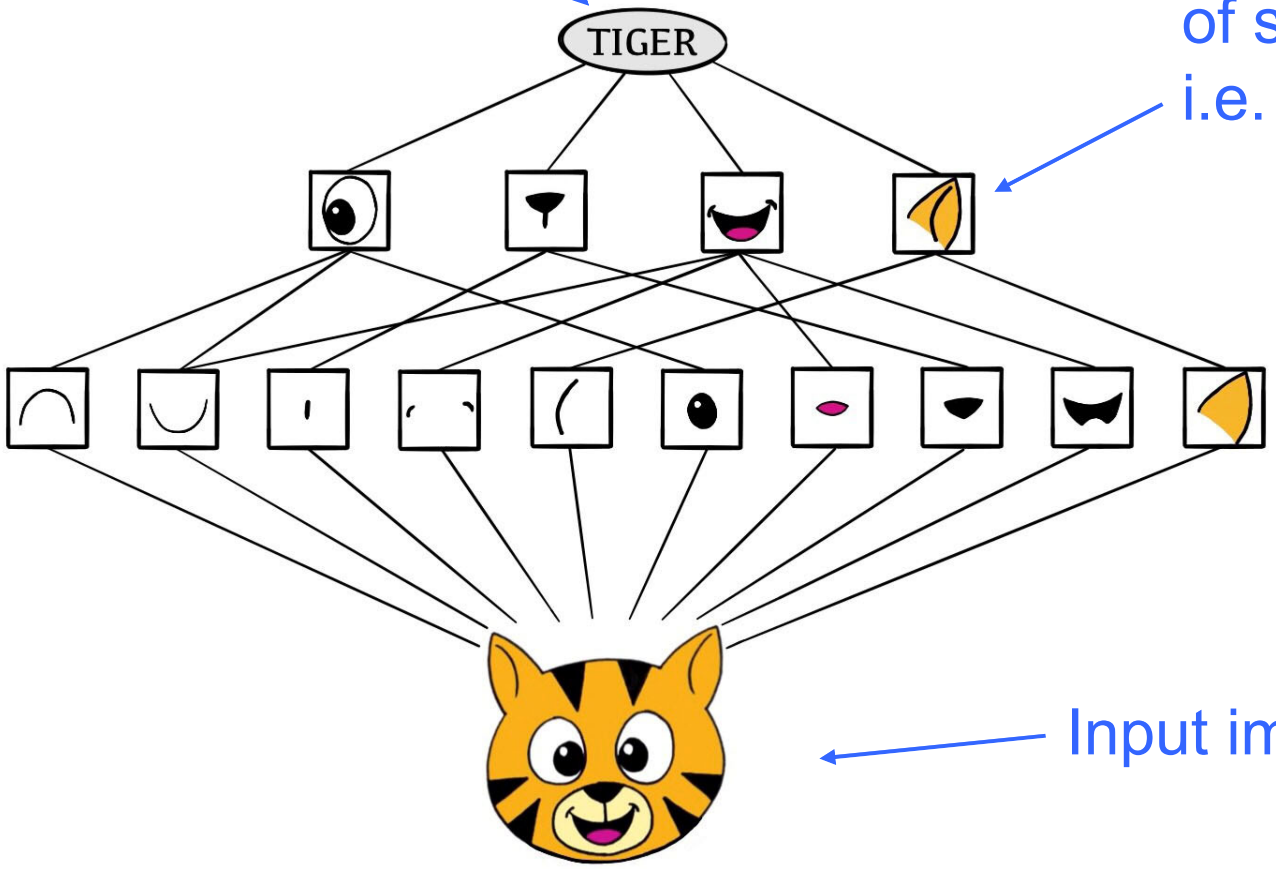
Today

- Neural networks and deep learning
  - Single layer neural networks
    - using keras package
    - architectures for different questions
      - regression, classification, multifunction
  - Multi-layer neural networks
  - Convolutional neural networks
  - U-net
  - Transformers

deep

Convolutional NNs

Output is high-level concept



Conv layer 2
learns combinations
of small features
i.e. objects

Conv layer 1
learns small
features
e.g. edges

Input image

ISLR 10.6

Convolutional NNs

- Example: CIFAR100
  - standard benchmark dataset: 100 categories
  - ecology subset: 61 categories
- 08_3_convolutional_nnet.R

