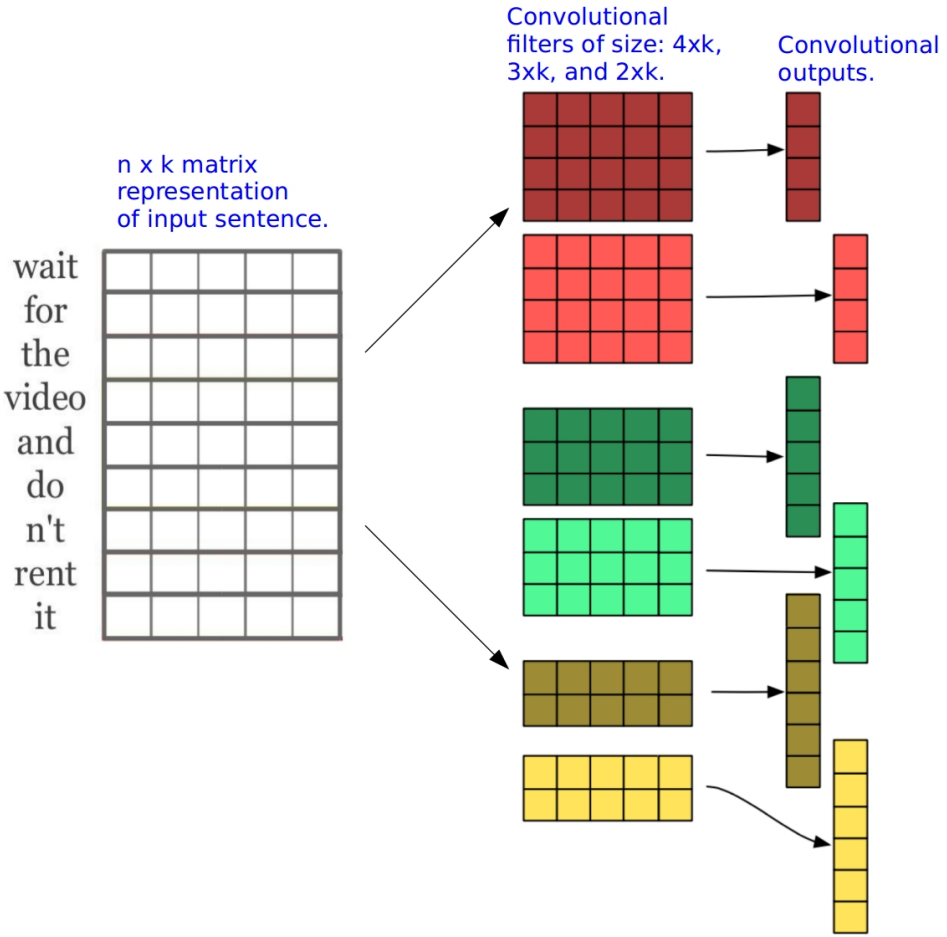
Convolution



Convolution

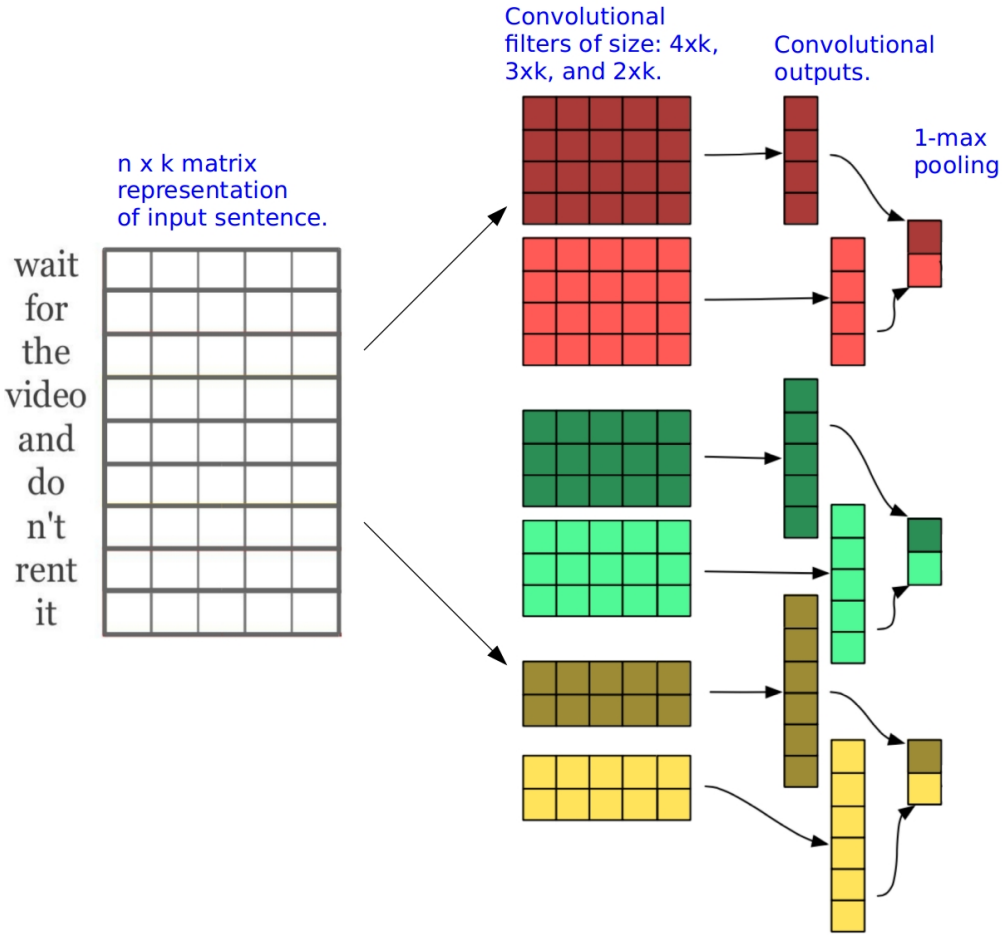


A Simple CNN Model for Text Mining (Kim et al., 2014)



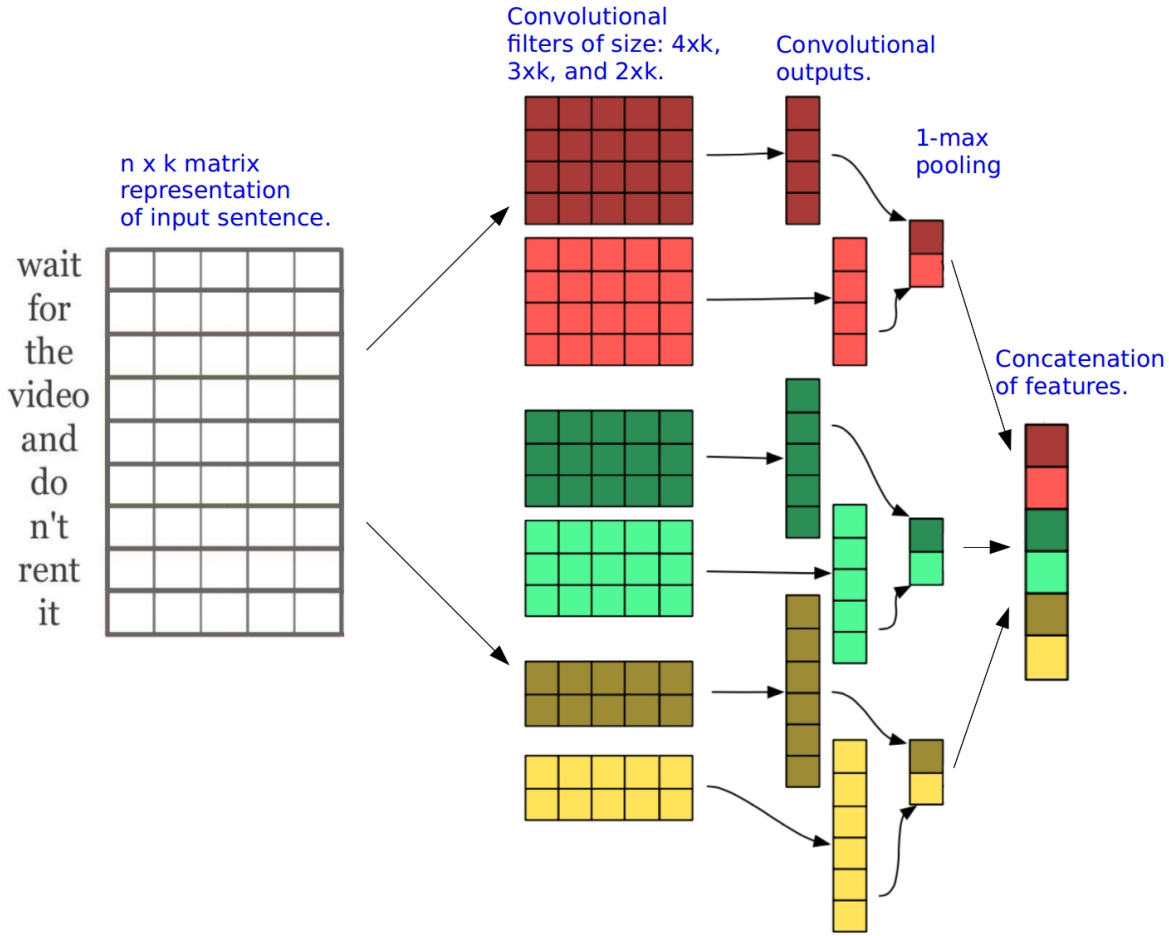
- Kim et al. use window filter sizes of 3,4,5 words (number of rows).
- Each filter size generates 100 feature maps.
- How many types of features do we learn with this filter architecture?

