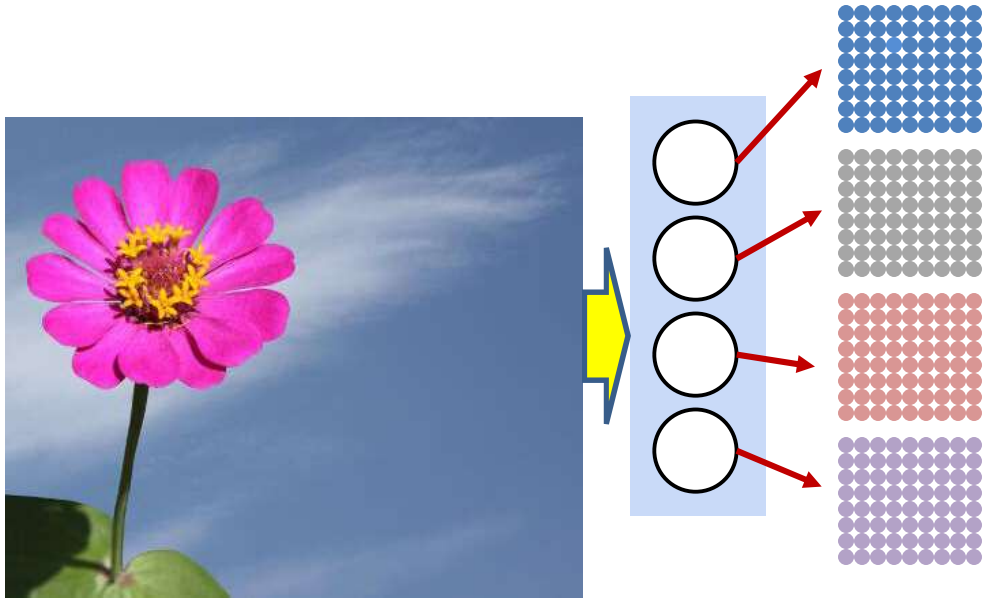
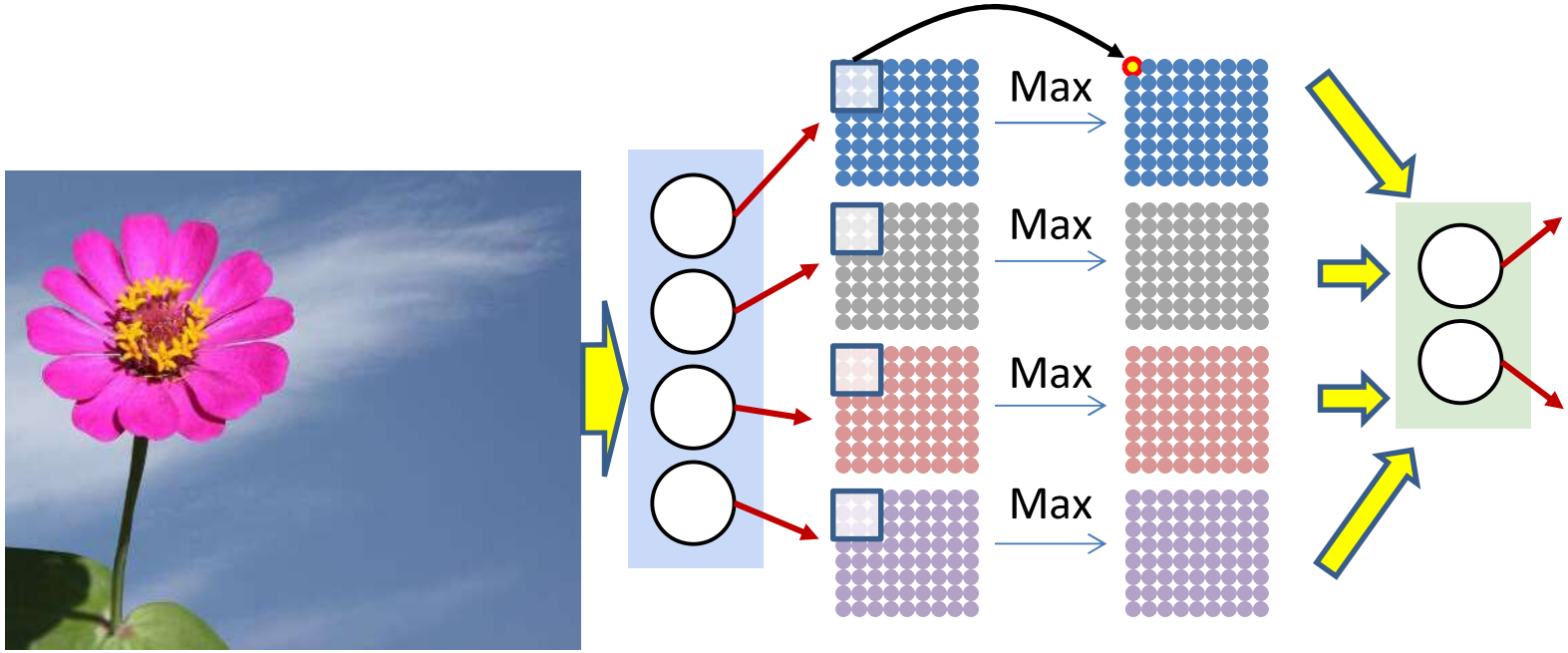
Modification 2: Accounting for jitter



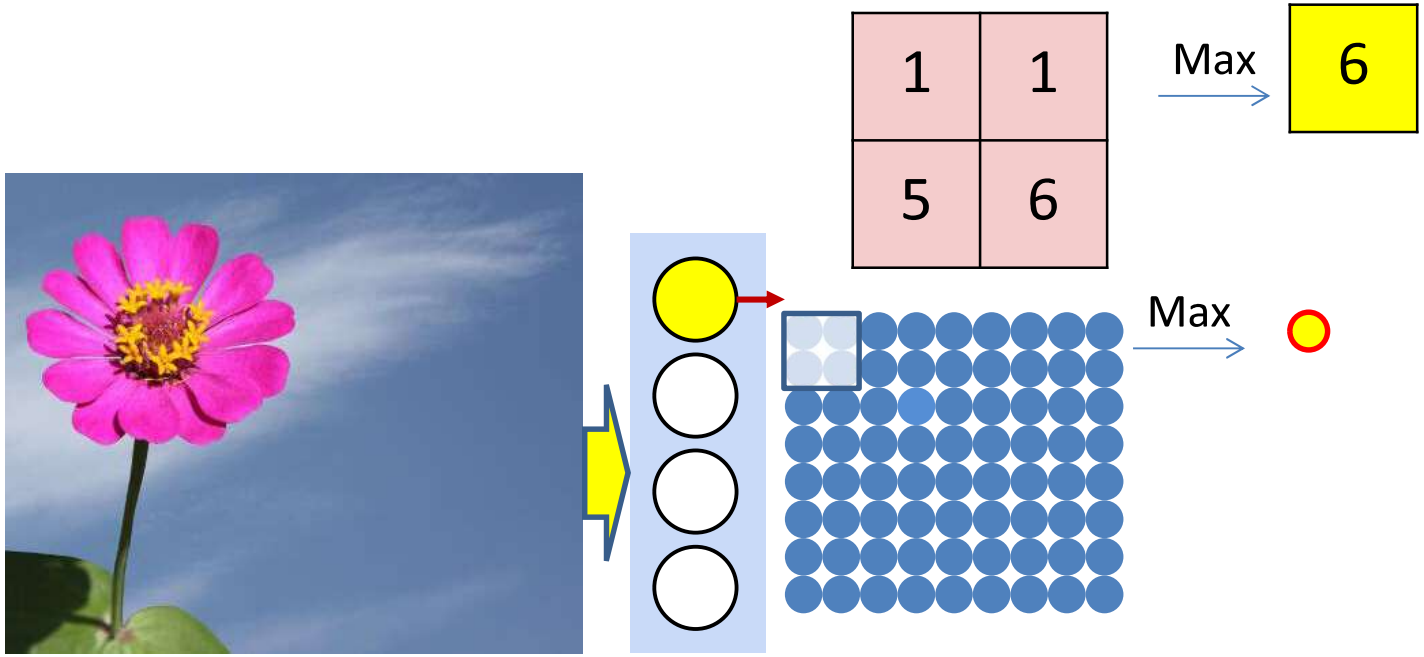
- We would like to account for some jitter in the first-level patterns
  - If a pattern shifts by one pixel, is it still a petal?