Convolution layer

Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

↓ convolve

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer

Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

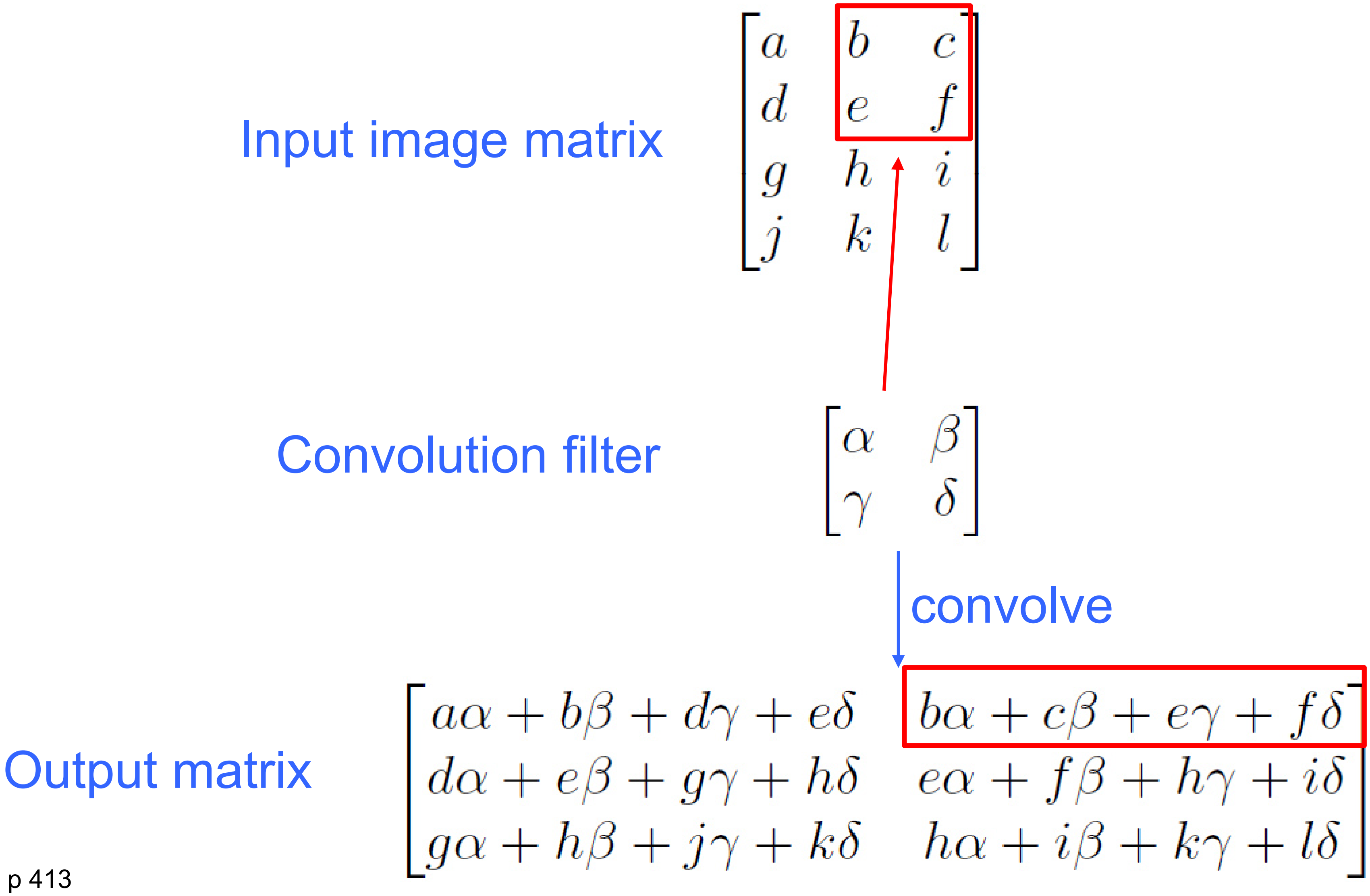
$$\Big\downarrow \text{ convolve}$$

Output matrix

$$\begin{bmatrix} \boxed{a\alpha + b\beta + d\gamma + e\delta} & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer



Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

convolve

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer

Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

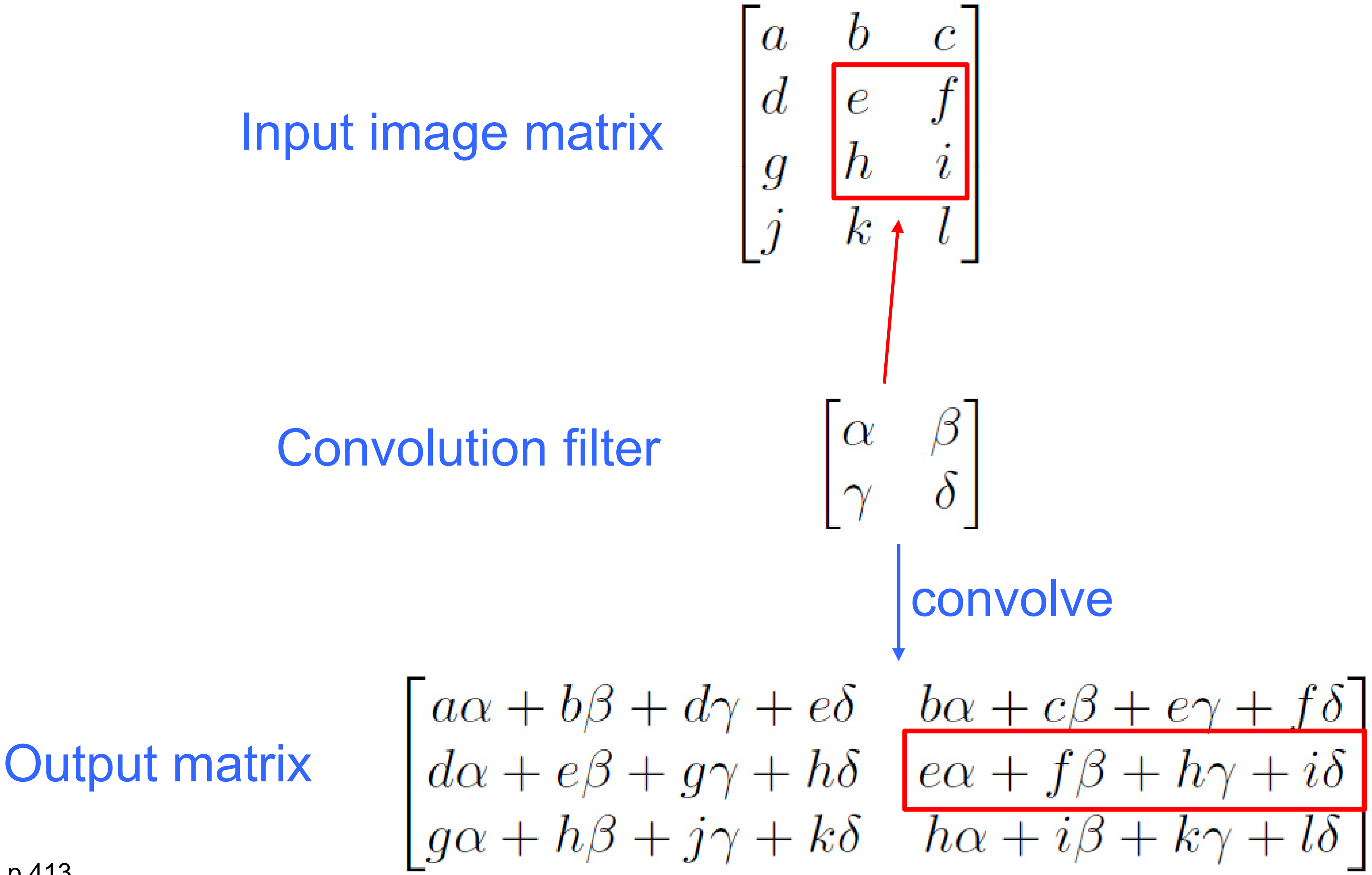
↓ convolve

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ \boxed{d\alpha + e\beta + g\gamma + h\delta} & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer



Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

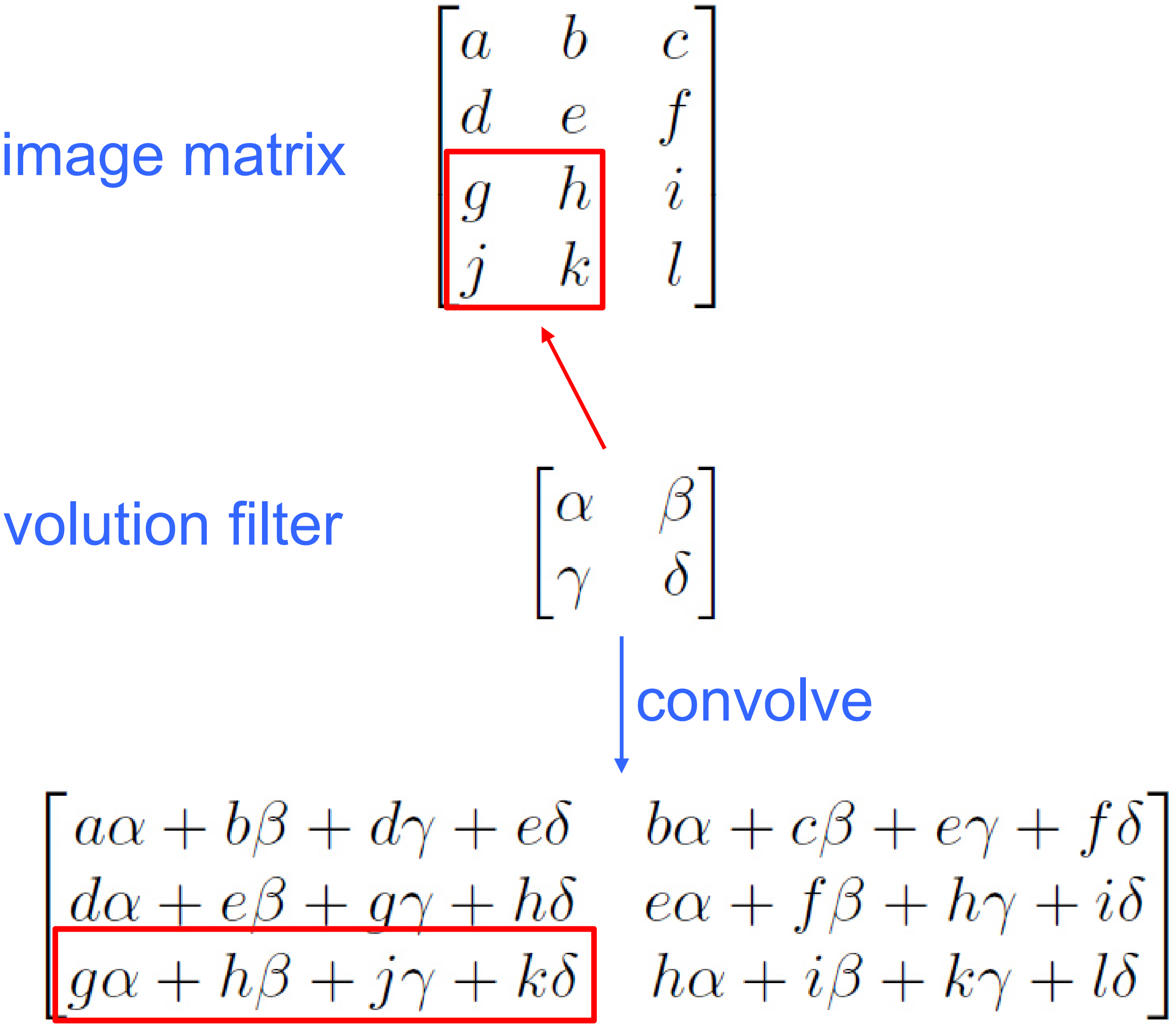
convolve

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & \boxed{e\alpha + f\beta + h\gamma + i\delta} \\ g\alpha + h\beta + j\gamma + k\delta & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer



Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

convolve

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ \boxed{g\alpha + h\beta + j\gamma + k\delta} & h\alpha + i\beta + k\gamma + l\delta \end{bmatrix}$$