A Simple CNN Model for Text Mining (Kim et al., 2014)



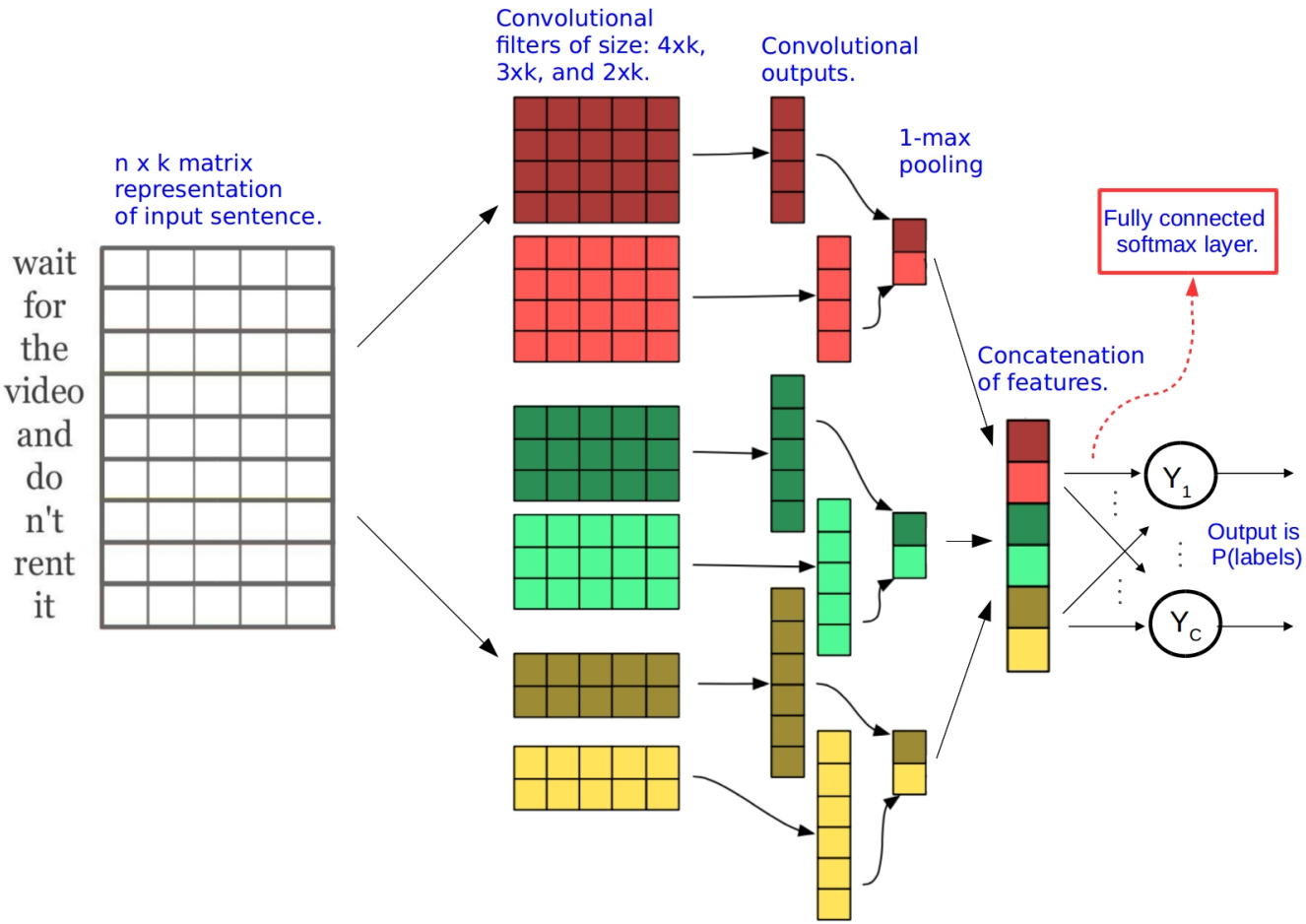
- Outputs from each filter are 1-max pooled (winner-takes-all).
- This pooling scheme naturally deals with variable sentence lengths.
- How this pooling affect the sentence model? (Hint: think about word order in input sentence).