Accounting for jitter



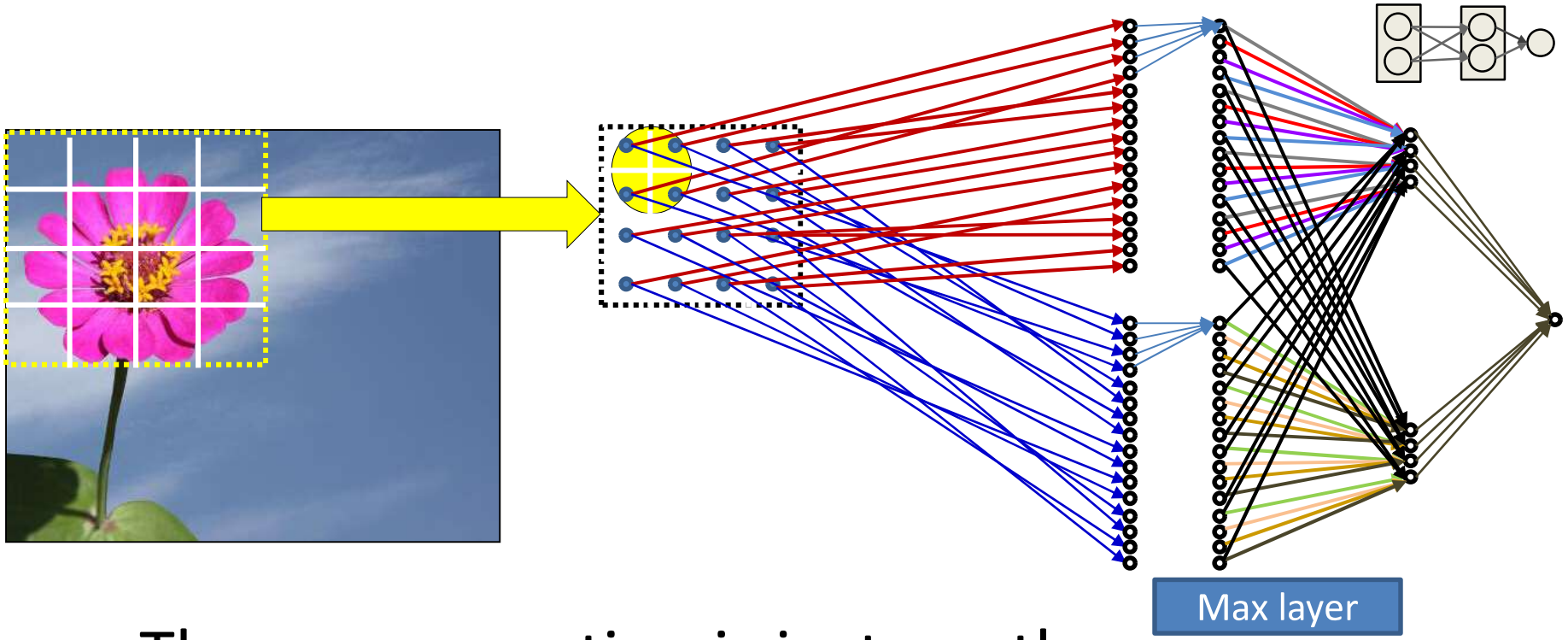
- We would like to account for some jitter in the first-level patterns
  - If a pattern shifts by one pixel, is it still a petal?
  - A small jitter is acceptable
    - Replace each value by the maximum of the values within a small region around it
      - ***Max filtering or Max pooling***

Max filter



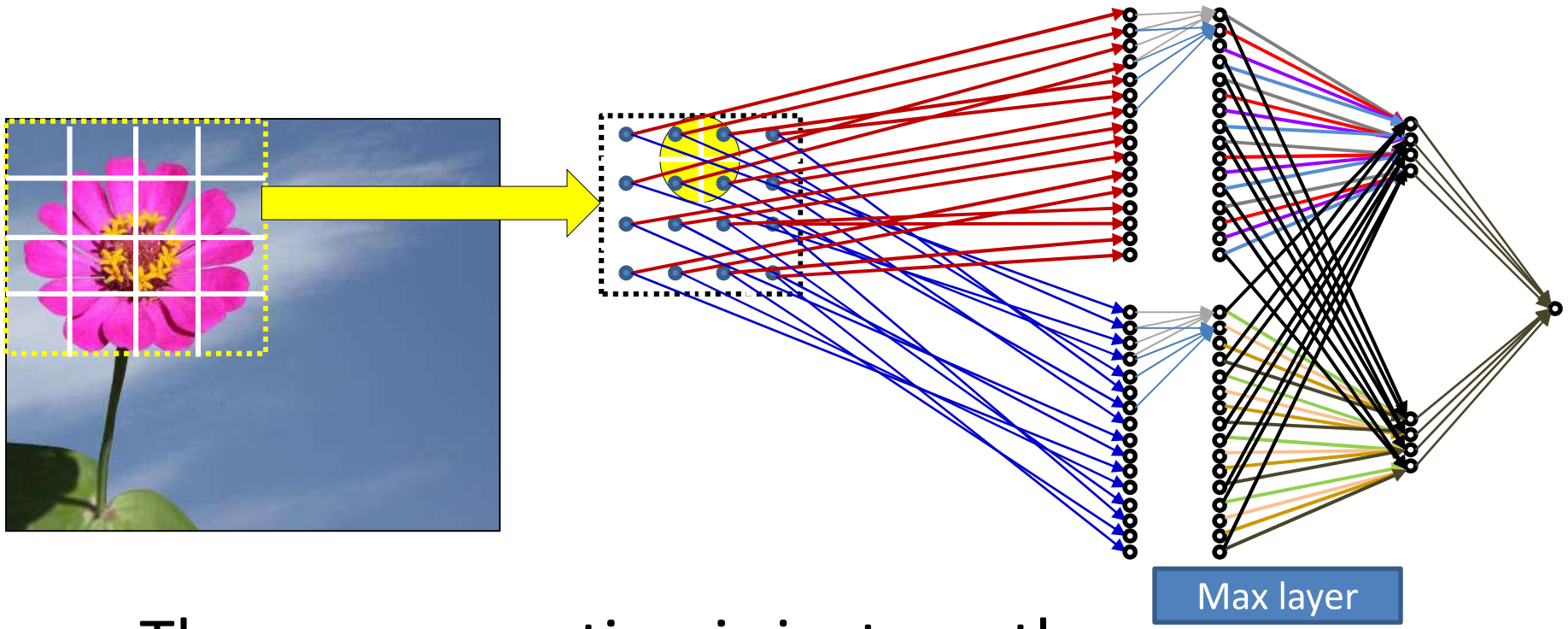
- We would like to account for some jitter in the first-level patterns
  - If a pattern shifts by one pixel, is it still a petal?
  - A small jitter is acceptable
    - Replace each value by the maximum of the values within a small region around it
      - ***Max filtering*** or ***Max pooling***

