Convolutional Neural Networks

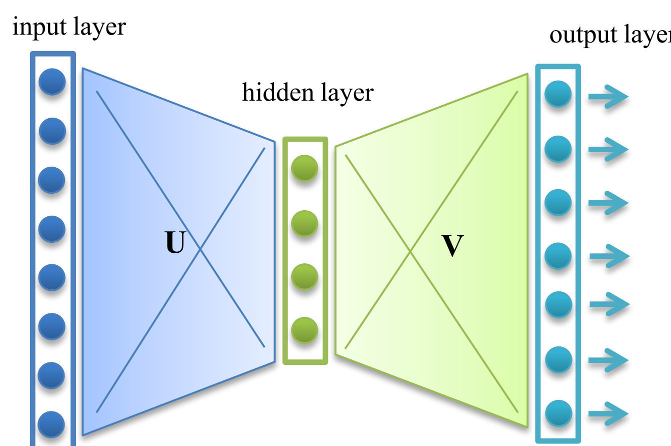
Kuan-Yu Chen (陳冠宇)

2018/04/26 @ NTUST

Parsimonious Neural Networks

- For image processing, the conventional neural networks have to estimate too many parameters
  - For a 1024*1024 image, the size of the input layer is up to 1,048,576
  - If the size of the first hidden layer is 100, the number of model parameter is over 104,857,600

1,024*1,024



Convolution Neural Networks – 1.

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum

1	0	1	1
0	1	0	0
1	0	0	1
1	0	1	1
0	0	0	1

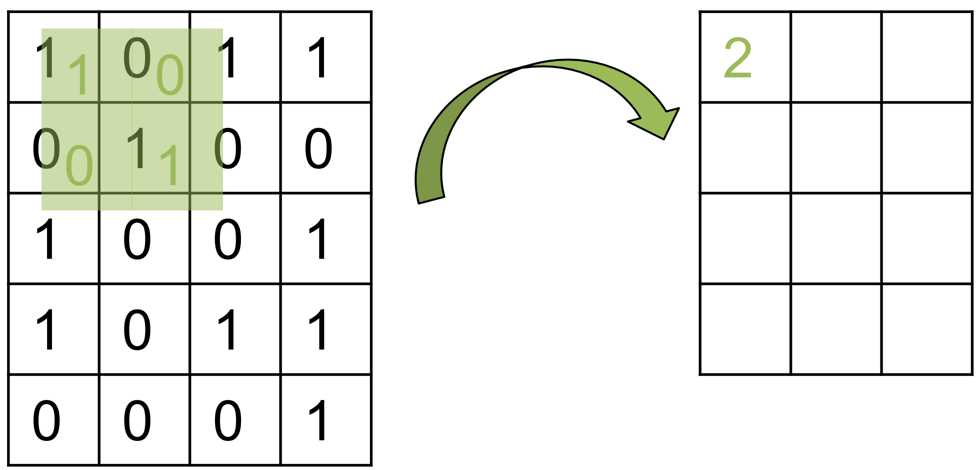
Filter or Kernel

1	0
0	1

1	0	1
0	1	0
1	0	0

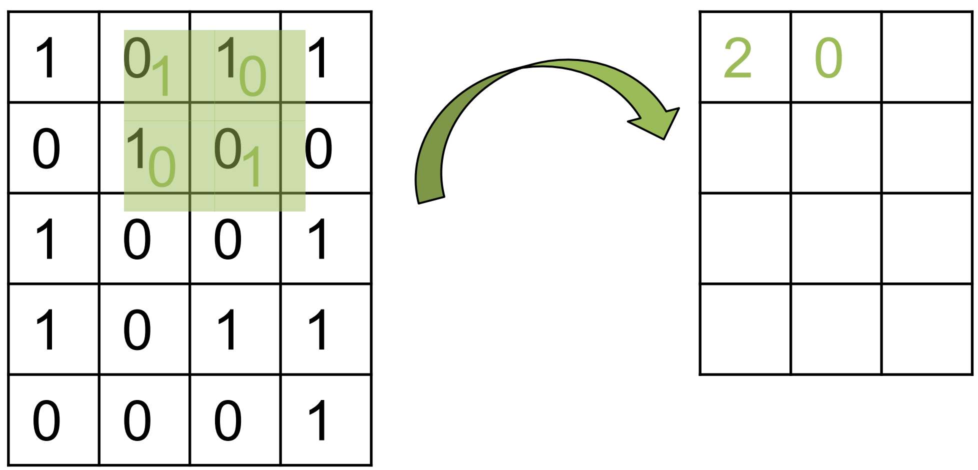
Convolution Neural Networks – 1..

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



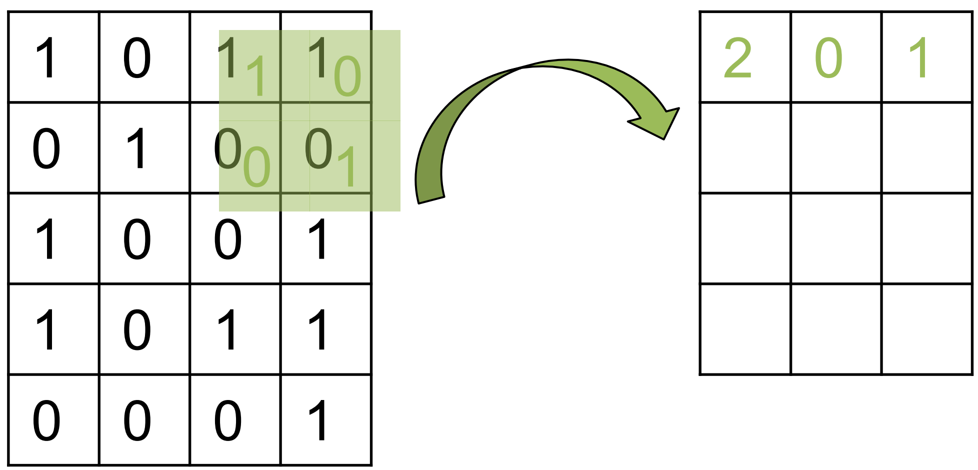
Convolution Neural Networks – 1…

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



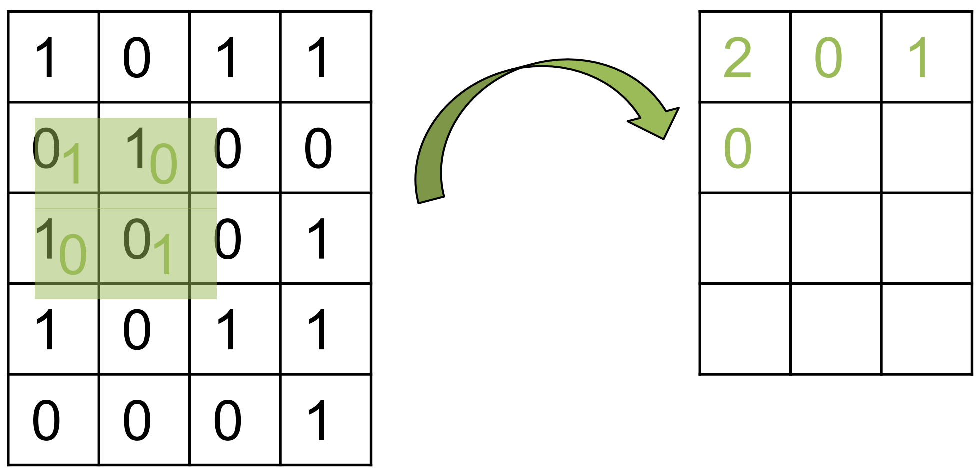
Convolution Neural Networks – 1….

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



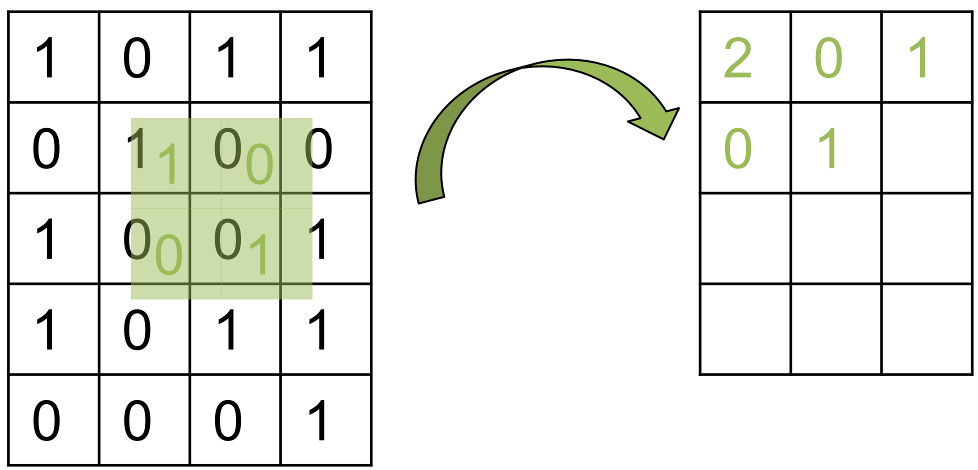
Convolution Neural Networks – 1…..

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



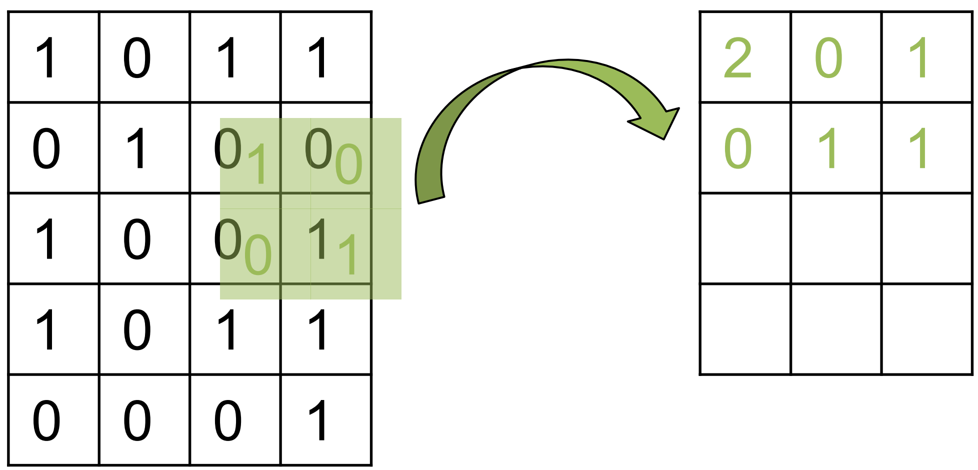
Convolution Neural Networks – 1……

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



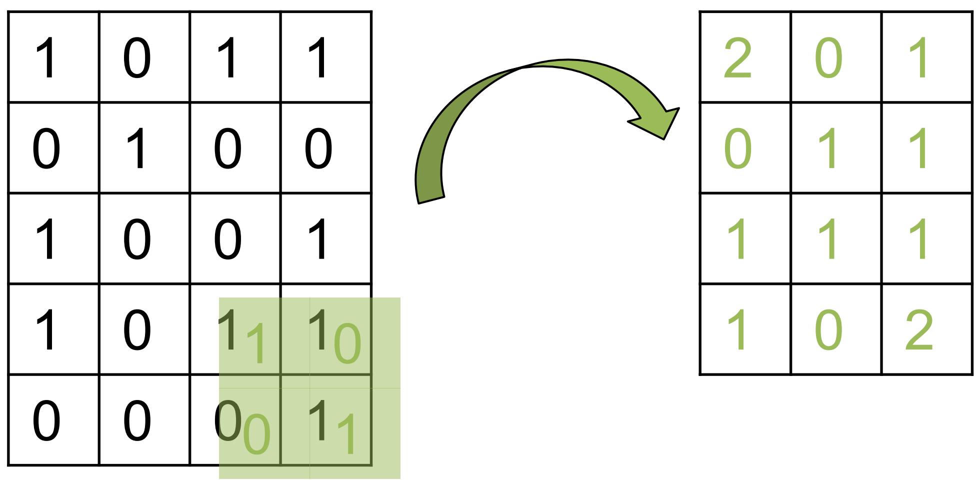
Convolution Neural Networks – 1…….