ISLR p 413

Convolution layer

Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

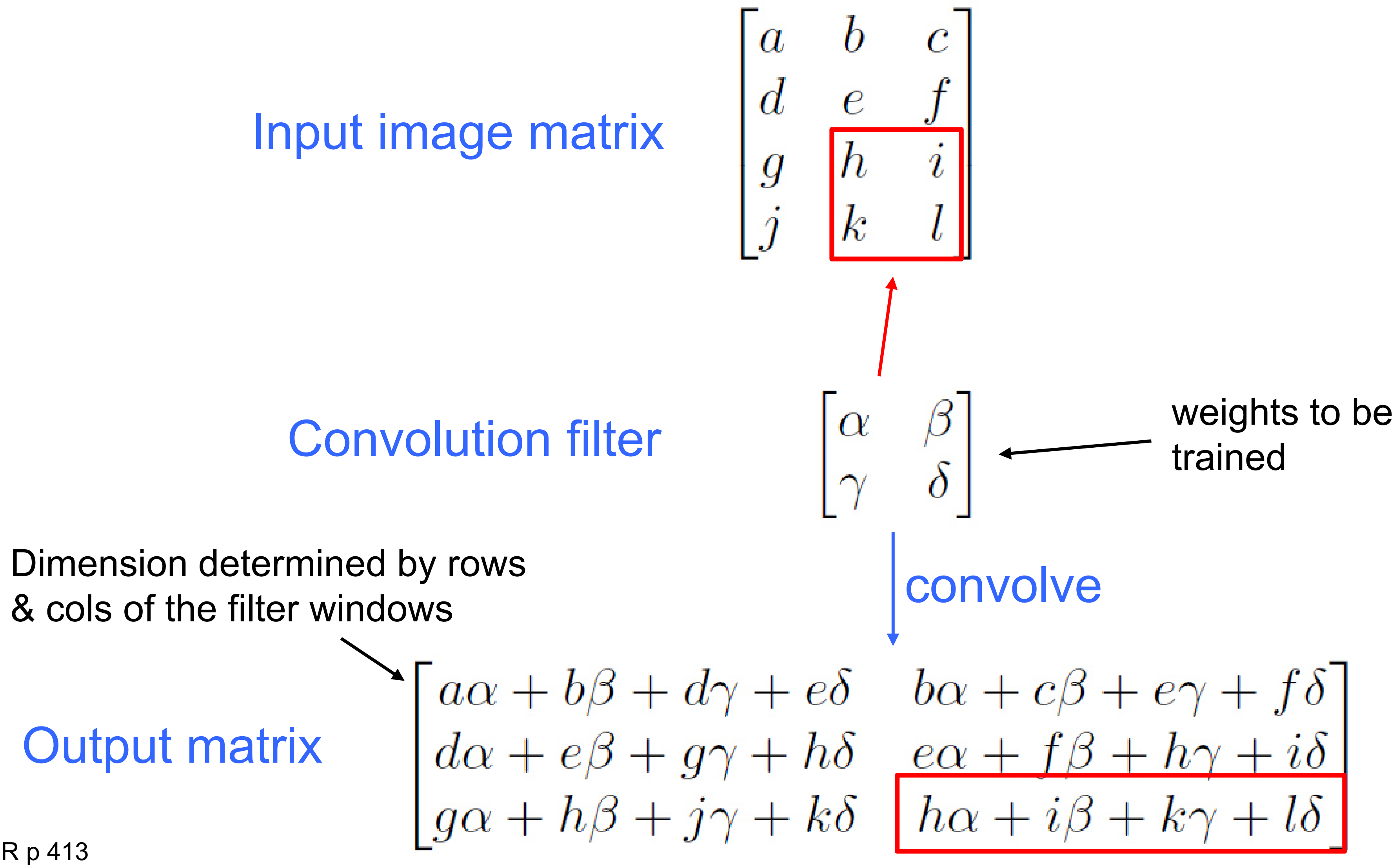
$$\Big\downarrow \text{convolve}$$

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & \boxed{h\alpha + i\beta + k\gamma + l\delta} \end{bmatrix}$$