The max operation is just a neuron



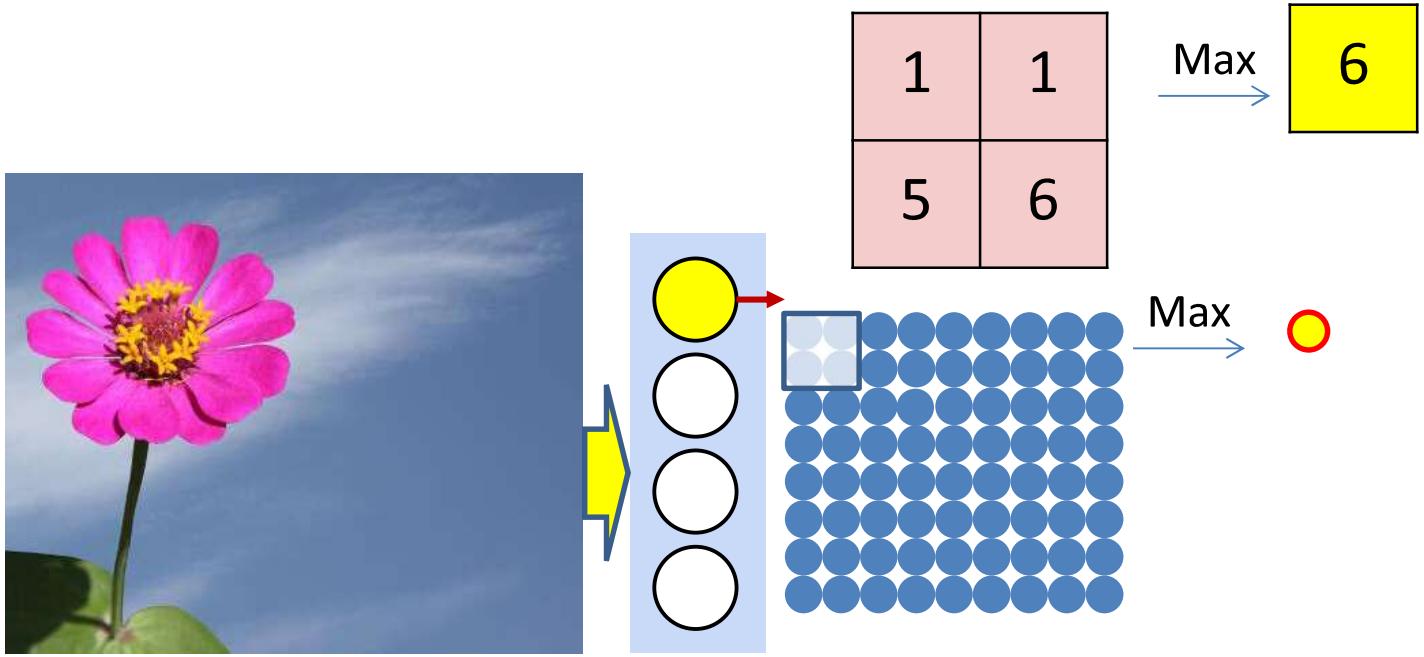
- The max operation is just another neuron
- Instead of applying an activation to the weighted sum of inputs, each neuron just computes the maximum over all inputs

The max operation is just a neuron



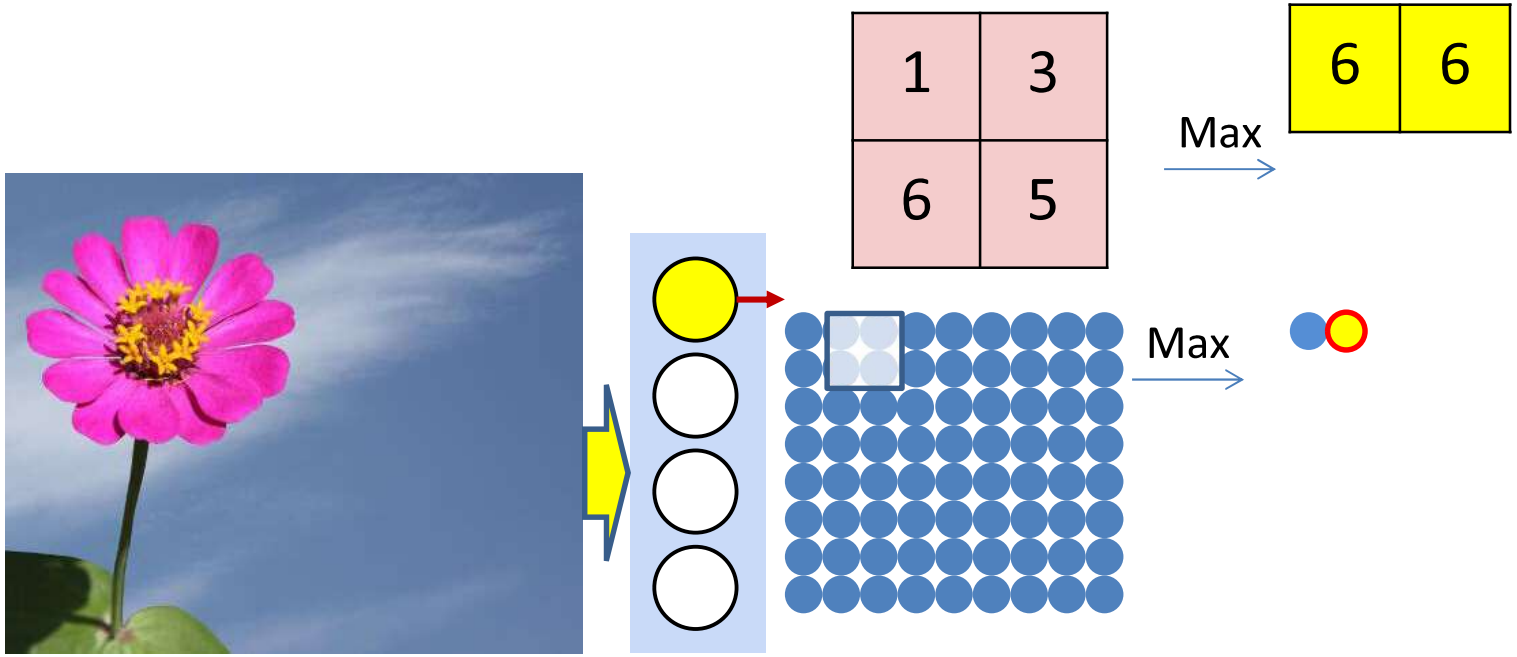
- The max operation is just another neuron
- Instead of applying an activation to the weighted sum of inputs, each neuron just computes the maximum over all inputs

Scanning with the max



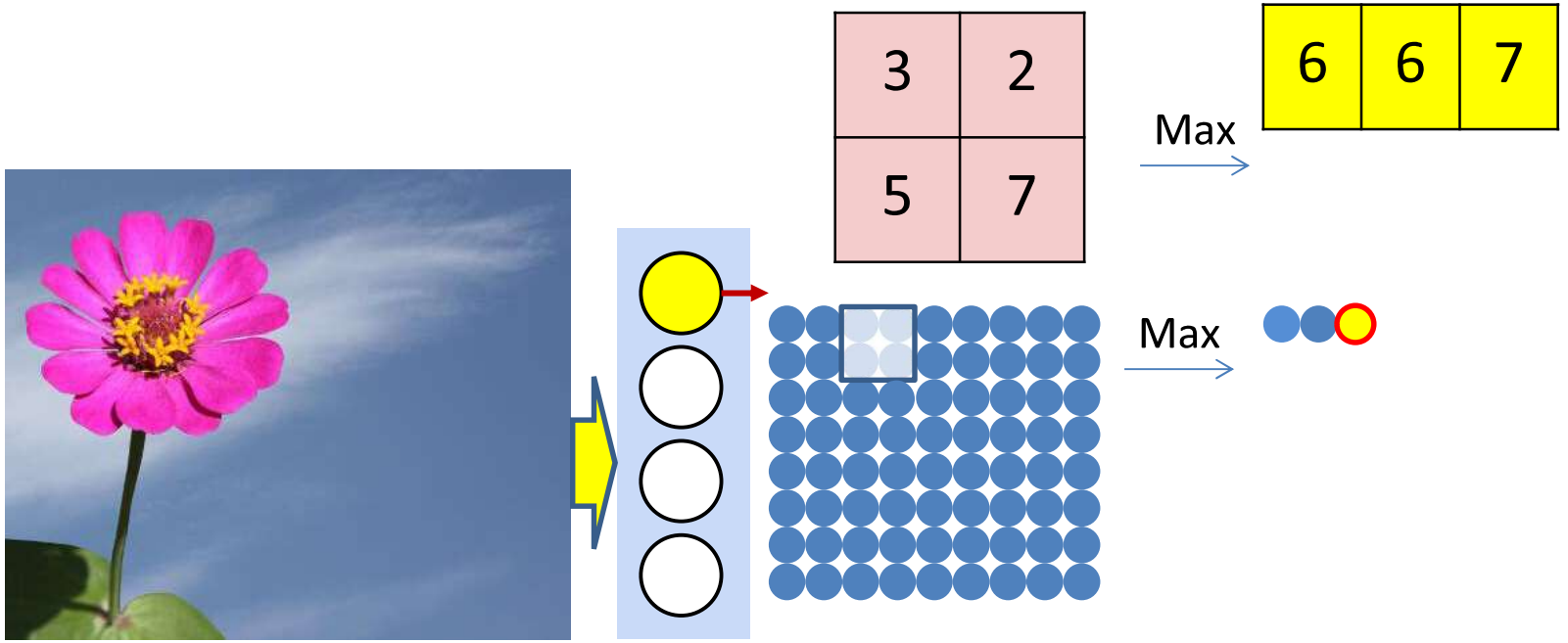
- The max filtering can also be performed as a scan

