Introduction to CNN

Weihao Qiu

Introduction to CNN

Agenda

Introduction to CNN

Agenda

How do we recognize digit? How does the CNN?

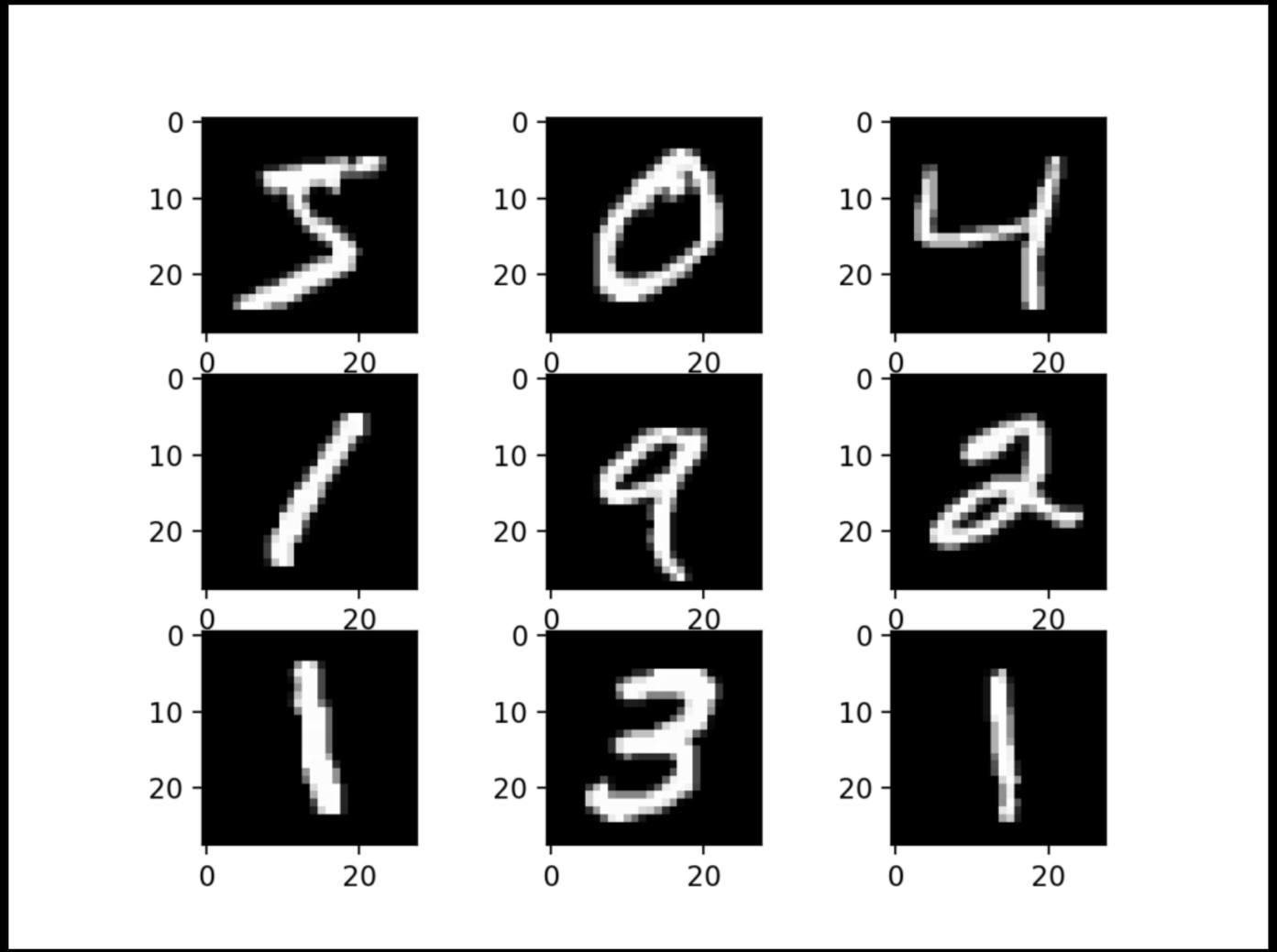
Convolutions for feature extraction

Convolutional Neural Network (CNN)

Terminology

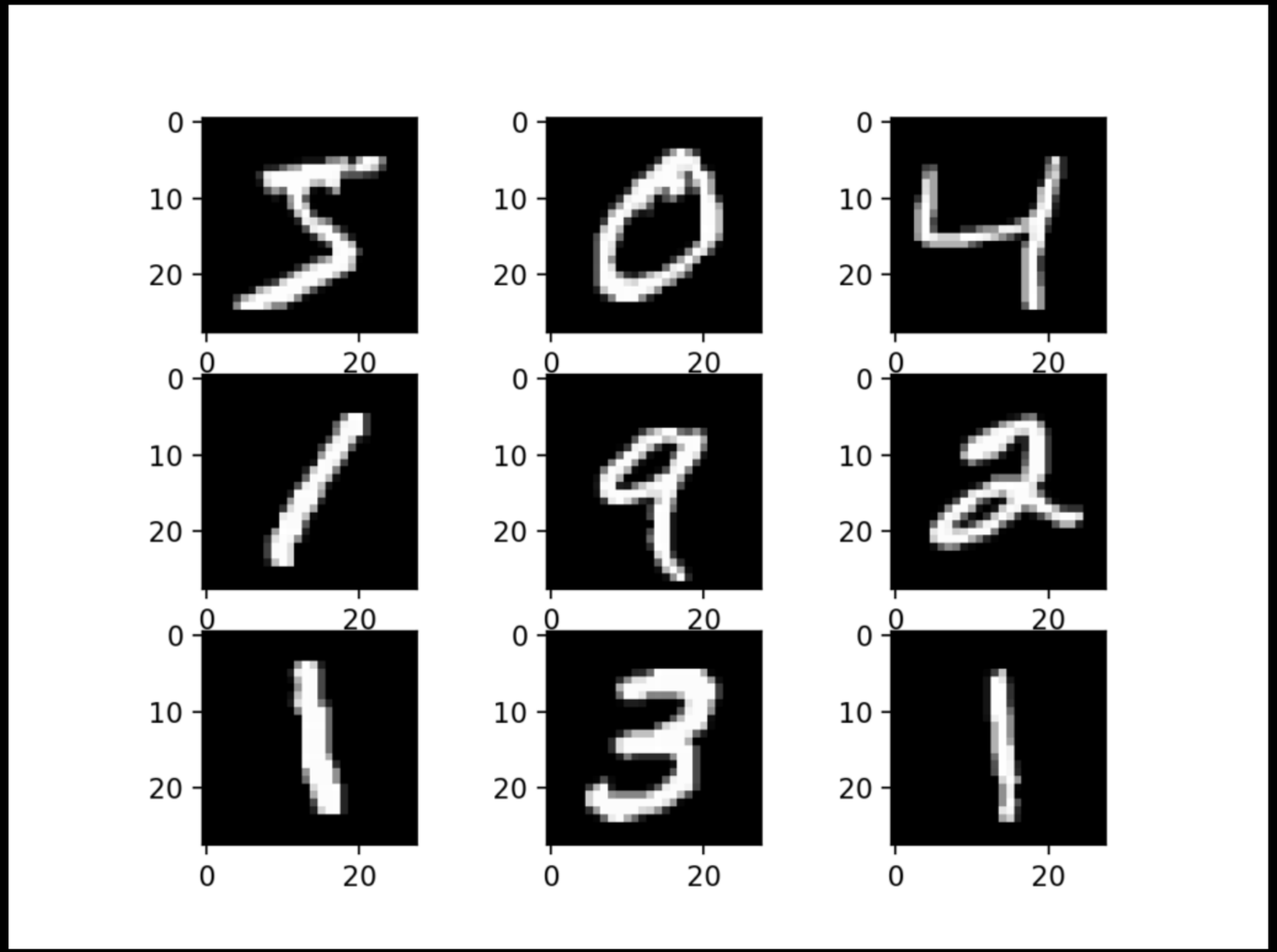
CNN explainer

How do we recognize the digit?



How do we recognize the digit?

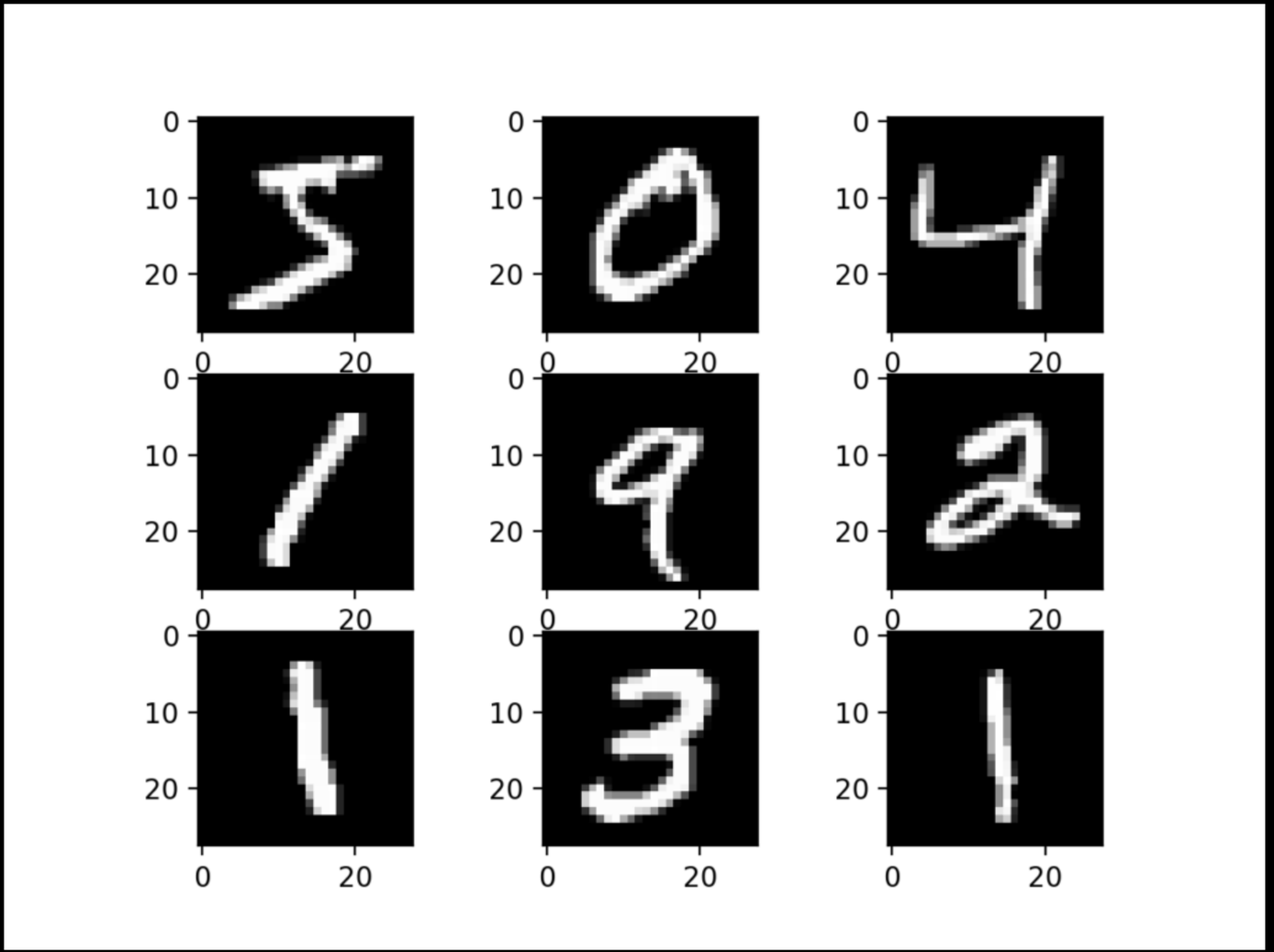
By summarized patterns from experience



How do we recognize the digit?

By summarized patterns from experience

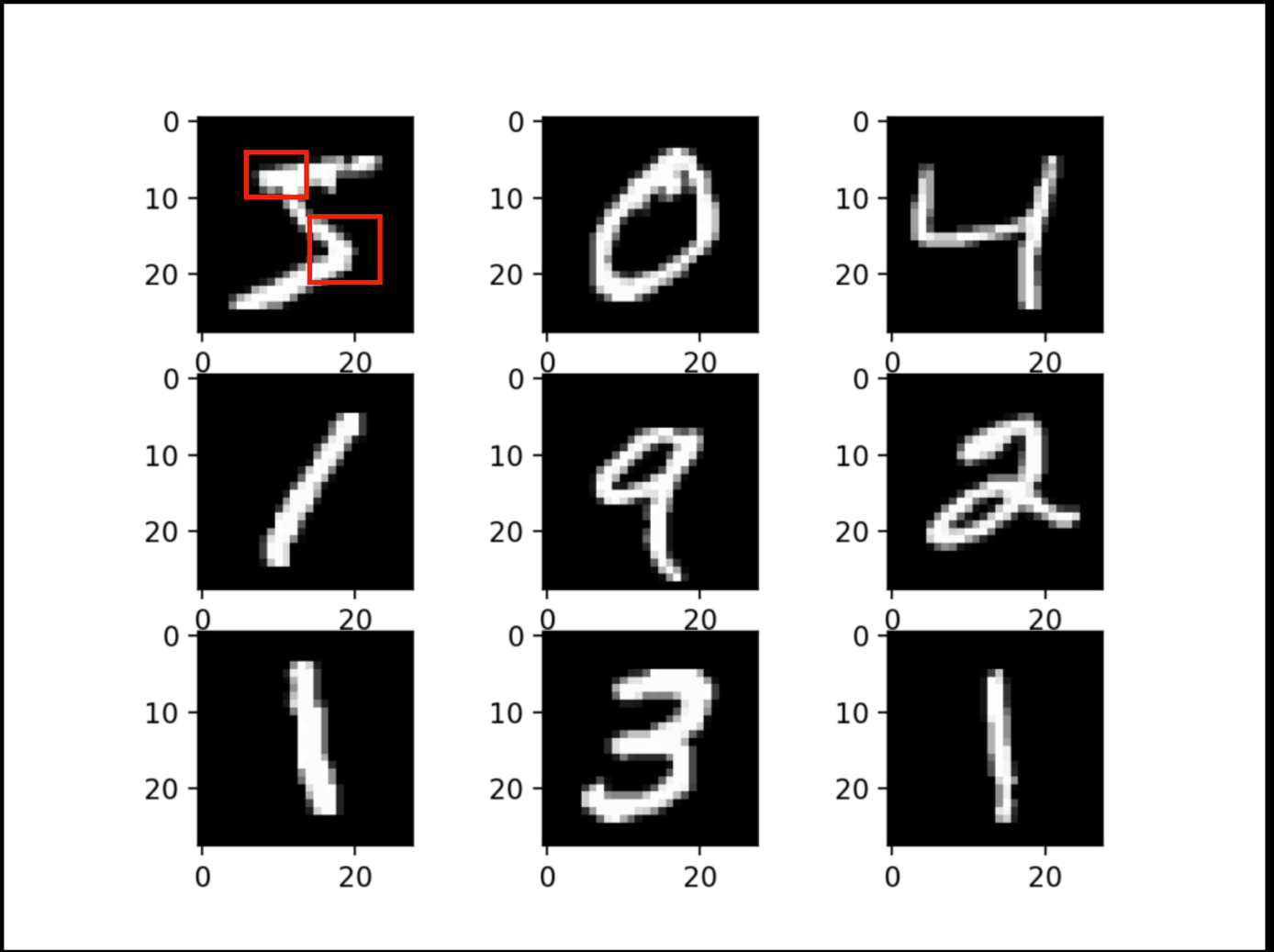
Patterns: the features shared by all samples of one digit



How do we recognize the digit?

By summarized patterns from experience

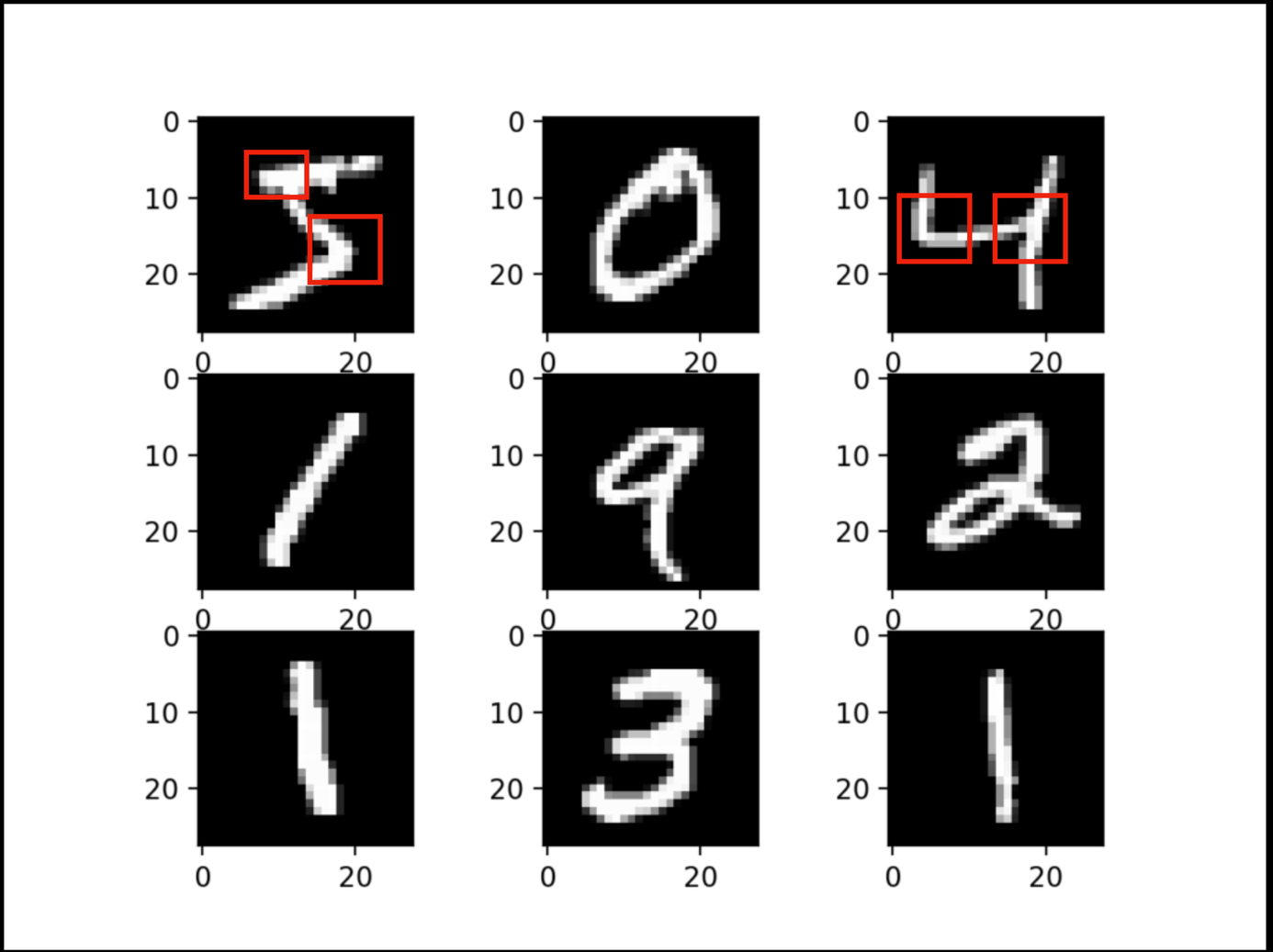
Patterns: the features shared by all samples of one digit



How do we recognize the digit?

By summarized patterns from experience

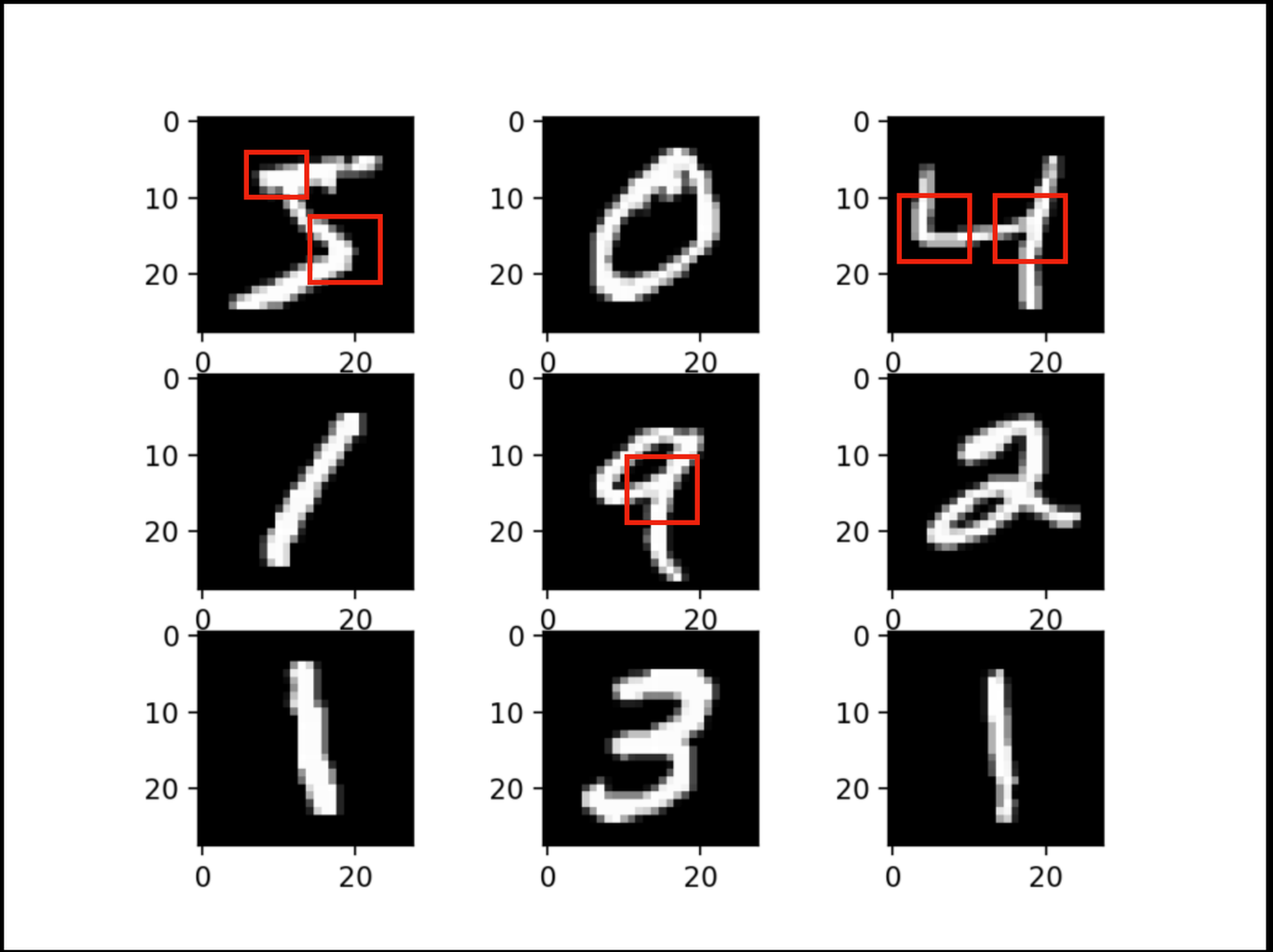
Patterns: the features shared by all samples of one digit



How do we recognize the digit?

By summarized patterns from experience

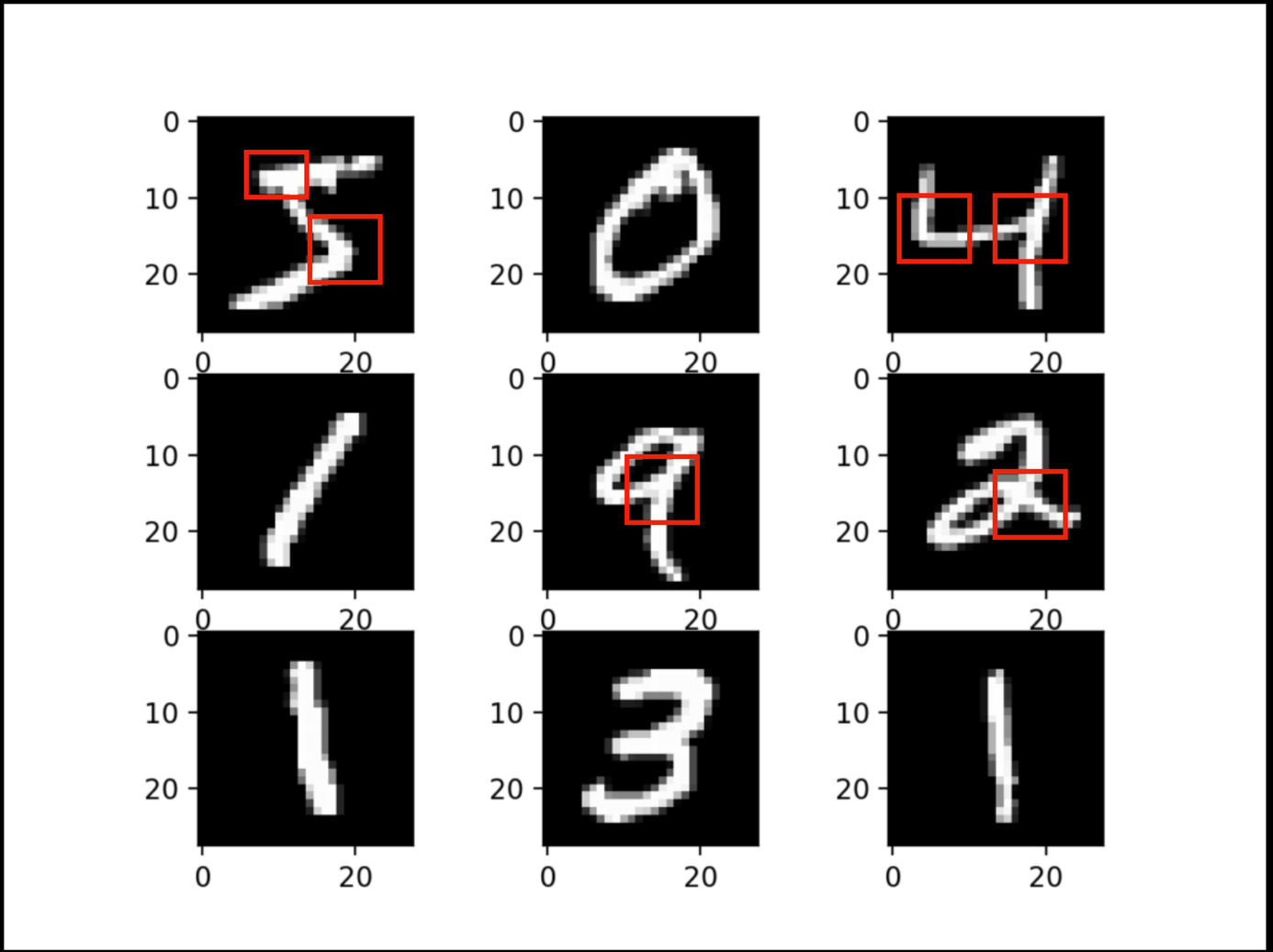
Patterns: the features shared by all samples of one digit



How do we recognize the digit?