ISLR p 413

Convolution layer



Input image matrix

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

Convolution filter

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

weights to be trained

convolve

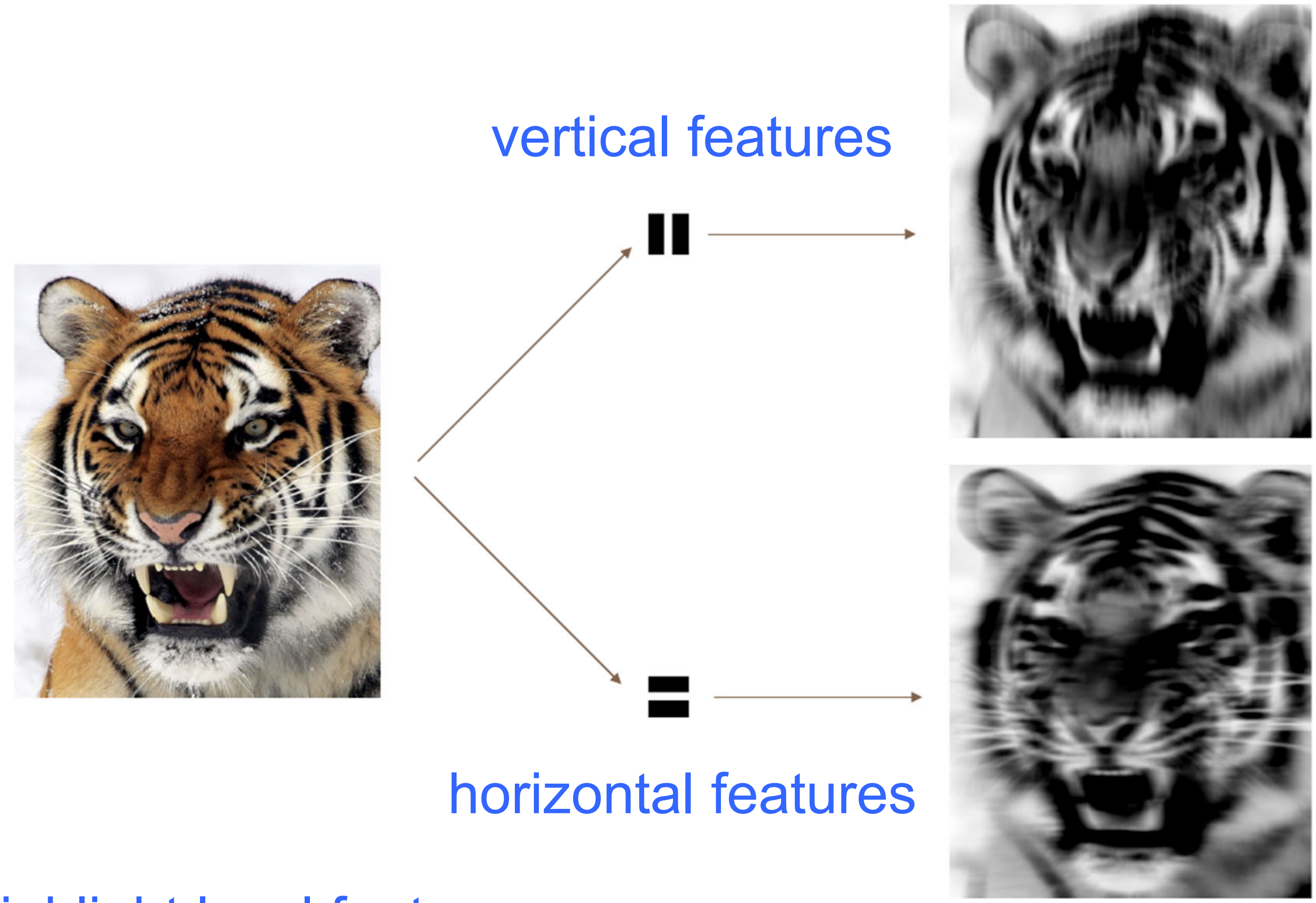
Dimension determined by rows & cols of the filter windows

Output matrix

$$\begin{bmatrix} a\alpha + b\beta + d\gamma + e\delta & b\alpha + c\beta + e\gamma + f\delta \\ d\alpha + e\beta + g\gamma + h\delta & e\alpha + f\beta + h\gamma + i\delta \\ g\alpha + h\beta + j\gamma + k\delta & \boxed{h\alpha + i\beta + k\gamma + l\delta} \end{bmatrix}$$

ISLR p 413

Convolution layer



input image

filter

convolved image

vertical features

horizontal features

filters highlight local features

ISLR 10.7

Pooling layer

Condenses information

e.g. “Max pooling”

$$\begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 5 \\ 2 & 4 \end{bmatrix}$$

Input matrix
4x4

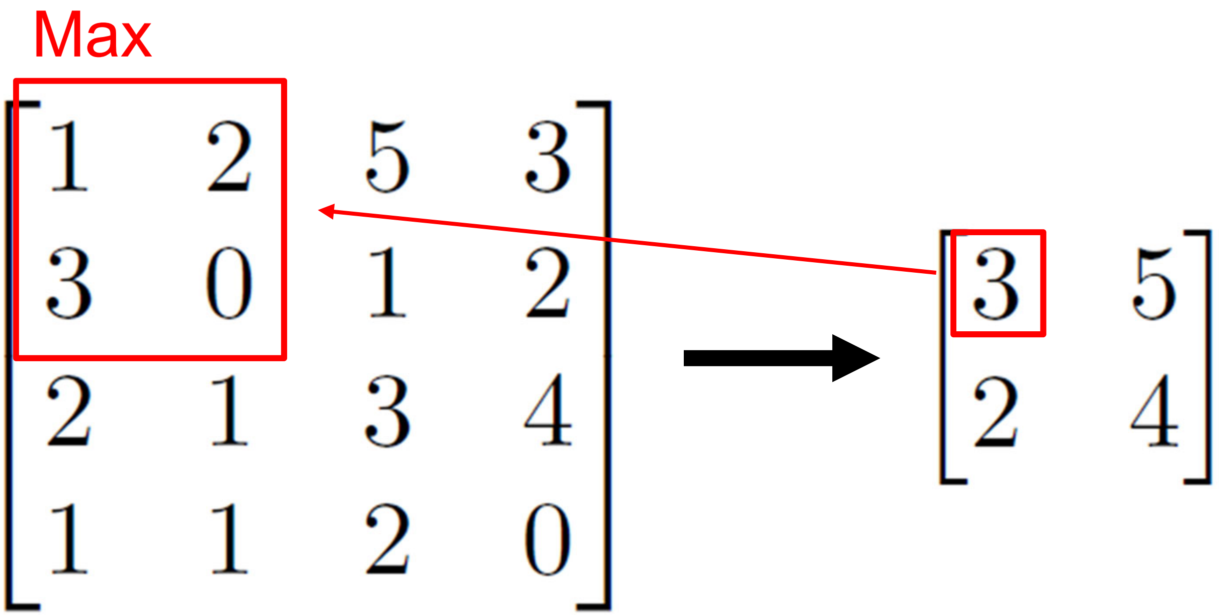
Output matrix
2x2

ISLR p 415

Pooling layer

Condenses information

e.g. “Max pooling”



$$\begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} \boxed{3} & 5 \\ 2 & 4 \end{bmatrix}$$