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



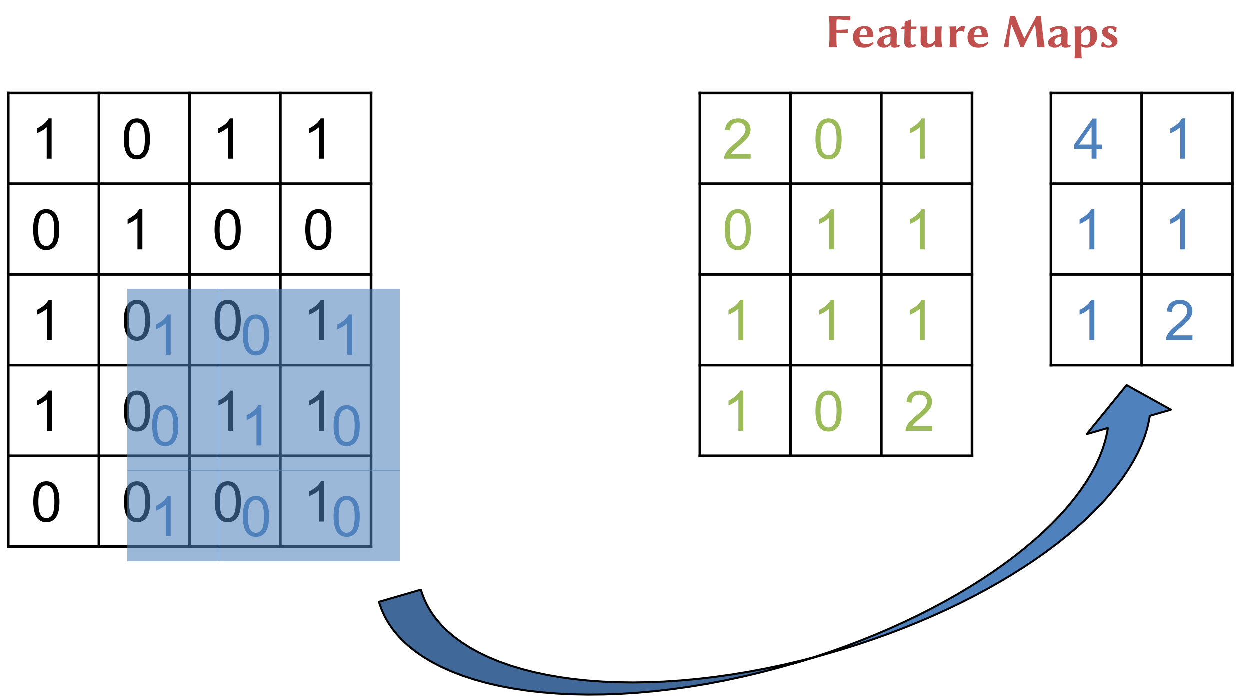
Convolution Neural Networks – 1……..

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



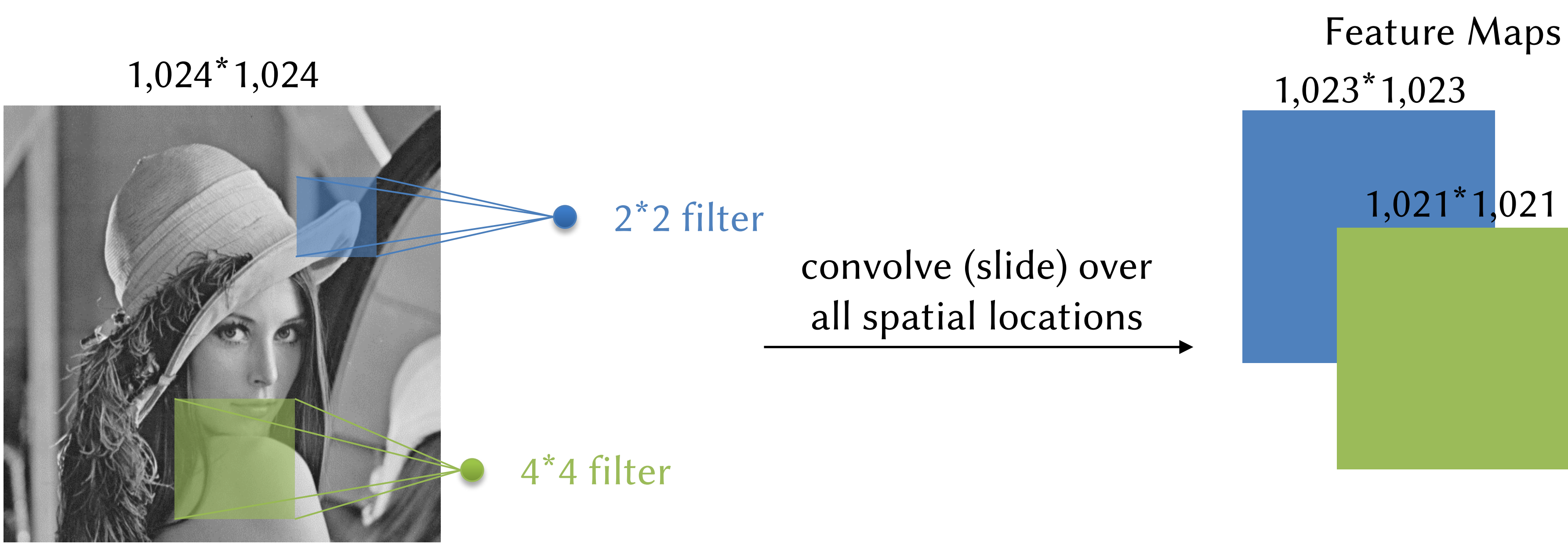
Convolution Neural Networks – 1………

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum



Convolution Neural Networks – 2

- Inspired from the visual cortex, each neuron can only perceive a sub-region (perceptive field) at a time
  - **Convolve** the filter with the image
  - Convolution = Element-wise Product then Sum
  - If we have two filters (2*2 and 4*4), the total parameters are 4+16=20
    - Parameters Sharing!