A Simple CNN Model for Text Mining (Kim et al., 2014)



- Outputs from filters are concatenated to form the feature vector that will feet the final classification step.

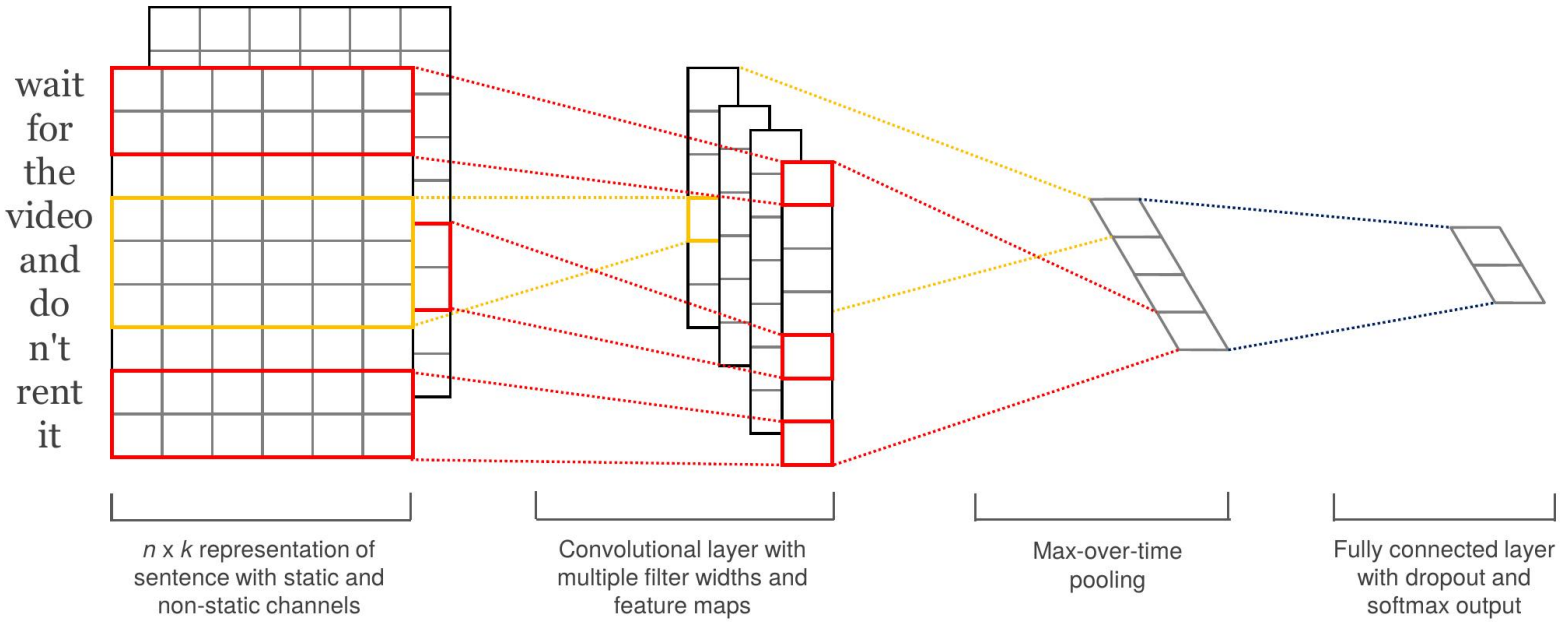
A Simple CNN Model for Text Mining (Kim et al., 2014)