Input matrix
4x4

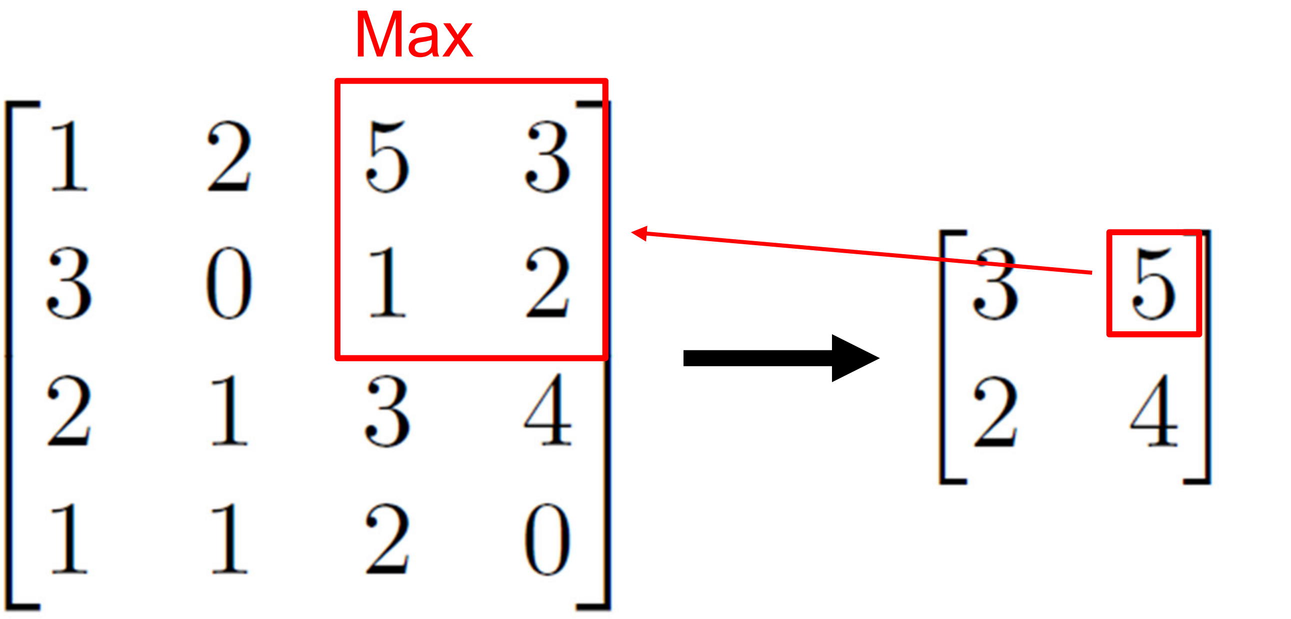
Output matrix
2x2

ISLR p 415

Pooling layer

Condenses information

e.g. “Max pooling”



Max

$$\begin{bmatrix}1 & 2 & 5 & 3\\ 3 & 0 & 1 & 2\\ 2 & 1 & 3 & 4\\ 1 & 1 & 2 & 0\end{bmatrix} \longrightarrow \begin{bmatrix}3 & \boxed{5}\\ 2 & 4\end{bmatrix}$$

Input matrix
4x4

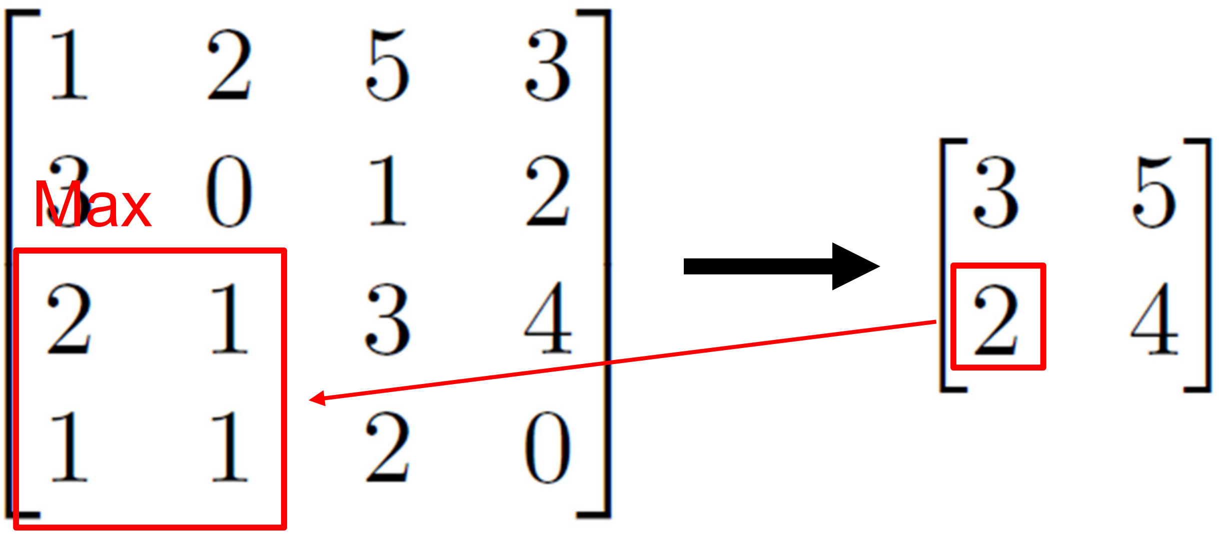
Output matrix
2x2

ISLR p 415

Pooling layer

Condenses information

e.g. “Max pooling”



$$\begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 5 \\ \boxed{2} & 4 \end{bmatrix}$$