Convolution Neural Networks – 3

- CNN is a special case of DNN

Input

1	0	1	1
0	1	0	0
1	0	0	1
1	0	1	1
0	0	0	1

Filters

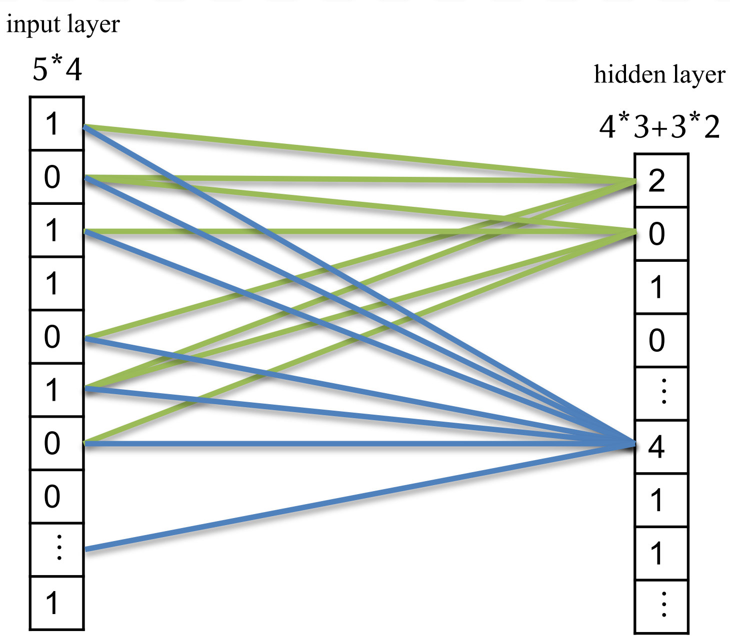
1	0
0	1

1	0	1
0	1	0
1	0	0

Feature Maps

2	0	1
0	1	1
1	1	1
1	0	2

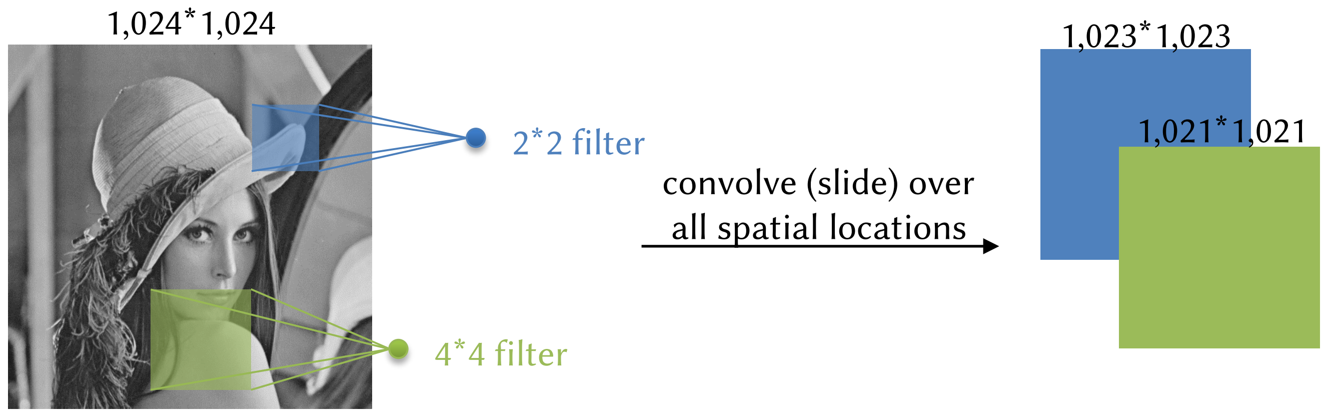
4	1
1	1
1	2

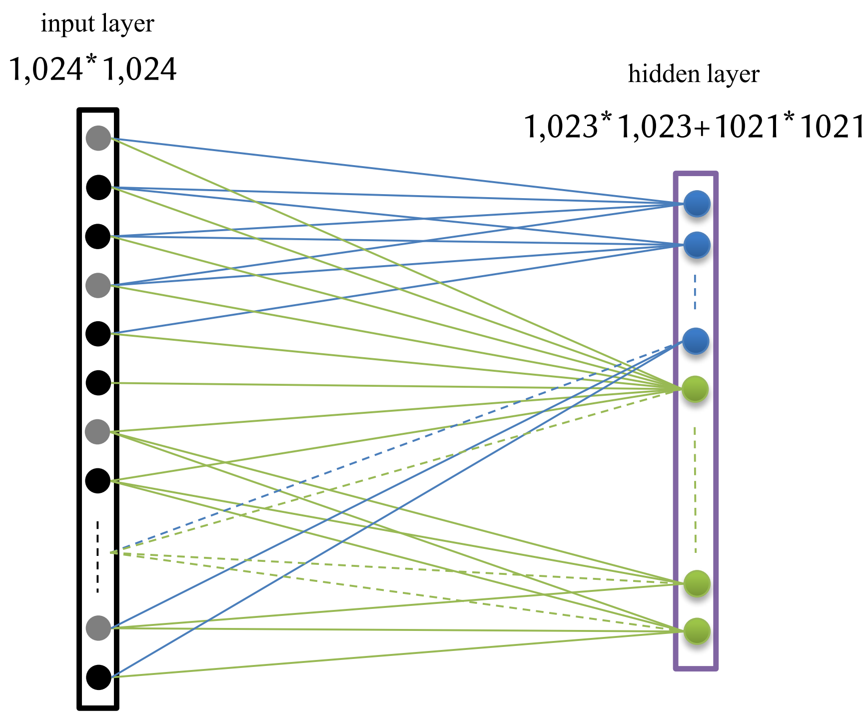


$$(5^*4)^*(4^*3+3^*2)$$

$$2^*2+3^*3$$

Convolution Neural Networks – 4



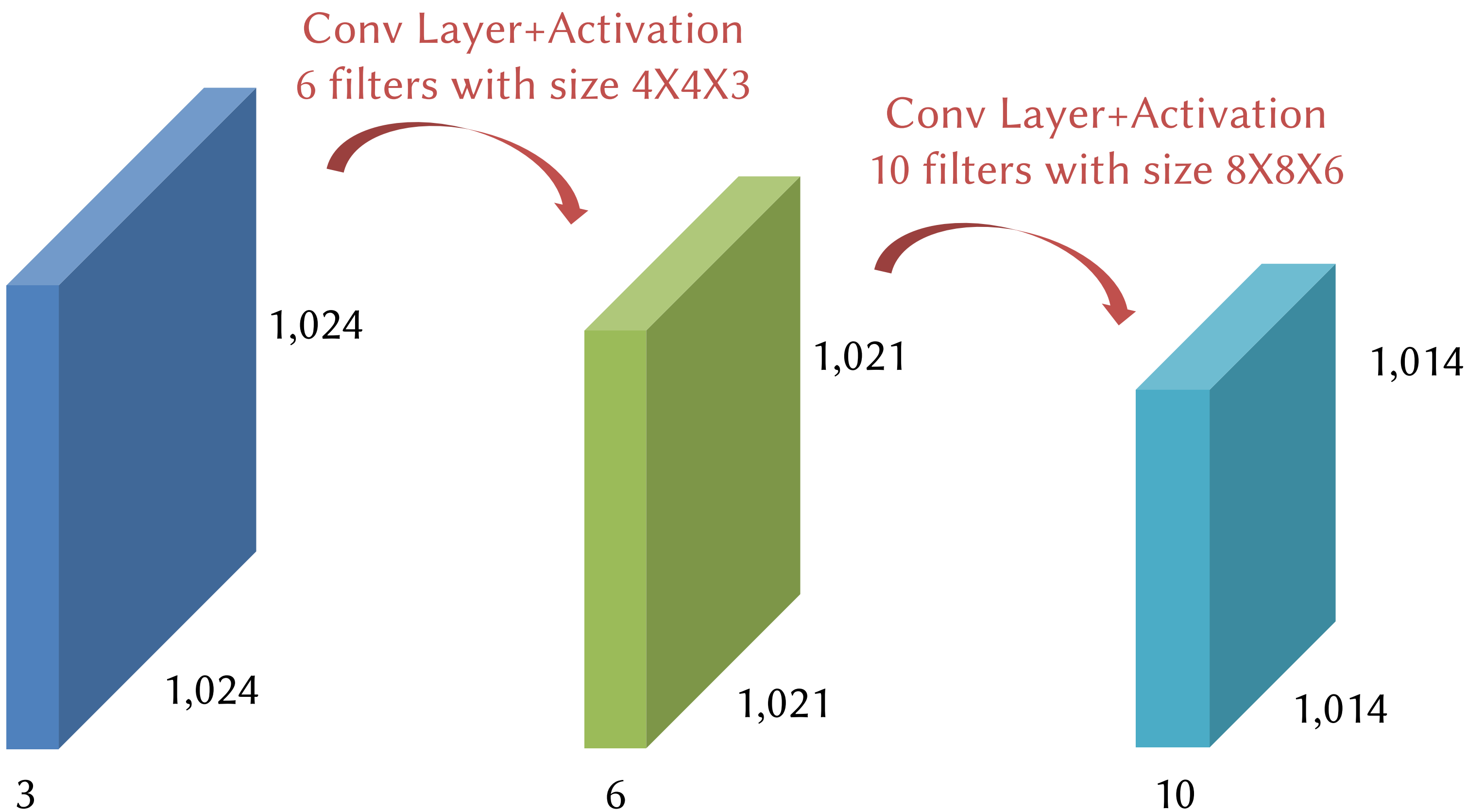


$$(1024*1024)*(1023*1023+1021*1021)$$

$$2*2+4*4$$

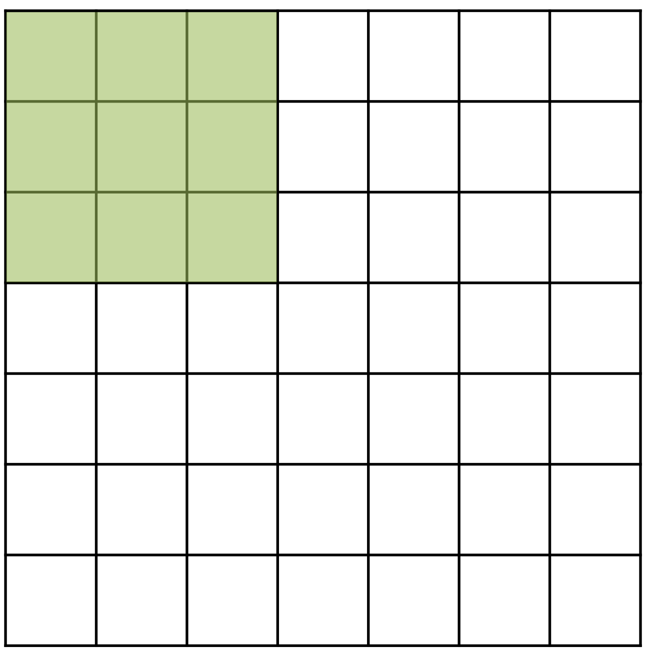
ConvNet

- Convolutional neural networks also call ConvNets or CNNs
  - It is a sequence of convolutional layers, interspersed with activation functions



Pooling & Stride

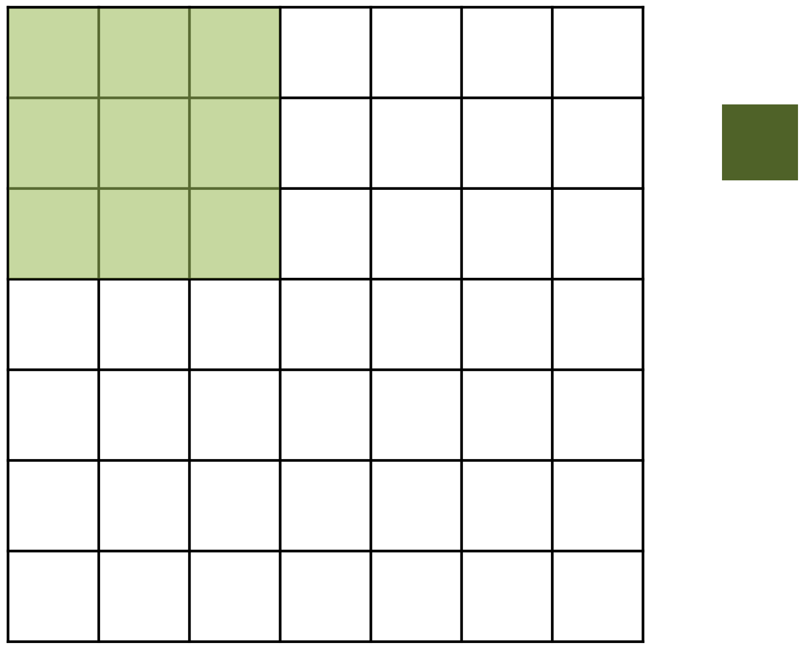
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride.