By summarized patterns from experience

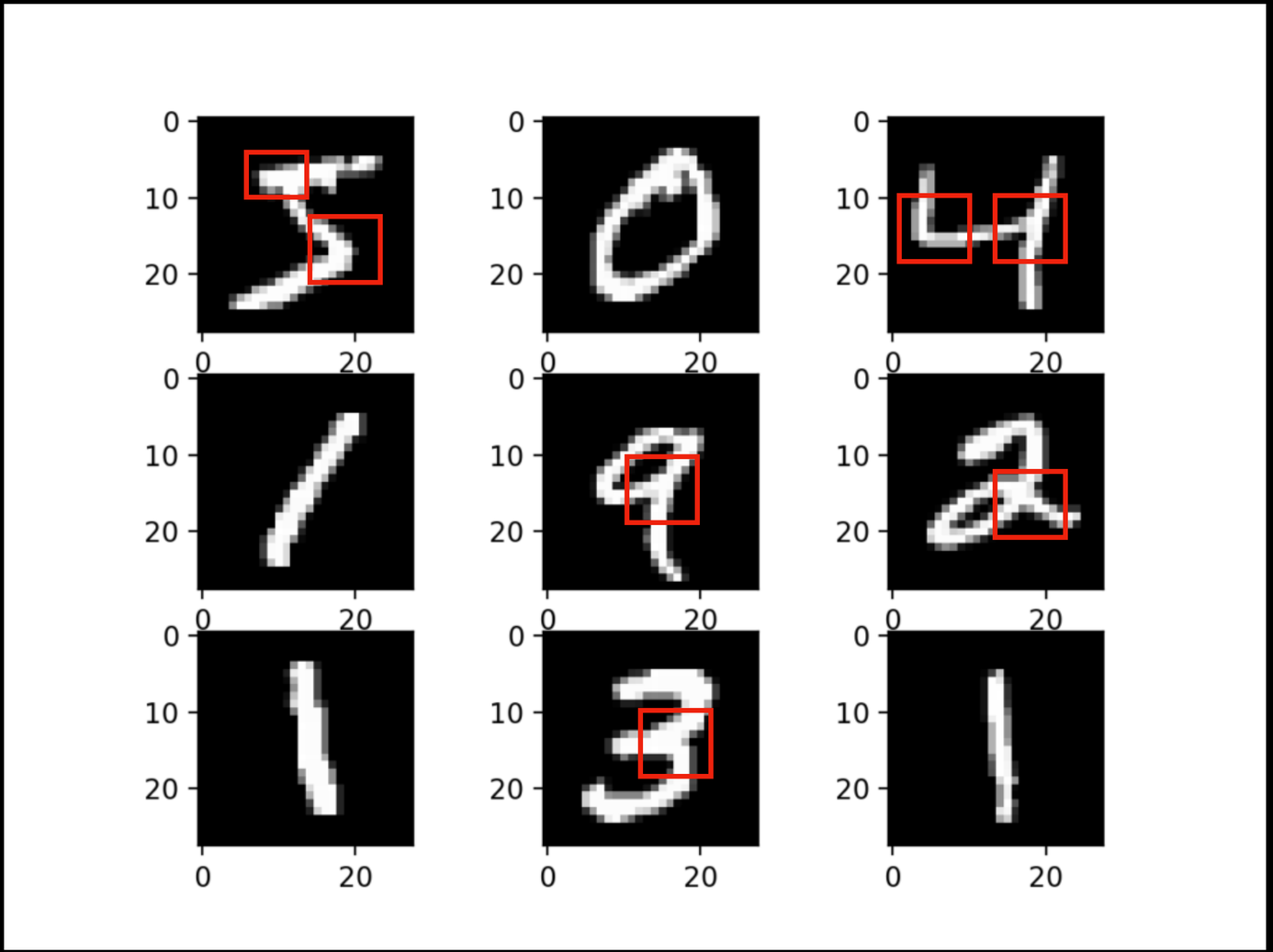
Patterns: the features shared by all samples of one digit



How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

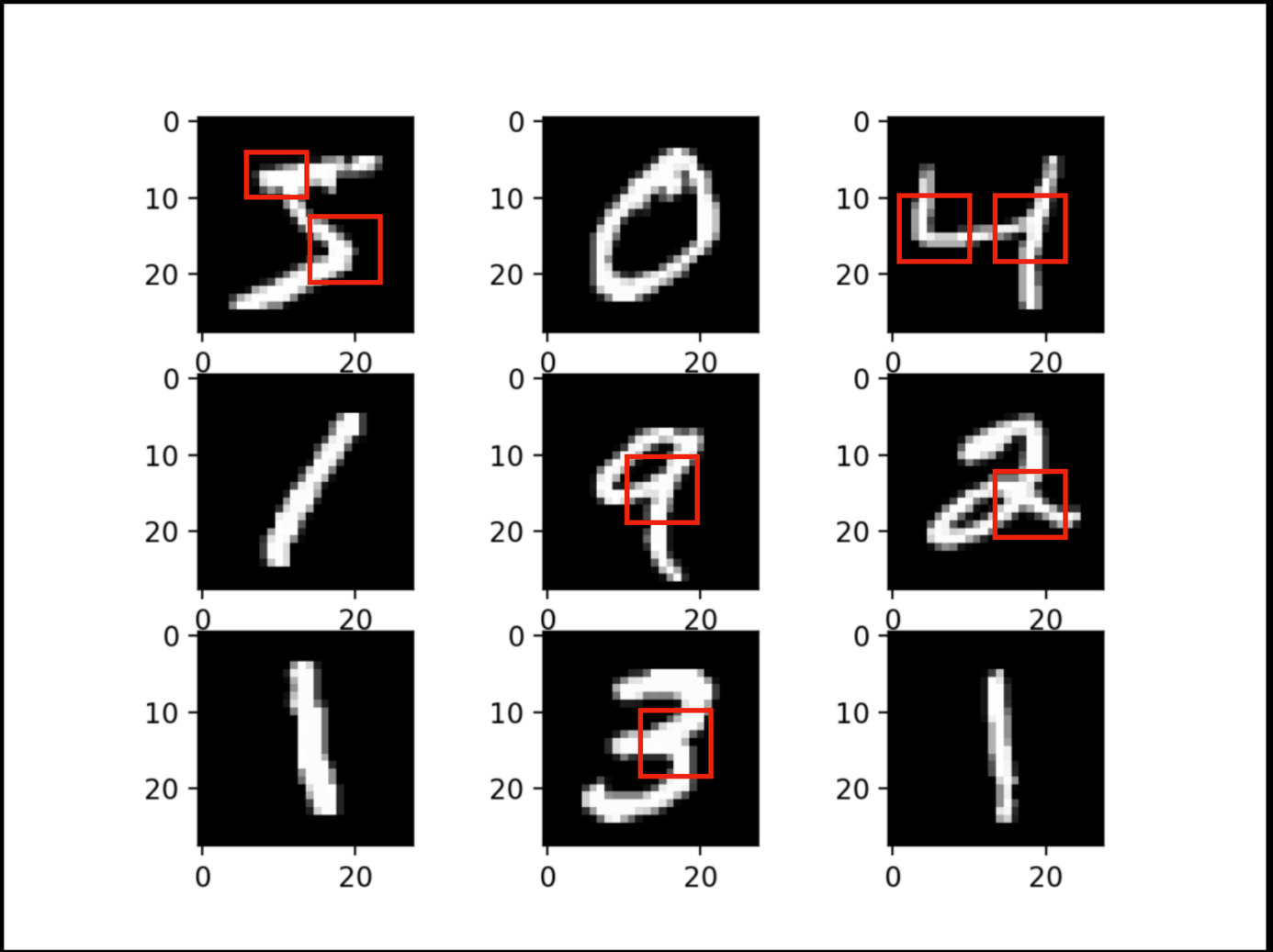


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”

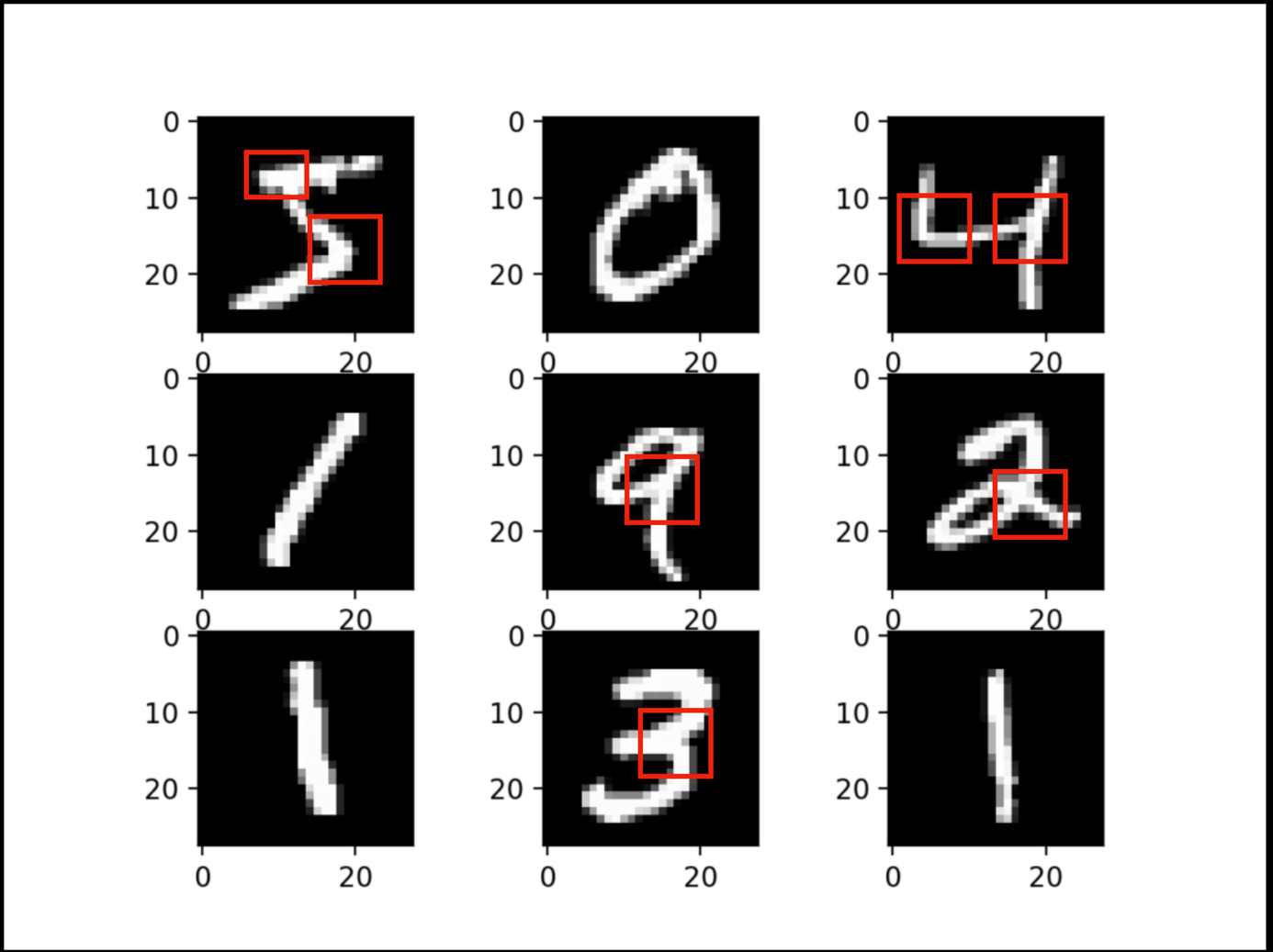


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”

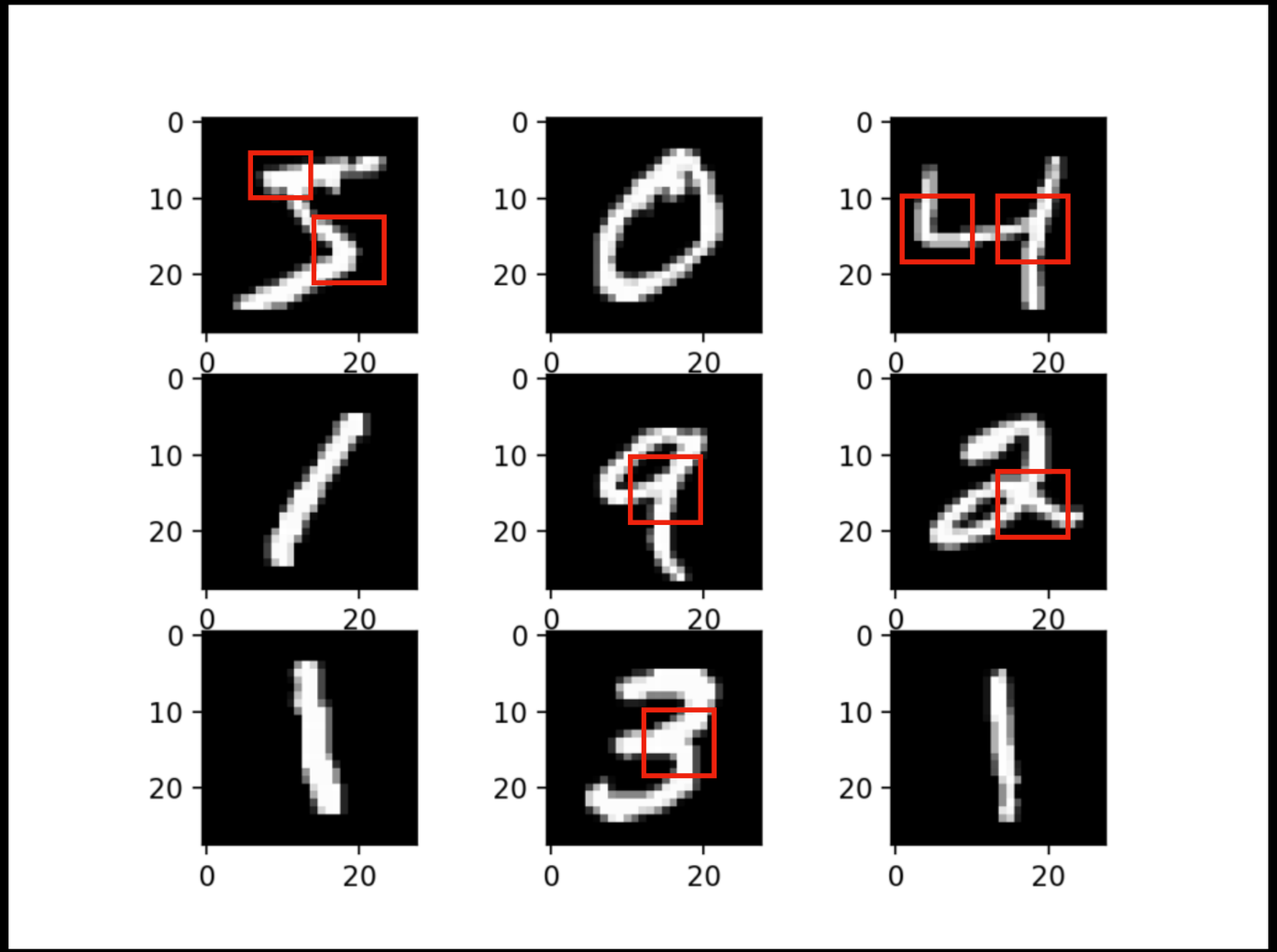


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”

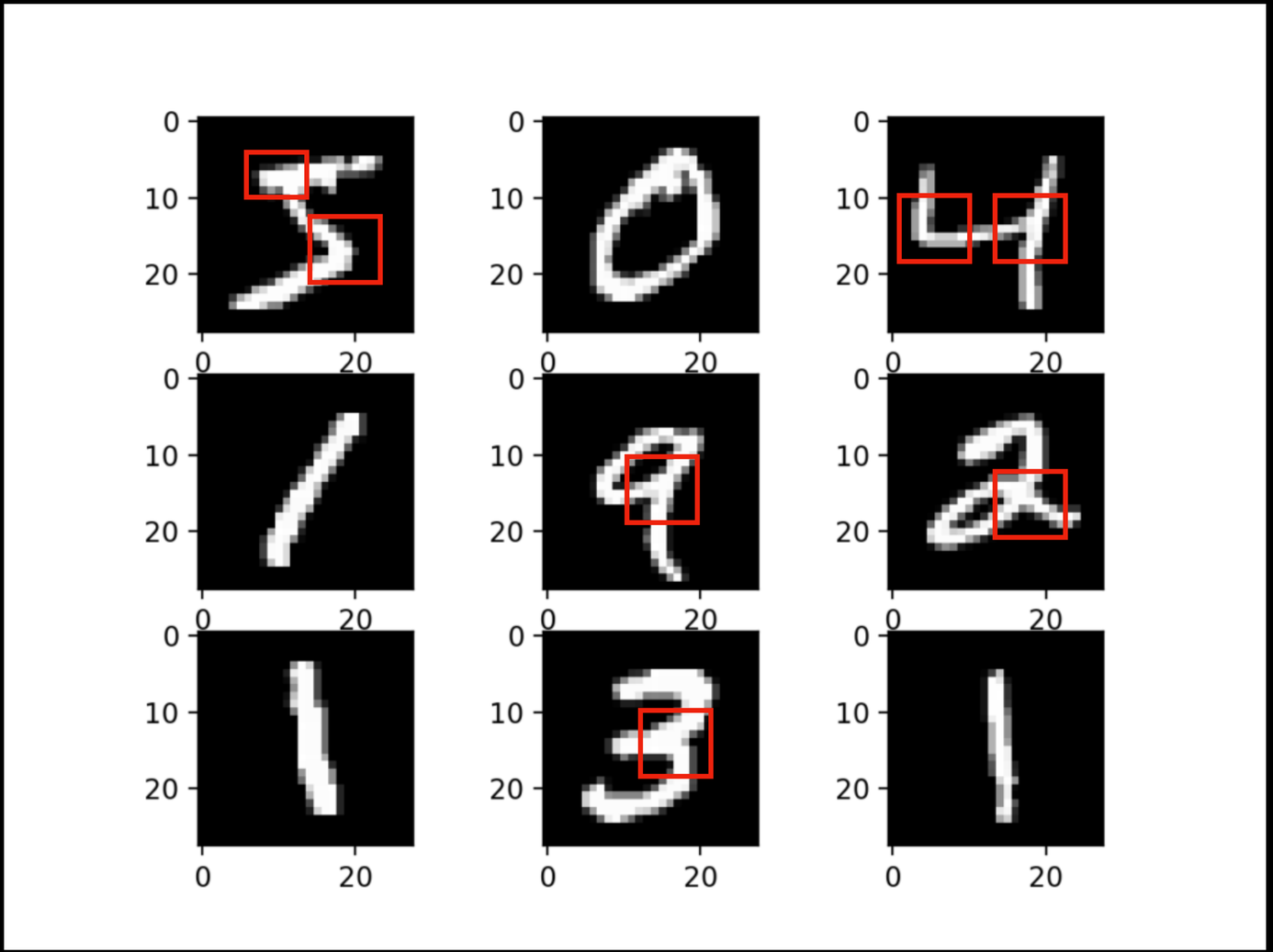


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”

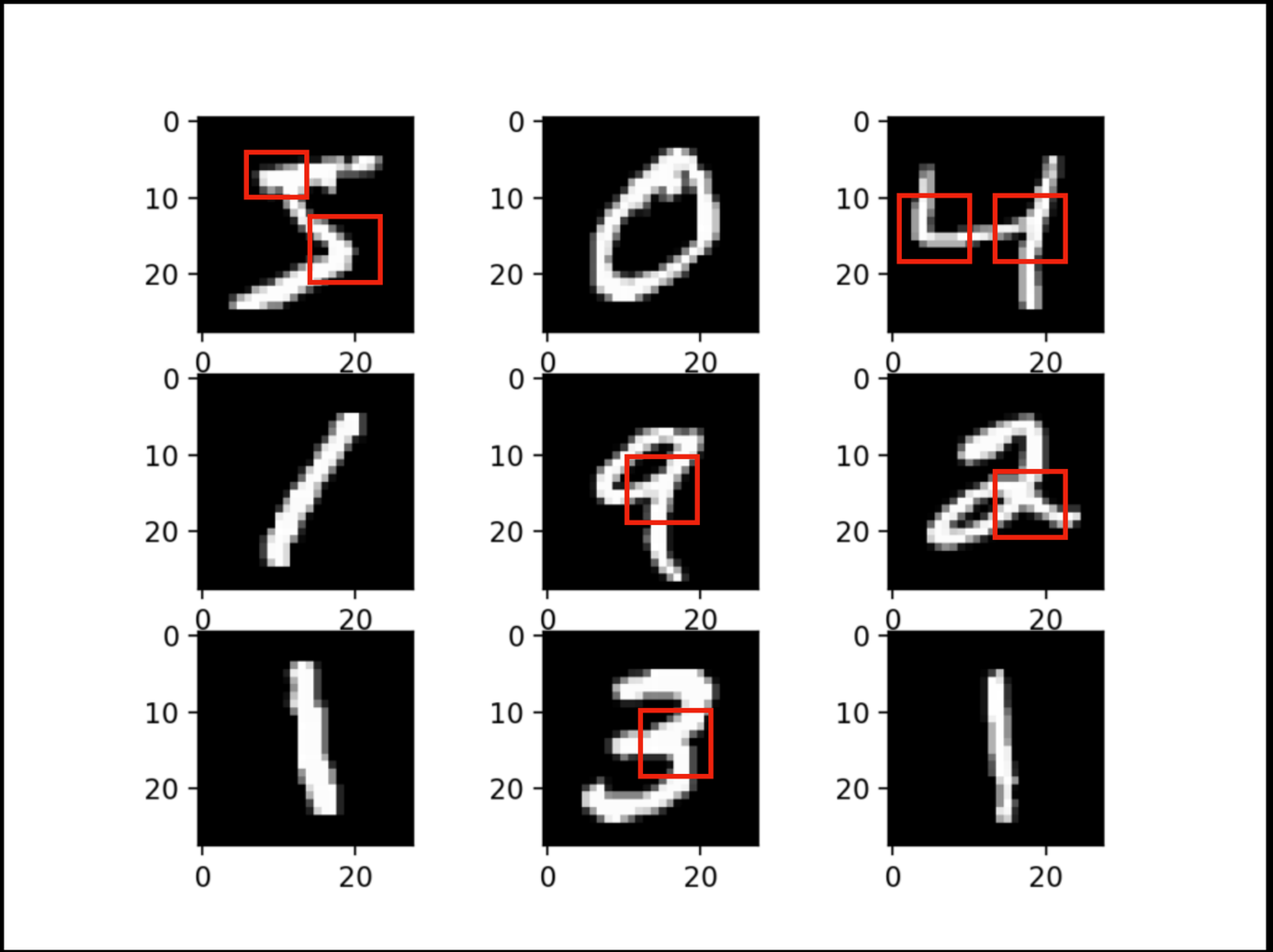


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”

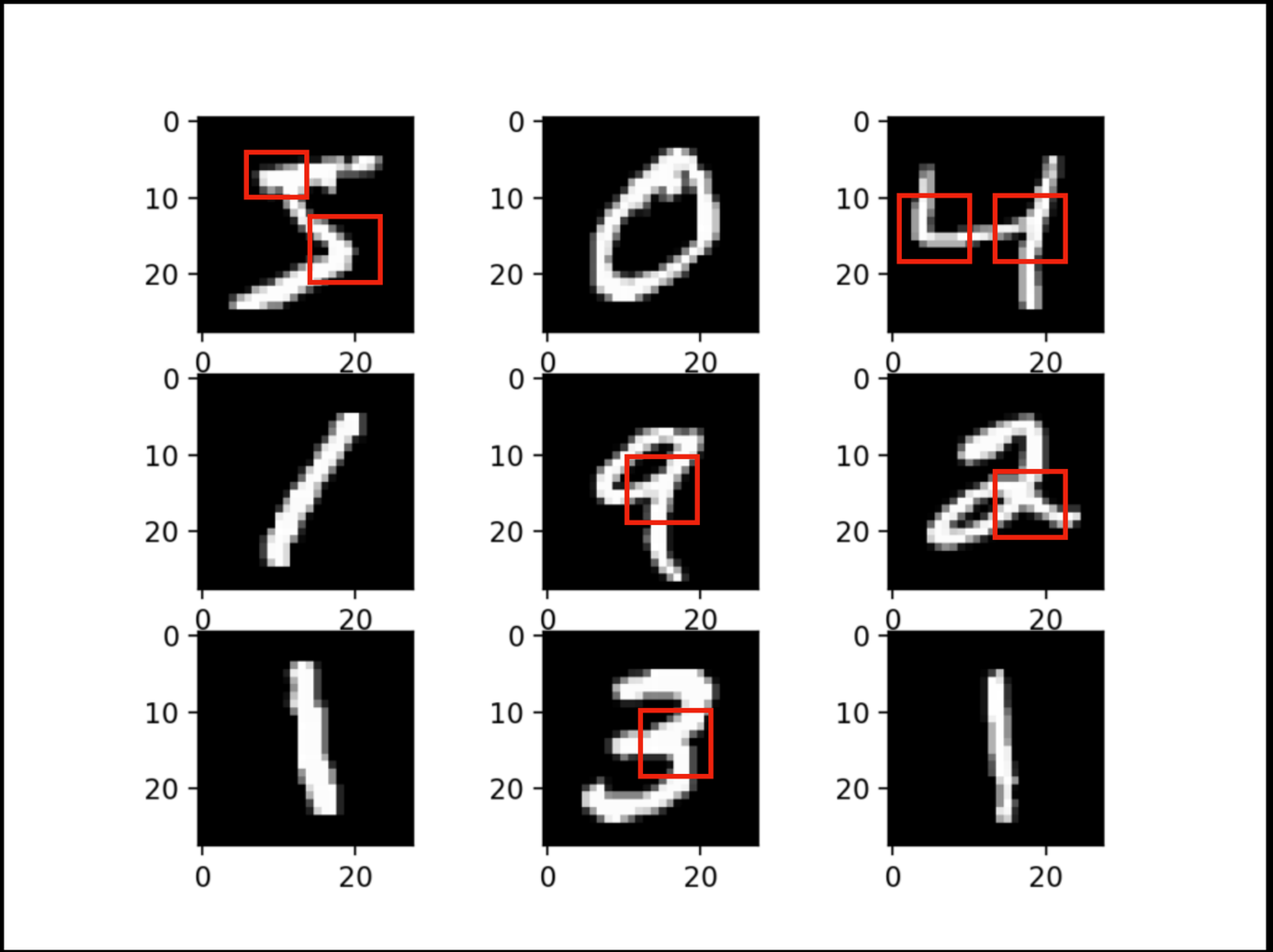


How do we recognize the digit?

By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”



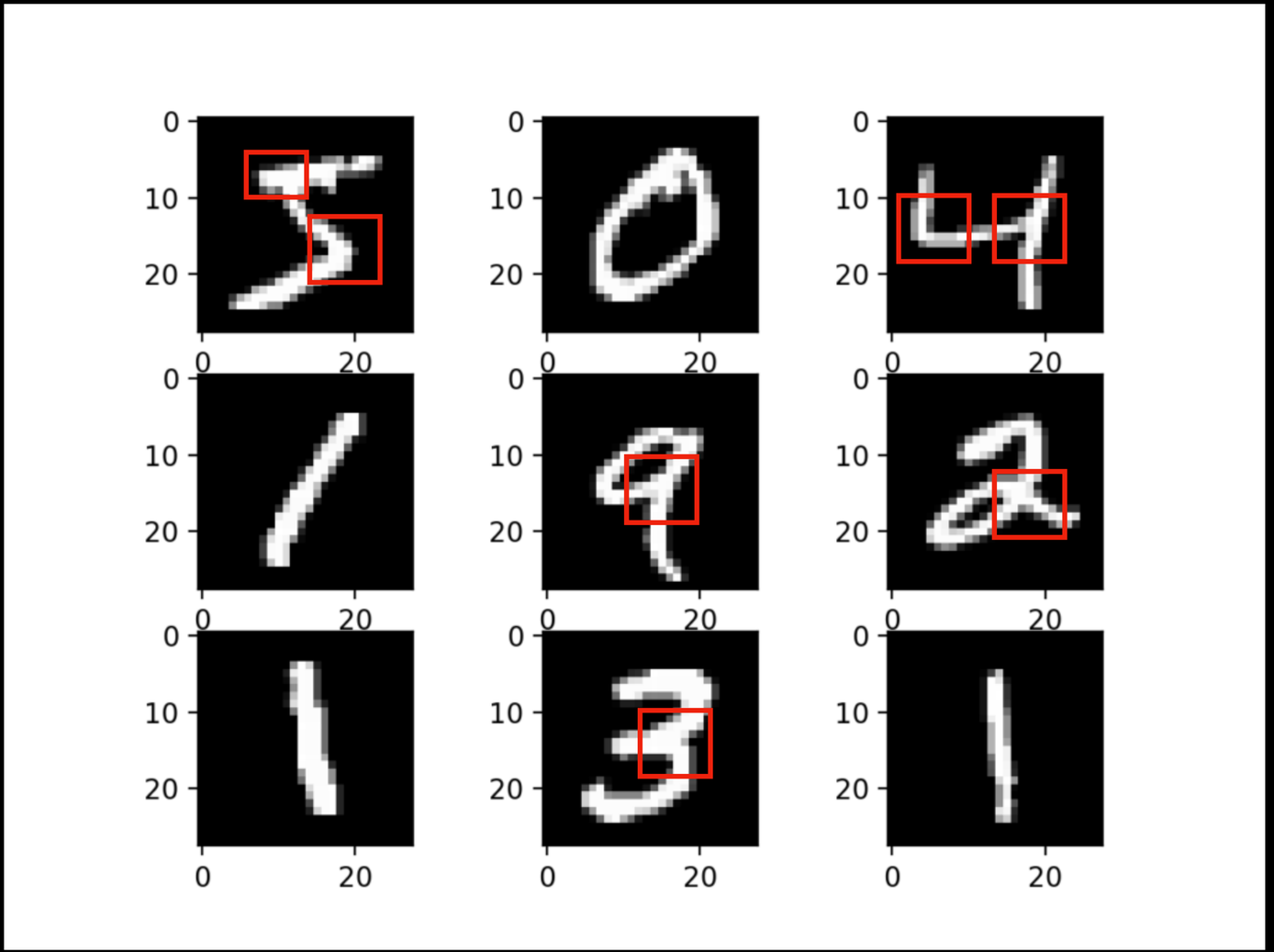
How do we recognize the digit?