- Complete architecture including a final fully connected layer with soft-max output.
- Soft-max units provide a probability estimate over target classes.

A Simple CNN Model for Text Mining (Kim et al., 2014)

Final model in Kim et al., 2014.



- They use two channels of input word vectors, one that is kept static throughout training and one that is fine-tuned via backpropagation.

A Simple CNN Model for Text Mining (Kim et al., 2014)

Further design issues:

Dropout

- After each convolution, they apply a ReLU activation function.
- They apply dropout on the penultimate layer. They use $p = 0.5$.
- At test time, there is no dropout, so feature vectors are larger. They avoid this problem by scaling the weights by p.
- By using drop-out, they report 2-4% improved accuracy and ability to use very large networks without overfitting.

Training

- They train the network using backpropagation.
- They use Adadelta to quickly search the hyperparameter space and then build final model with Adagrad.
- Mini batch size for SGD training: 50.
- Words not in word2vec are initialized randomly.
- They apply a constraint on L2-norms of the weight vectors.

Benchmarks

- **MR**: Movie reviews with one sentence per review. Classification involves detecting positive/negative reviews (Pang and Lee, 2005).[3]
- **SST-1**: Stanford Sentiment Treebank—an extension of MR but with train/dev/test splits provided and fine-grained labels (very positive, positive, neutral, negative, very negative), re-labeled by Socher et al. (2013).[4]
- **SST-2**: Same as SST-1 but with neutral reviews removed and binary labels.
- **Subj**: Subjectivity dataset where the task is to classify a sentence as being subjective or objective (Pang and Lee, 2004).
- **TREC**: TREC question dataset—task involves classifying a question into 6 question types (whether the question is about person, location, numeric information, etc.) (Li and Roth, 2002).[5]
- **CR**: Customer reviews of various products (cameras, MP3s etc.). Task is to predict positive/negative reviews (Hu and Liu, 2004).[6]
- **MPQA**: Opinion polarity detection subtask of the MPQA dataset (Wiebe et al., 2005).[7]

Benchmarks

Data	c	l	N	$\|V\|$	$\|V_{pre}\|$	*Test*
MR	2	20	10662	18765	16448	CV
SST-1	5	18	11855	17836	16262	2210
SST-2	2	19	9613	16185	14838	1821
Subj	2	23	10000	21323	17913	CV
TREC	6	10	5952	9592	9125	500
CR	2	19	3775	5340	5046	CV
MPQA	2	3	10606	6246	6083	CV

c: Number of target classes.

l: Average sentence length.

N: Dataset size.

$|V|$: Vocabulary size.

$|V_{pre}|$: n_words in set of pre-trained word vectors.

Test: Test set size (CV: 10-fold CV instead of train/test split).

Results

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

CNN-rand: initial word vectors are random.
CNN-static: initial word vector given by Word2Vec coding.
CNN-non-static: initial word vector given by Word2Vec are fine-tuned for each task.
CNN-multichannel: contain two sets of word vectors, one is kept fixed, other one is fine-tuned.

Results

	Most Similar Words for	
	Static Channel	Non-static Channel
bad	*good* *terrible* *horrible* *lousy*	*terrible* *horrible* *lousy* *stupid*
good	*great* *bad* *terrific* *decent*	*nice* *decent* *solid* *terrific*
n't	*os* *ca* *ireland* *wo*	*not* *never* *nothing* *neither*
!	*2,500* *entire* *jez* *changer*	*2,500* *lush* *beautiful* *terrific*
,	*decasia* *abysmally* *demise* *valiant*	*but* *dragon* *a* *and*

Based on cosine similarity for vectors in the static channel and fine-tuned vectors in the non-static channel. Results using the multichannel model and the SST-2 dataset.