Input matrix
4x4

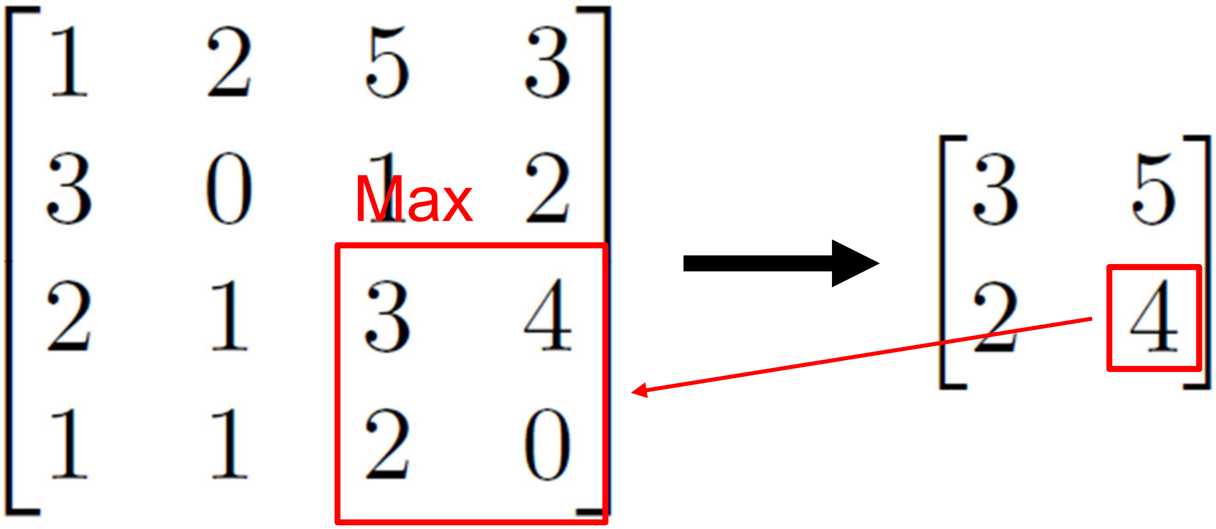
Output matrix
2x2

ISLR p 415

Pooling layer

Condenses information

e.g. “Max pooling”



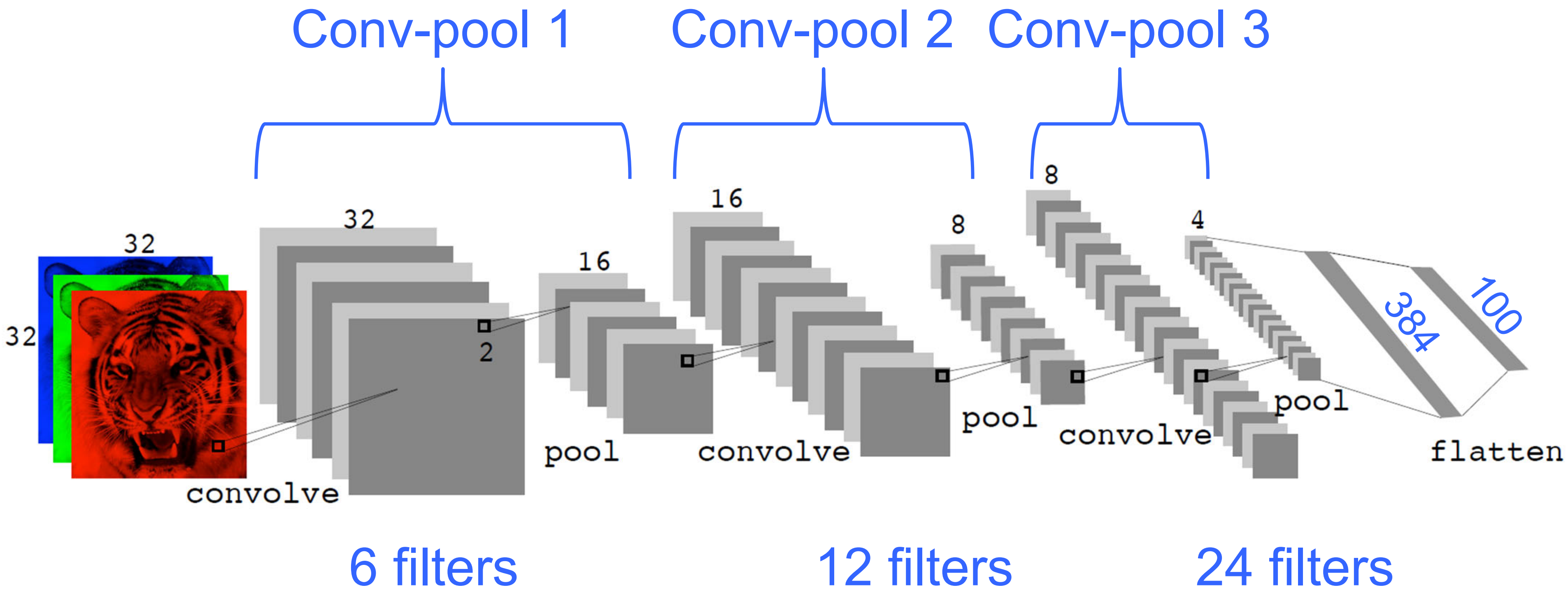
$$\begin{bmatrix} 1 & 2 & 5 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 1 & 3 & 4 \\ 1 & 1 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 3 & 5 \\ 2 & \boxed{4} \end{bmatrix}$$