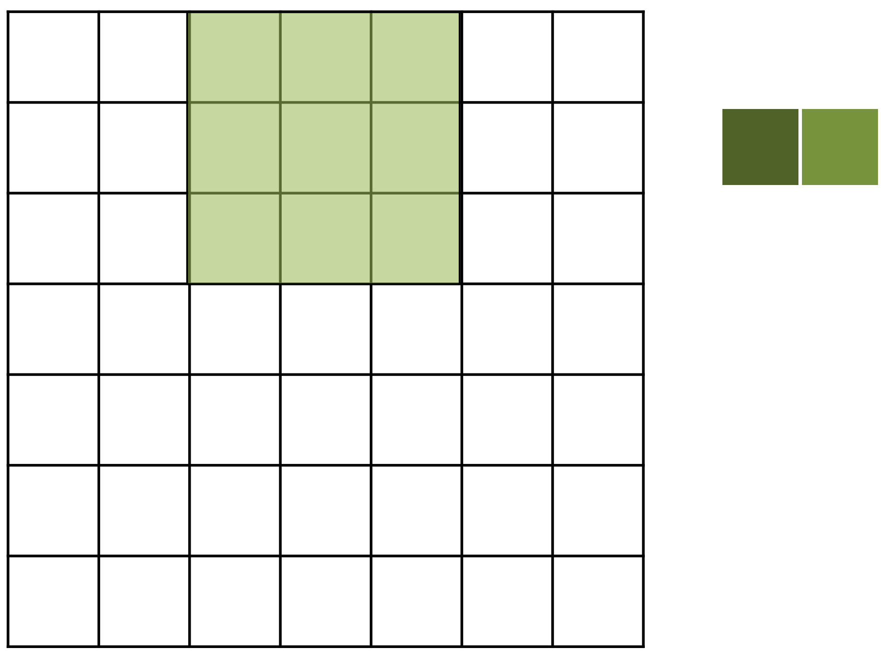
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride..

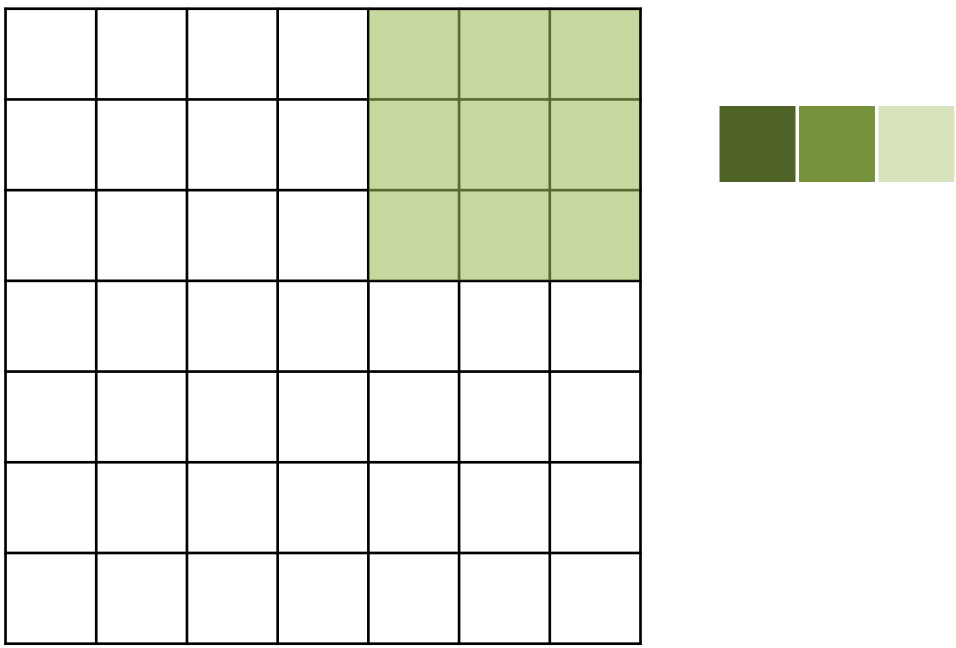
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride...

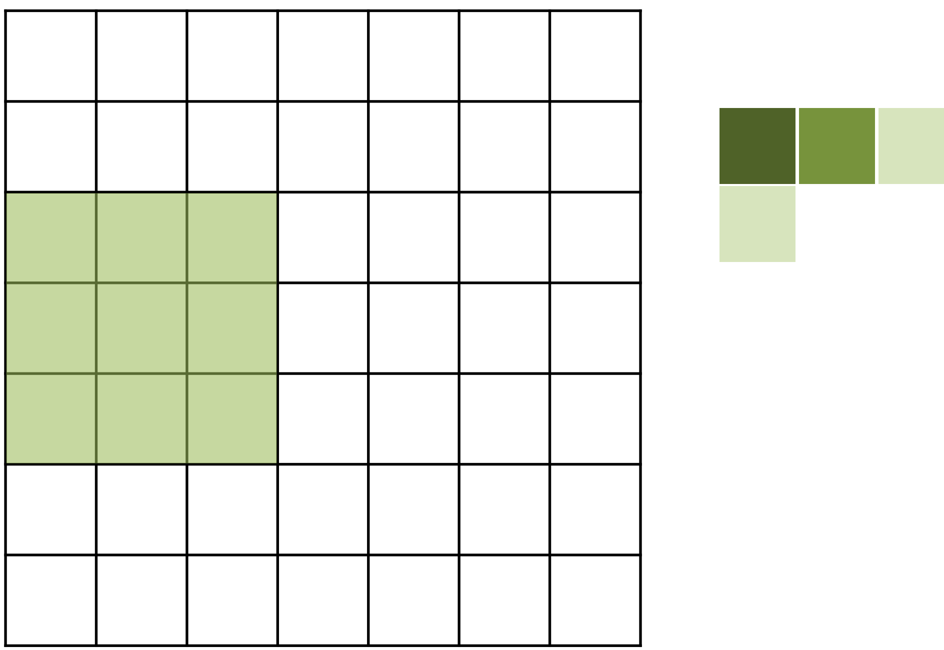
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride....

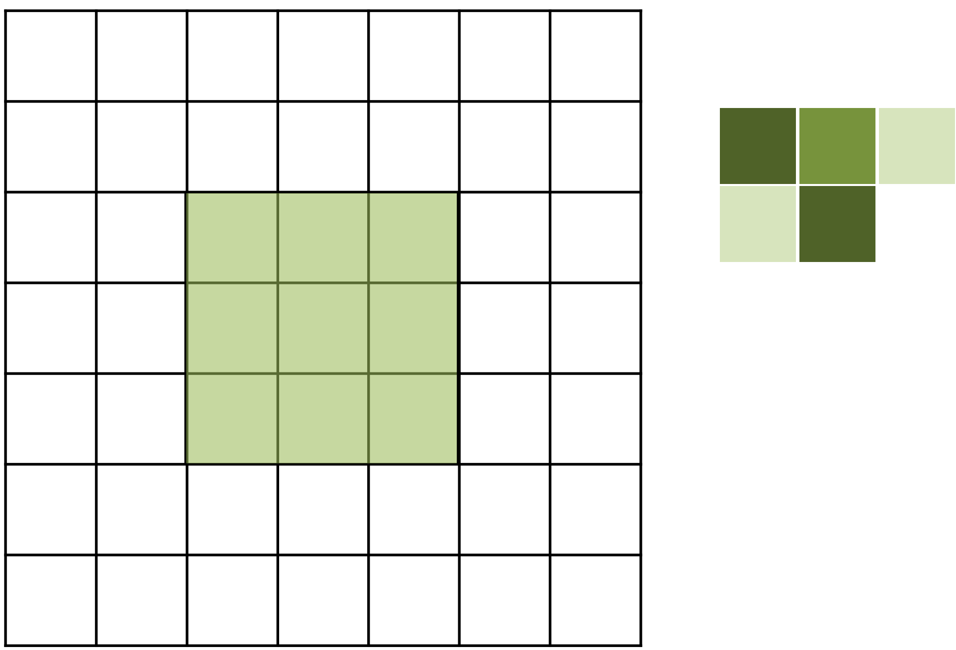
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride.....

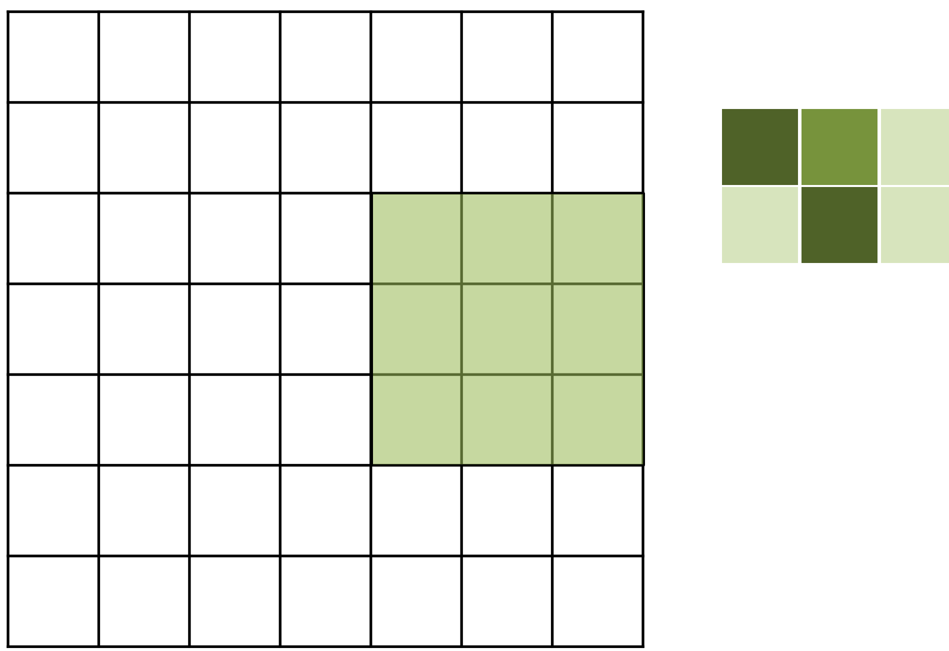
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride……

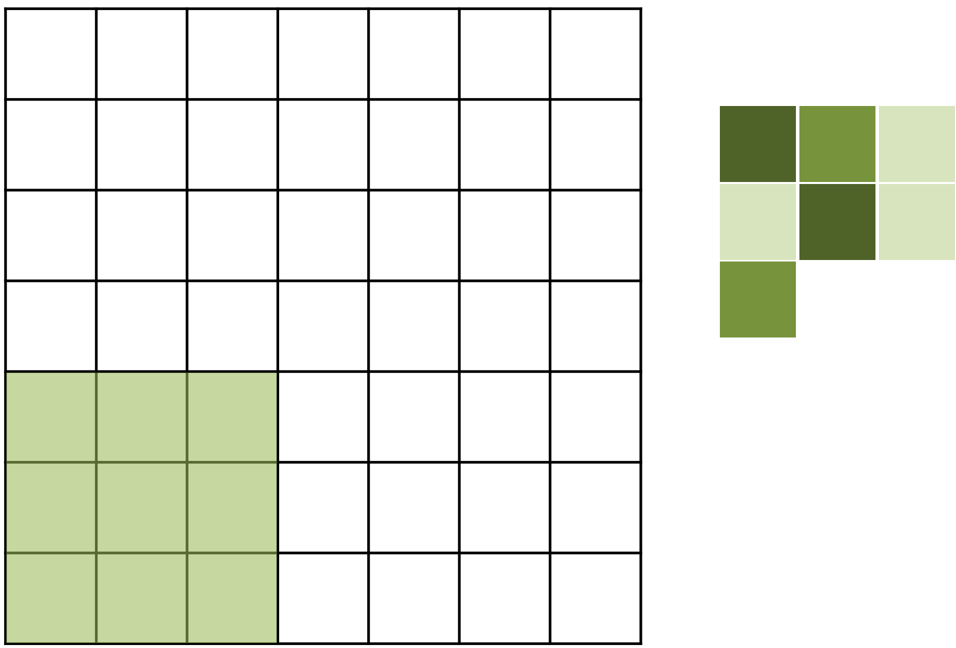
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$



7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride…….