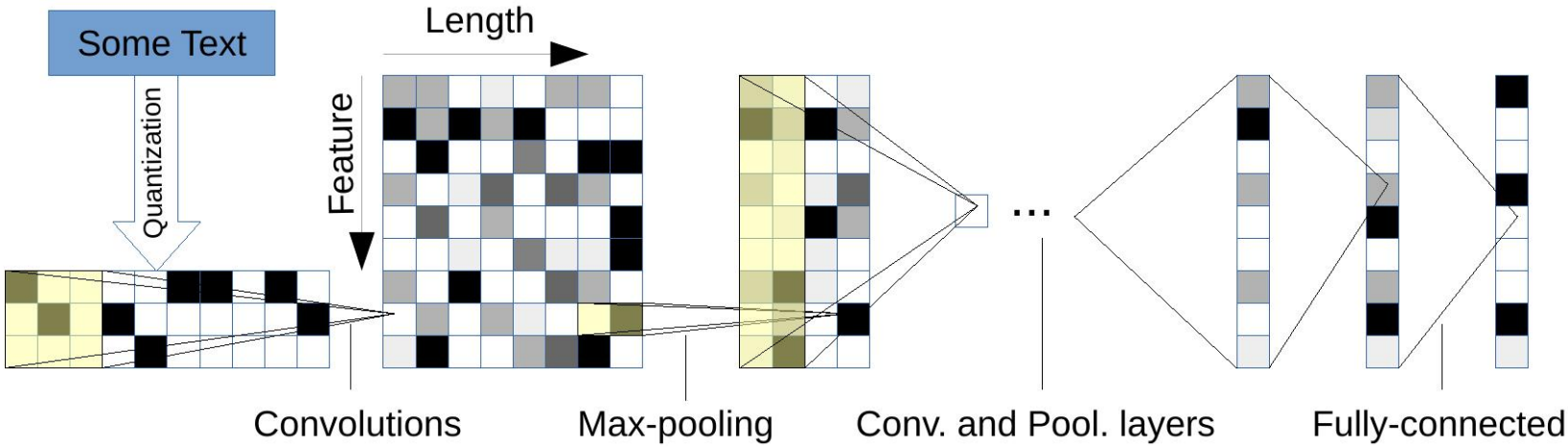
Further Example

Text Mining and Convolutional Neural Networks

Zhang et al., 2015

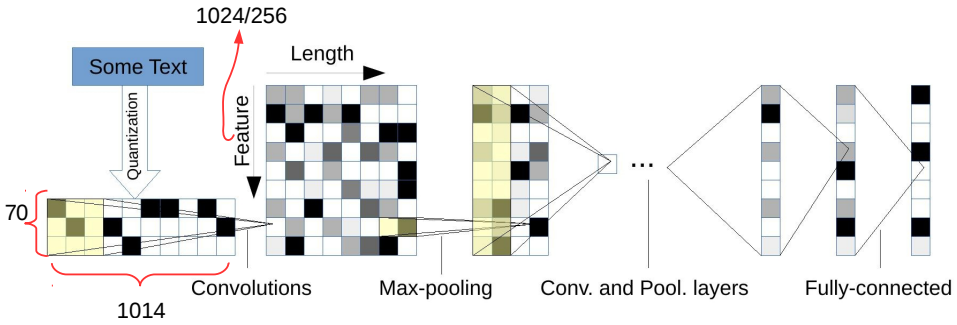
Deeper Architectures (Zhang et al., 2015)

A character-level CNN for text classification. This has the advantage that abnormal character combinations such as misspellings and emoticons may be naturally learnt.



- Character encoding is done by defining an alphabet of size $|m|$ according to input language. Then each character is quantized using a 1-of-|m| encoding (one-hot).
- They use an alphabet that consists of 70 characters (m=70), including 26 english letters, 10 digits, 33 other characters and the new line character.
- Characters that are not in the alphabet (including blank spaces) are quantized as all-zero vectors.