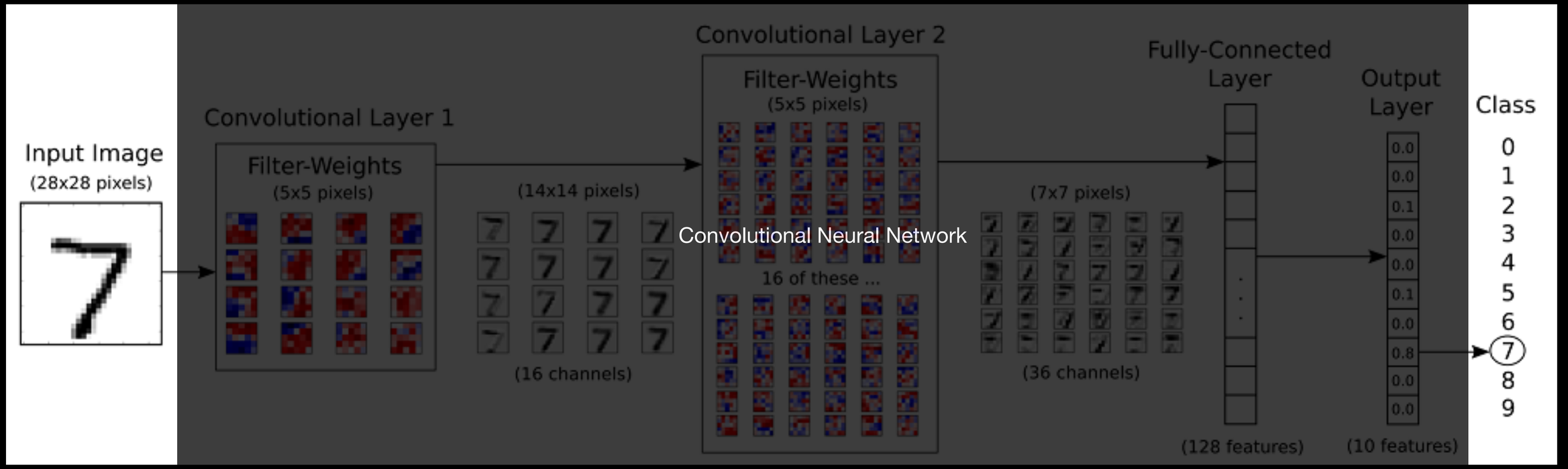
By summarized patterns from experience

Patterns: the features shared by all samples of one digit

“0” — a “circle”
“1” — a “close-to-vertical line”
“9” — a “circle” + a “close-to-vertical line”
“5” — two reversed “corners”
“3” — the “connection between up and bottom”

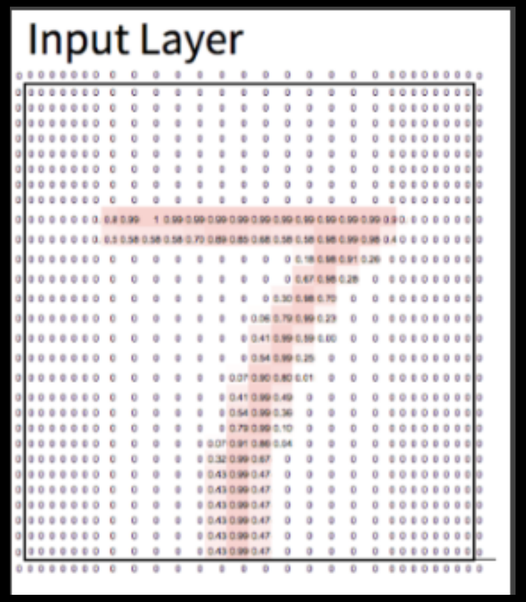
features

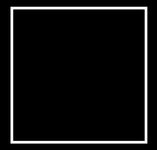



Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected Layer
(128 features)
Output Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9
Convolutional Neural Network

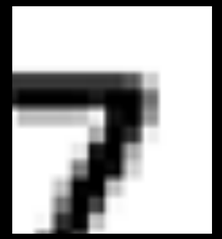
Input Layer

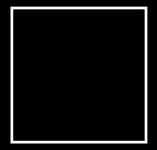
Patterns



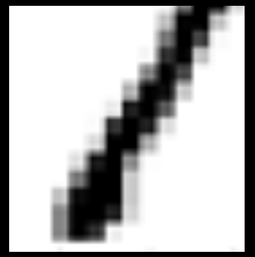


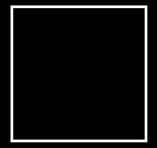
feature 1





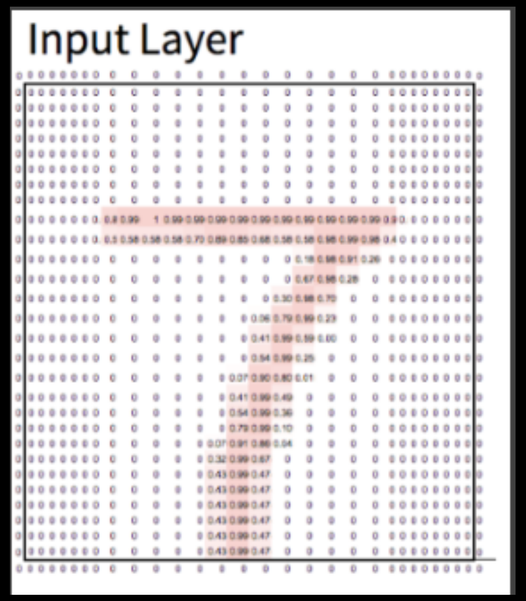
feature 2

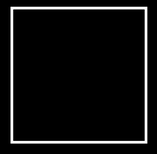




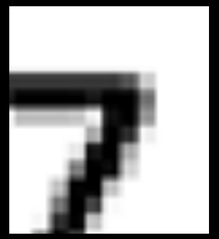
feature 3

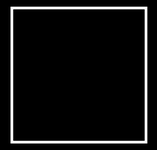
Patterns



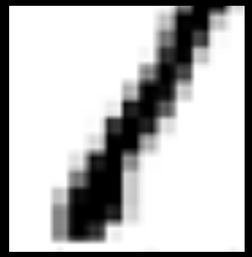


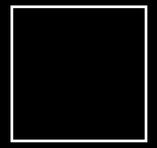
feature 1





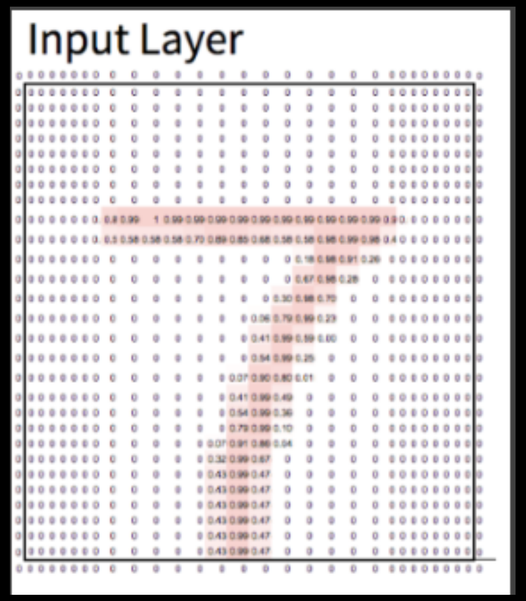
feature 2



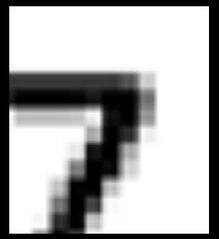


feature 3

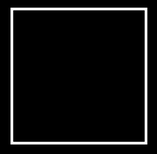
Patterns

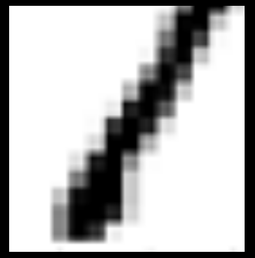


feature 1

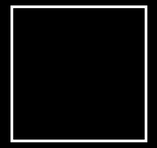


feature 2

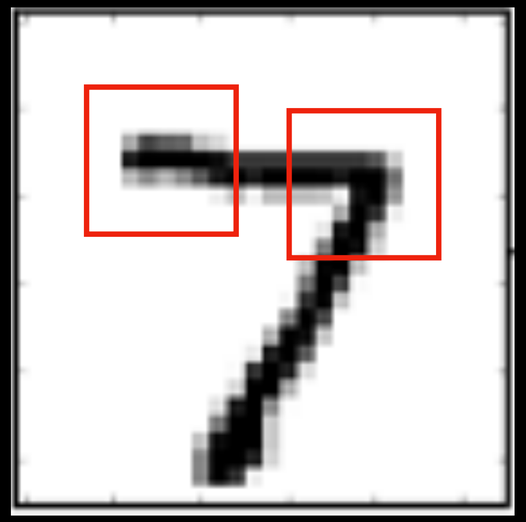


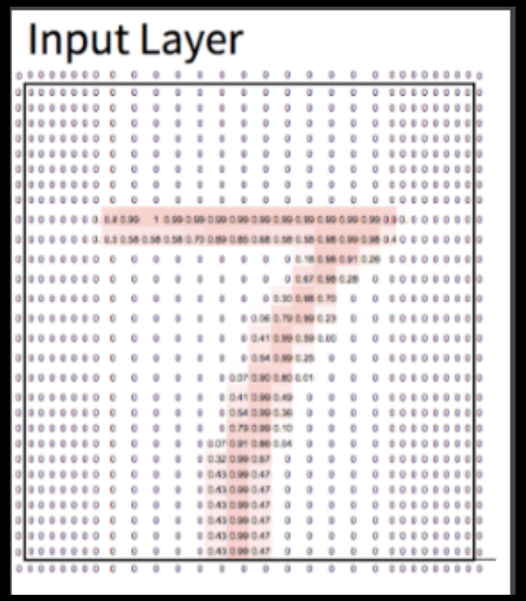


feature 3

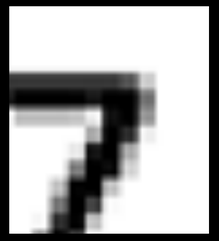


Patterns

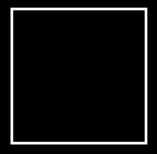


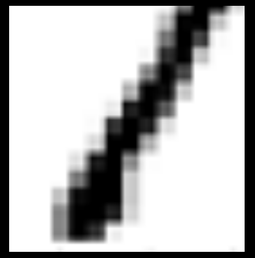


feature 1

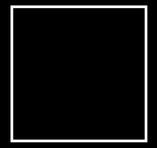


feature 2

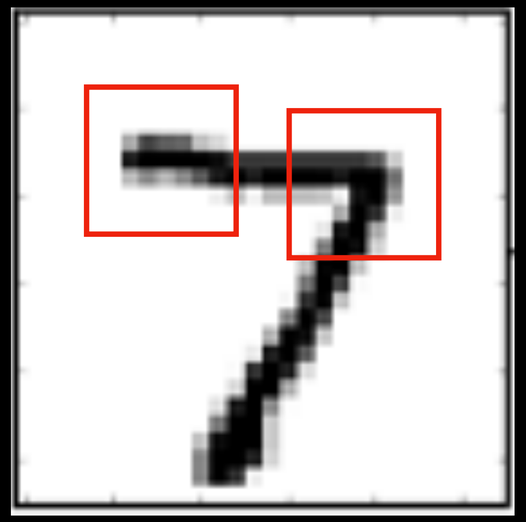


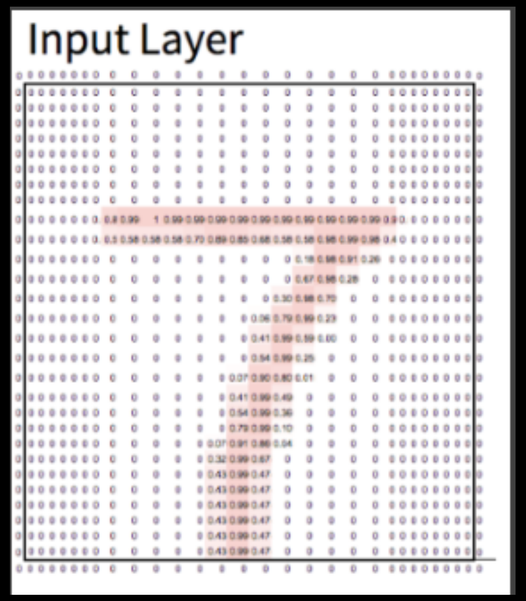


feature 3

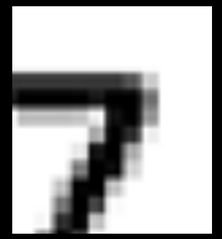


Patterns

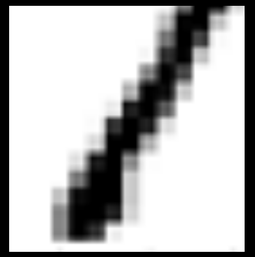




feature 1

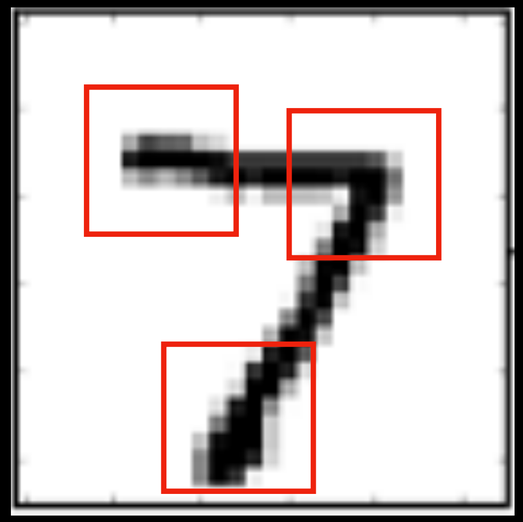


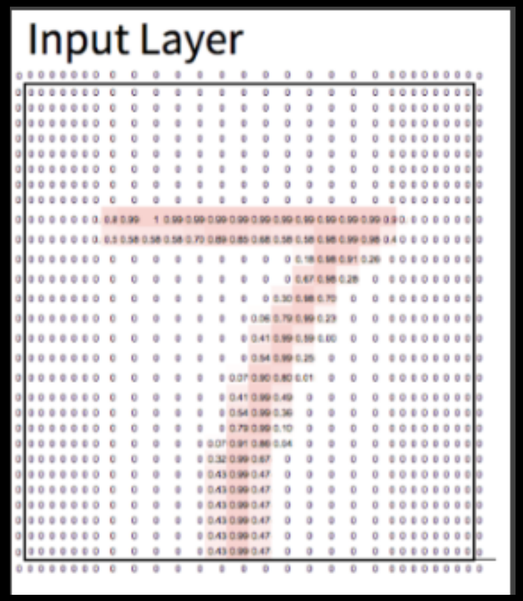
feature 2



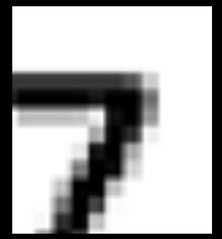
feature 3

Patterns

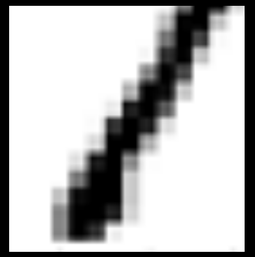




feature 1

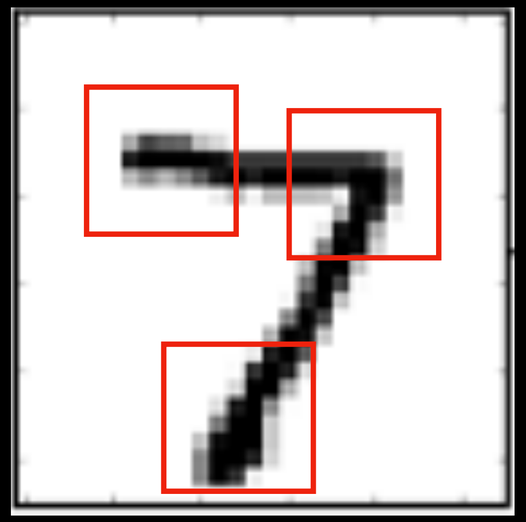


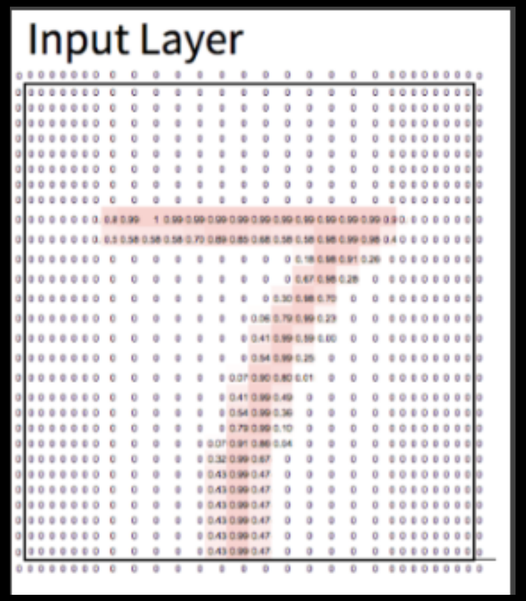
feature 2



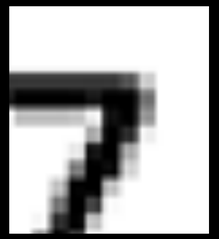
feature 3

Patterns

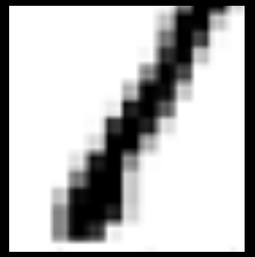




feature 1

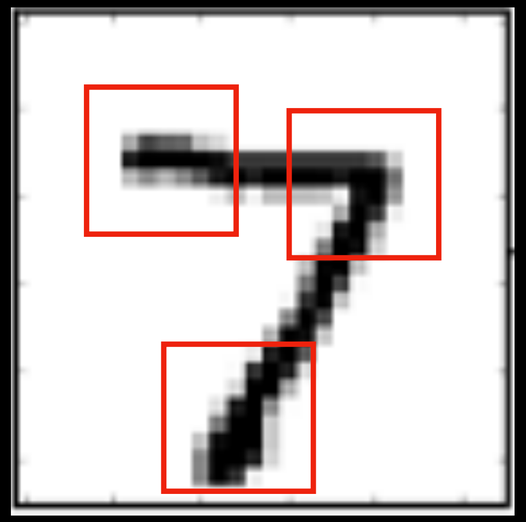


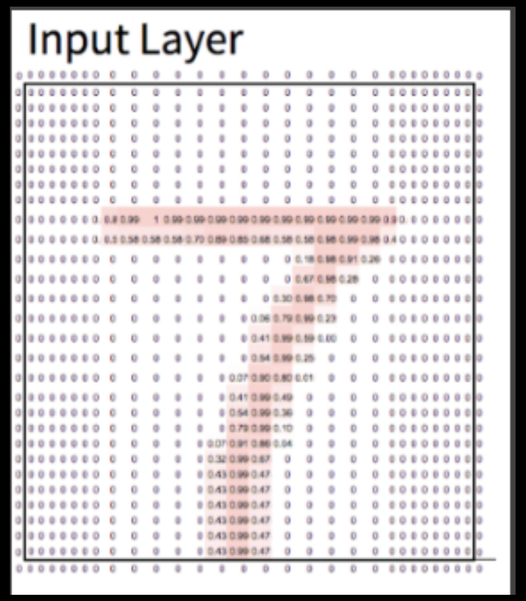
feature 2



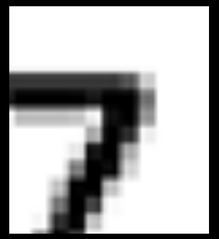
feature 3

Patterns

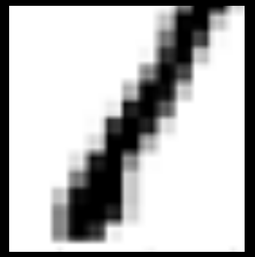




feature 1

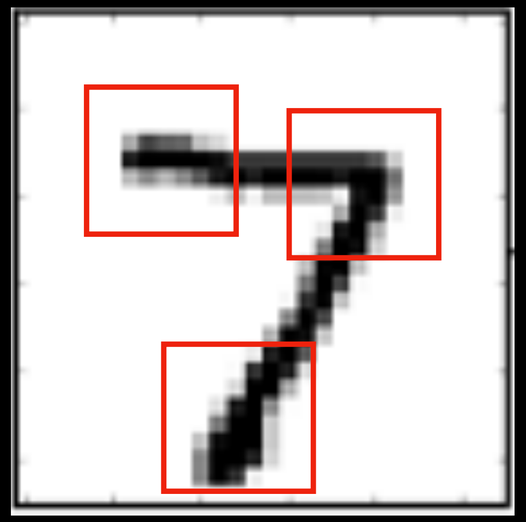


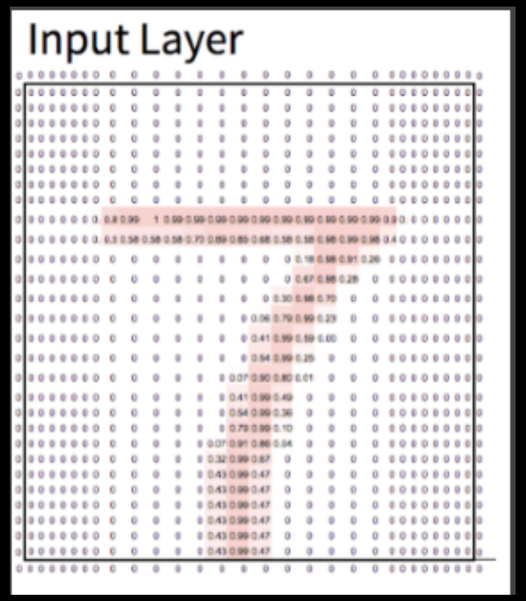
feature 2



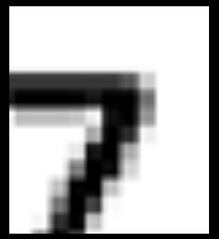
feature 3

Patterns

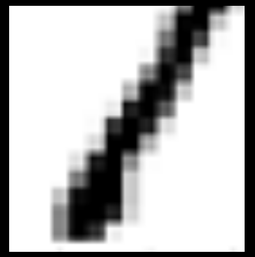




feature 1



feature 2



feature 3

It is a “7”

Neural Network is an algorithm used to **recognize patterns** in data.

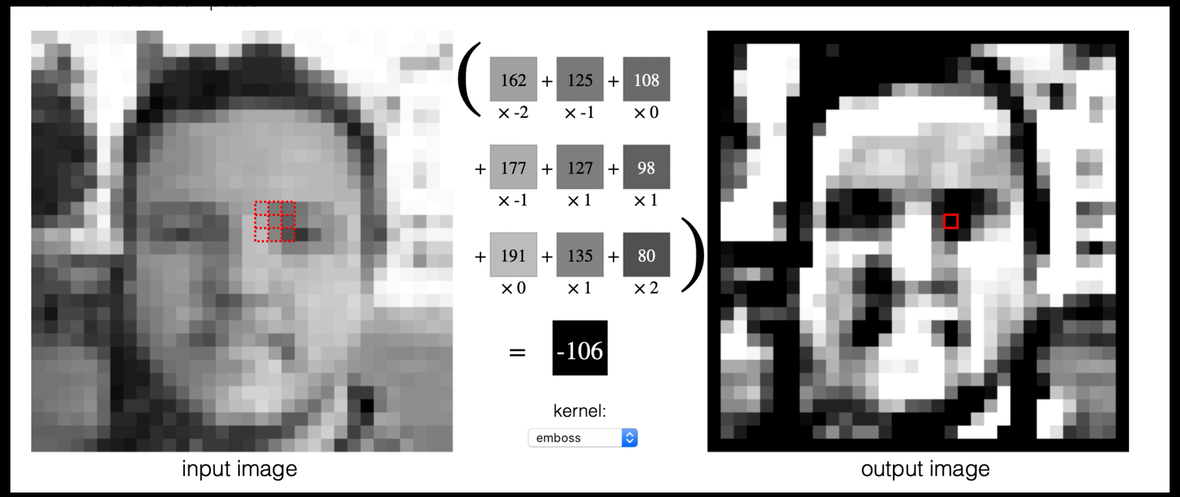
Convolutional Neural Network uses convolutions to extract **image features** and **recognize feature patterns**.

Neural Network is an algorithm used to **recognize patterns** in data.
Convolutional Neural Network uses convolutions to extract **image features** and **recognize feature patterns**.

Extract image features -> find feature patterns for each category

Extract Features

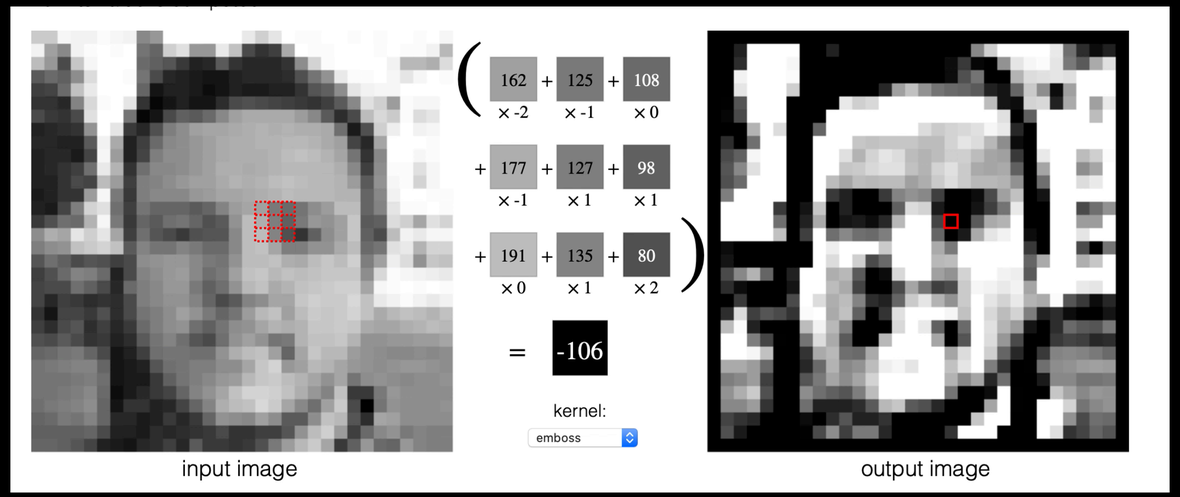
Convolution



https://setosa.io/ev/image-kernels/

Extract Features

Convolution



https://setosa.io/ev/image-kernels/

Extract Features

Convolution

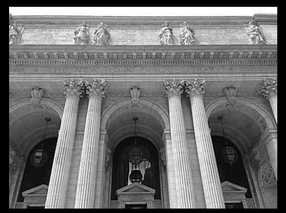
blur

sobel

outline

emboss

sharpen



Extract Features

Convolution

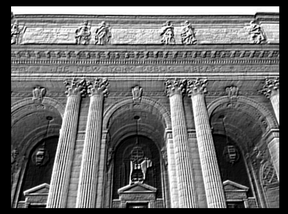
blur

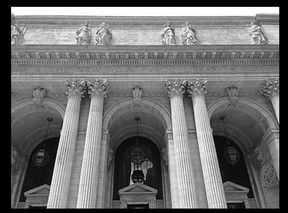
sobel

outline

emboss

sharpen





Edges

Extract Features

Convolution

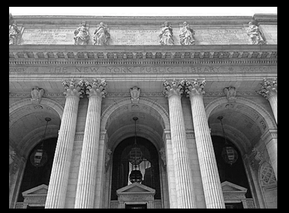
blur

sobel

outline

emboss

sharpen



Edges

Corners

Extract Features

Convolution

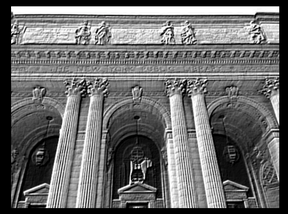
blur

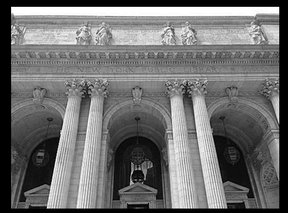
sobel

outline

emboss

sharpen





Edges

Corners

Lines

Extract Features

Convolution

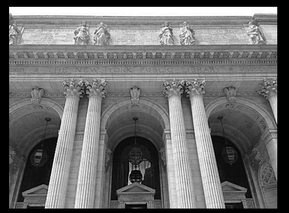
blur

sobel

outline

emboss

sharpen



Edges

Corners

Lines

Circles

Extract Features

Convolution

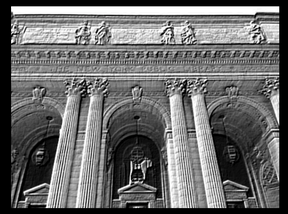
blur

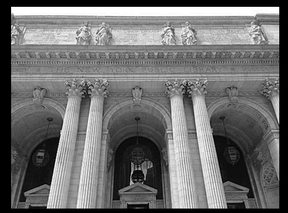
sobel

outline

emboss

sharpen





Edges

Corners

Lines

Circles

...

Extract Features

Convolution

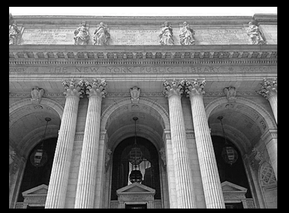
blur

sobel

outline

emboss

sharpen



Edges

Corners

Lines

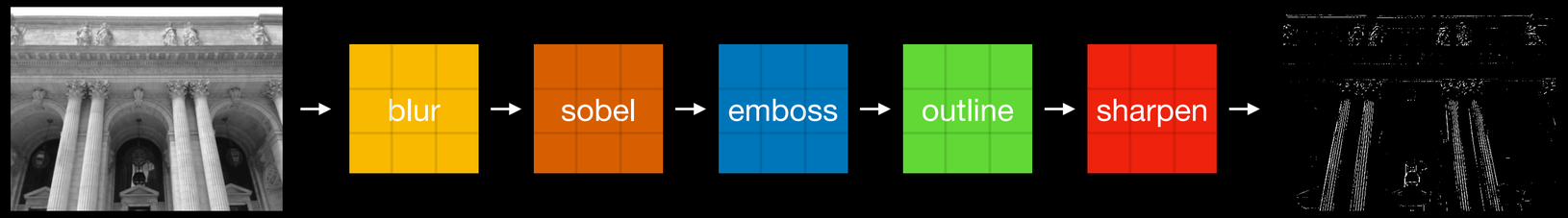
Circles

...

Basic Features

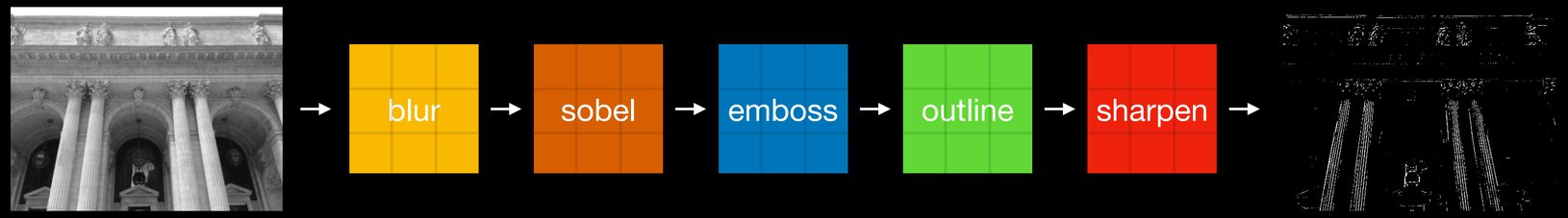
Extract Features

Convolution for high-level feature



Extract Features

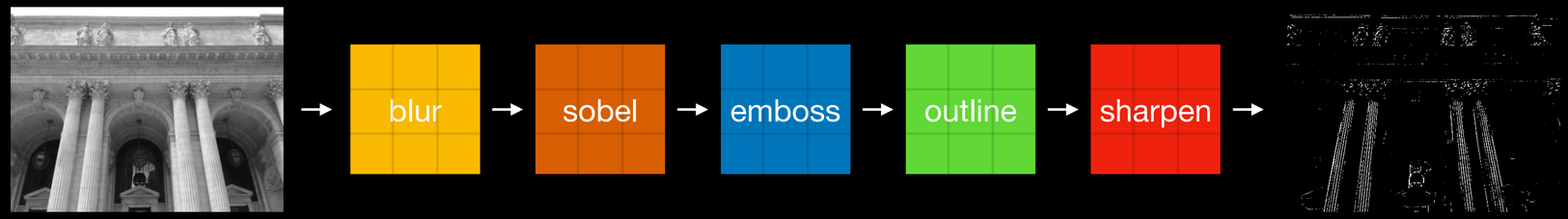
Convolution for high-level feature



Columns

Extract Features

Convolution for high-level feature

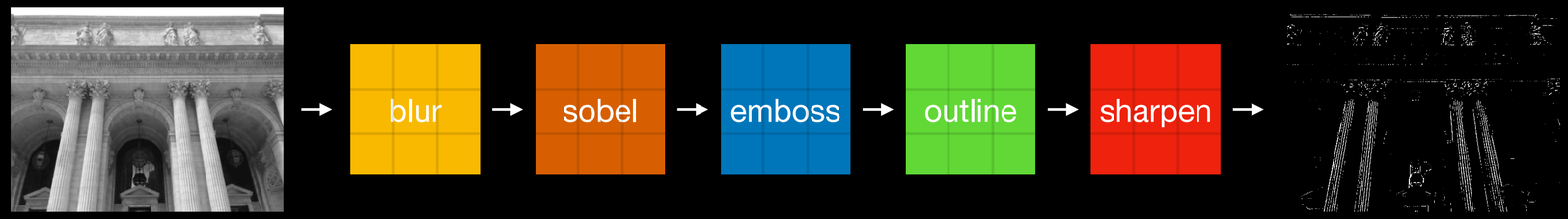


Columns

Statues

Extract Features

Convolution for high-level feature



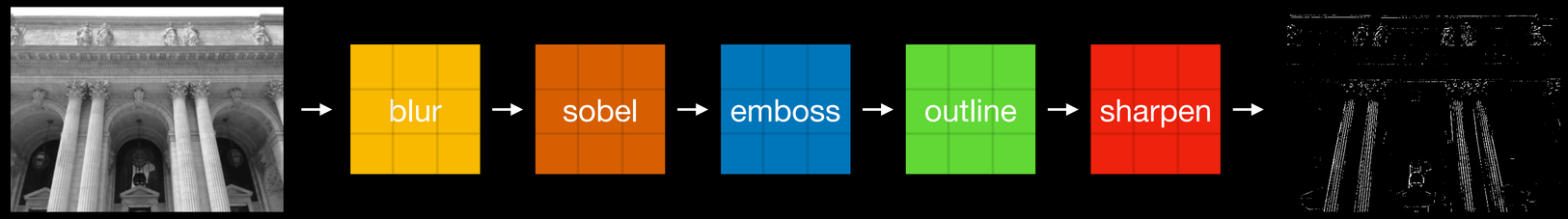
Columns

Statues

...