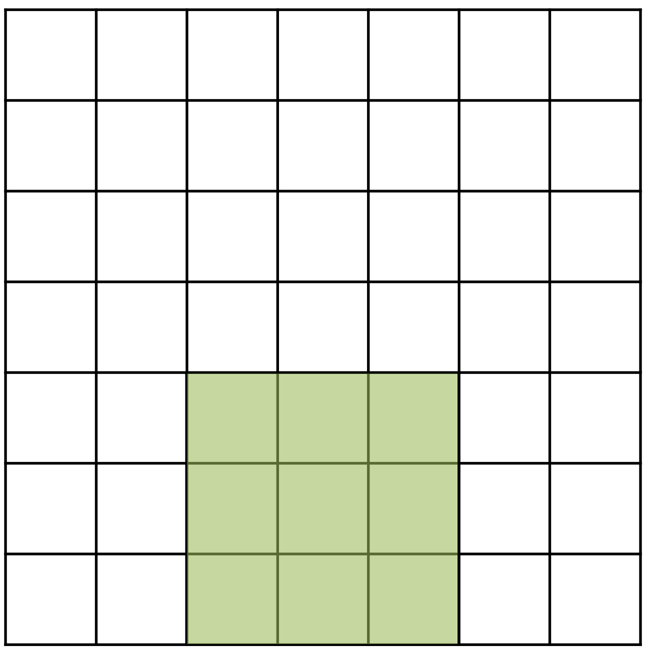
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$

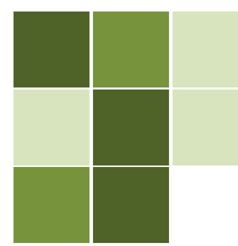


7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride……..

- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$

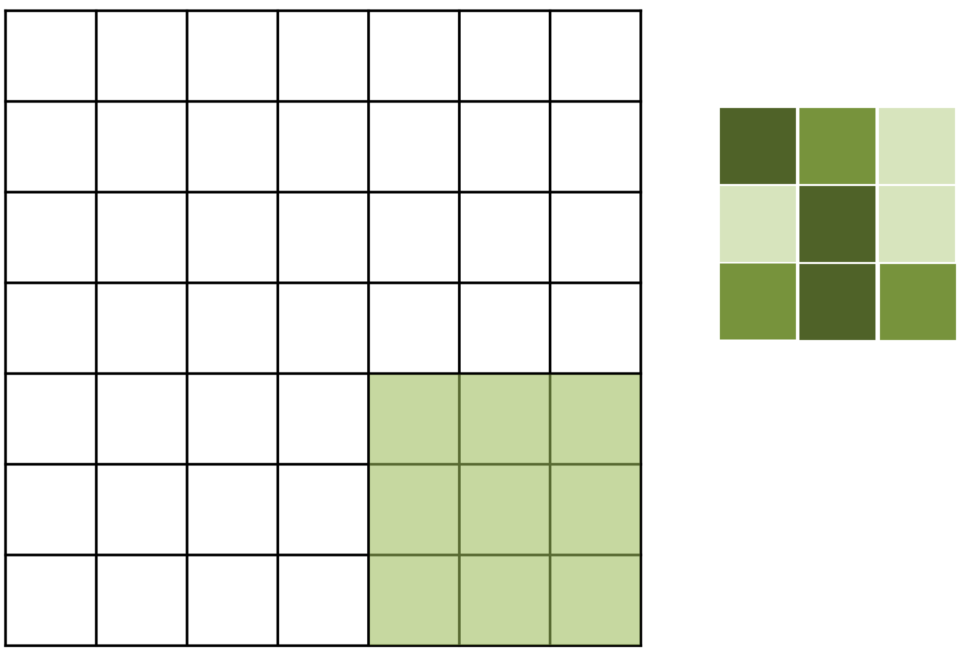




7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride………

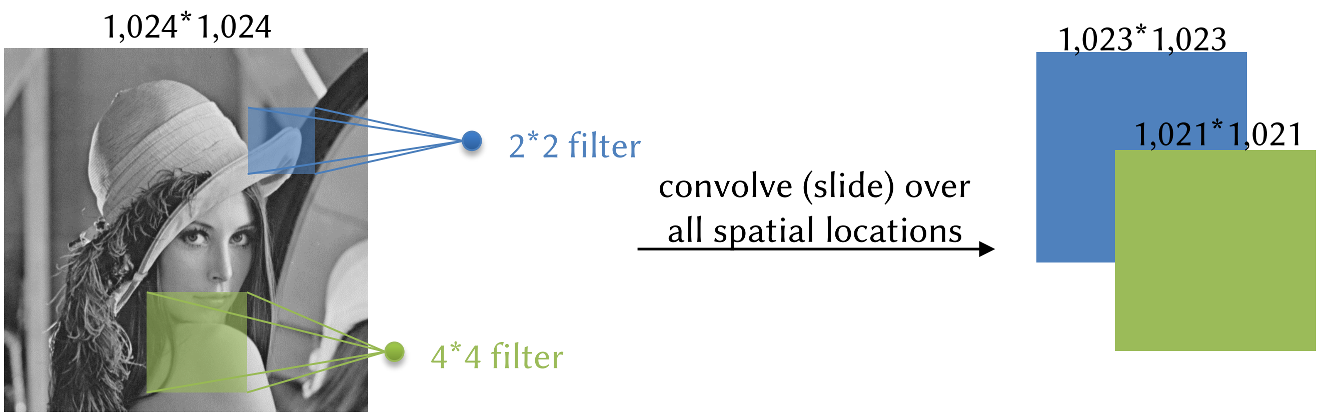
- Although the model parameters can be reduced, the feature dimension is still very large
  - Pooling
  - Stride
    - $Output\ Size = \frac{(Input\ Size - Filter\ Size)}{Stride\ Size} + 1$

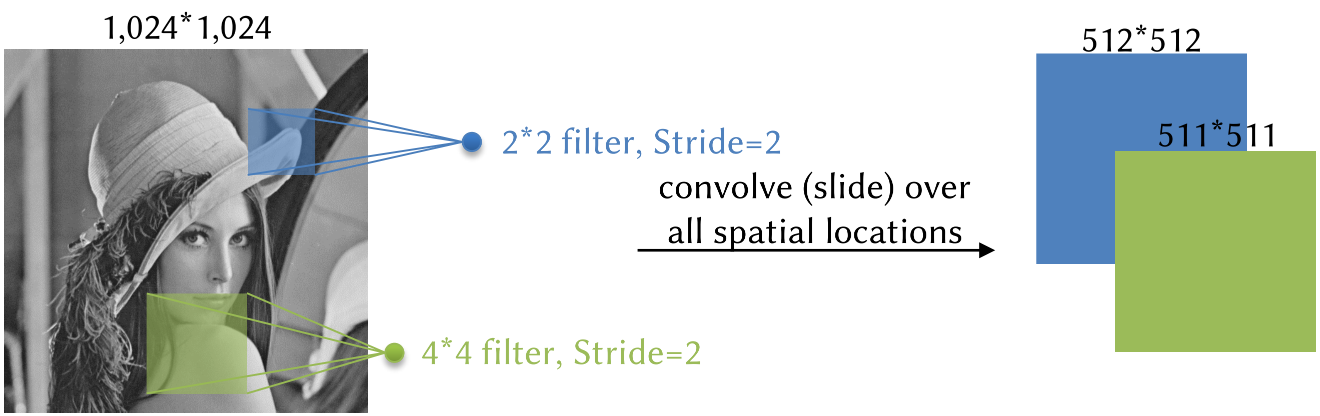


7x7 input
3x3 filter
stride 2
=> 3x3 output!

Stride

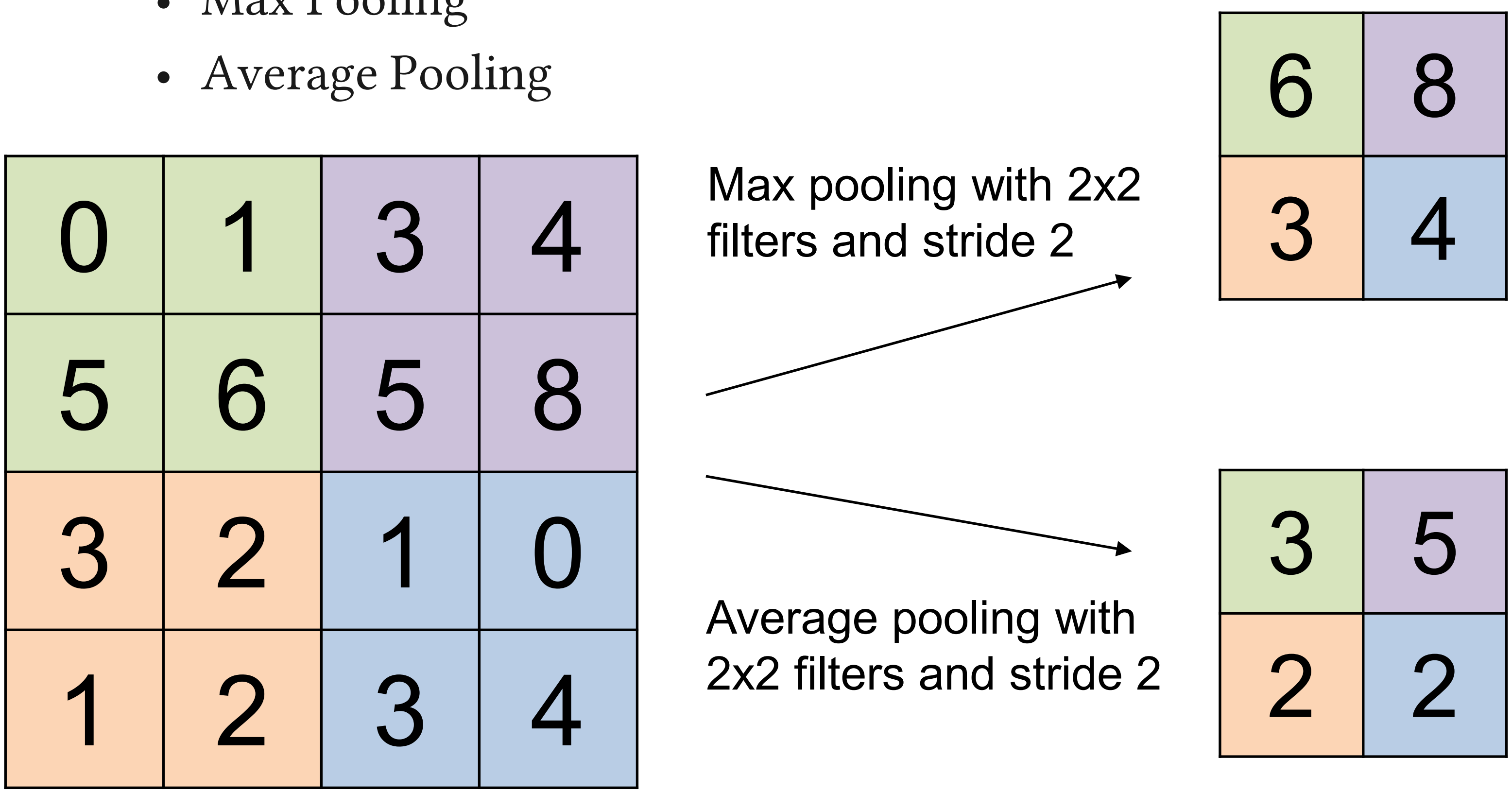
$$Output\ Size = \frac{(Input\ Size\ - Filter\ Size)}{Stride\ Size} + 1$$