Scanning with the max



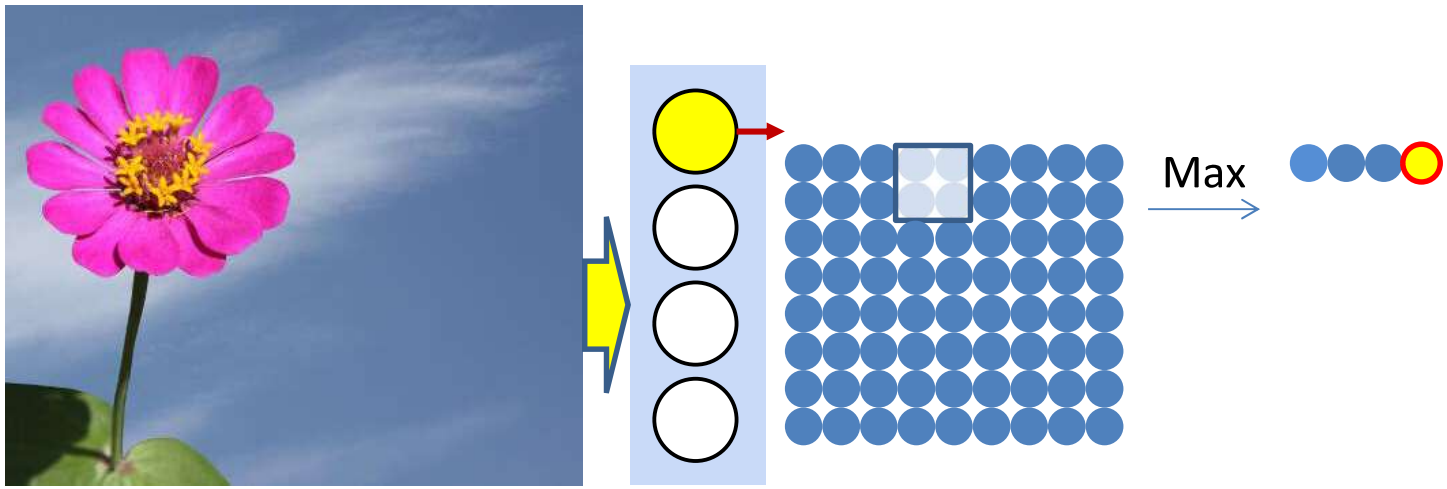
- The “max filter” operation too “scans” the picture

Scanning with the max



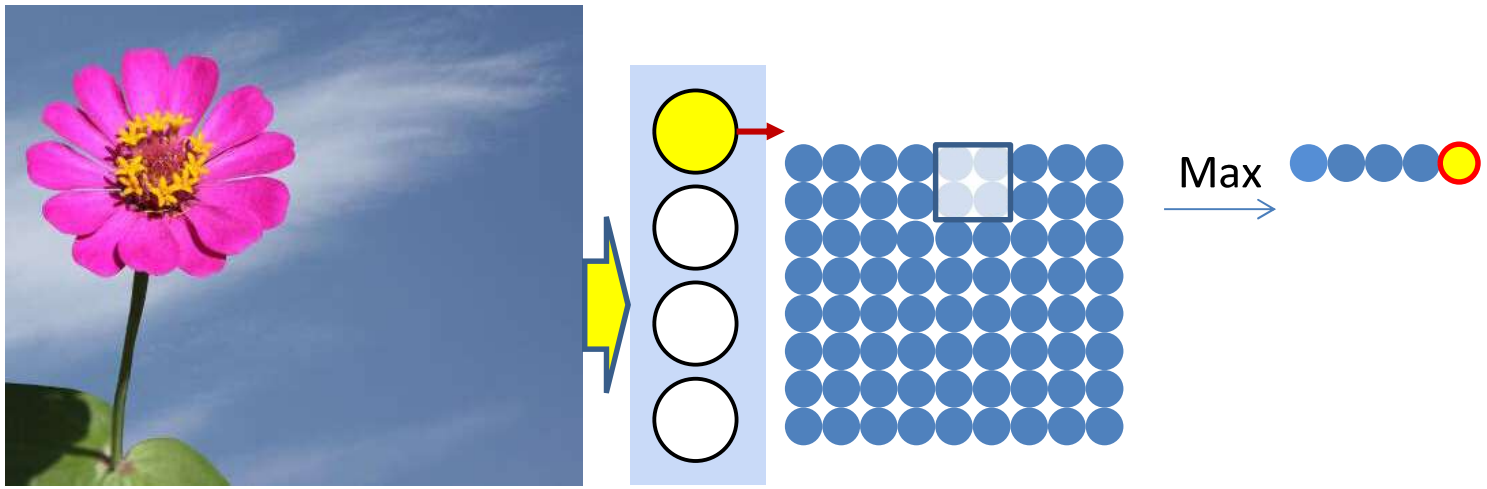
- The “max filter” operation too “scans” the picture

Scanning with the max



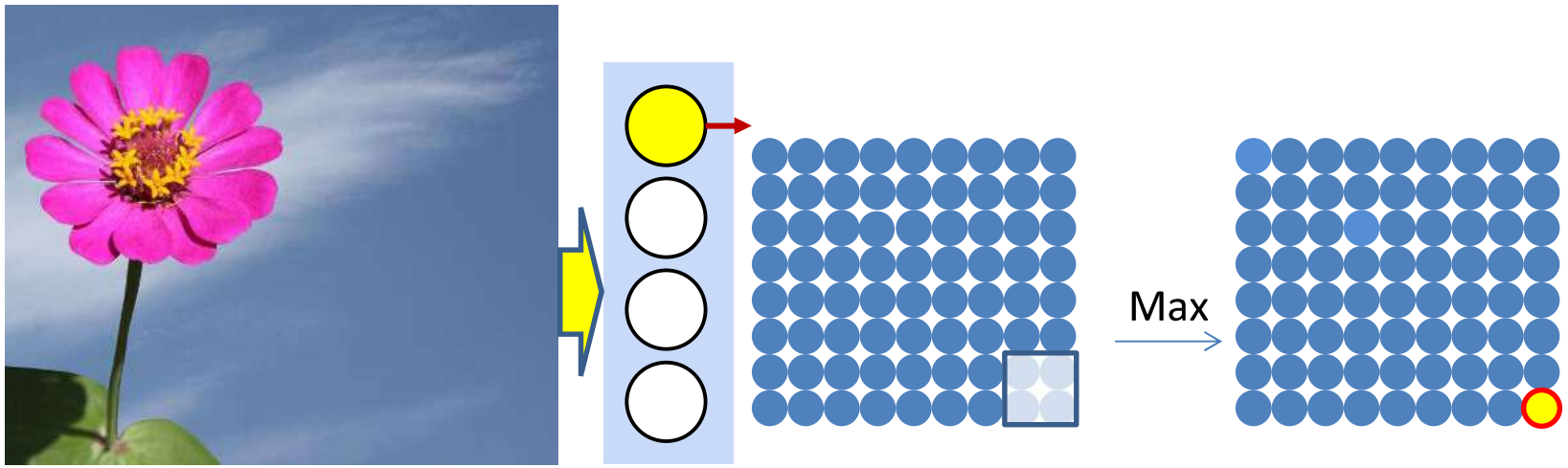
- The “max filter” operation too “scans” the picture