Extract Features

Convolution for high-level feature



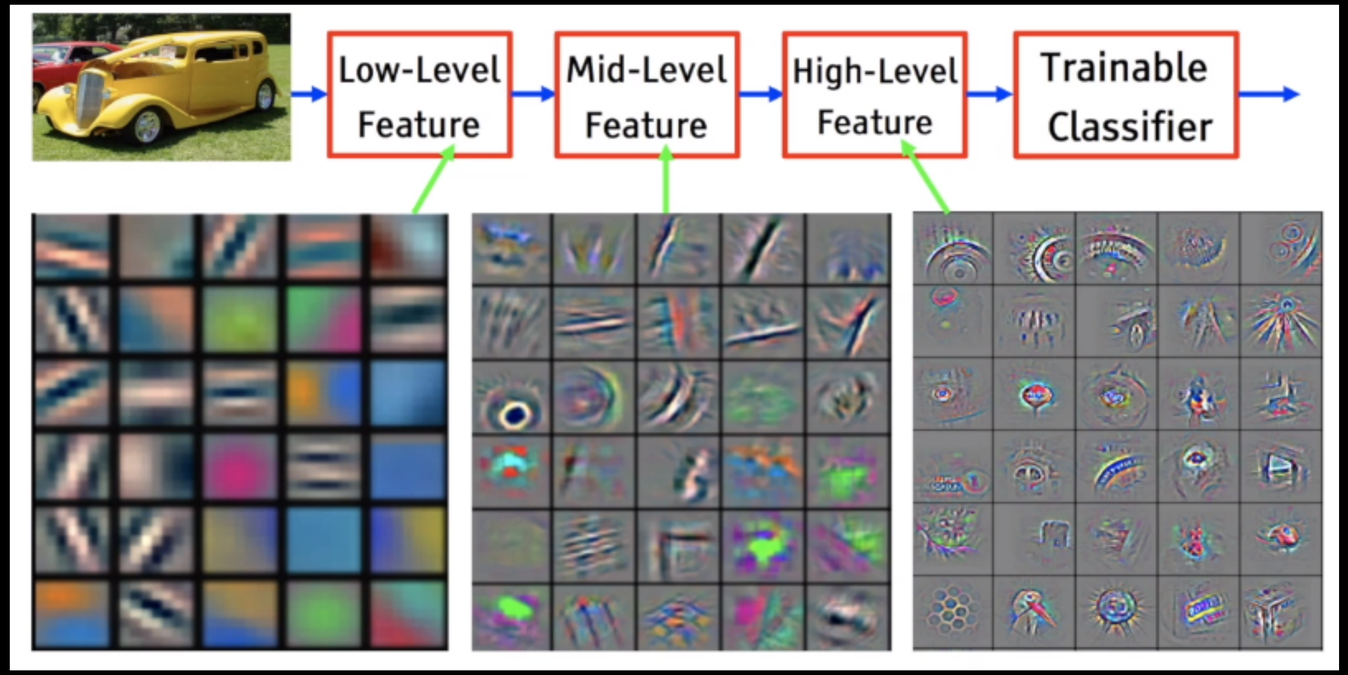
Columns

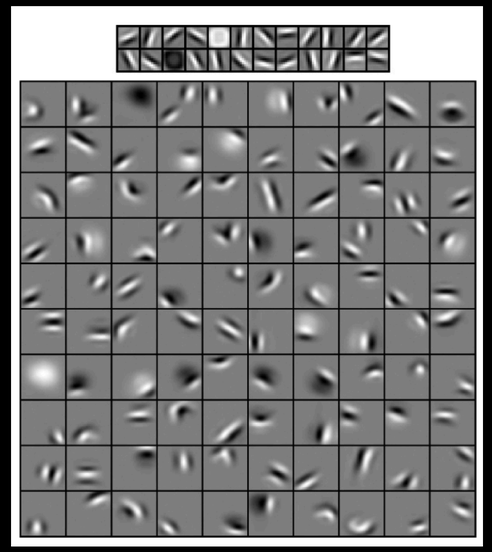
Statues

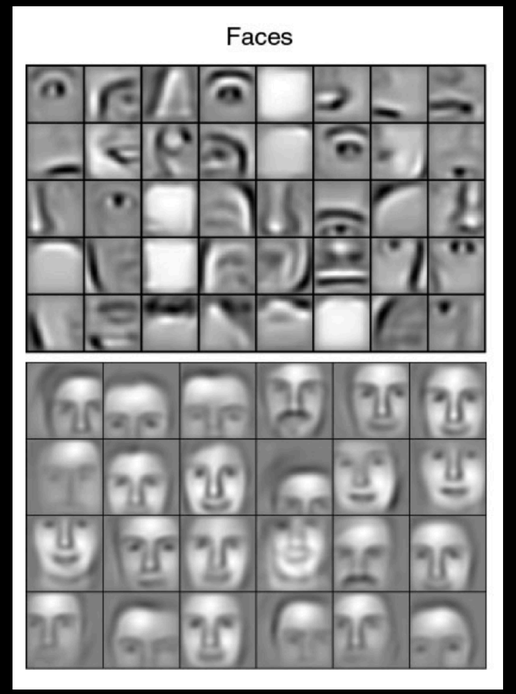
...