Pooling

- Pooling can make the representations smaller and more manageable
  - It operates over each feature map independently
    - Max Pooling
    - Average Pooling



Padding & Color Image

Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	2	2	1	0	2	0
0	2	0	2	1	2	0
0	0	2	2	2	0	0
0	0	1	0	1	0	0
0	2	0	0	1	2	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	2	0	1	1	0
0	2	2	2	1	1	0
0	1	1	0	2	0	0
0	0	0	0	1	0	0
0	2	2	1	2	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	0	2	0	2	0
0	1	1	2	0	0	0
0	0	2	1	1	1	0
0	1	1	0	0	2	0
0	1	1	2	1	0	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

1	0	0
-1	1	1
0	-1	0

w0[:,:,1]

1	-1	0
-1	1	-1
0	-1	-1

w0[:,:,2]

1	-1	1
0	1	0
0	-1	-1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

-1	-1	-1
-1	1	1
1	-1	0

w1[:,:,1]

1	-1	1
0	1	-1
0	-1	1

w1[:,:,2]

-1	1	1
-1	1	0
1	-1	-1

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

-1	-8	2
-1	-1	-3
4	3	1

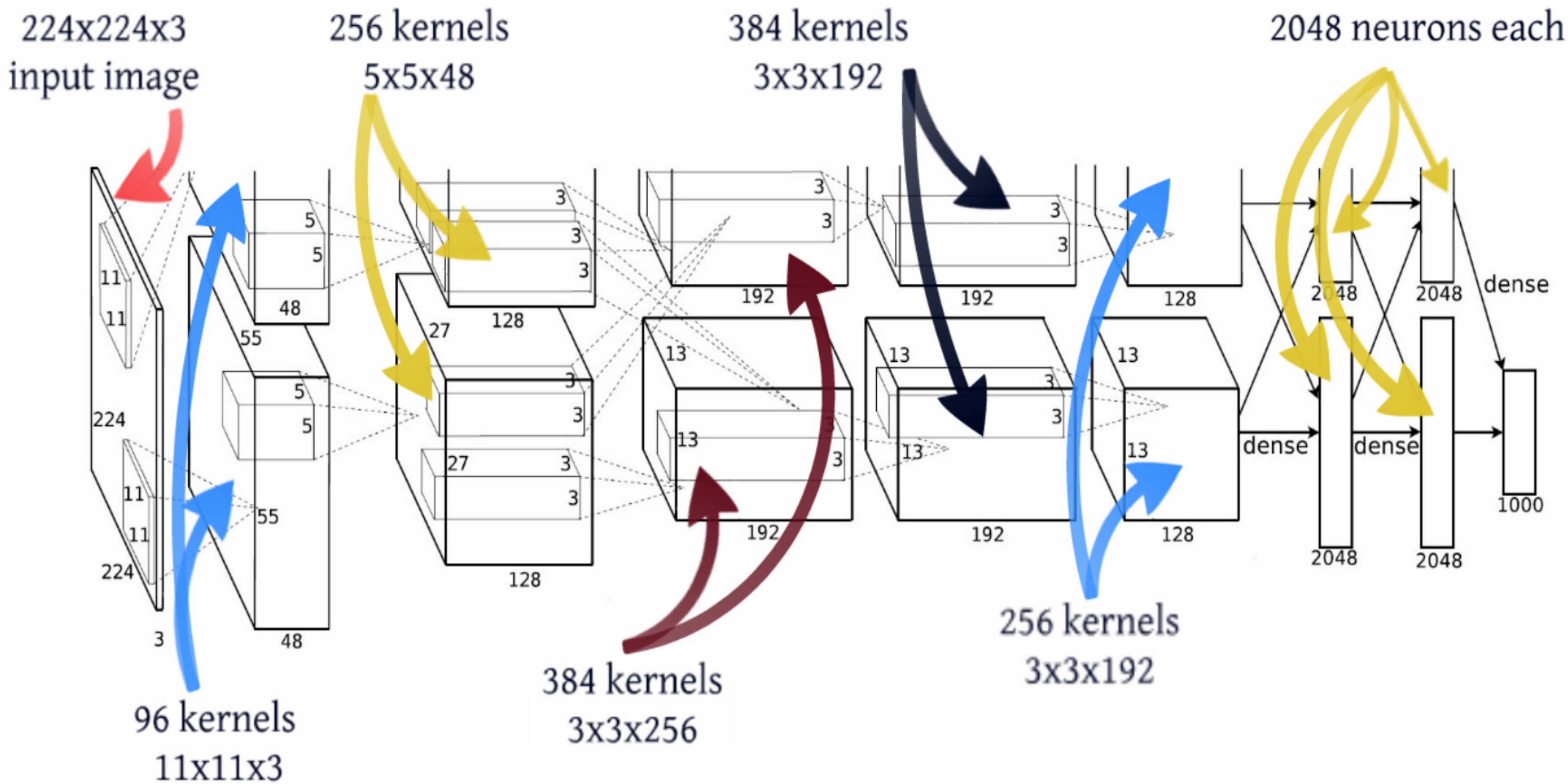
o[:,:,1]

2	-4	3
0	1	-6
4	-1	3

toggle movement

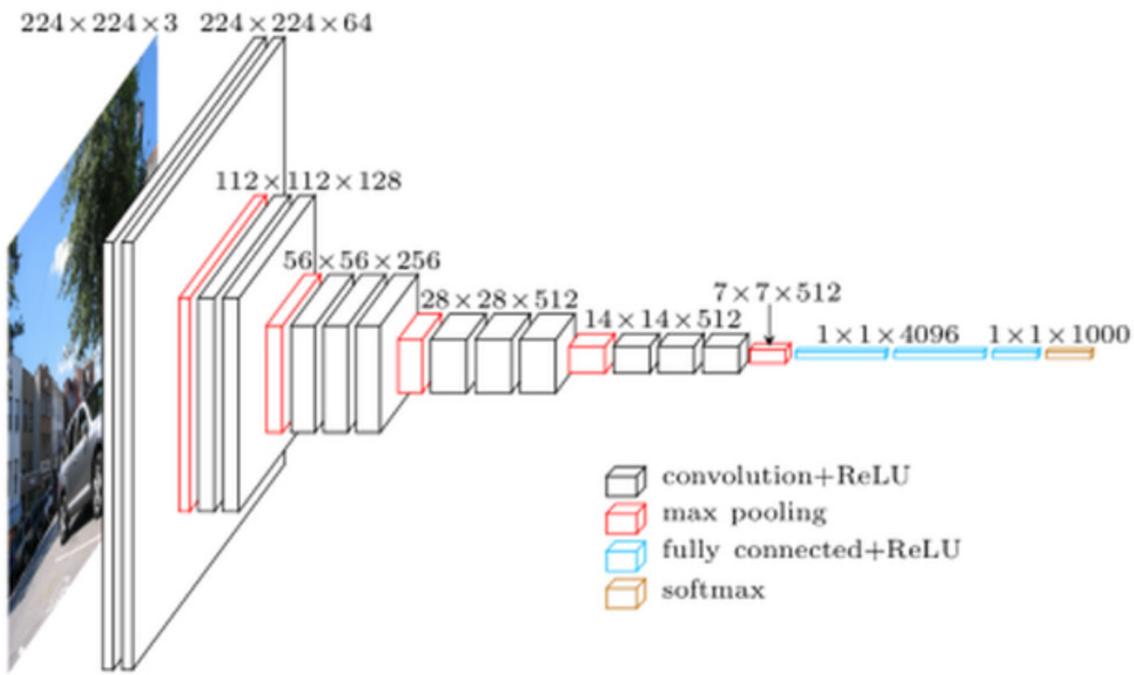
http://vision.stanford.edu/teaching/cs231b_spring1415/slides/alexnet_tugce_kyunghee.pdf

AlexNet



https://openclassrooms.com/courses/classez-et-segmentez-des-donnees-visuelles/tp-implementez-votre-premier-reseau-de-neurones-avec-keras