Scanning with the max



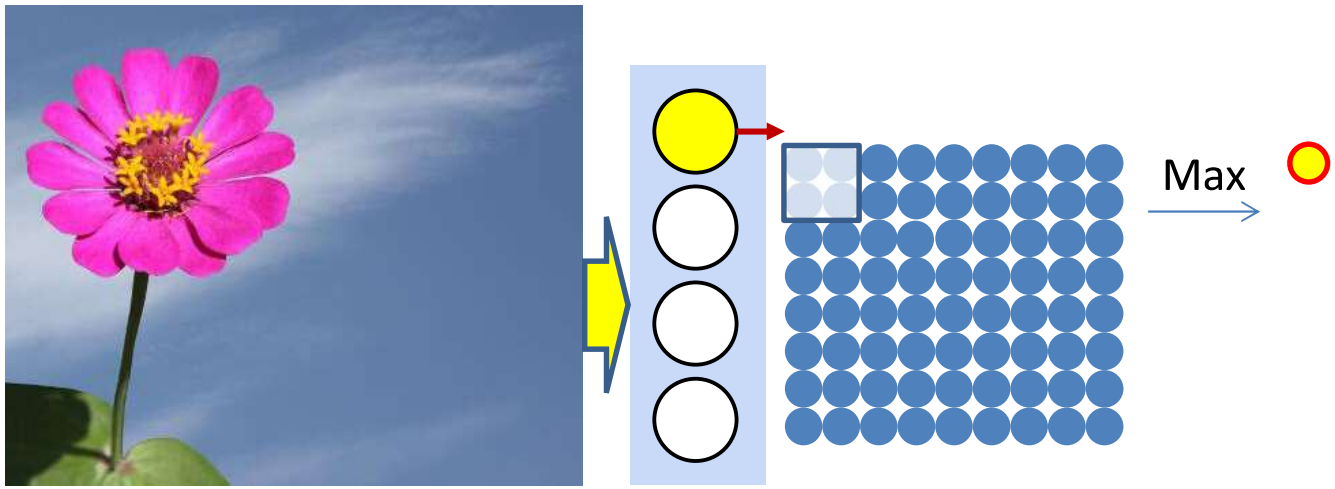
- The “max filter” operation too “scans” the picture

Scanning with the max



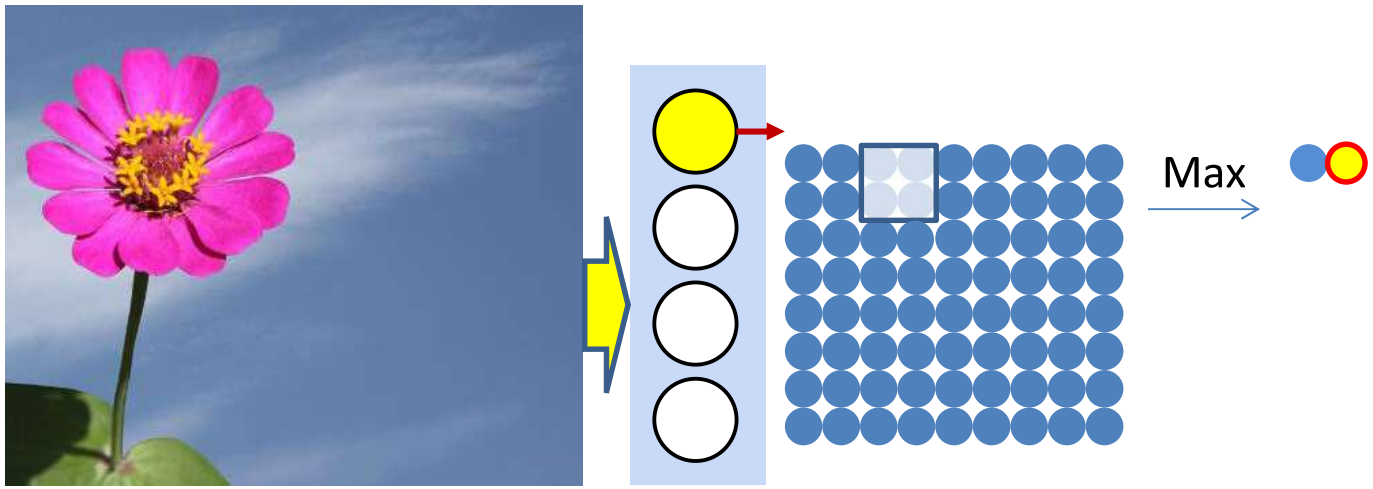
- The “max filter” operation too “scans” the picture

Max pooling “Strides”



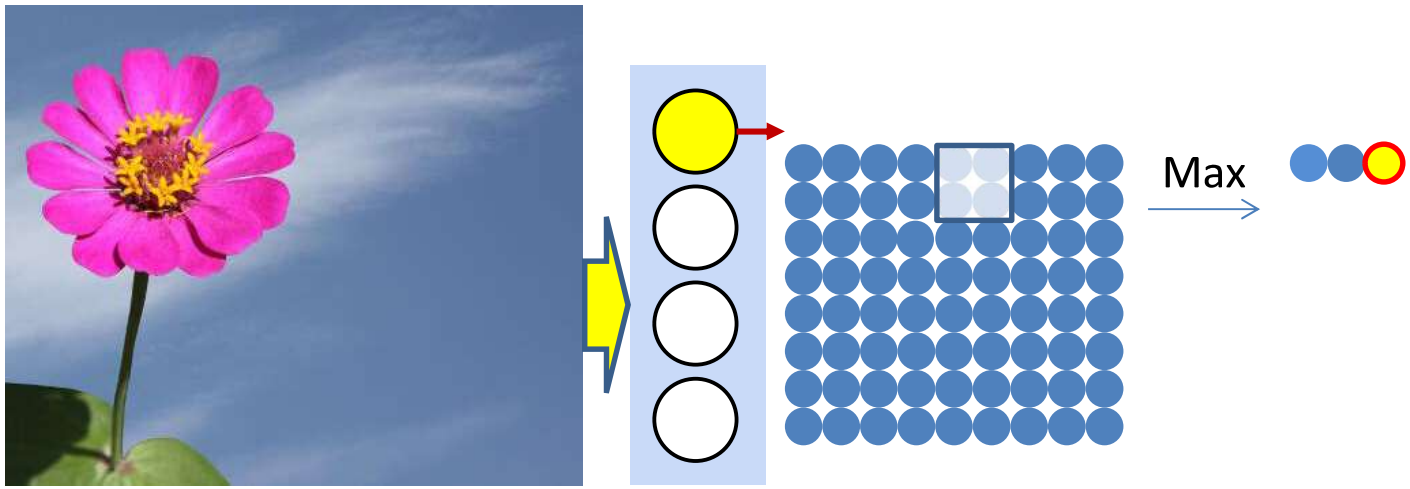
- The “max” operations may “stride” by more than one

Max pooling “Strides”