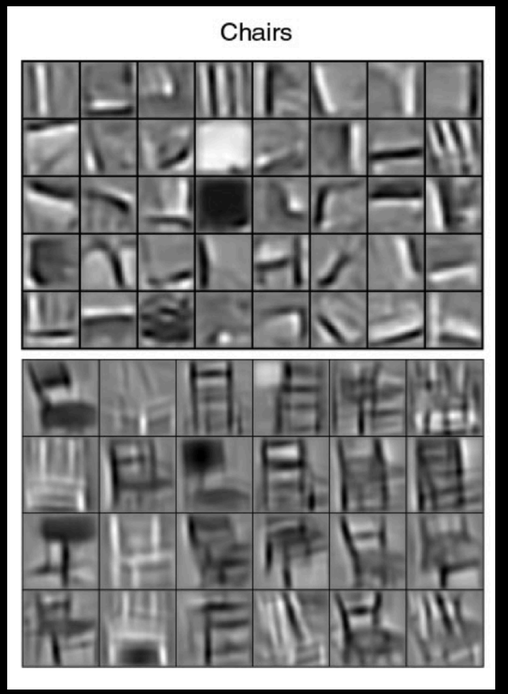
Complex Features

Features in different levels









Feature Map

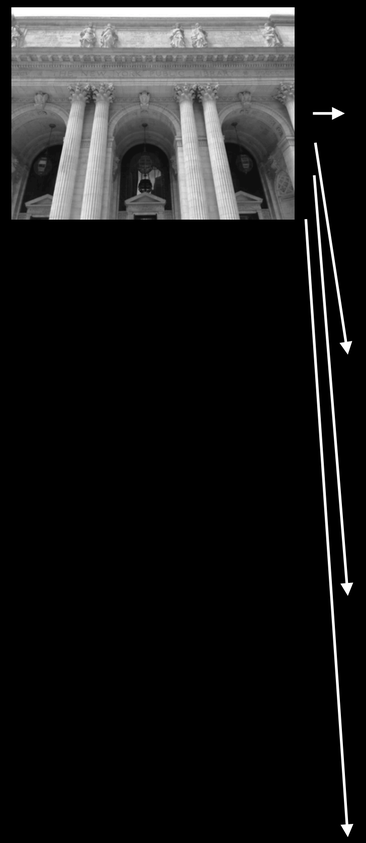
Convolutional Neural Network

Feature Map

Convolutional Neural Network

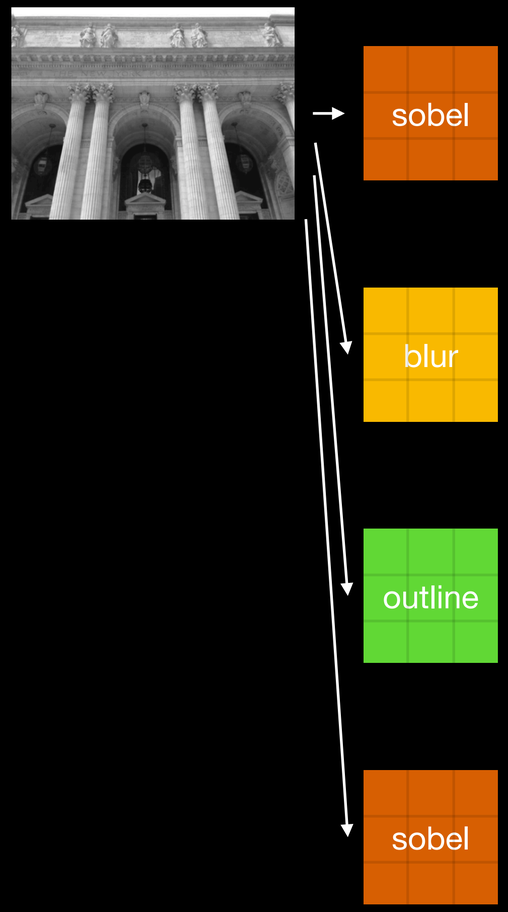
Feature Map

Convolutional Neural Network



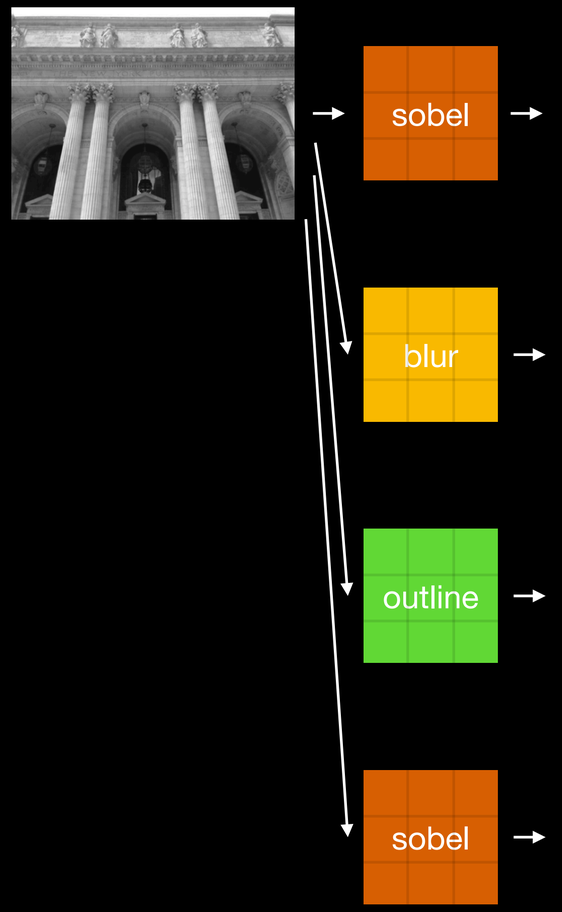
Feature Map

Convolutional Neural Network



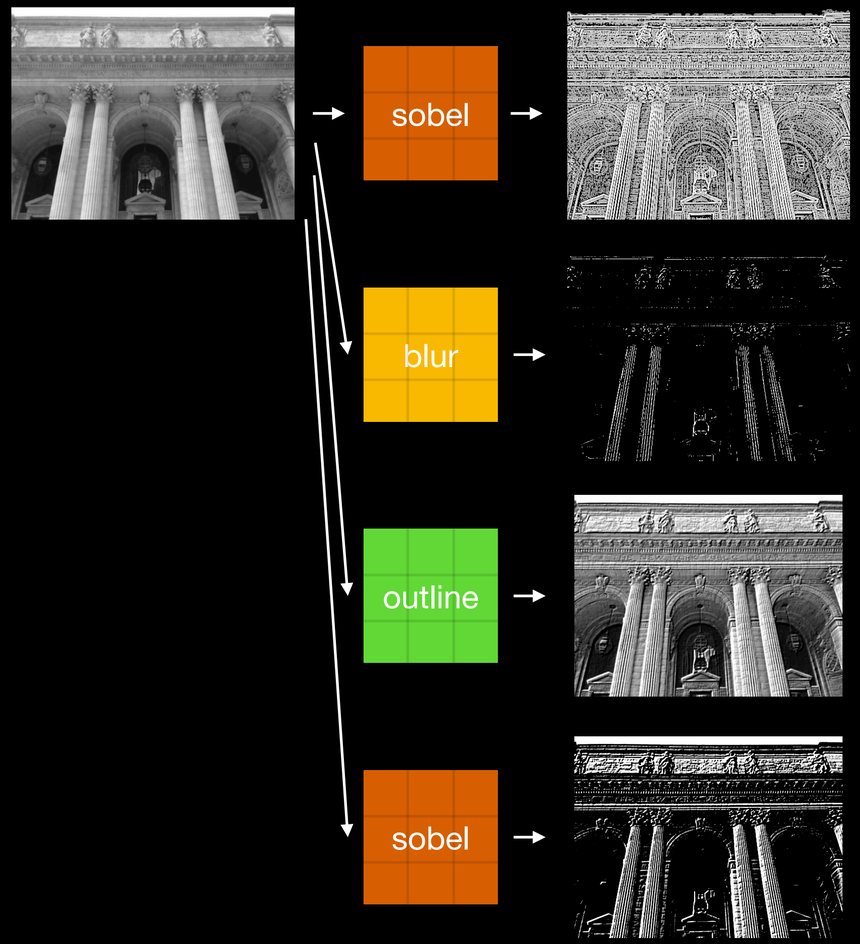
Feature Map

Convolutional Neural Network



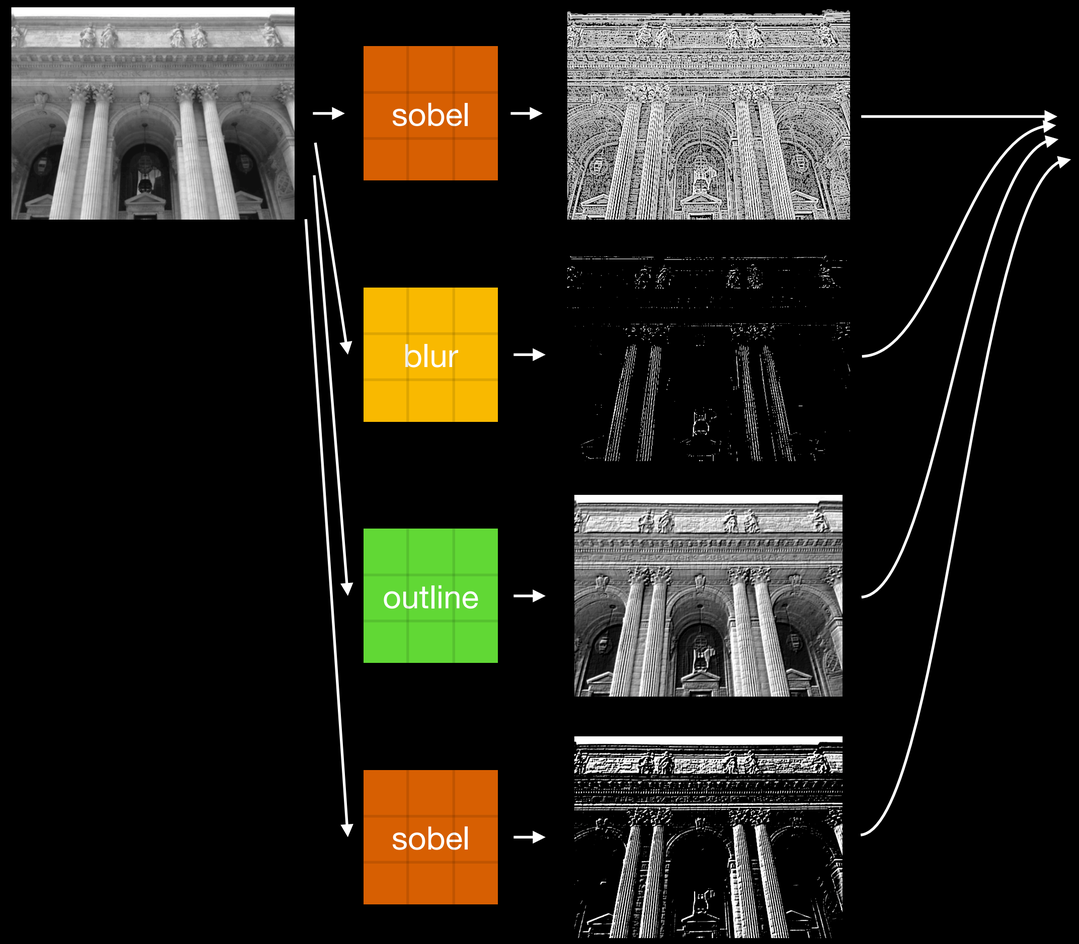
Feature Map

Convolutional Neural Network



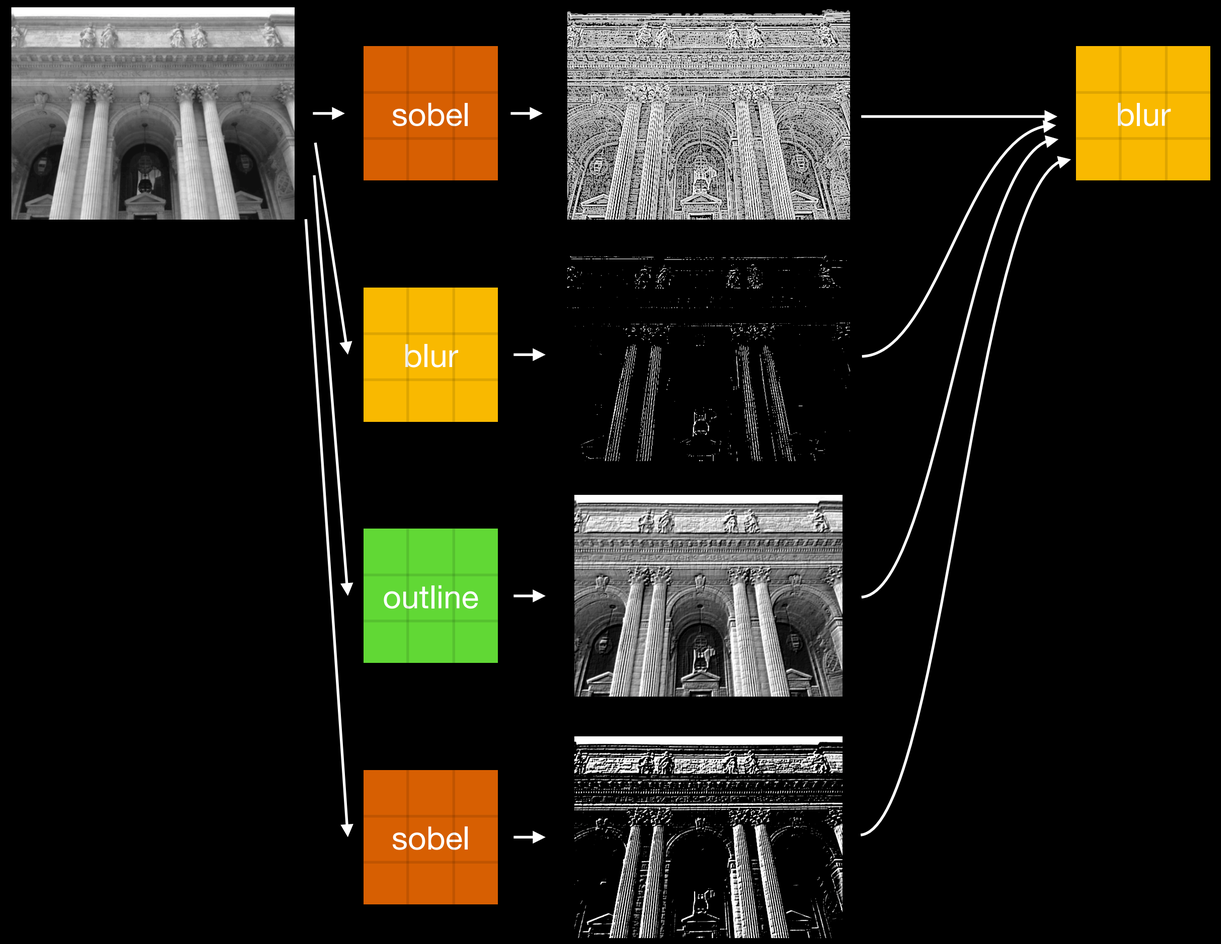
Feature Map

Convolutional Neural Network



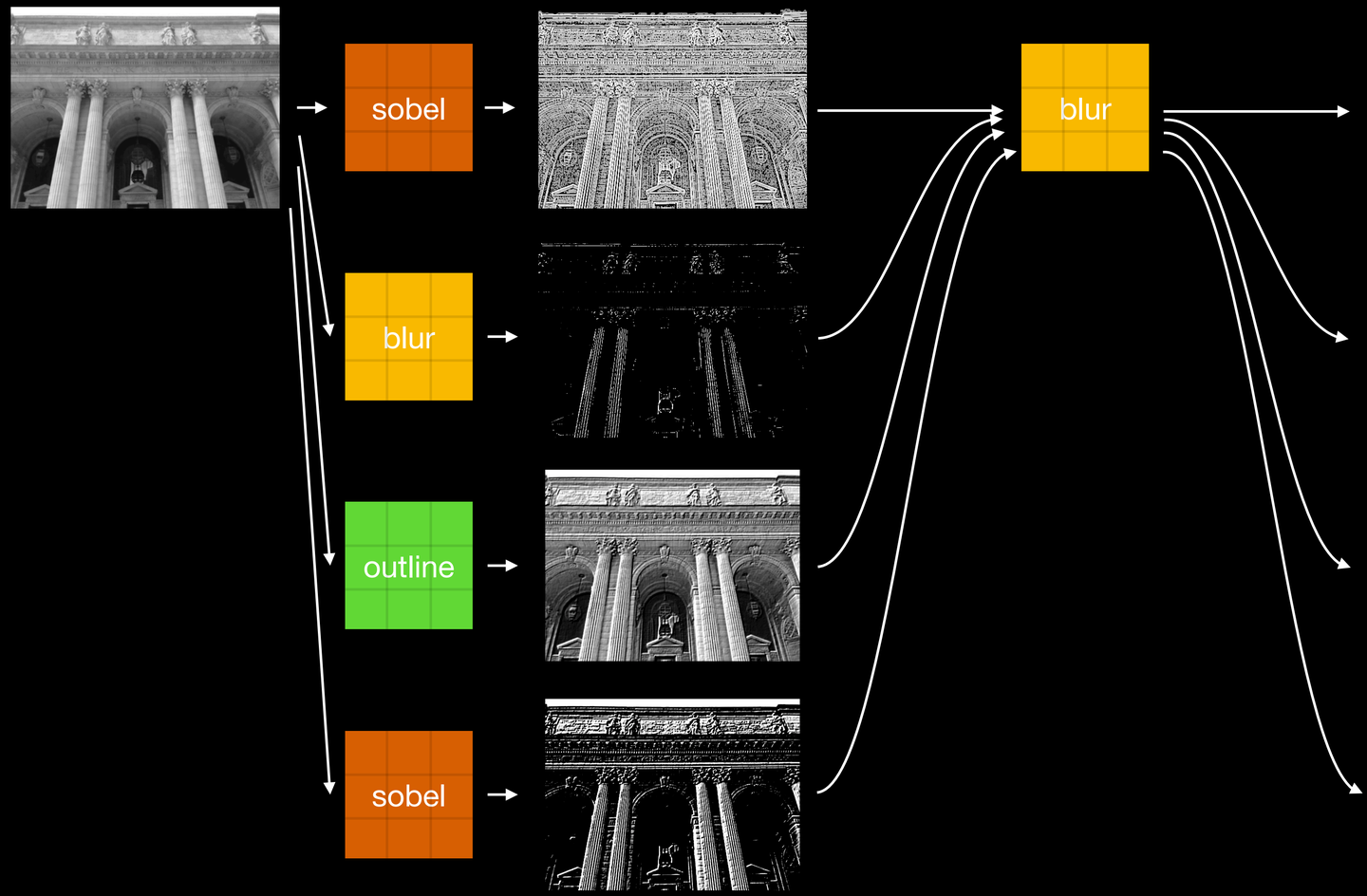
Feature Map

Convolutional Neural Network



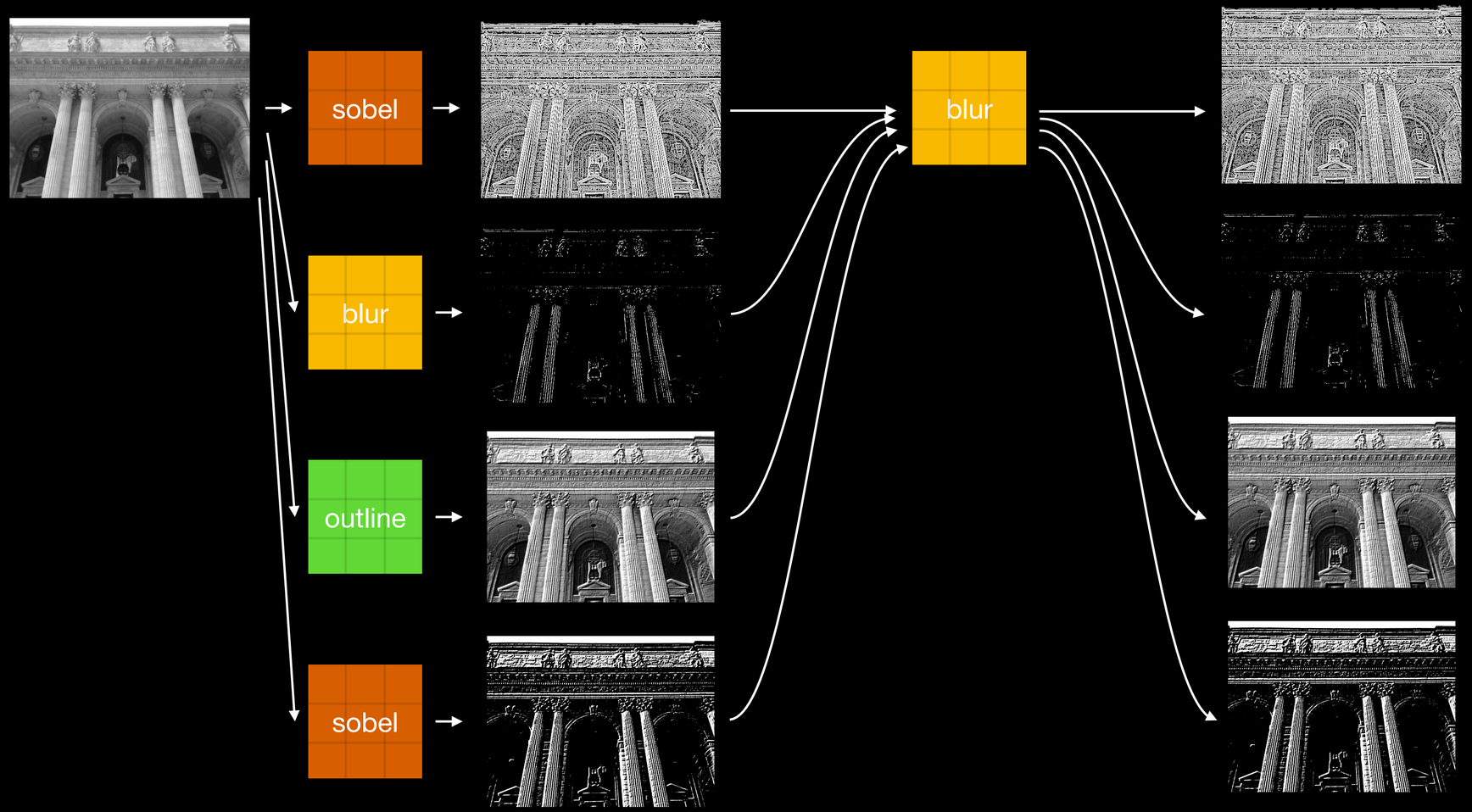
Feature Map

Convolutional Neural Network



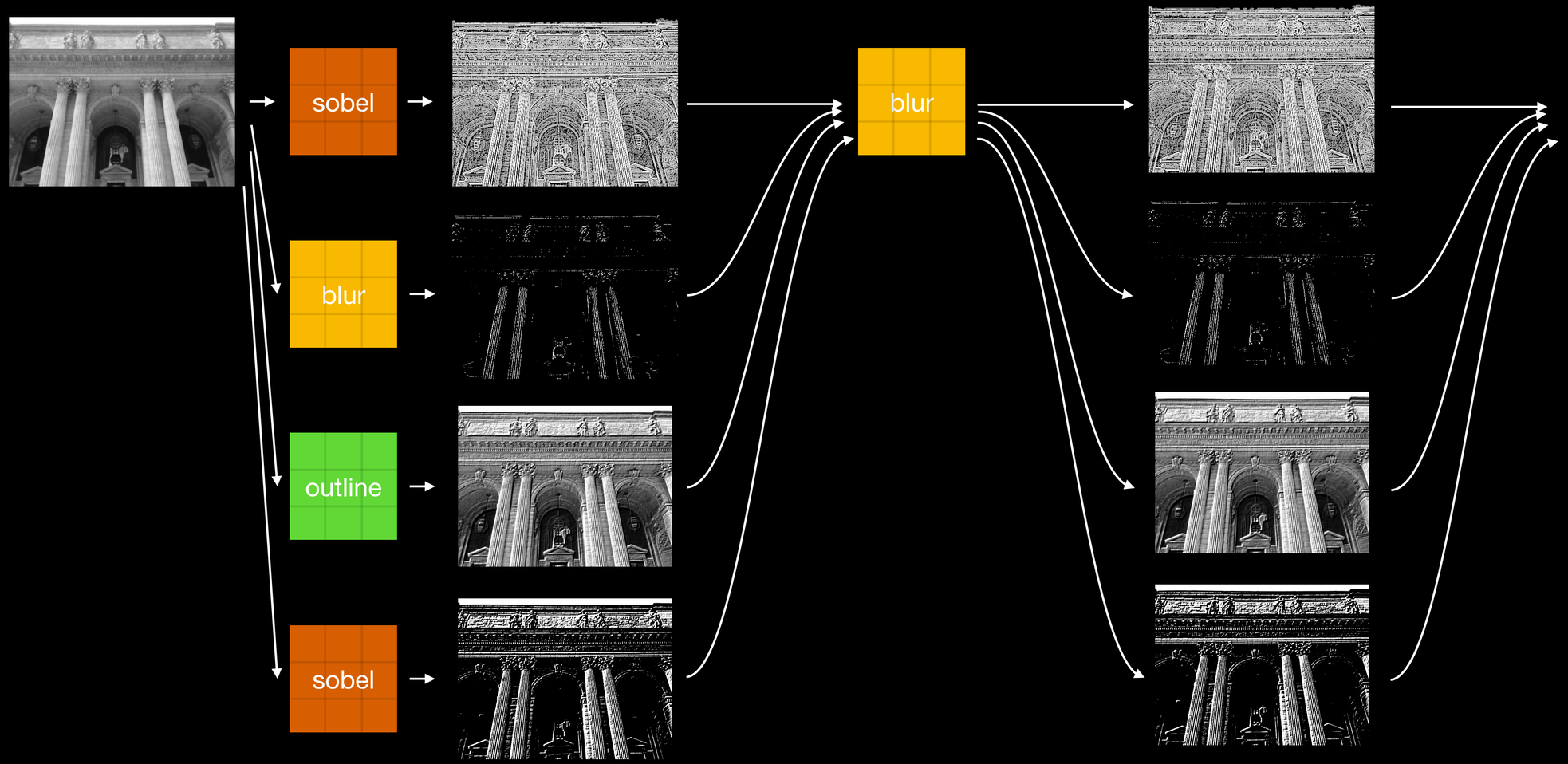
Feature Map

Convolutional Neural Network



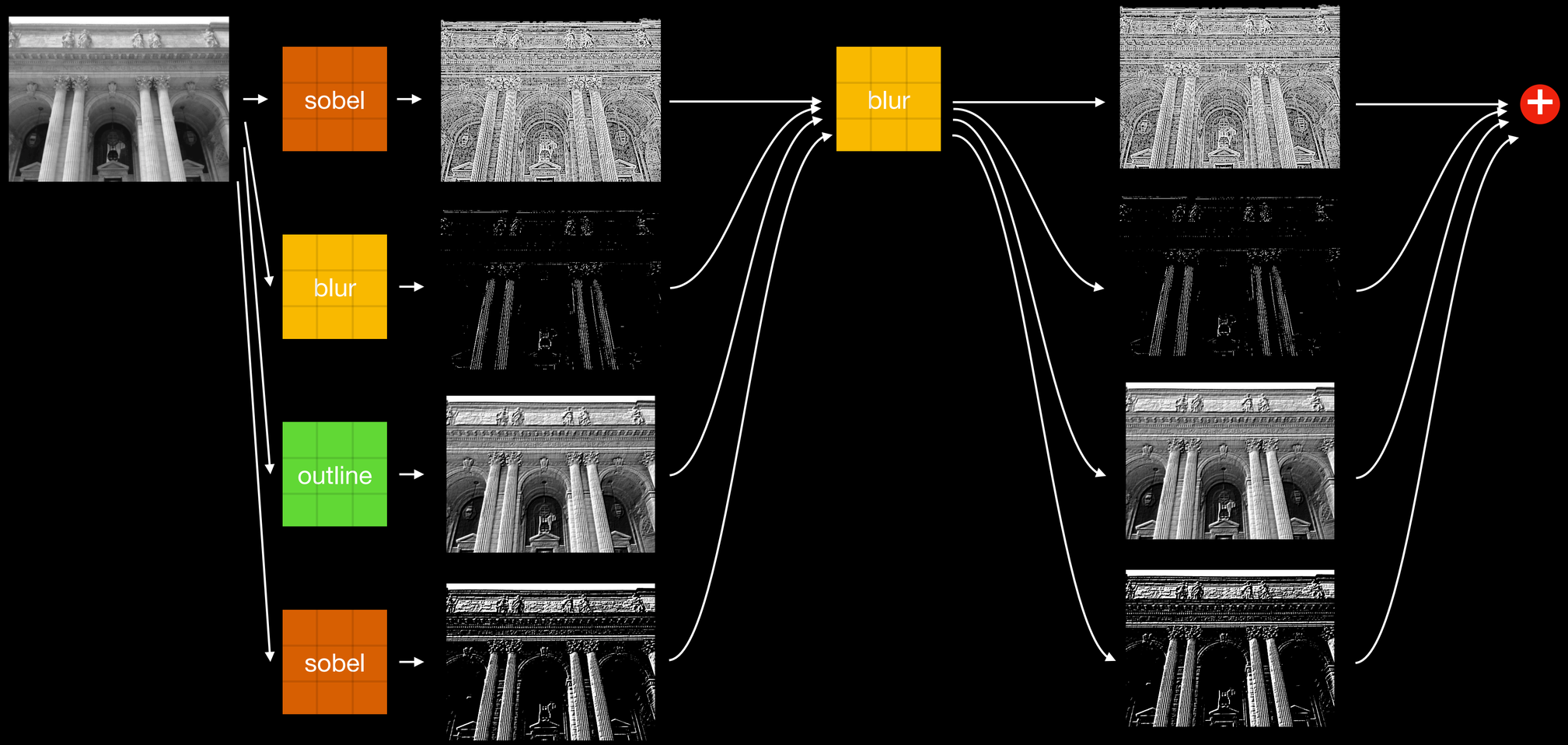
Feature Map

Convolutional Neural Network



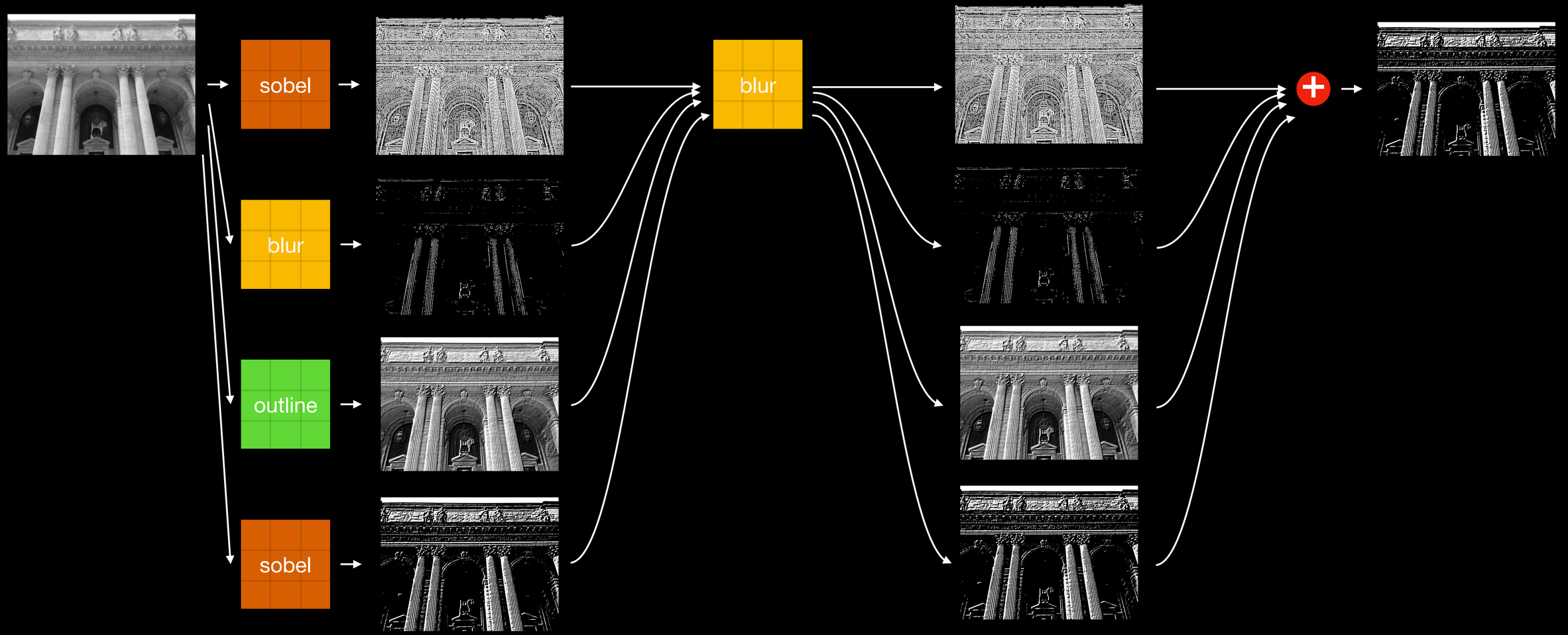
Feature Map

Convolutional Neural Network



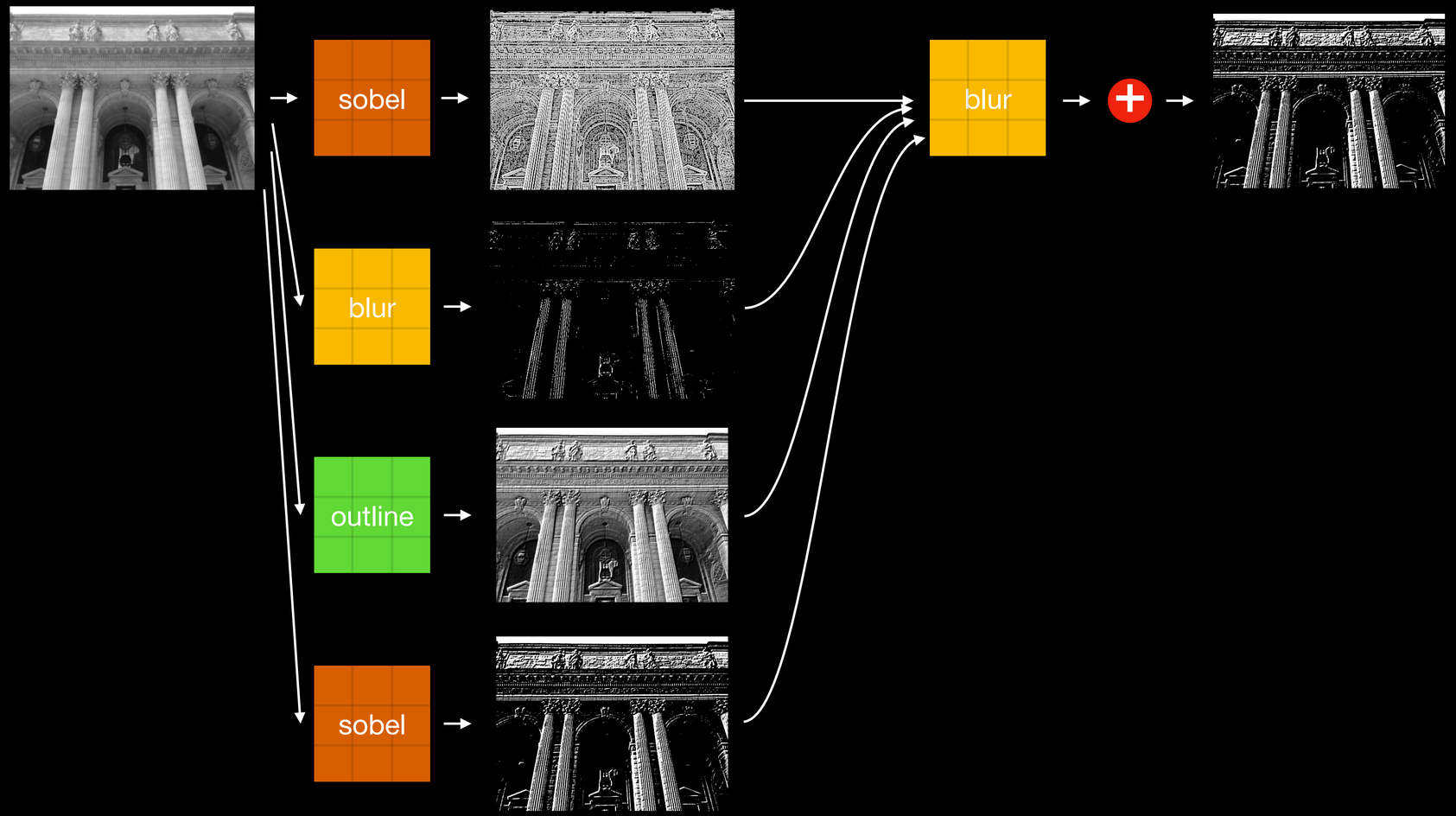
Feature Map

Convolutional Neural Network



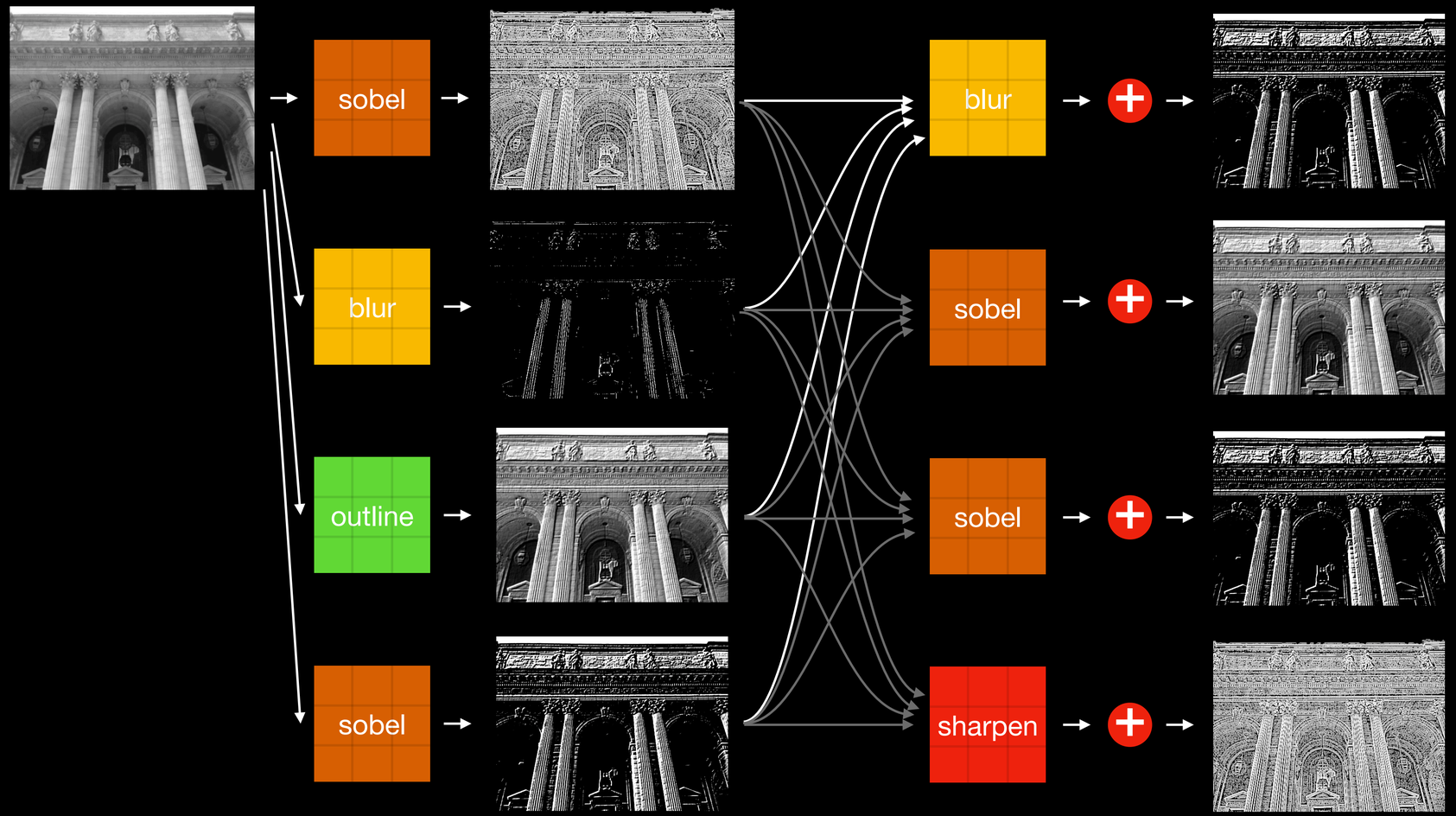
Feature Map

Convolutional Neural Network



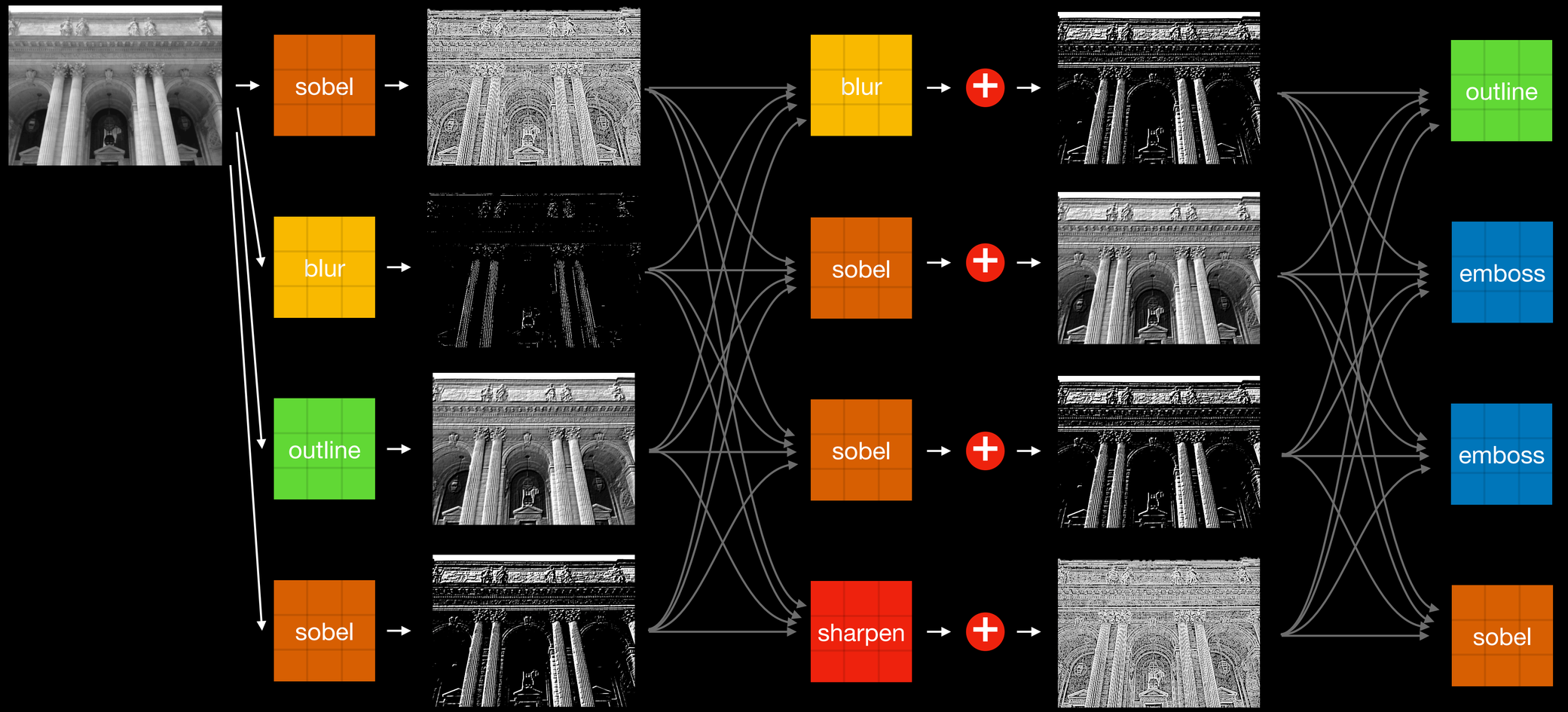
Feature Map

Convolutional Neural Network



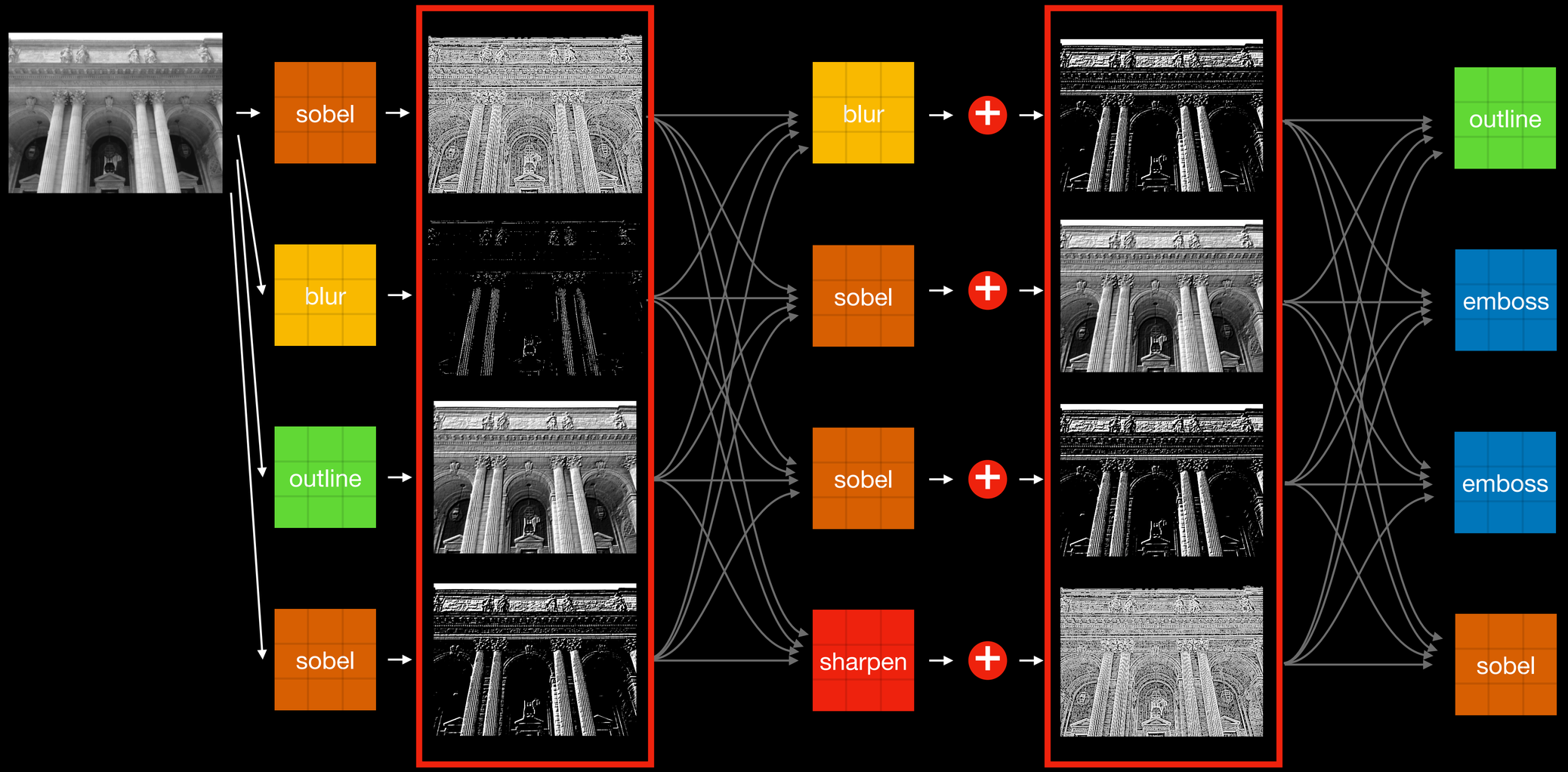
Feature Map

Convolutional Neural Network



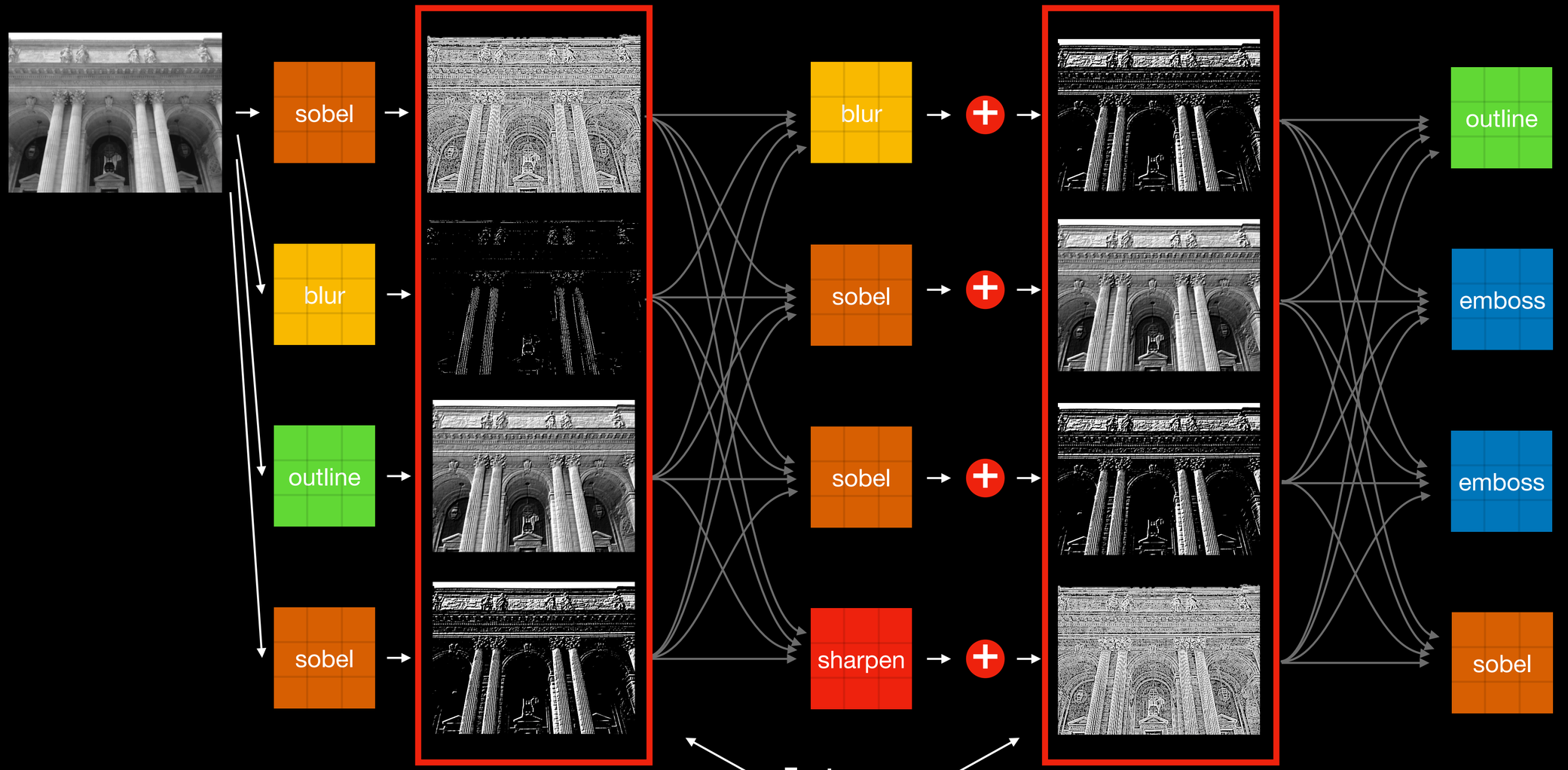
Feature Map

Convolutional Neural Network



Feature Map

Convolutional Neural Network



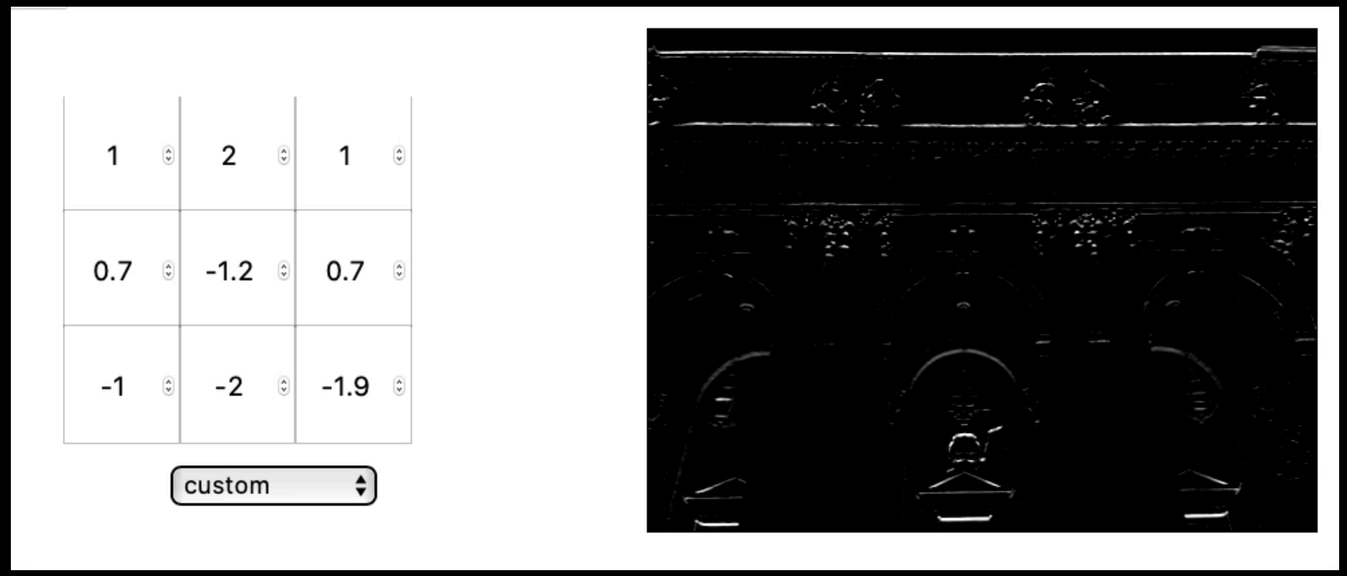
Feature map

Different processed versions of the original images

Training CNN

Customized Convolution Filter (kernel)