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:’’’/\|_@#$%ˆ&*˜‘+-=<>()[]{}
```



- They consider input as a sequence of 1014 chars.
- For the intended applications, 1014 chars capture most of the texts of interest.
- Then, input to the NW consists of 1014 attributes of 70 dims each, why?

- Nw consists of 6 convolutional layers and 3 fully-connected layers.
- They test using 1024 and 256 filters. Filter sizes are 3 or 7 (see table).
- They insert 2 dropout modules in between the 3 fully-connected layers (p=0.5).

Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

Layer	Output Units Large	Output Units Small
7	2048	1024
8	2048	1024
9	Depends on the problem	

Deeper Architectures (Zhang et al., 2015)

Data Augmentation

- In text mining applications, it is not reasonable to augment the data using signal transformations as done in image or speech recognition, why?.
- One possibility is to ask humans to rephrase a word or sentence, but this does not scale well.
- In this work, they use an English thesaurus to replace words or phrases with their synonyms (Wordnet).
- Also, they convert all letters to lowercase.

Deeper Architectures (Zhang et al., 2015)

Dataset	Classes	Train Samples	Test Samples	Epoch Size
AG's News	4	120,000	7,600	5,000
Sogou News	5	450,000	60,000	5,000
DBPedia	14	560,000	70,000	5,000
Yelp Review Polarity	2	560,000	38,000	5,000
Yelp Review Full	5	650,000	50,000	5,000
Yahoo! Answers	10	1,400,000	60,000	10,000
Amazon Review Full	5	3,000,000	650,000	30,000
Amazon Review Polarity	2	3,600,000	400,000	30,000

Deeper Architectures (Zhang et al., 2015)

Model	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW	11.19	7.15	3.39	7.76	42.01	31.11	45.36	9.60
BoW TFIDF	10.36	6.55	2.63	6.34	40.14	28.96	44.74	9.00
ngrams	7.96	2.92	1.37	**4.36**	43.74	31.53	45.73	7.98
ngrams TFIDF	**7.64**	**2.81**	**1.31**	4.56	45.20	31.49	47.56	8.46
Bag-of-means	**16.91**	**10.79**	**9.55**	**12.67**	**47.46**	**39.45**	**55.87**	**18.39**
LSTM	13.94	4.82	1.45	5.26	41.83	29.16	40.57	6.10
Lg. w2v Conv.	9.92	4.39	1.42	4.60	40.16	31.97	44.40	5.88
Sm. w2v Conv.	11.35	4.54	1.71	5.56	42.13	31.50	42.59	6.00
Lg. w2v Conv. Th.	9.91	-	1.37	4.63	39.58	31.23	43.75	5.80
Sm. w2v Conv. Th.	10.88	-	1.53	5.36	41.09	29.86	42.50	5.63
Lg. Lk. Conv.	8.55	4.95	1.72	4.89	40.52	29.06	45.95	5.84
Sm. Lk. Conv.	10.87	4.93	1.85	5.54	41.41	30.02	43.66	5.85
Lg. Lk. Conv. Th.	8.93	-	1.58	5.03	40.52	28.84	42.39	5.52
Sm. Lk. Conv. Th.	9.12	-	1.77	5.37	41.17	28.92	43.19	5.51
Lg. Full Conv.	9.85	8.80	1.66	5.25	38.40	29.90	40.89	5.78
Sm. Full Conv.	11.59	8.95	1.89	5.67	38.82	30.01	40.88	5.78
Lg. Full Conv. Th.	9.51	-	1.55	4.88	38.04	29.58	40.54	5.51
Sm. Full Conv. Th.	10.89	-	1.69	5.42	**37.95**	29.90	40.53	5.66
Lg. Conv.	12.82	4.88	1.73	5.89	39.62	29.55	41.31	5.51
Sm. Conv.	15.65	8.65	1.98	6.53	40.84	29.84	40.53	5.50
Lg. Conv. Th.	13.39	-	1.60	5.82	39.30	**28.80**	40.45	**4.93**
Sm. Conv. Th.	14.80	-	1.85	6.49	40.16	29.84	**40.43**	5.67

Error rates in several benckmark datasets. Red: highest error; Blue: lowest error.

Notice that dataset size increases from left to right.