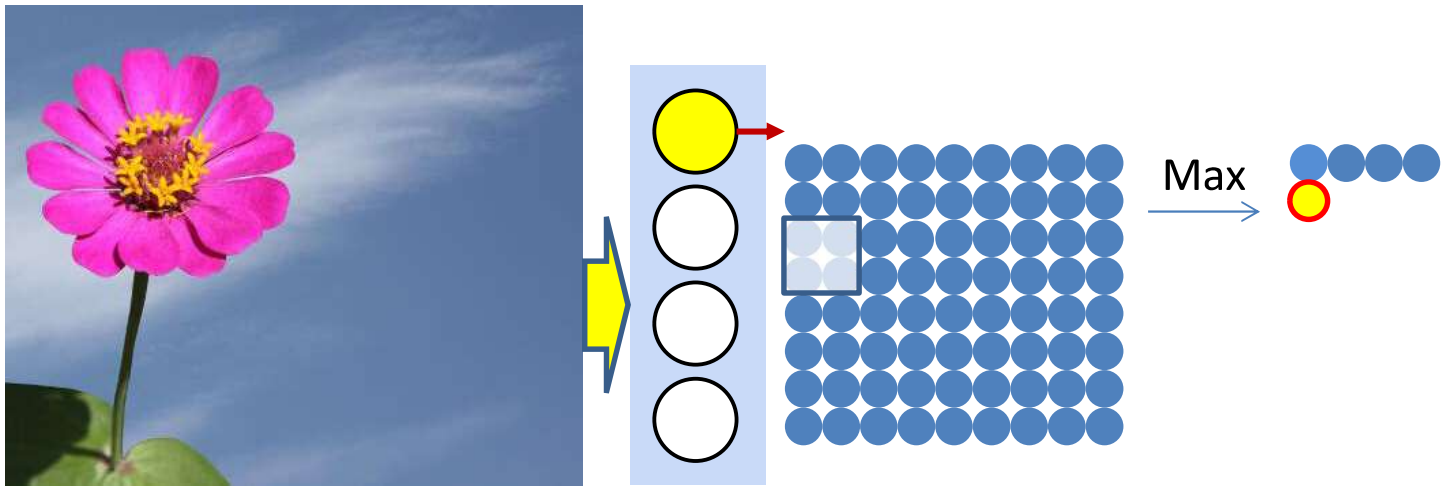
- The “max” operations may “stride” by more than one

Max pooling “Strides”



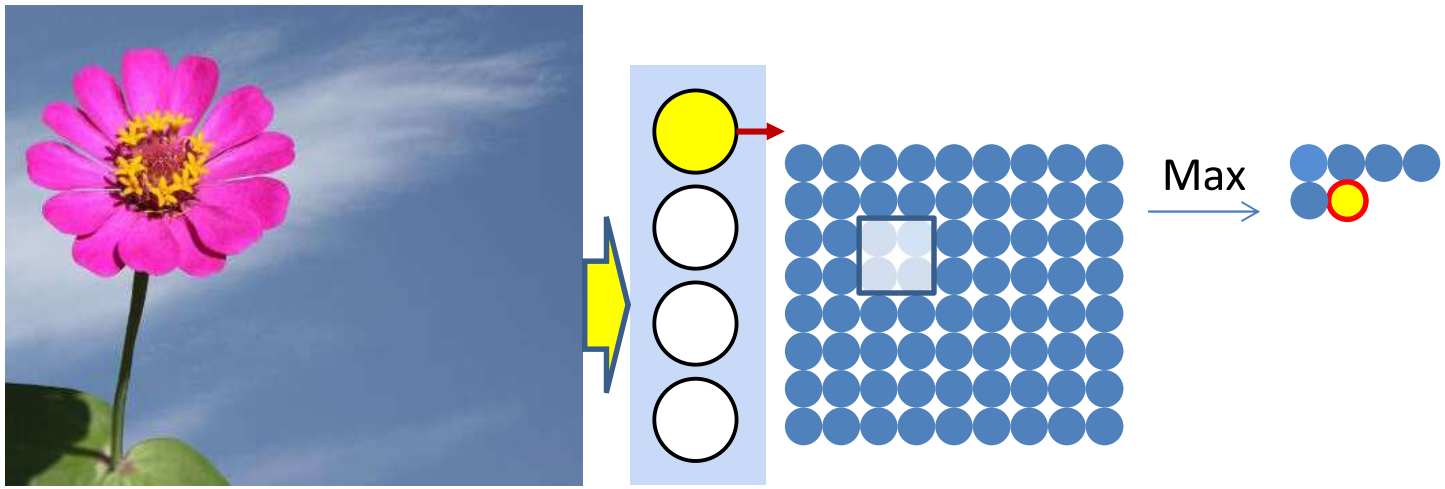
- The “max” operations may “stride” by more than one

Max pooling “Strides”



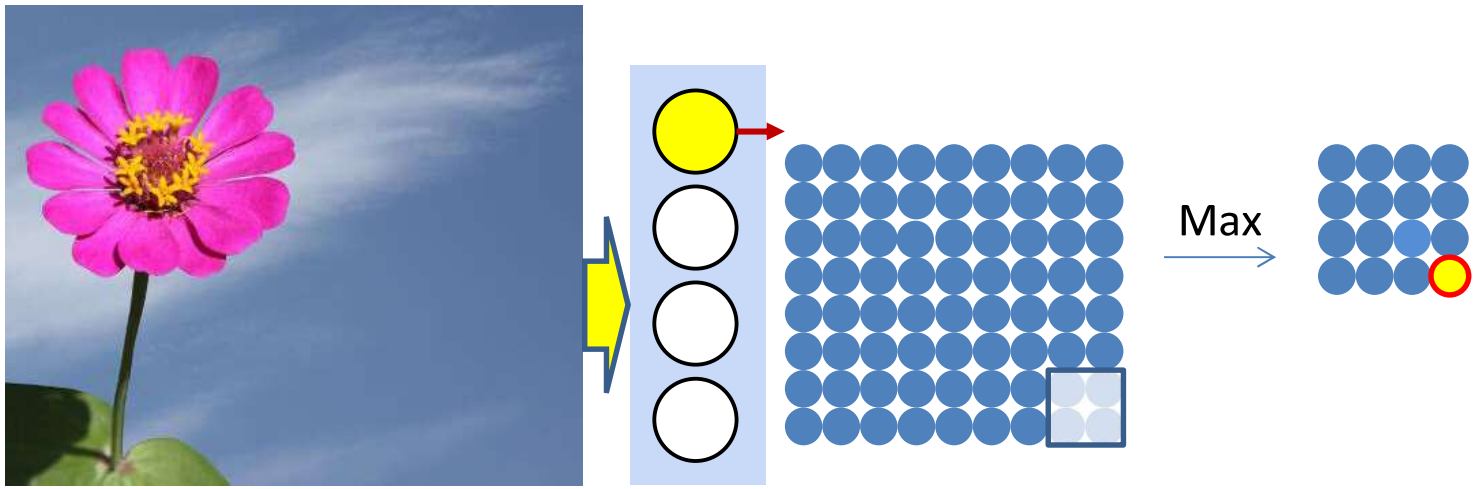
- The “max” operations may “stride” by more than one

Max pooling “Strides”



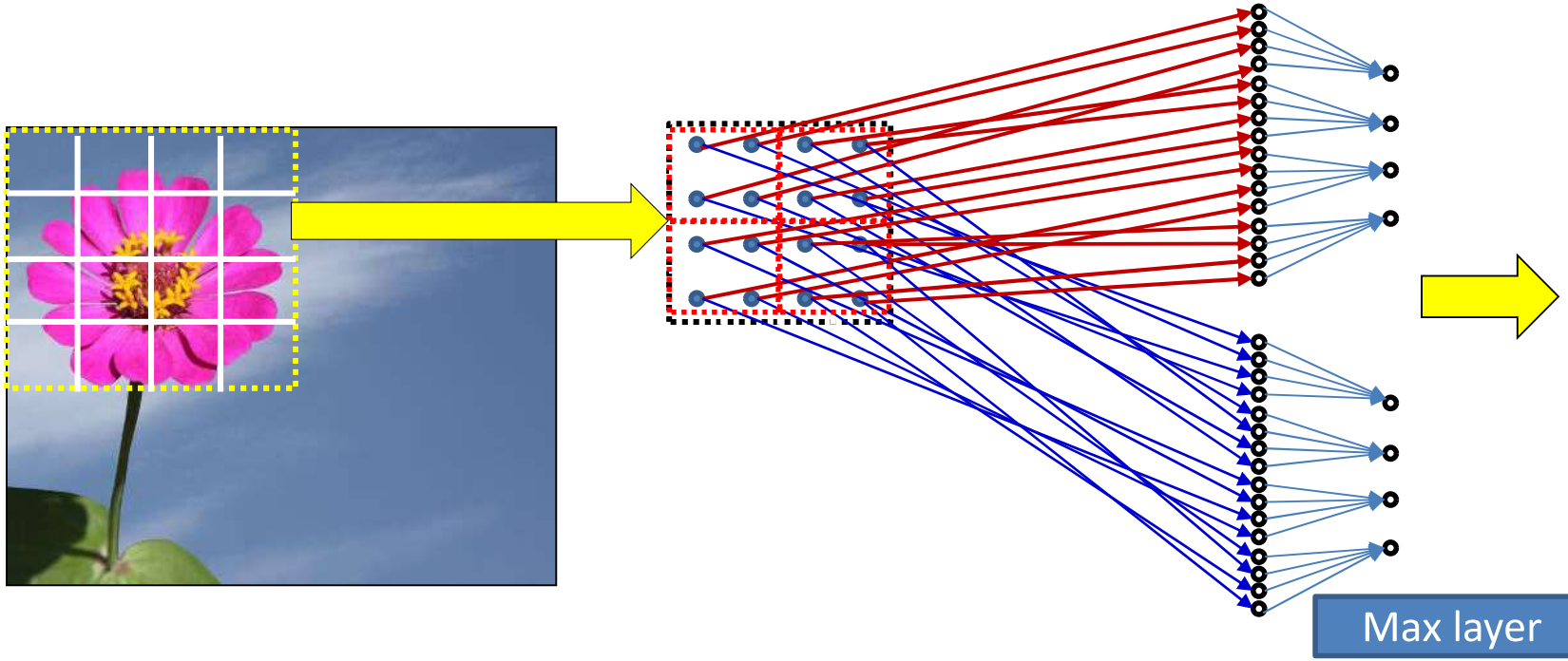
- The “max” operations may “stride” by more than one