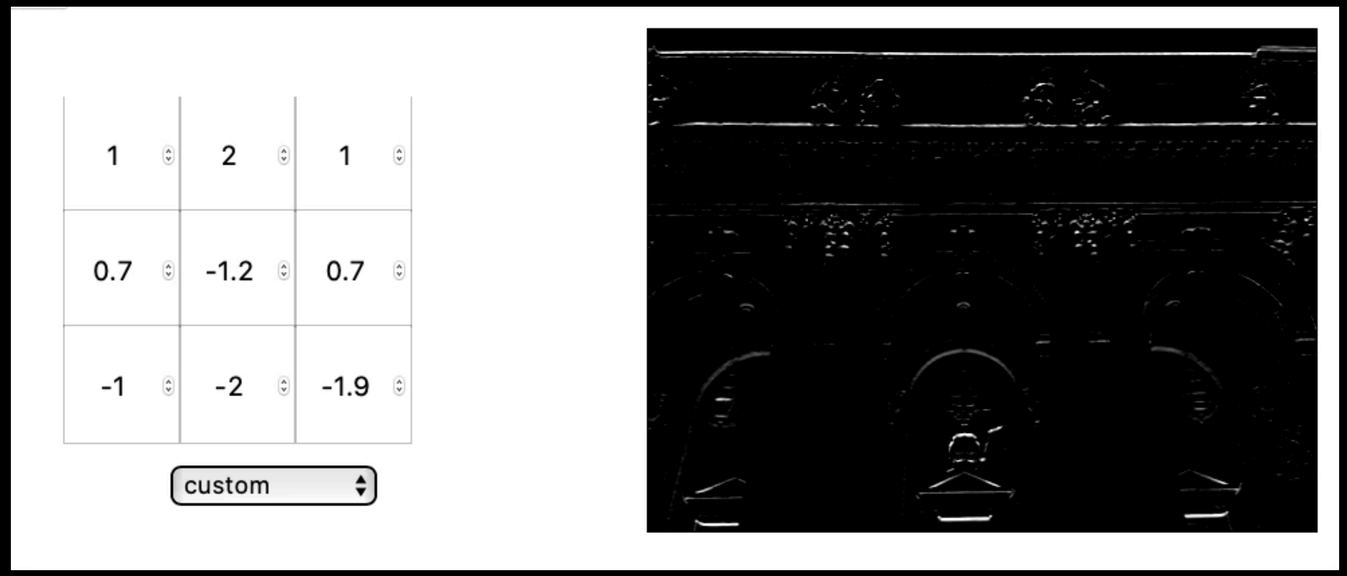
Change the kernel values



Training CNN

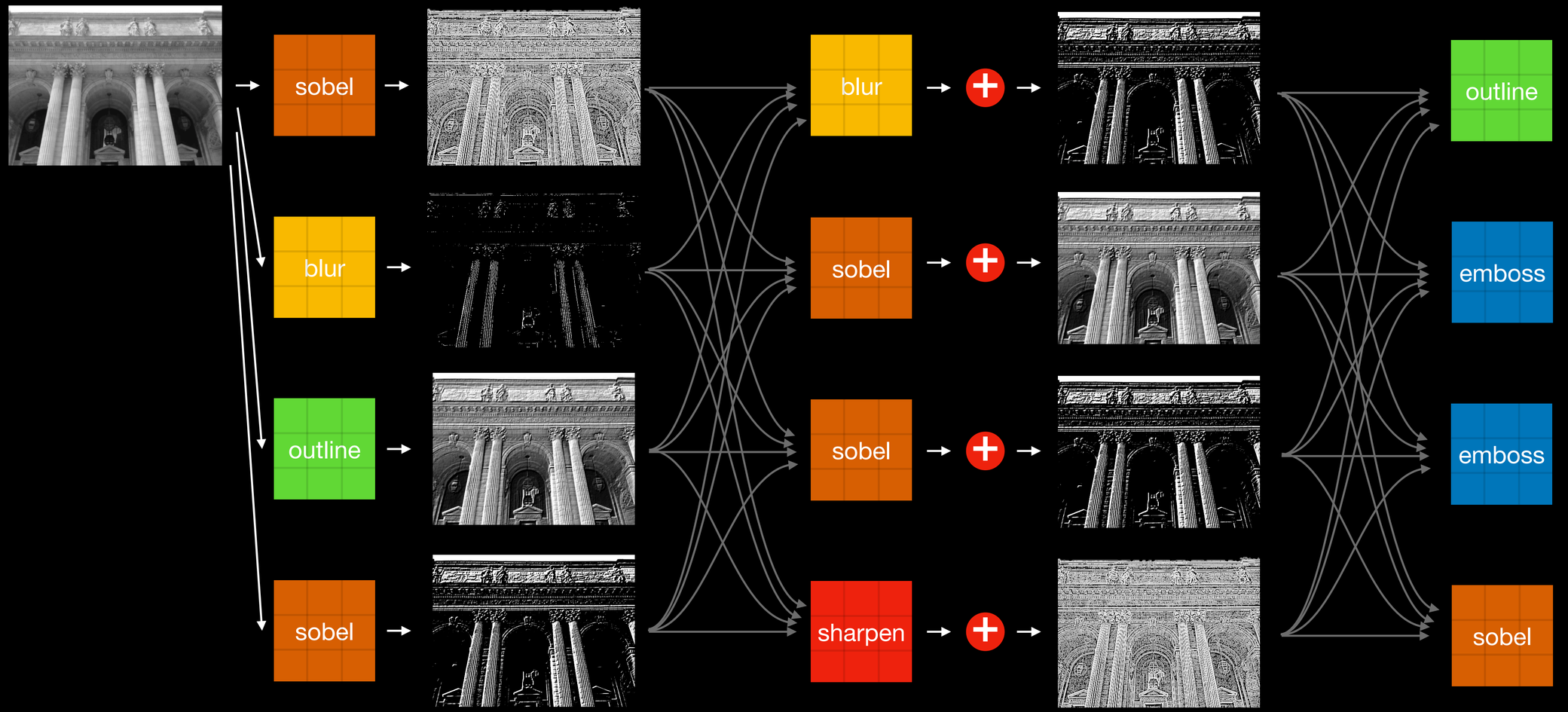
Customized Convolution Filter (kernel)

Change the kernel values



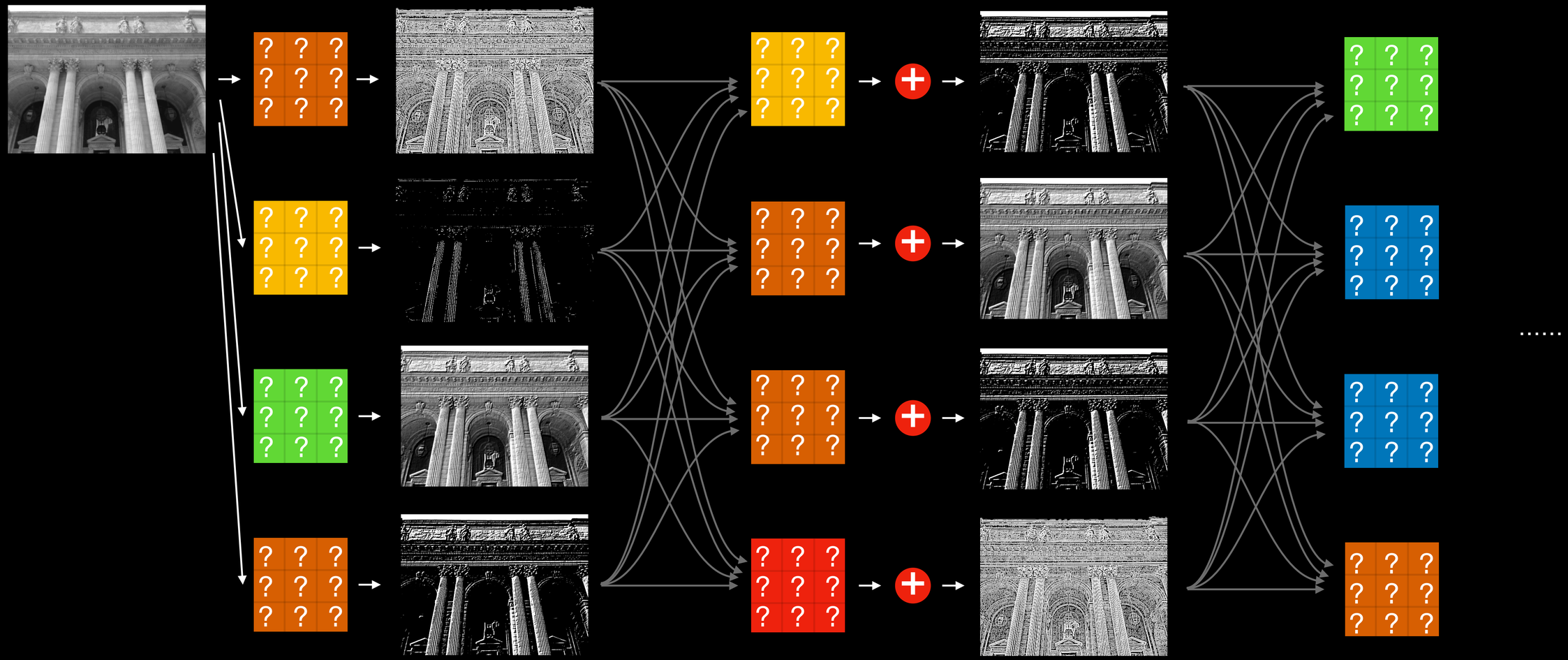
Training CNN

Customized Convolution Filter (kernel)



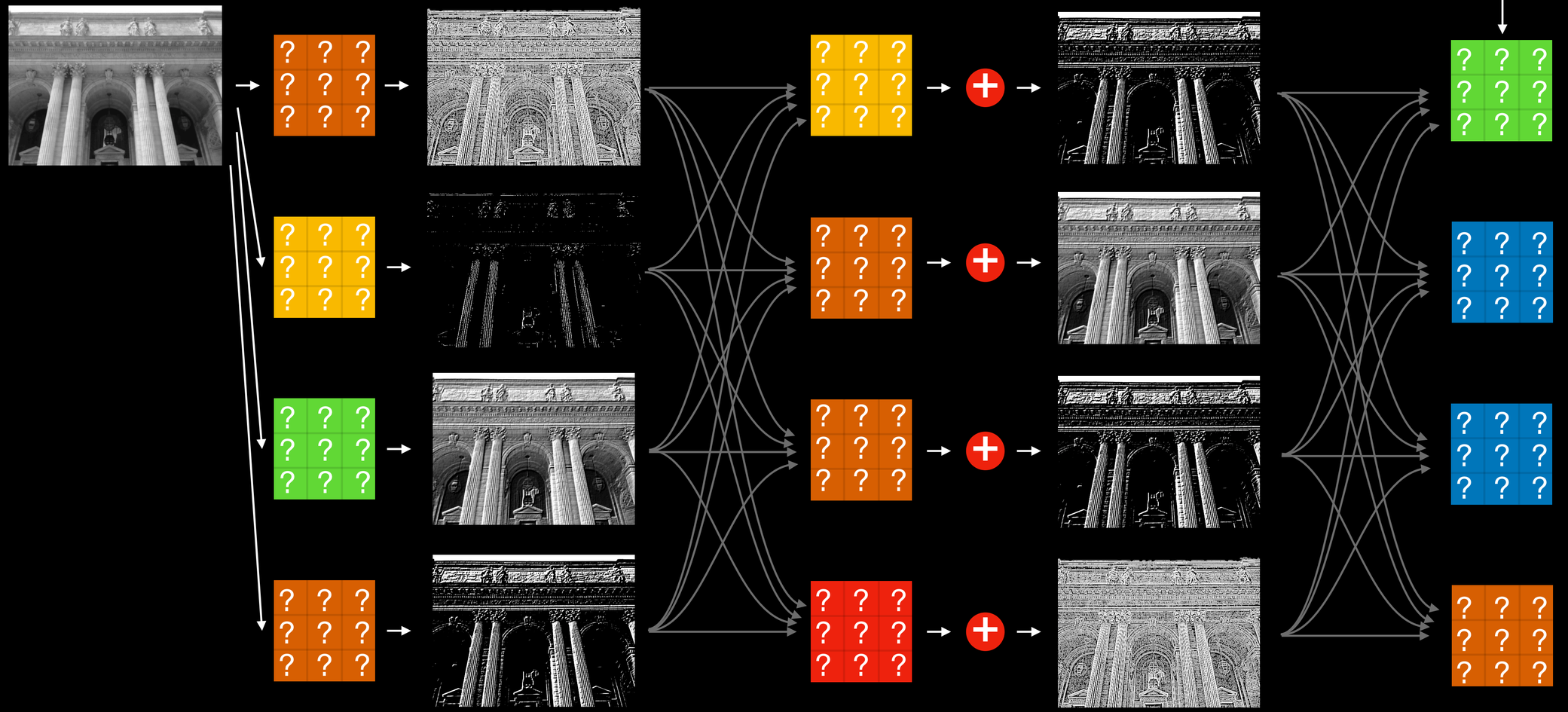
Training CNN

Customized Convolution Filter (kernel)



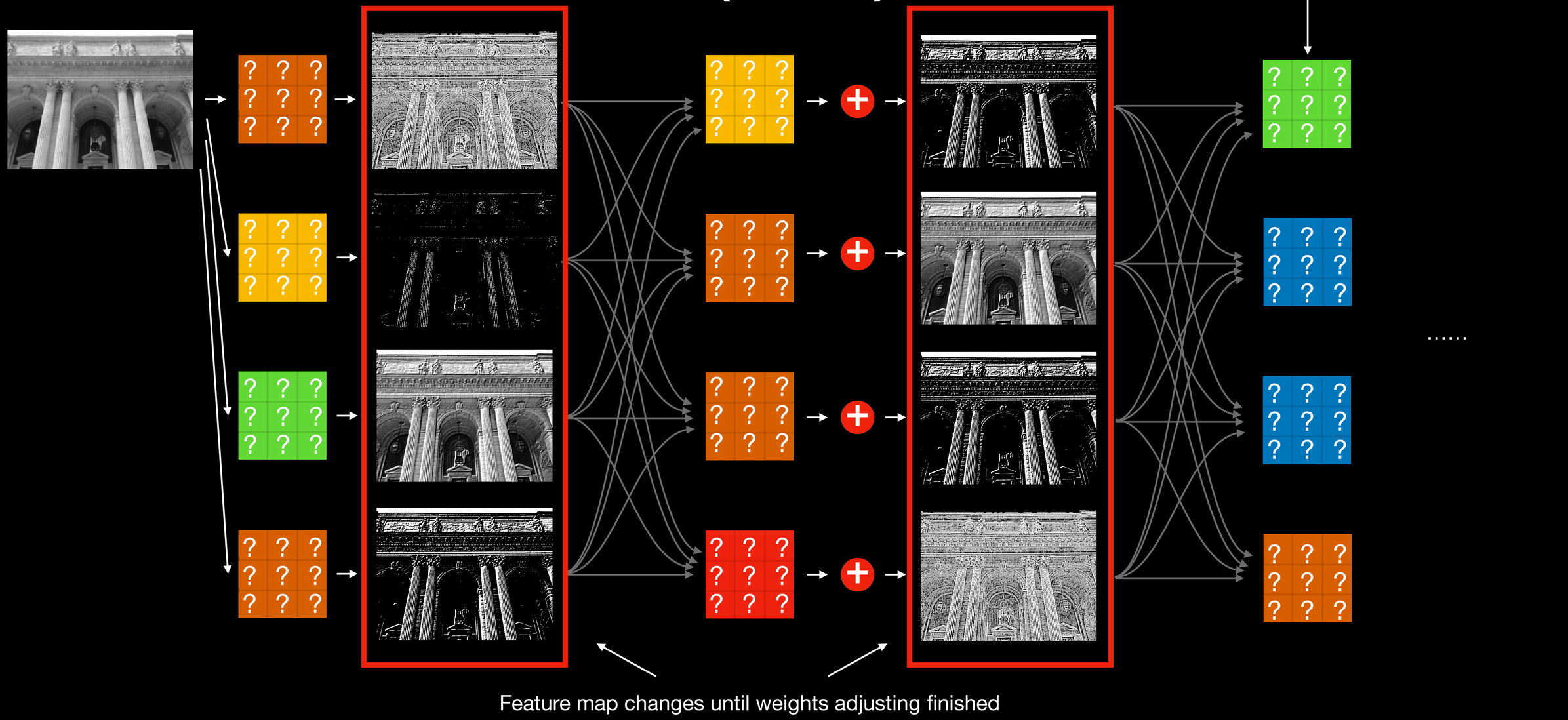
Training CNN

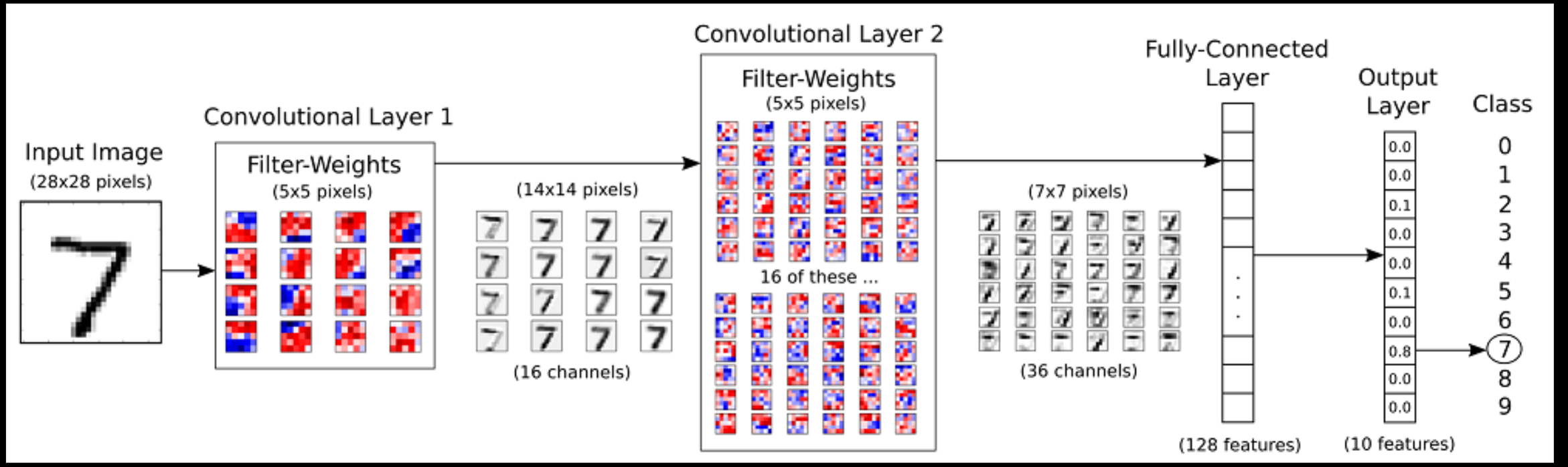
Customized Convolution Filter (kernel)

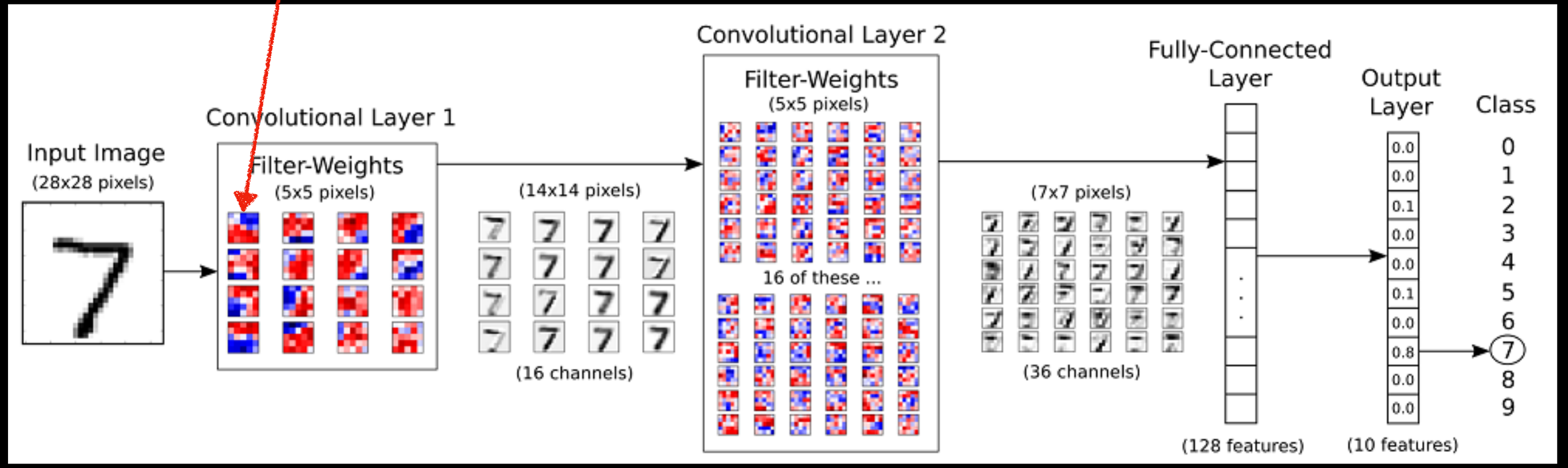


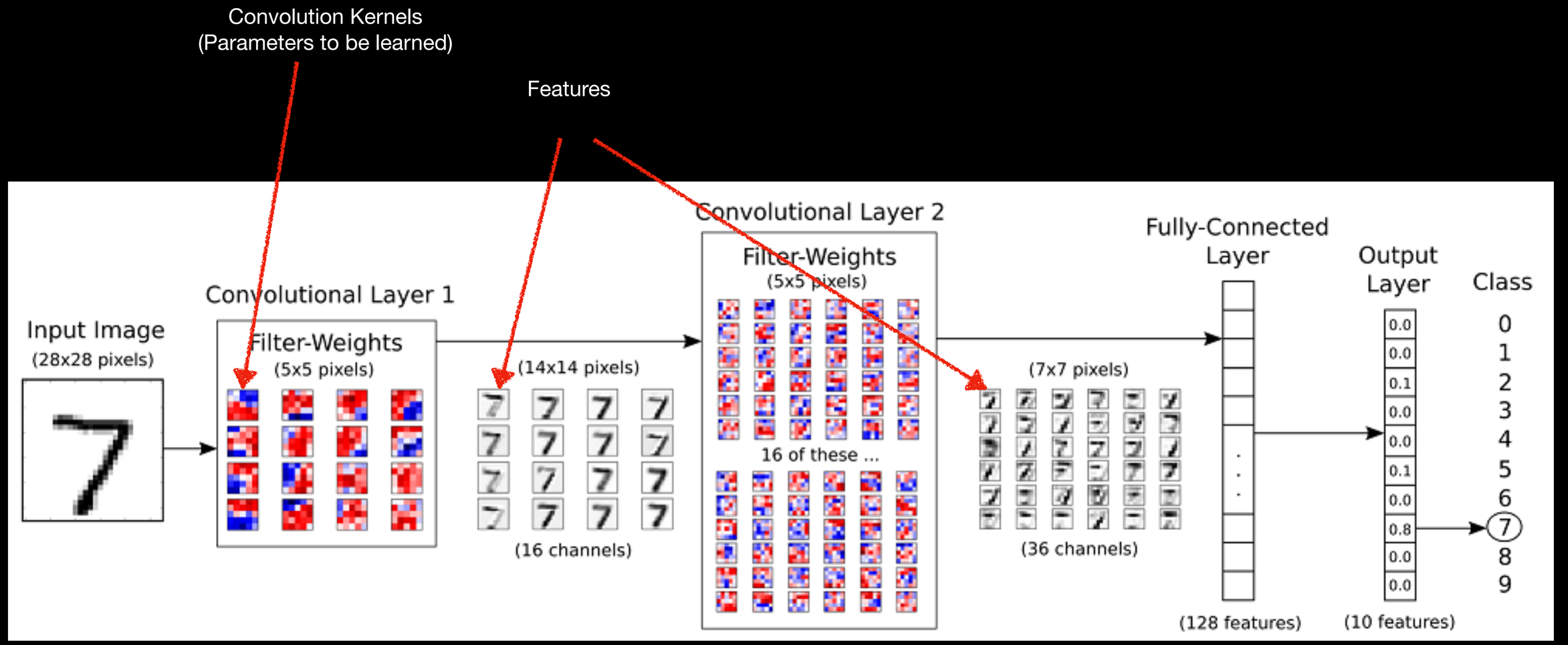
Training CNN

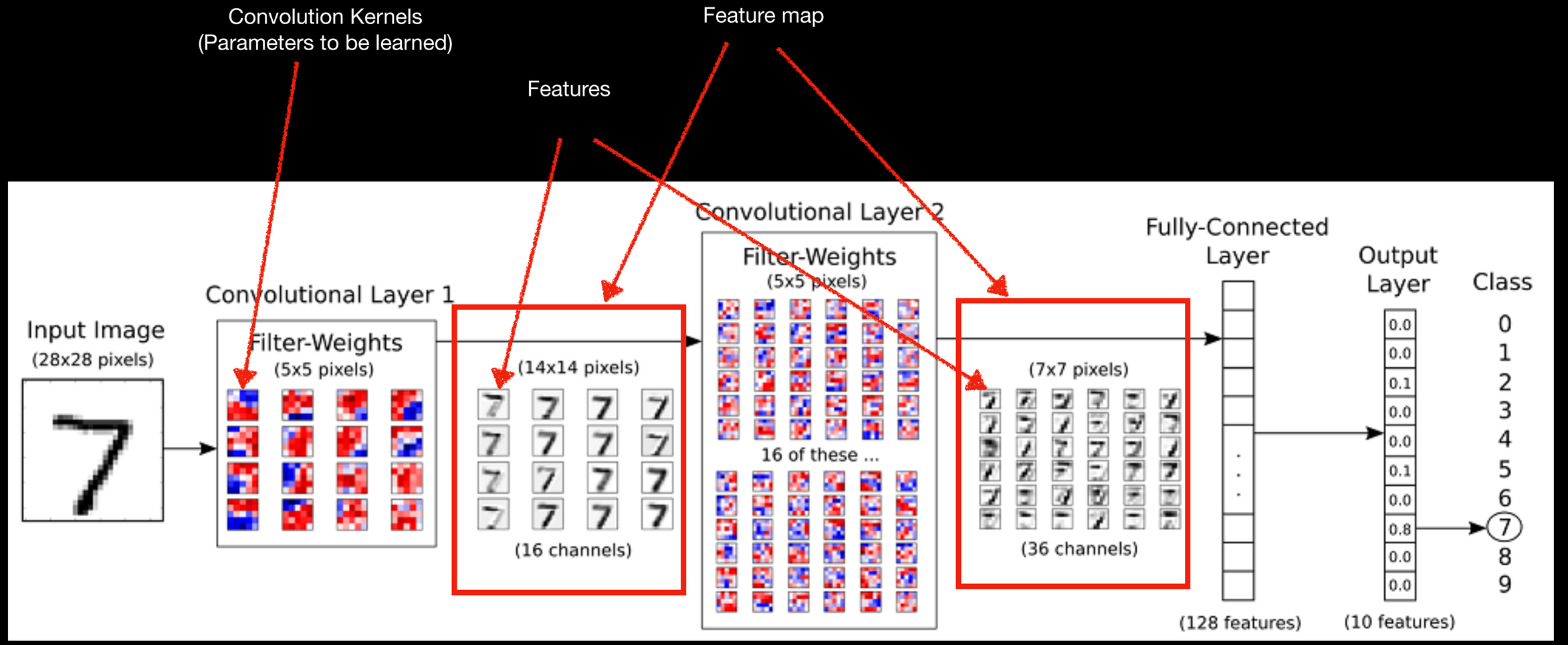
Customized Convolution Filter (kernel)

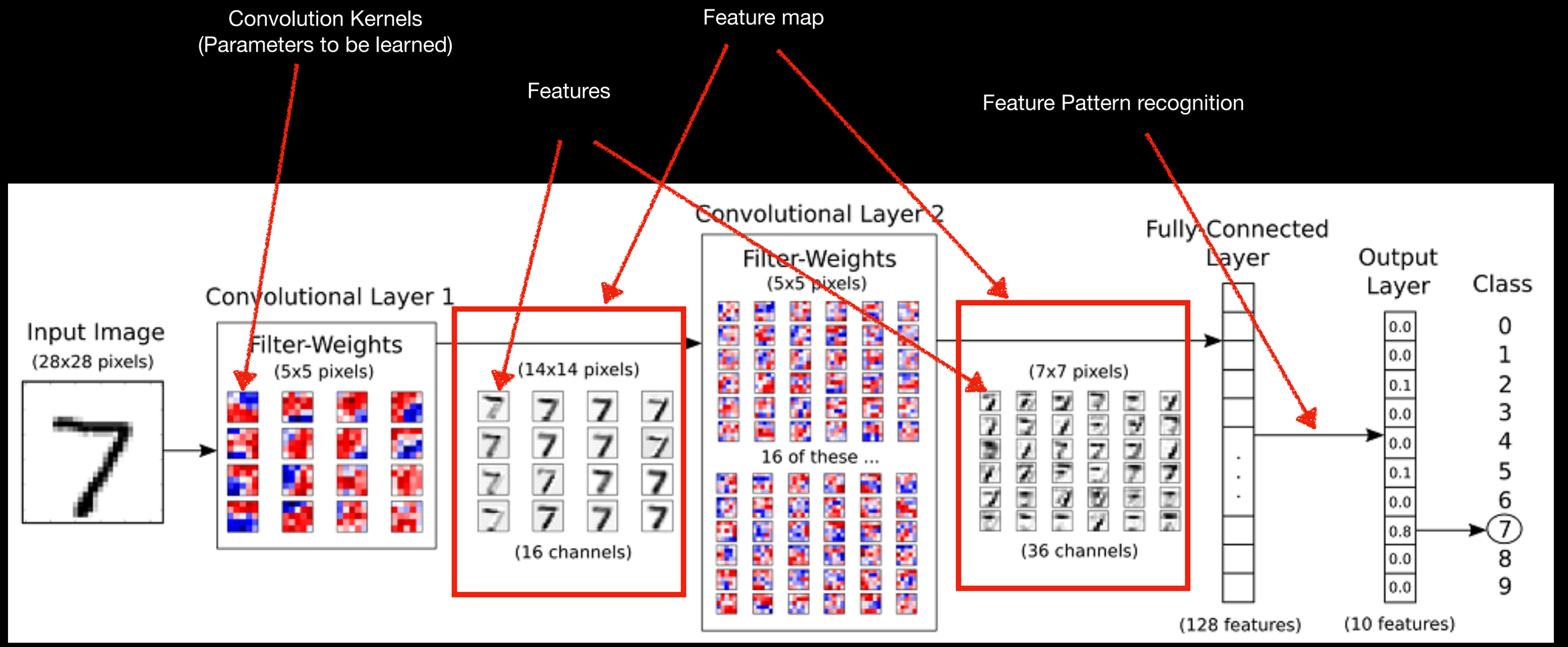


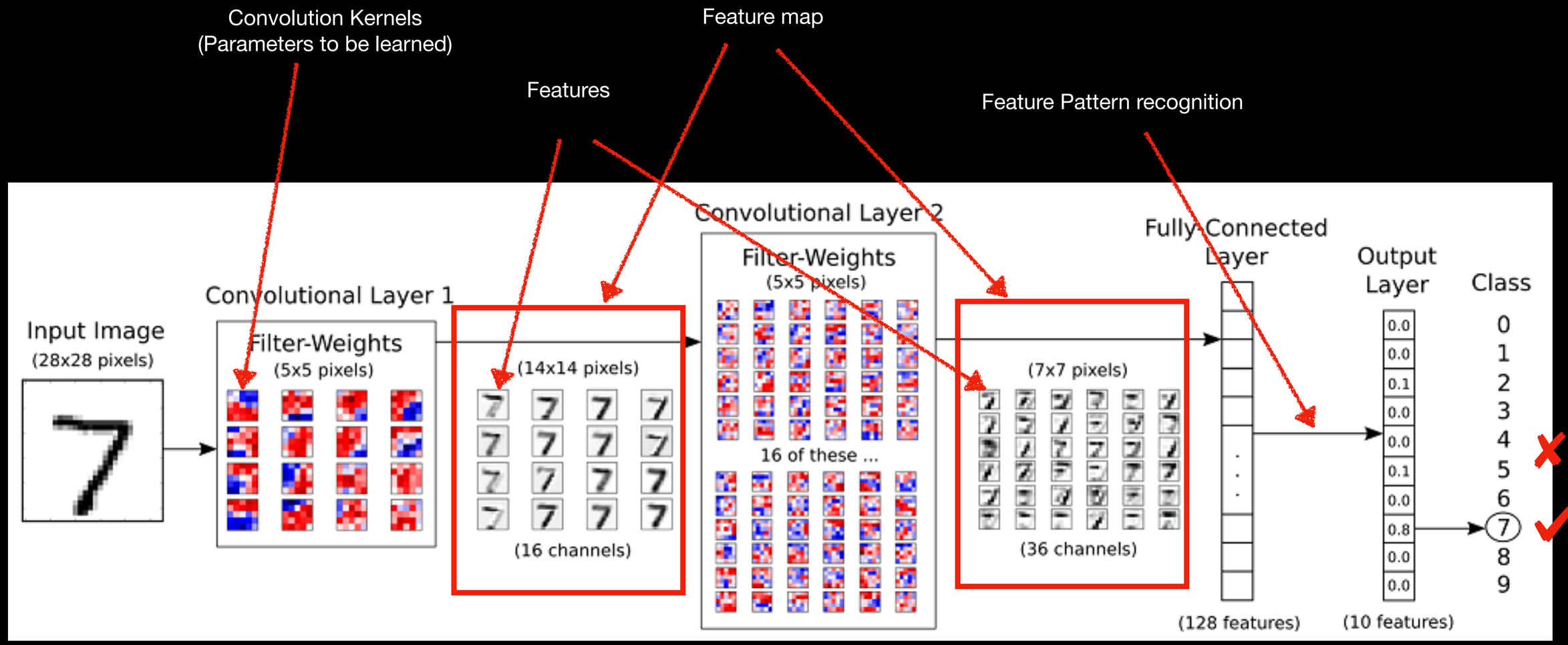

Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected Layer
(128 features)
Output Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9