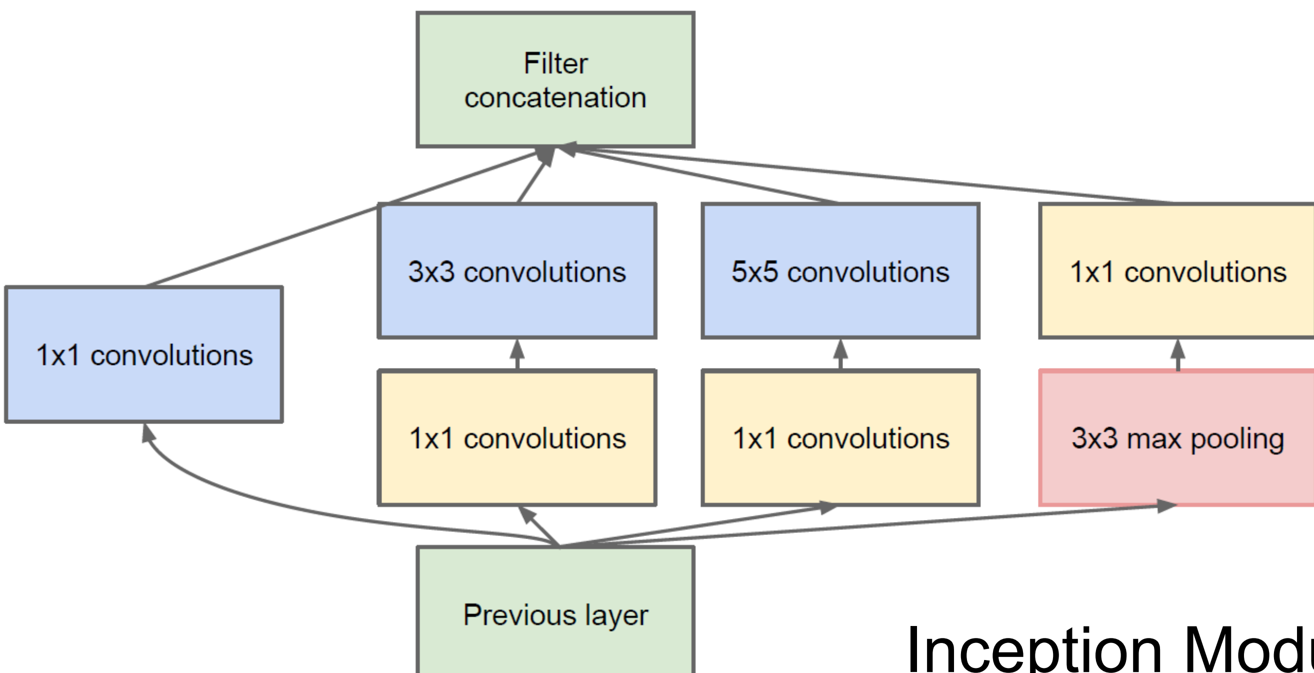
VGGNet



C. Szegedy, et al., “Going deeper with convolutions,” *arXiv:1409.4842*, 2014.

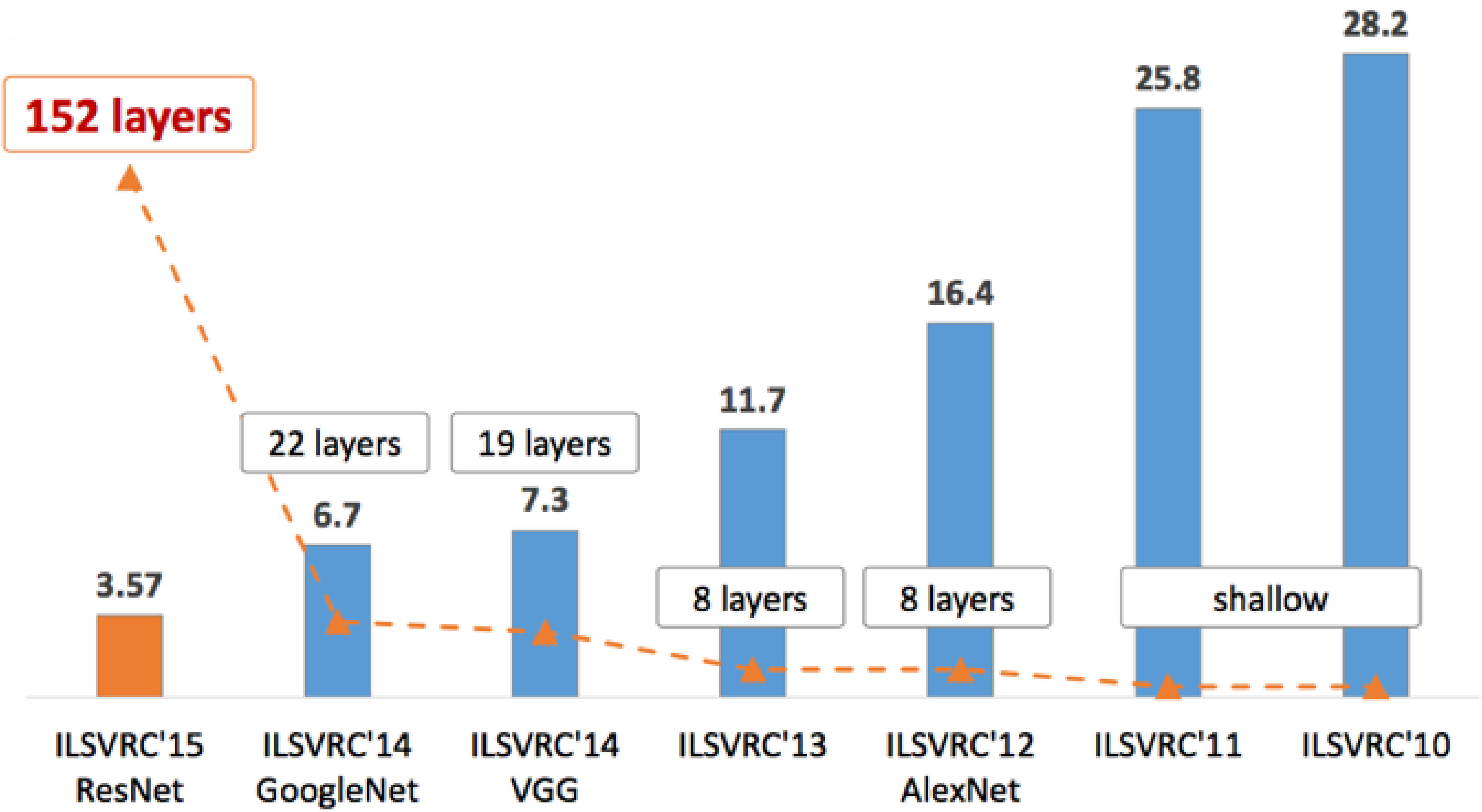
GoogLeNet

Convolution
Pooling
Softmax
Other



Inception Module

Comparisons



CNN in NLP

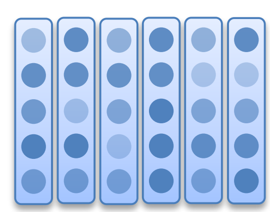
- A document is a sequence of words

there are books on the table

- Each word can be represented by a word embedding

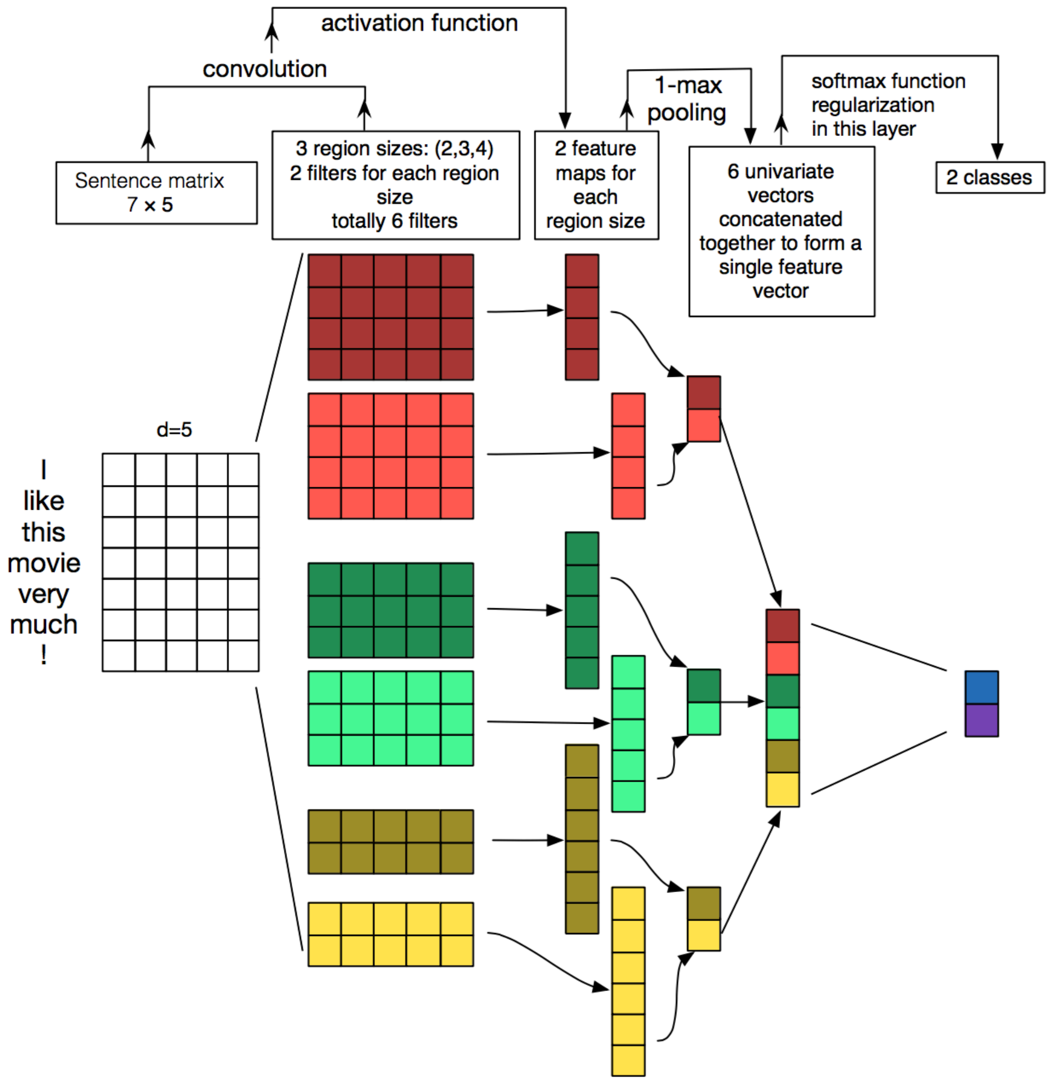
there are books on the table

- The document can be viewed as a image/matrix
  - Apply CNN!