Max pooling “Strides”



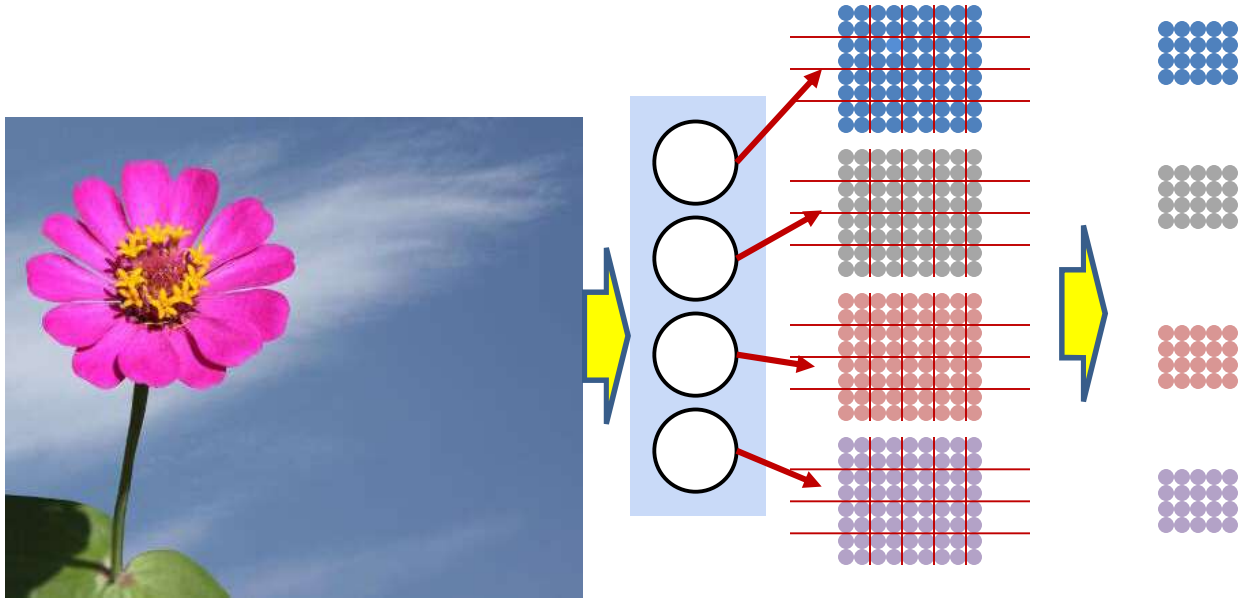
- The “max” operations may “stride” by more than one
  - This will result in a *shrinking* of the map
  - The operation is usually called “pooling”
    - Pooling a number of outputs to get a single output
    - When stride is greater than 1, also called “Down sampling”

Shrinking with a max



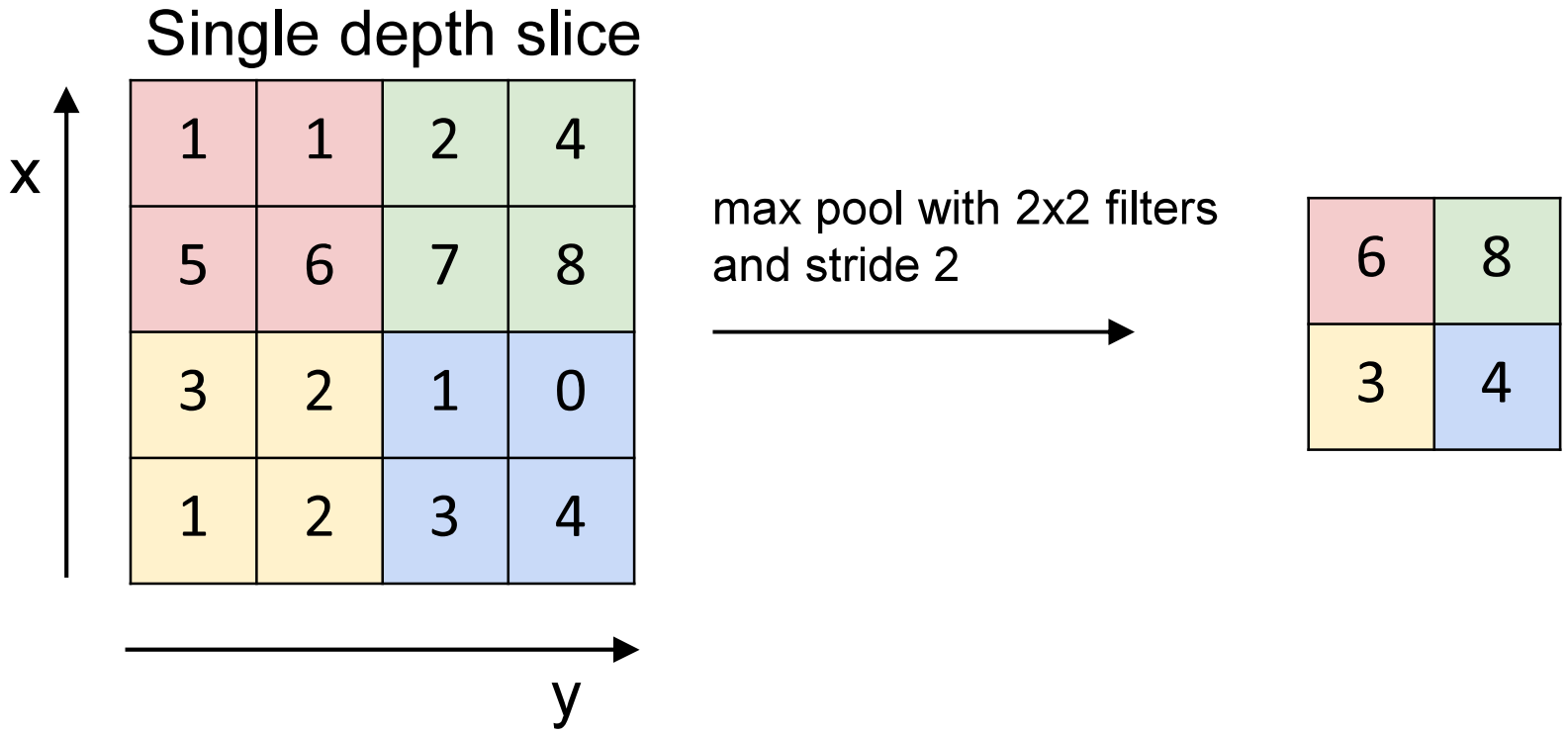
- In this example we *shrank* the image after the max
  - Adjacent “max” operators did not overlap
  - The stride was the size of the max filter itself