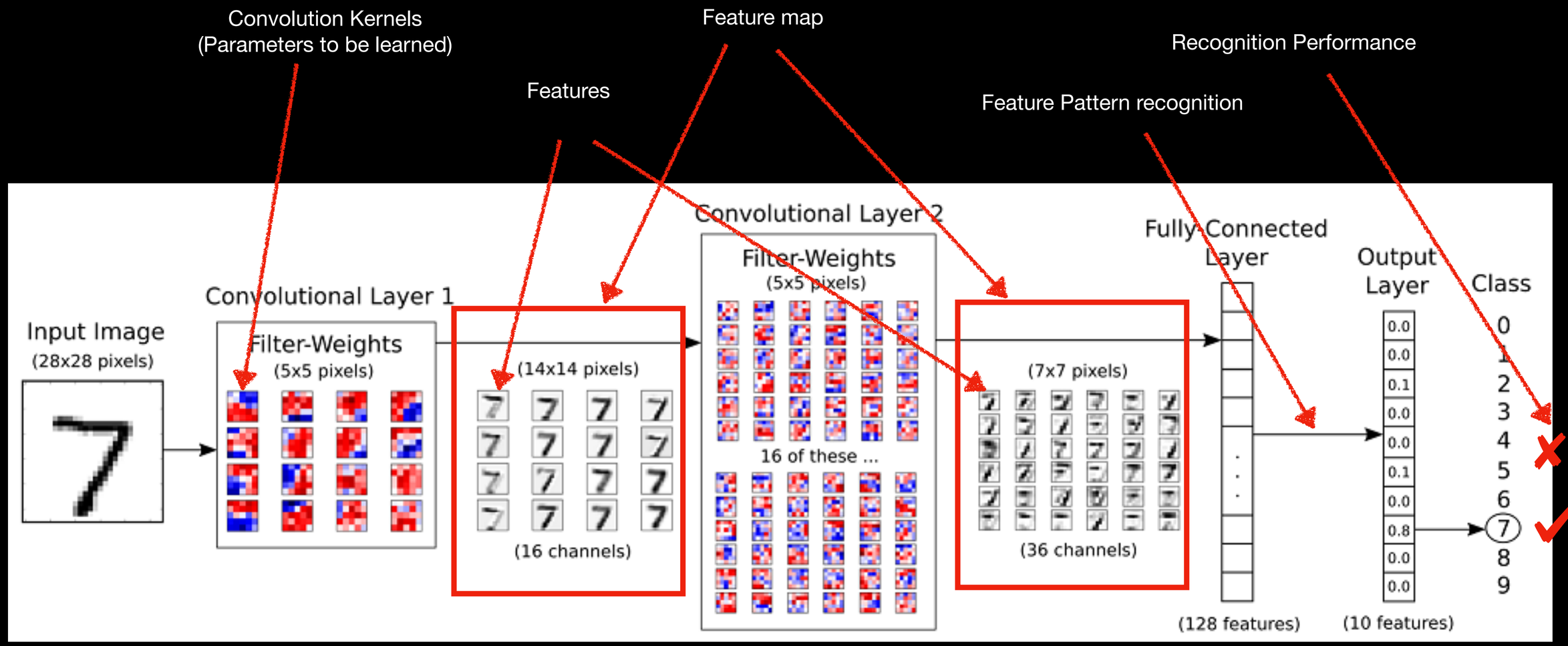

Convolution Kernels
(Parameters to be learned)
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected
Layer
(128 features)
Output
Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9

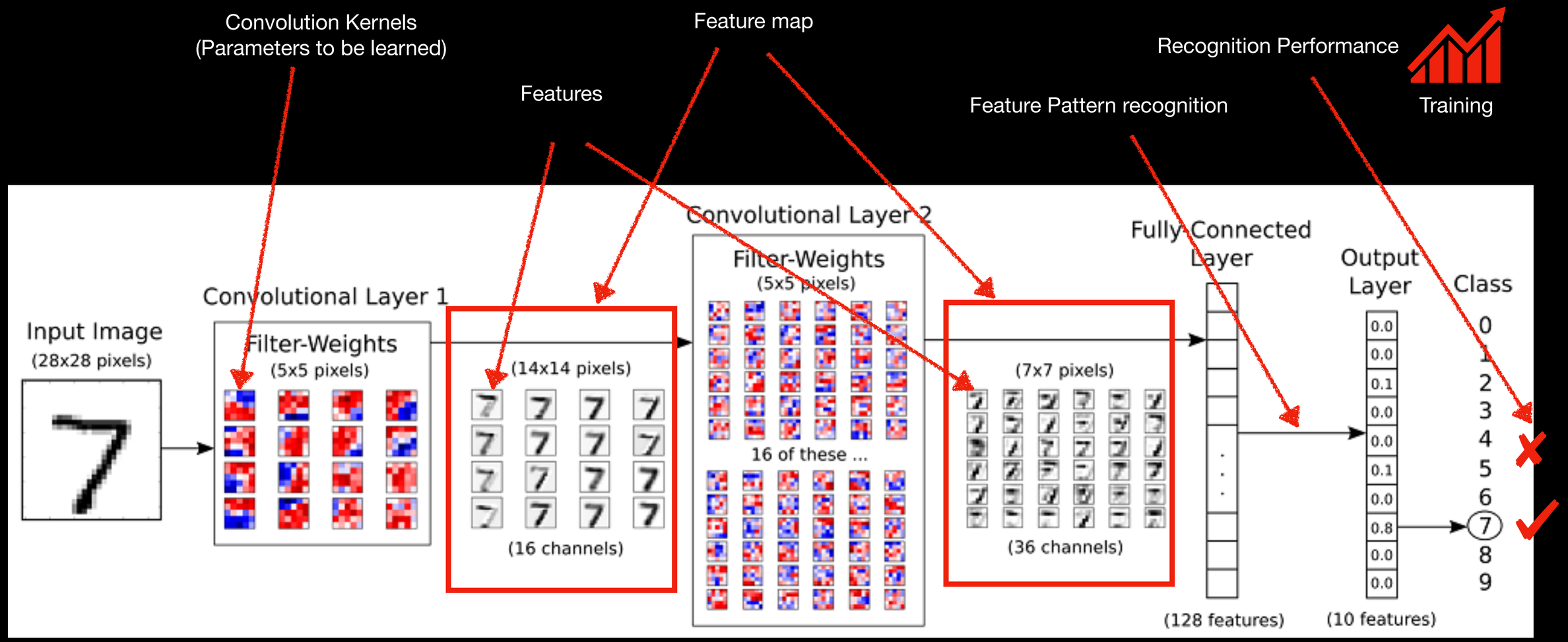

Convolution Kernels
(Parameters to be learned)
Features
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected
Layer
(128 features)
Output
Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9

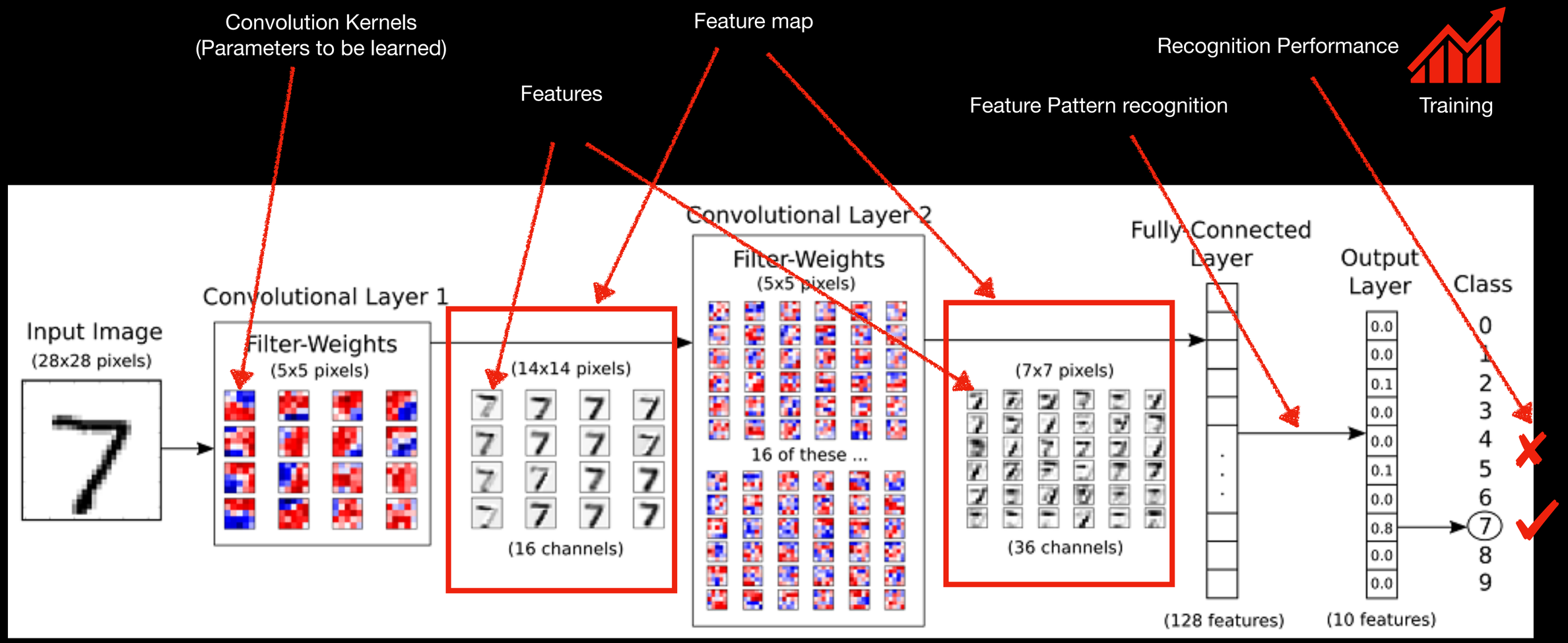

Convolution Kernels
(Parameters to be learned)
Feature map
Features
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected
Layer
(128 features)
Output
Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9


Convolution Kernels
(Parameters to be learned)
Feature map
Features
Feature Pattern recognition
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected
Layer
(128 features)
Output
Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9


Convolution Kernels
(Parameters to be learned)
Feature map
Features
Feature Pattern recognition
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected Layer
(128 features)
Output Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9

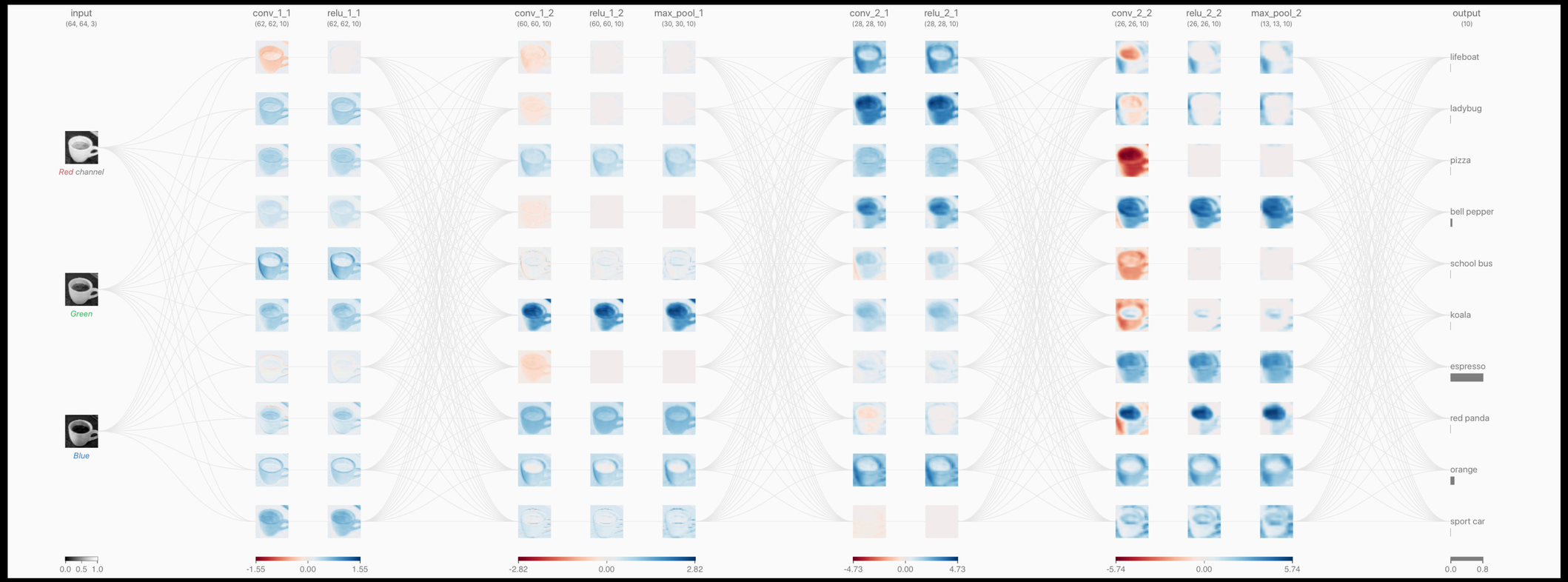

Convolution Kernels
(Parameters to be learned)
Feature map
Recognition Performance
Features
Feature Pattern recognition
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected Layer
(128 features)
Output Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9


Convolution Kernels
(Parameters to be learned)
Feature map
Recognition Performance
Features
Feature Pattern recognition
Training
Input Image
(28x28 pixels)
Convolutional Layer 1
Filter-Weights
(5x5 pixels)
(14x14 pixels)
(16 channels)
Convolutional Layer 2
Filter-Weights
(5x5 pixels)
16 of these ...
(7x7 pixels)
(36 channels)
Fully-Connected Layer
(128 features)
Output Layer
0.0
0.0
0.1
0.0
0.0
0.1
0.0
0.8
0.0
0.0
(10 features)
Class
0
1
2
3
4
5
6
7
8
9



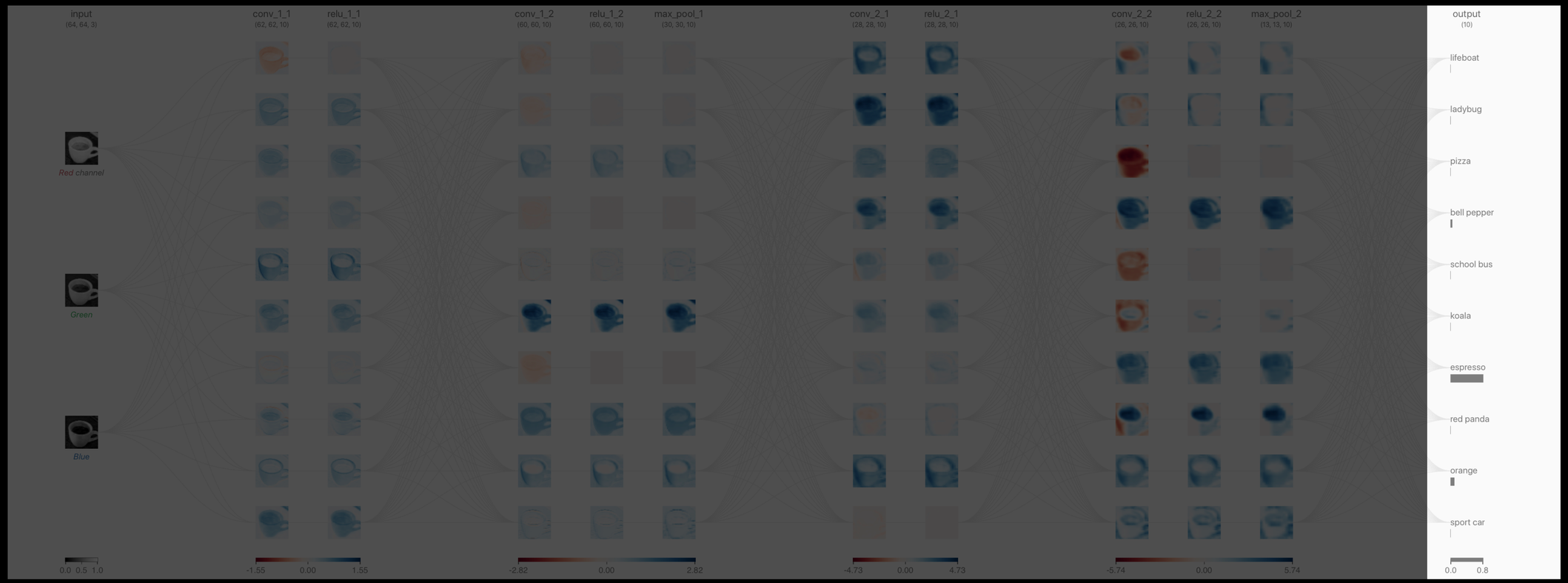
Model

- Num of layers (stages)
- Num of channels (filters) in each layer



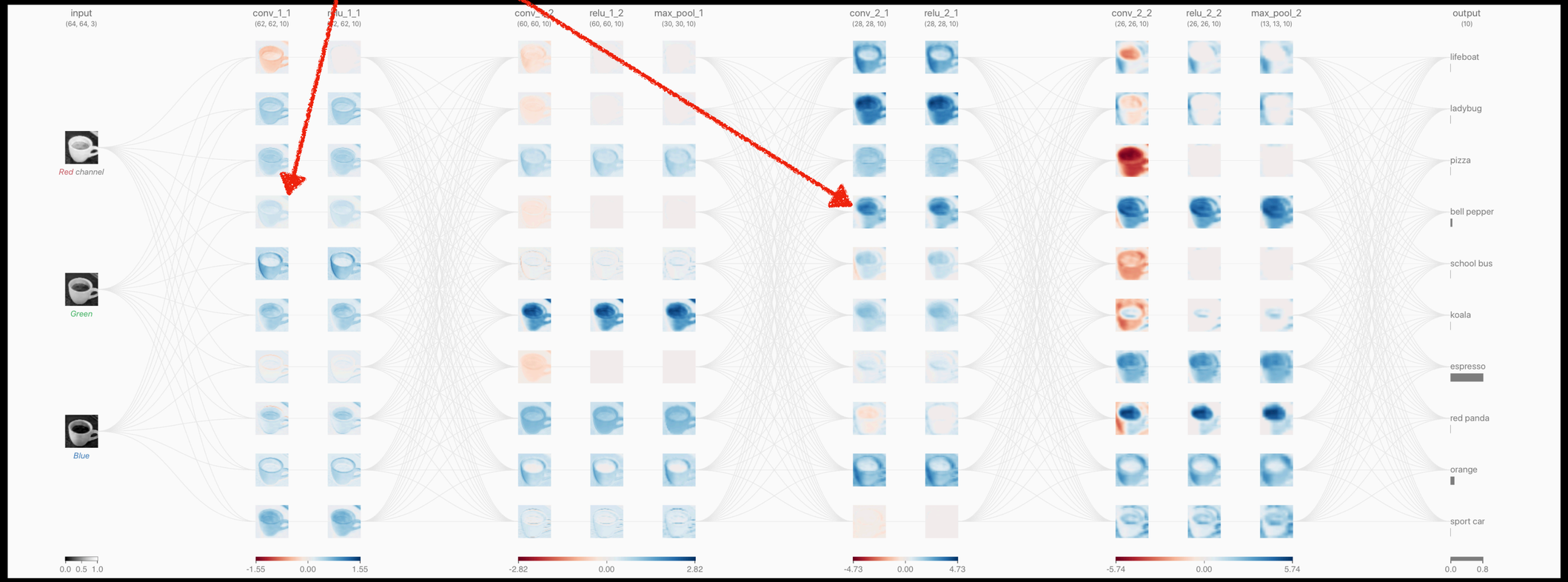
https://poloclub.github.io/cnn-explainer/#article-convolution

Scores on different categories



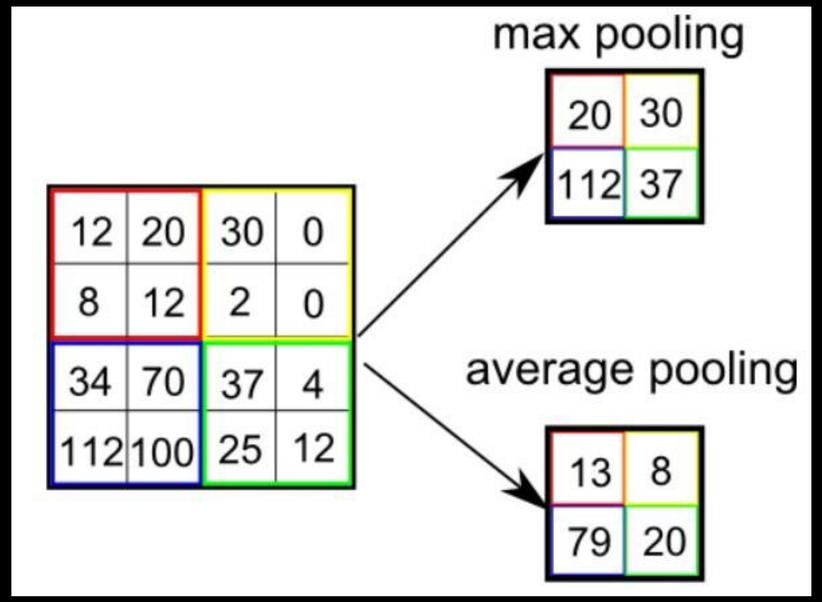
https://poloclub.github.io/cnn-explainer/#article-convolution

Features

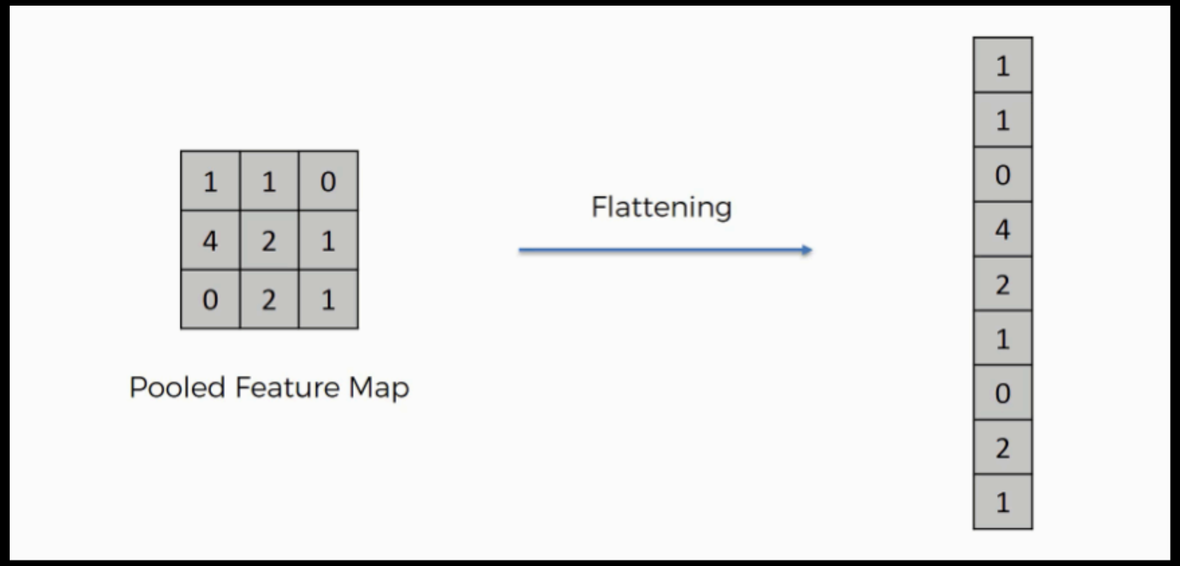


https://poloclub.github.io/cnn-explainer/#article-convolution

Other operations

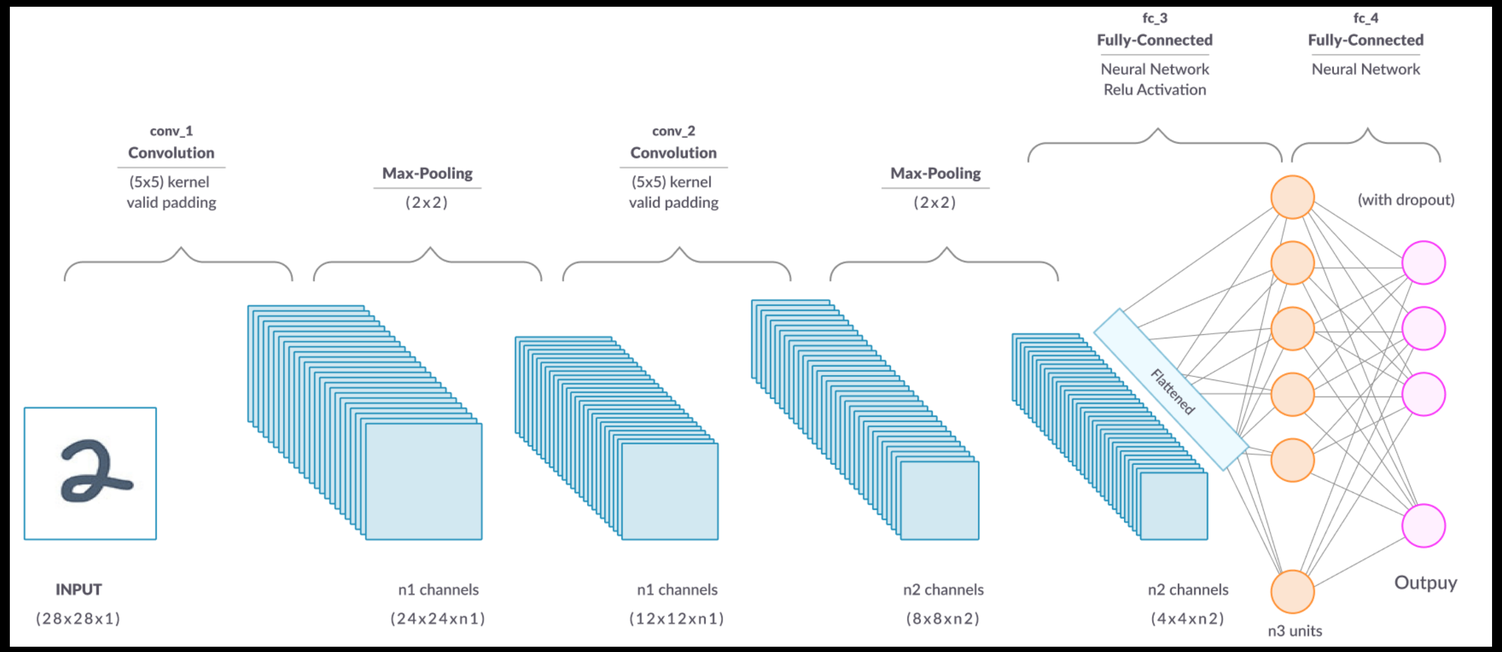


Pooling Layer (reduce the resolution)

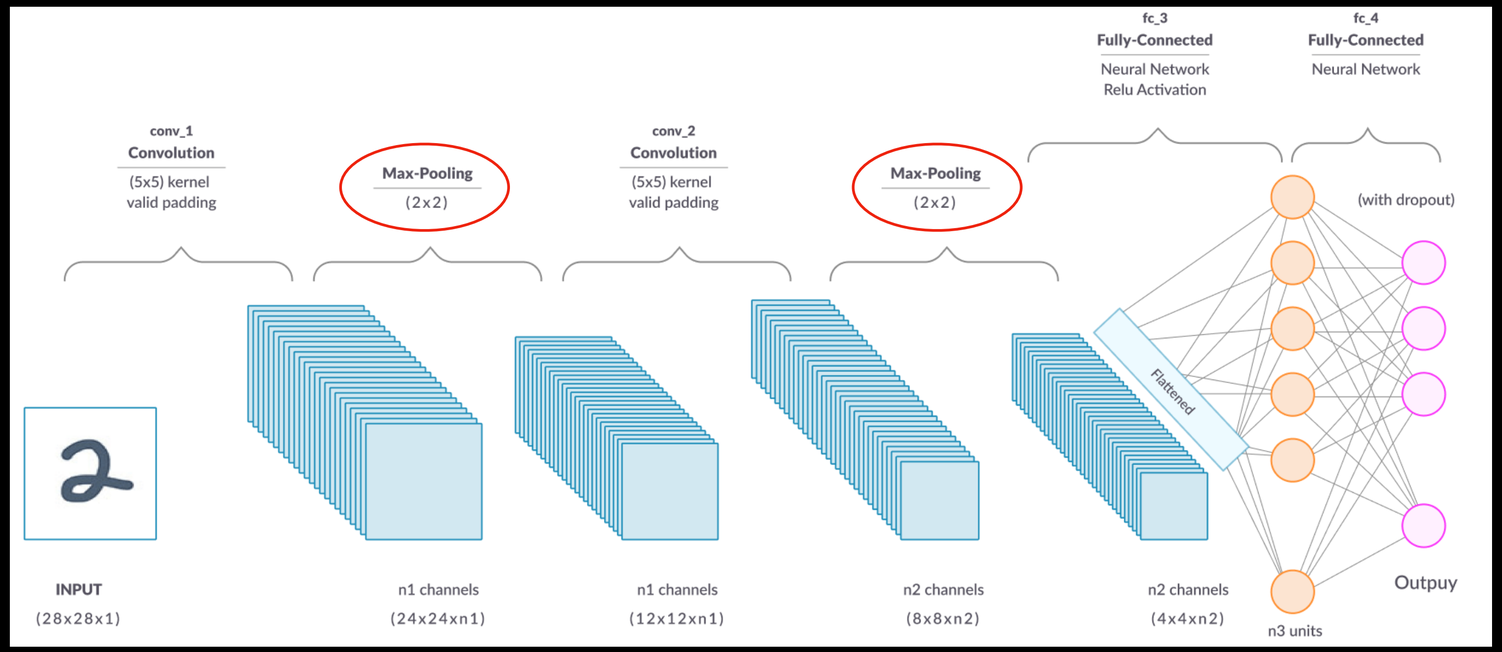


Flatten Layer (Convert 2D data to 1D for neural nets)