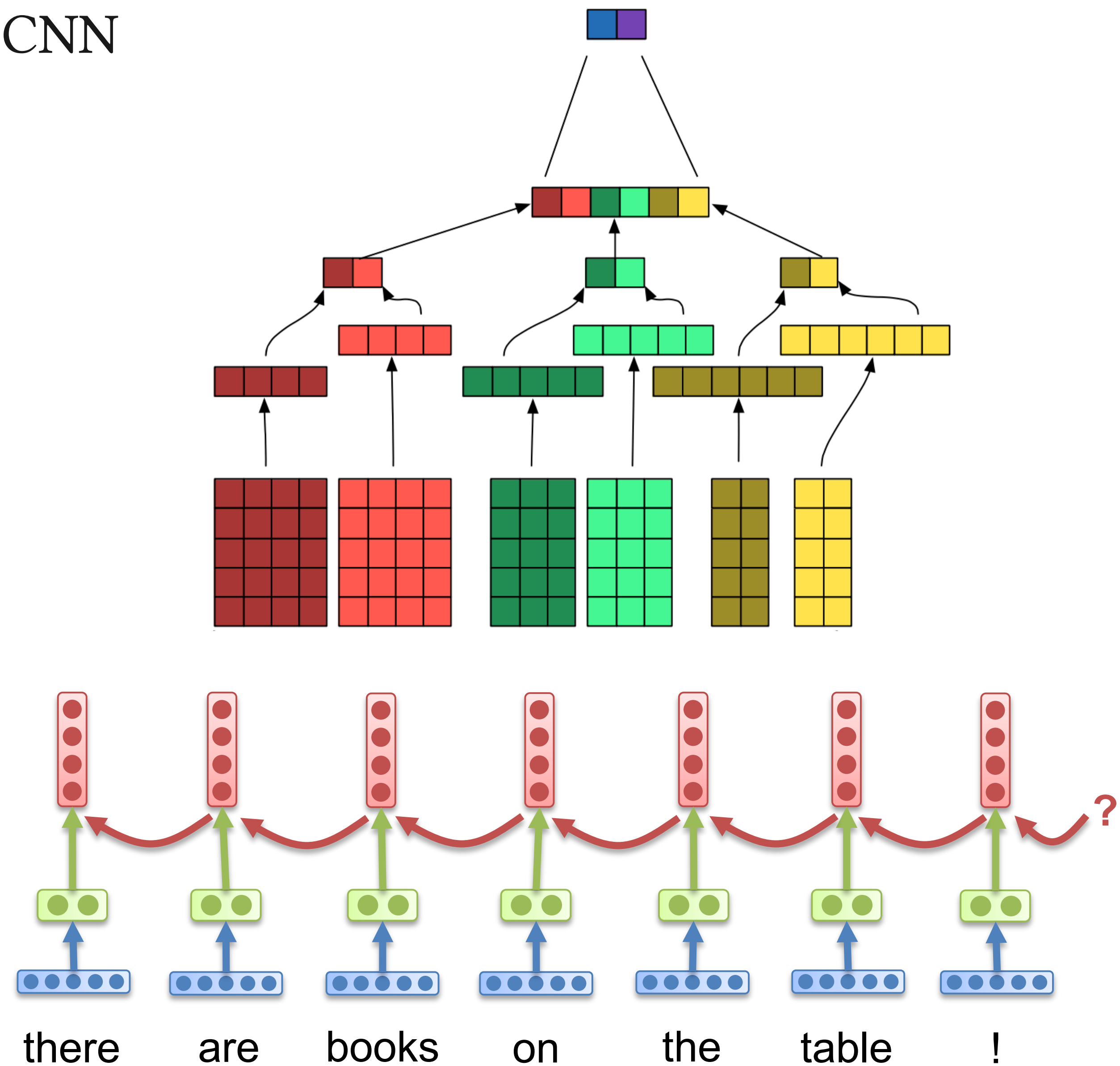
CNN for NLP – 1

- Using CNN to extract N-gram pattern



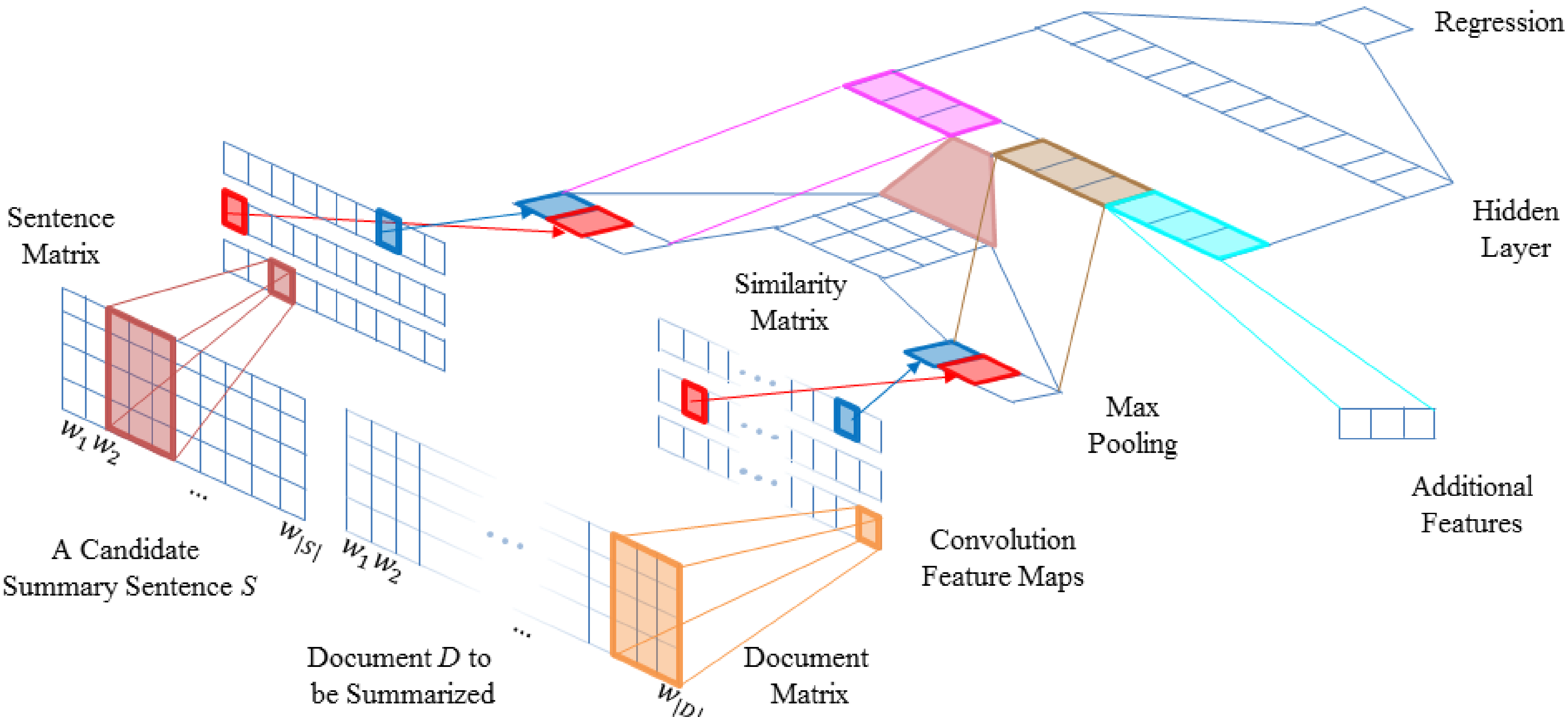
CNN for NLP – 2

- RNN+CNN

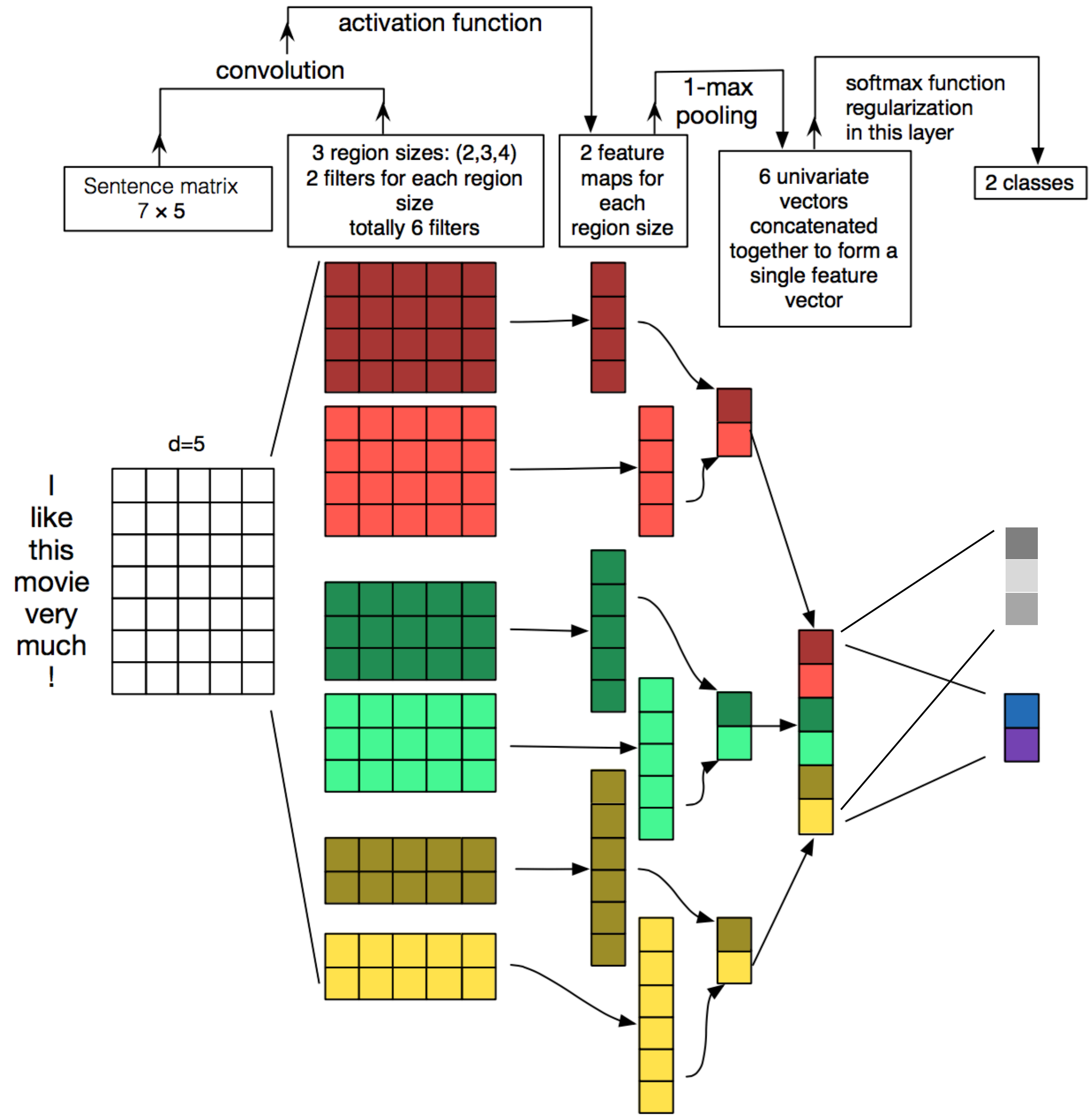


CNN for NLP – 3

- For Summarization



Multi-task Learning



Questions?

kychen@mail.ntust.edu.tw