Non-overlapped strides

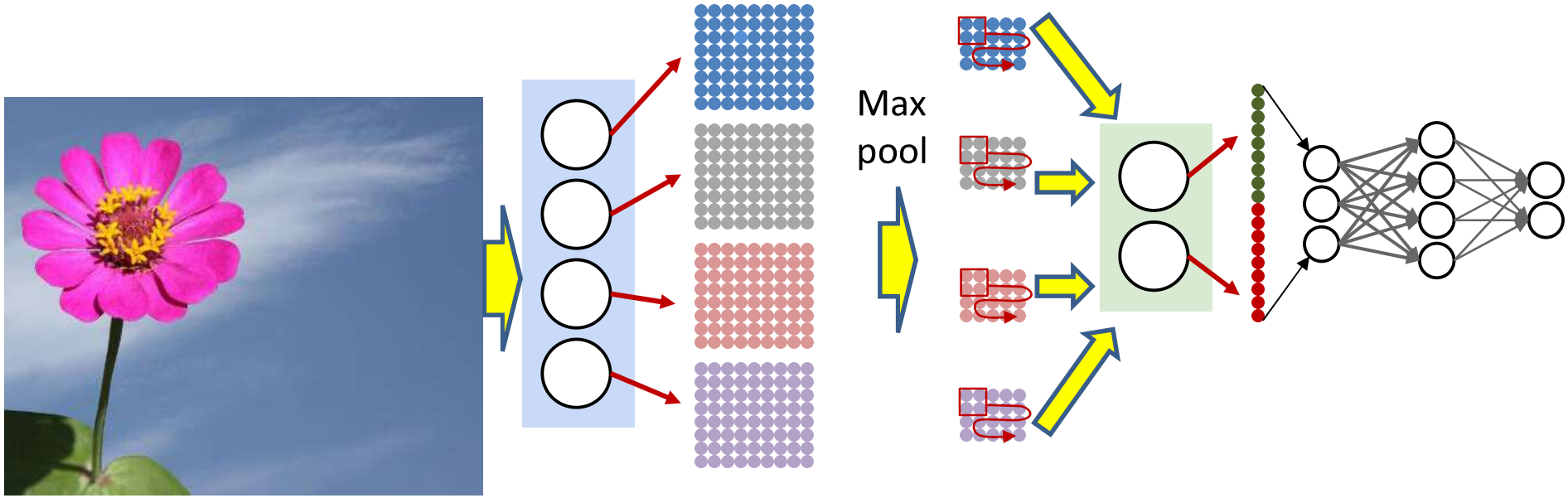


- Non-overlapping strides: Partition the output of the layer into blocks
- Within each block only retain the *highest* value
  - If you detect a petal anywhere in the block, a petal is detected..

Max Pooling

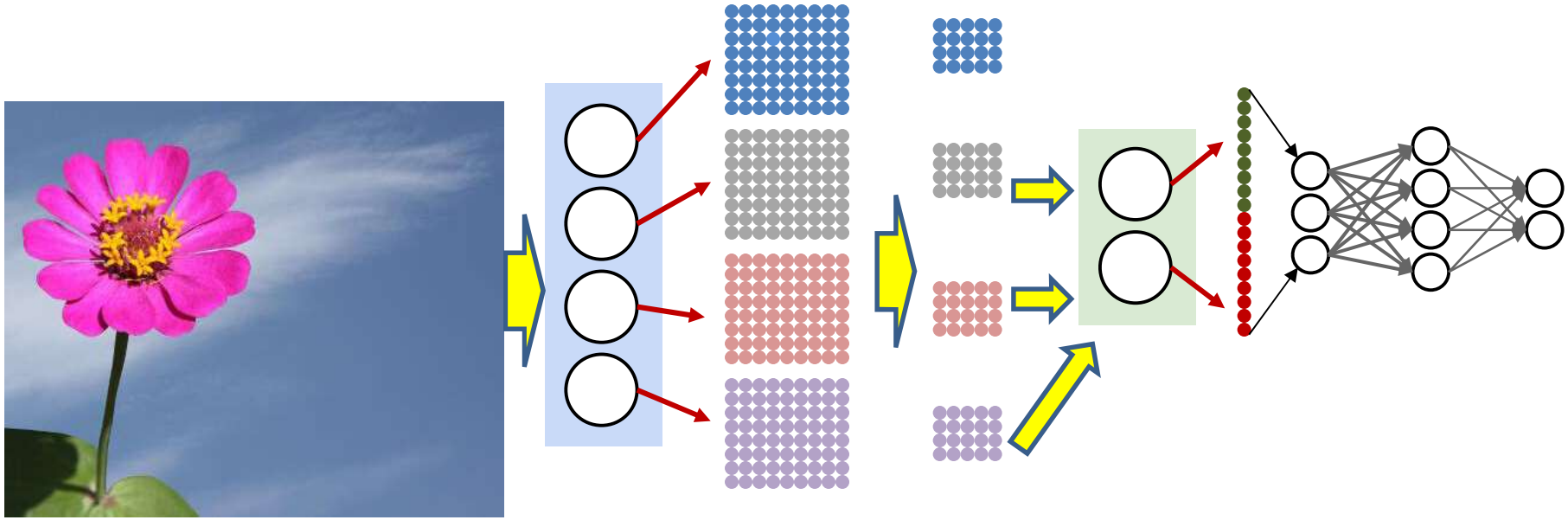

Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

Higher layers



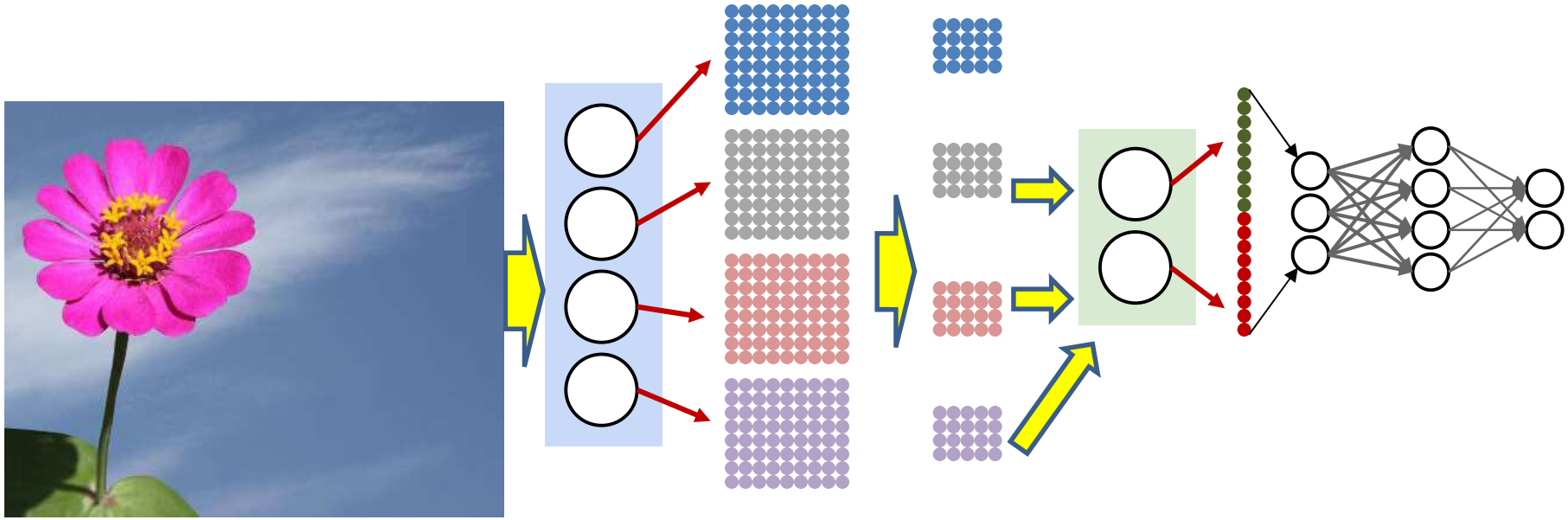
- The next layer works on the *max-pooled* maps

The overall structure



- We can have many layers of “convolution” (scanning) followed by max pooling (and reduction) before the final MLP
  - Not every convolutional layer needs to be followed by max pooling

The overall structure



- This entire structure is called a ***Convolutional Neural Network***

Convolutional Neural Network