Input matrix
4x4

Output matrix
2x2

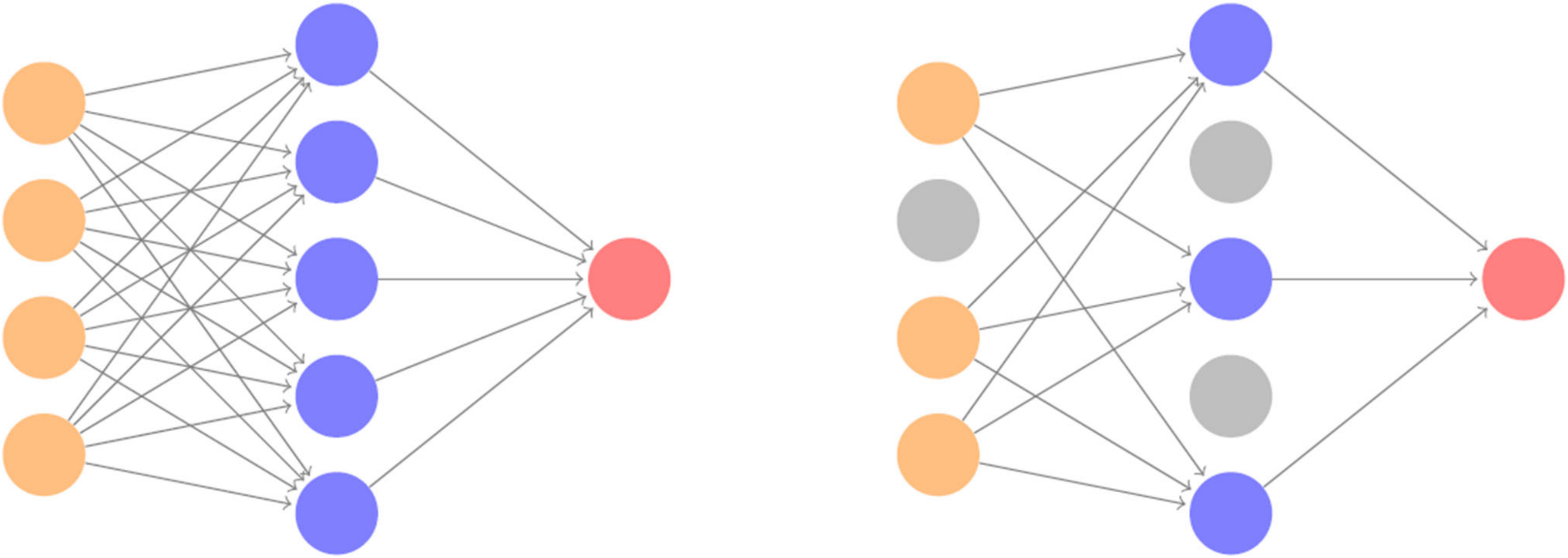
ISLR p 415

Architecture



ISLR 10.8

Training algorithm: dropout

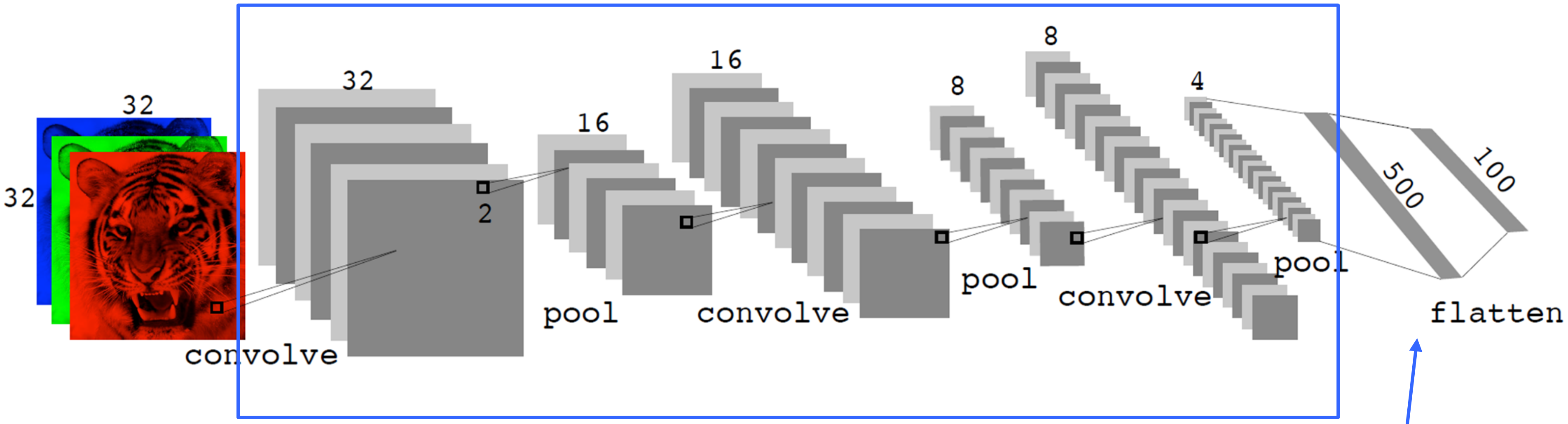


like randomforest sample columns

ISLR 10.19

Pretrained models

Transfer learning



Freeze convolve-pool layers (architecture, weights etc)

Reconfigure
and/or retrain
dense layers

ISLR 10.8