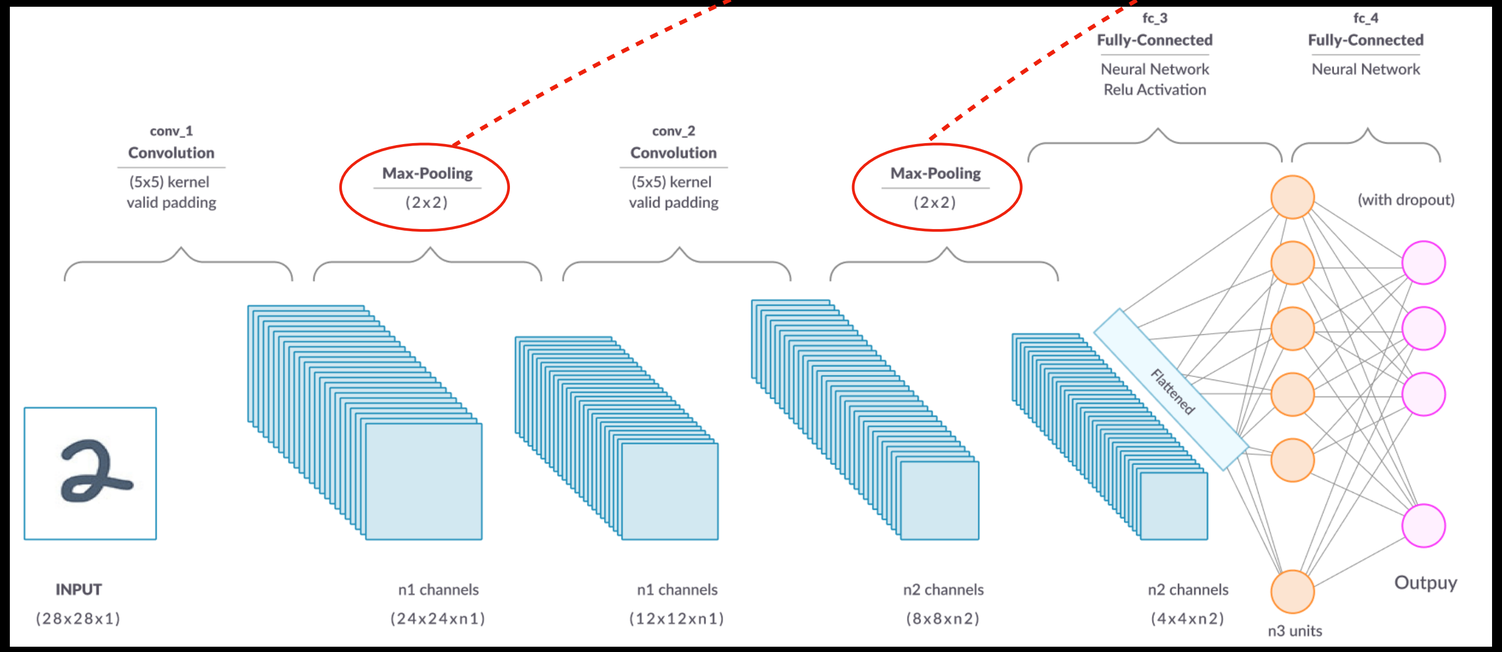
Other operations

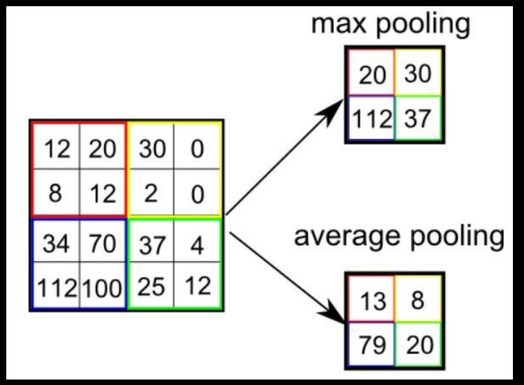


Other operations



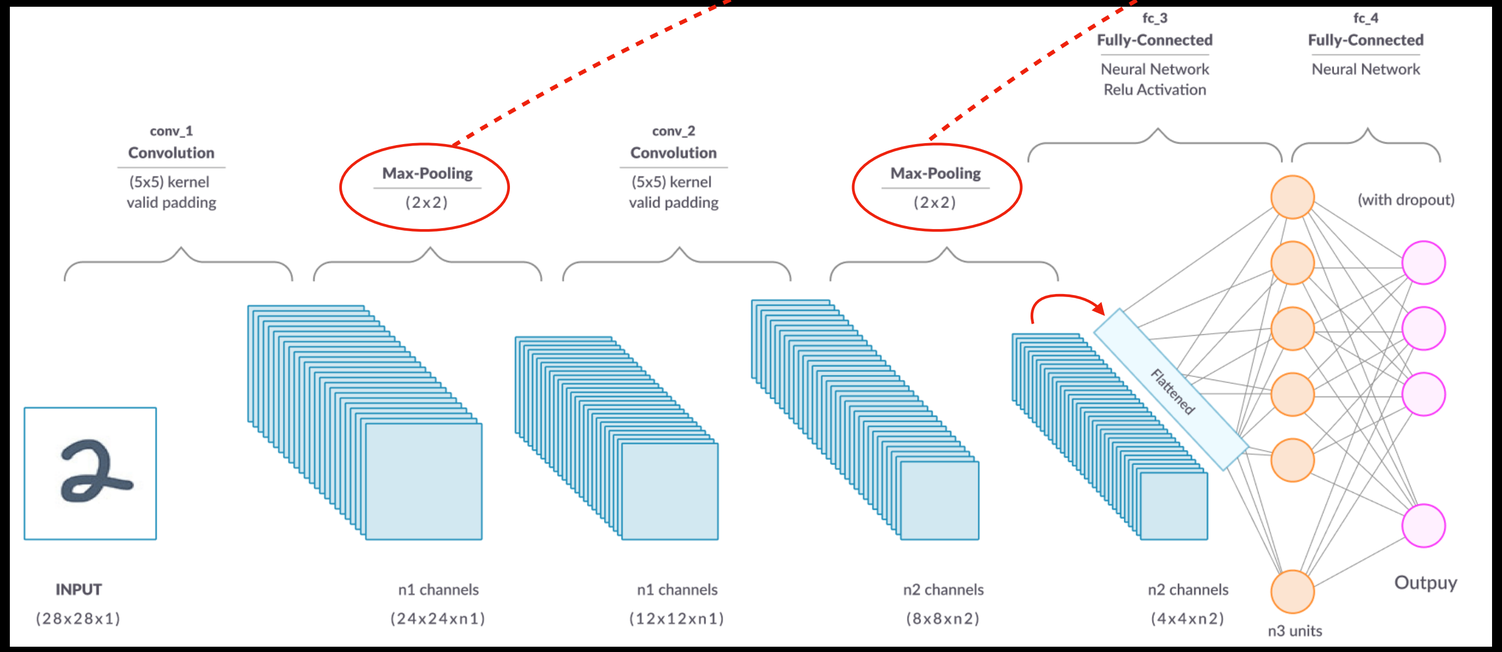
Other operations

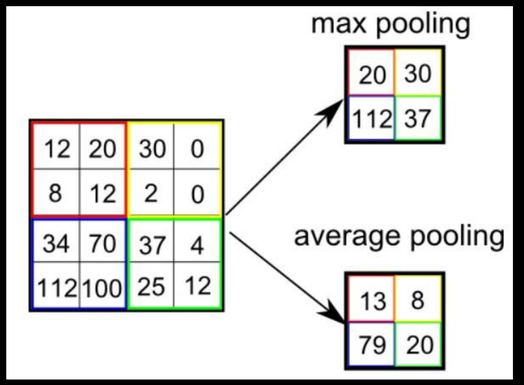




Pooling Layer
reduce the resolution

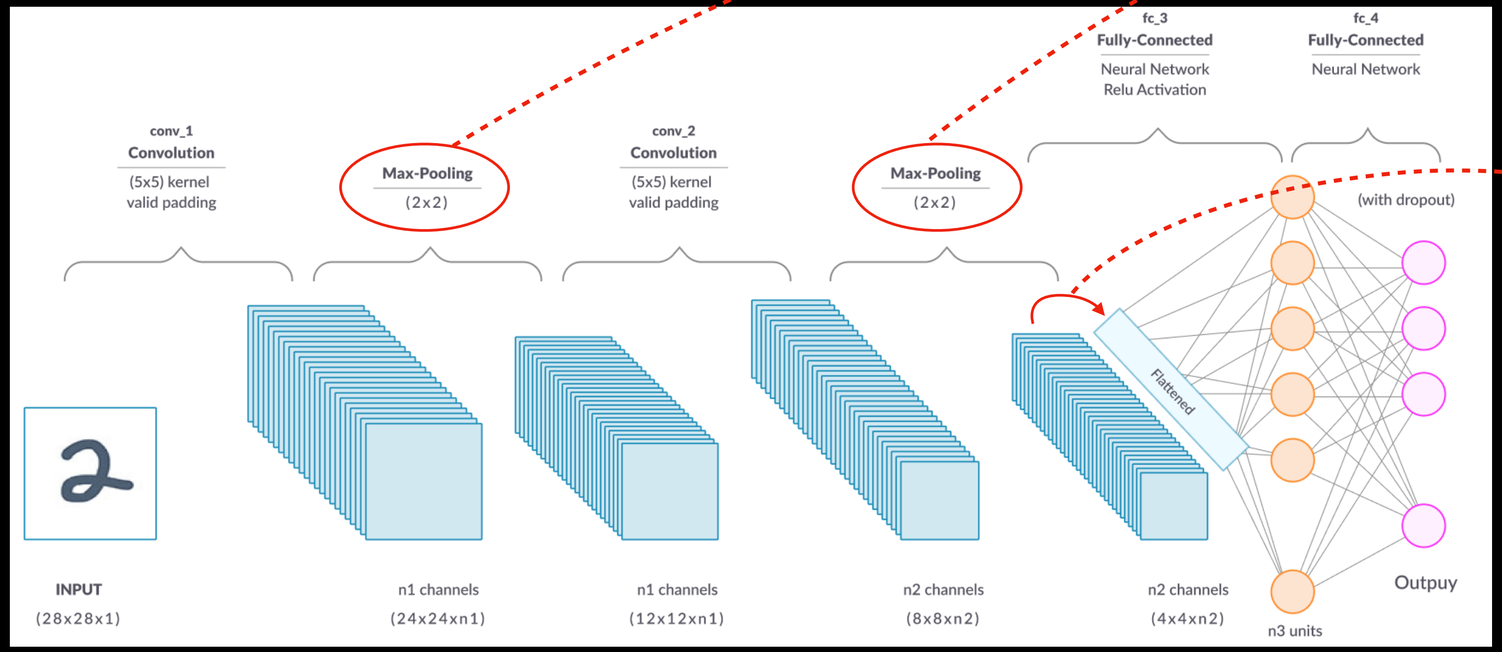
Other operations

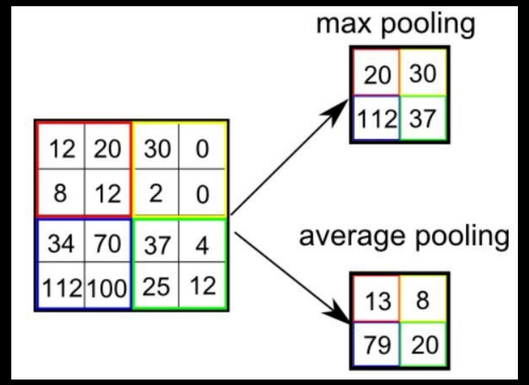




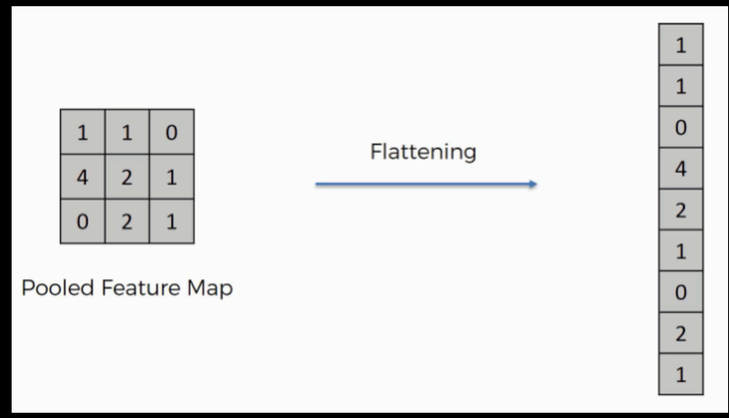
Pooling Layer
reduce the resolution

Other operations



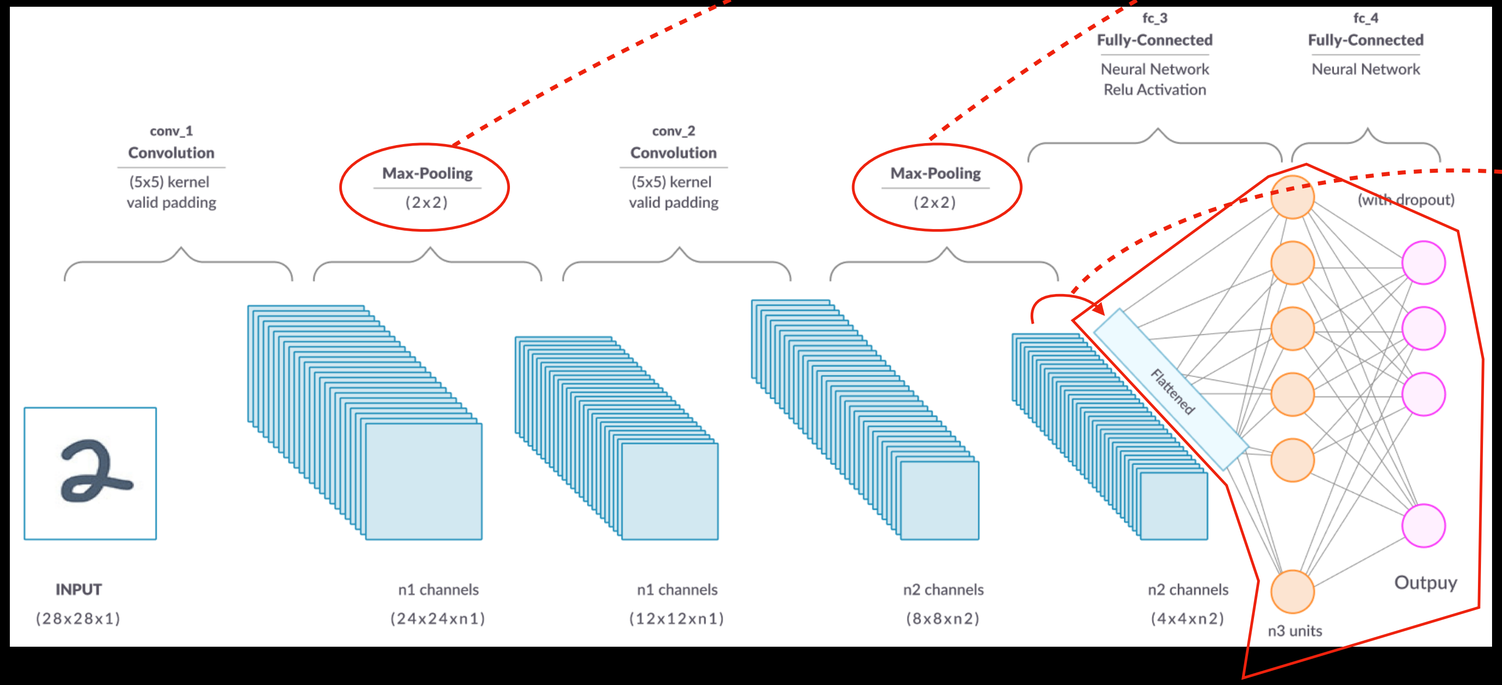


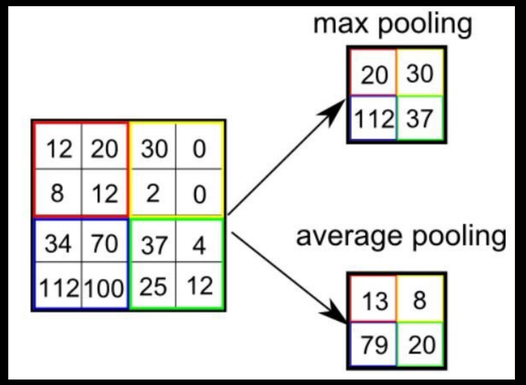
Pooling Layer
reduce the resolution



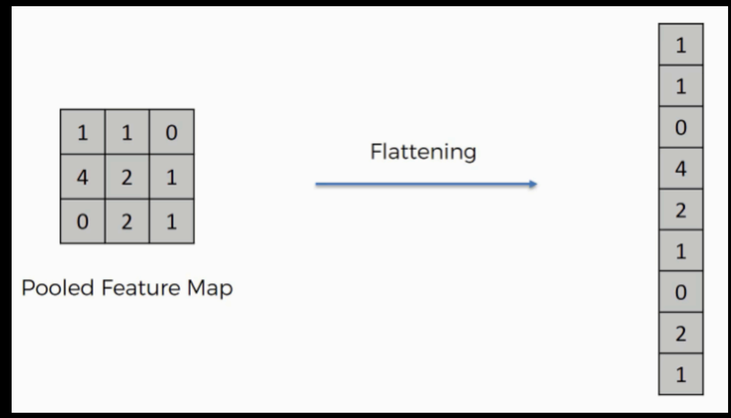
Flatten Layer
Convert 2D data to 1D for neural nets

Other operations



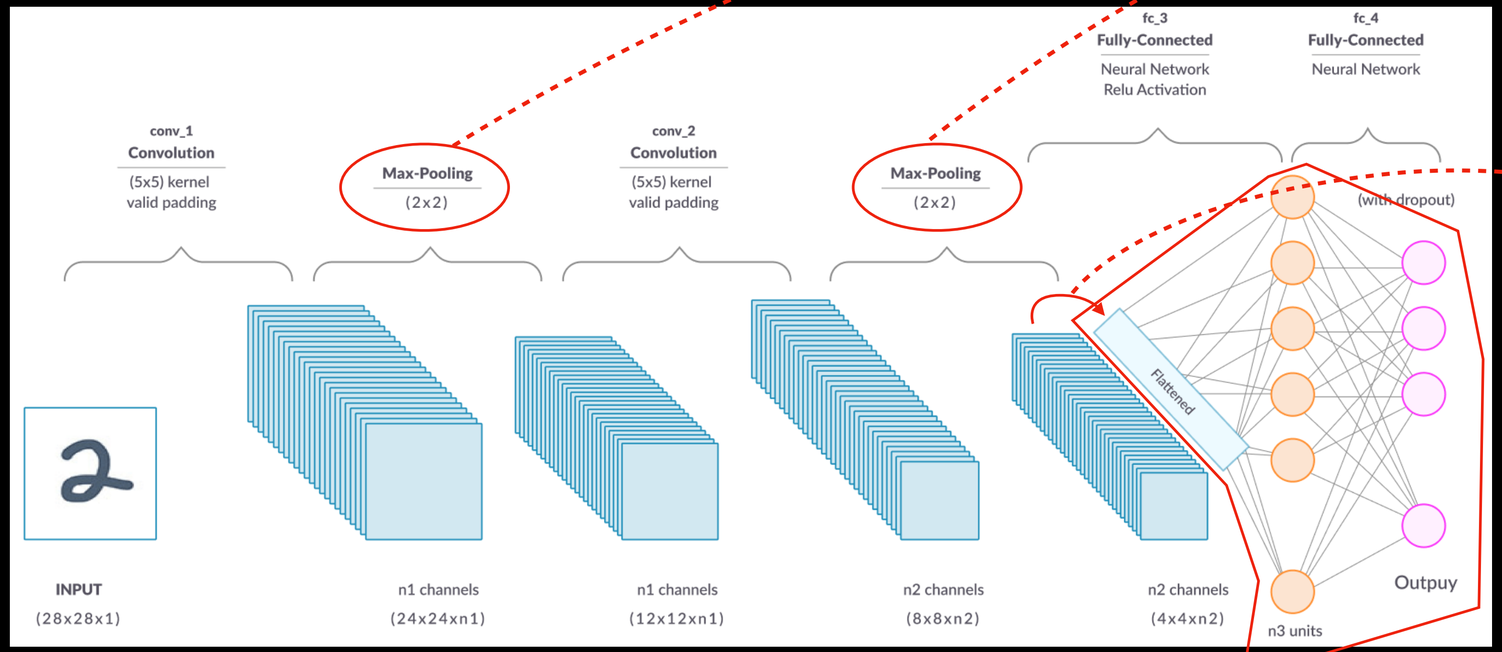


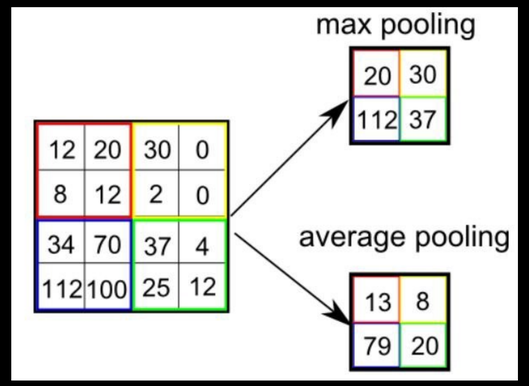
Pooling Layer
reduce the resolution



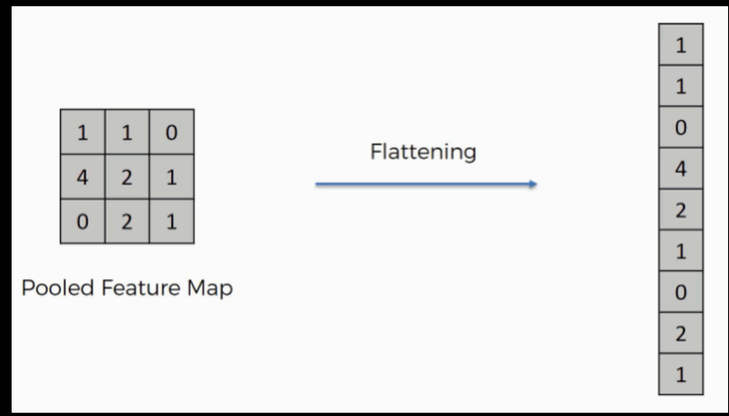
Flatten Layer
Convert 2D data to 1D for neural nets

Other operations

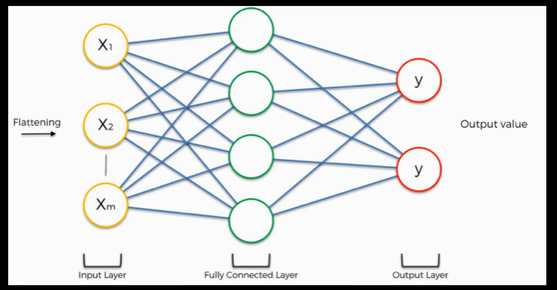




Pooling Layer
reduce the resolution



Flatten Layer
Convert 2D data to 1D for neural nets

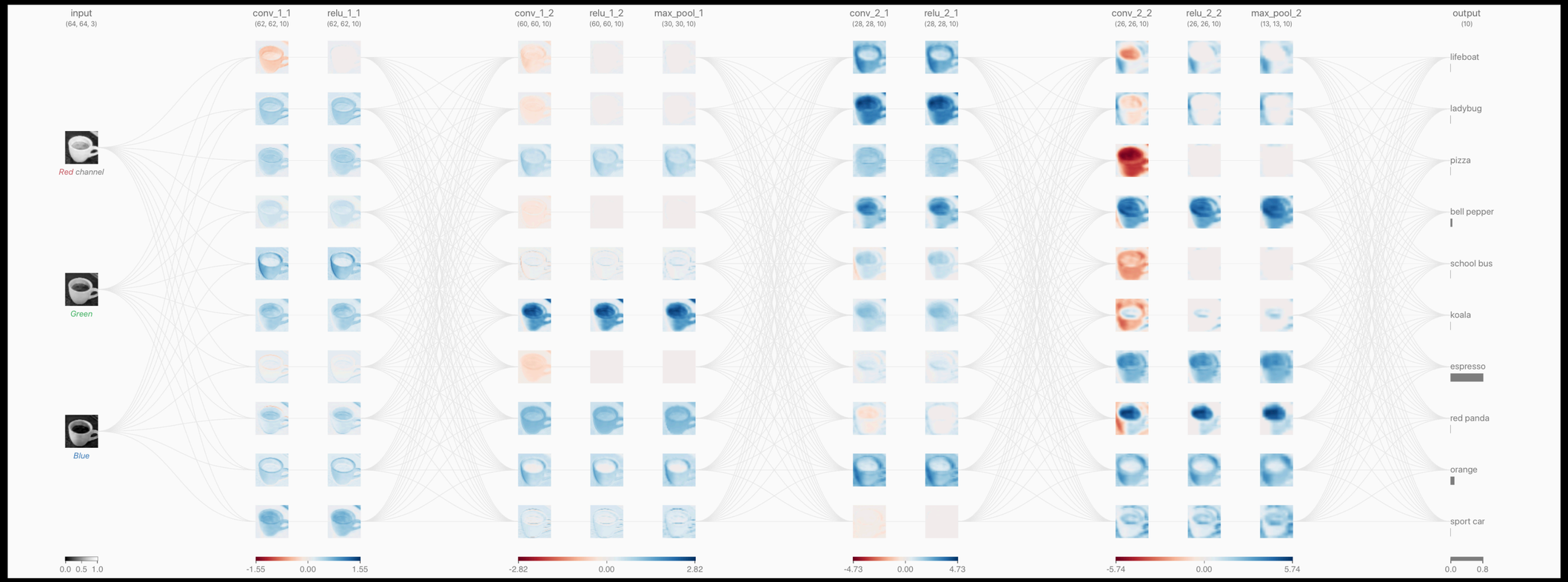


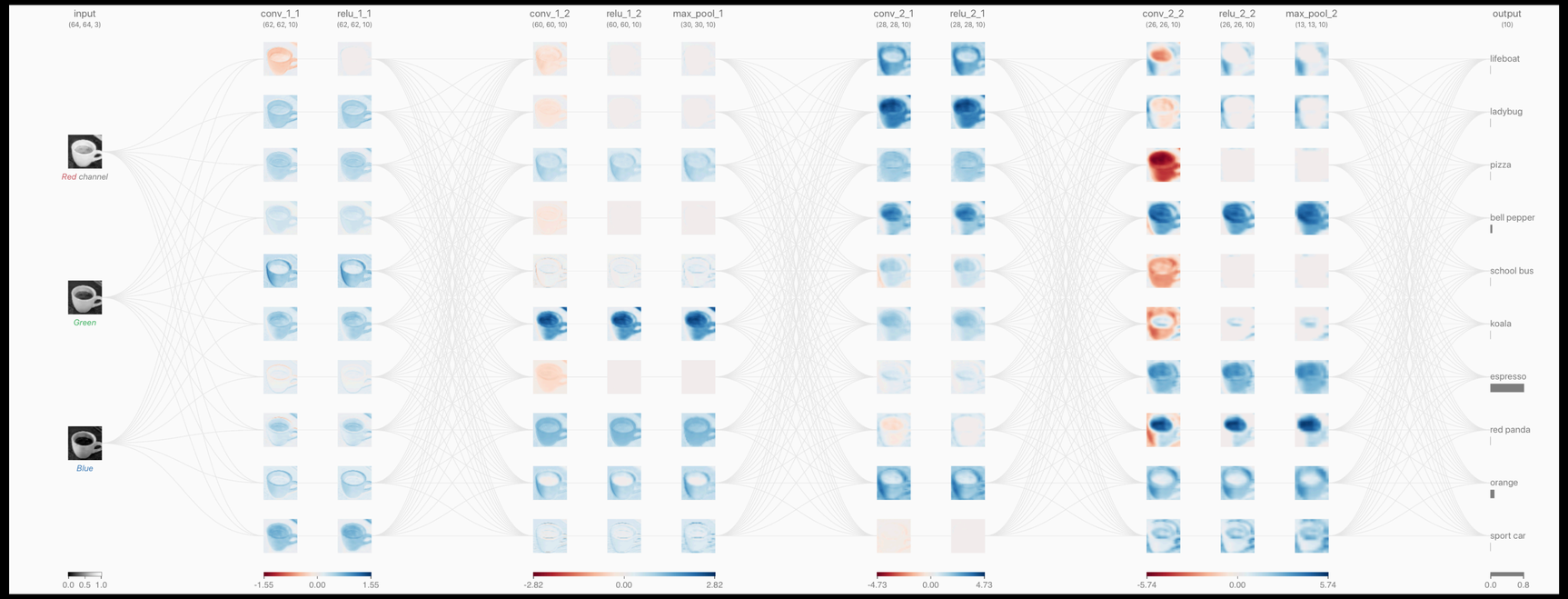
Fully-Connected Layer
Neural Nets for pattern recognition

Other operations

ReLu Activation

max(pixel_value, 0)
Add non-linearity




input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.55 0.00 1.55
-2.82 0.00 2.82
-4.73 0.00 4.73
-5.74 0.00 5.74
0.0 0.8


input
(64, 64, 3)
conv_1_1
(62, 62, 10)
relu_1_1
(62, 62, 10)
conv_1_2
(60, 60, 10)
relu_1_2
(60, 60, 10)
max_pool_1
(30, 30, 10)
conv_2_1
(28, 28, 10)
relu_2_1
(28, 28, 10)
conv_2_2
(26, 26, 10)
relu_2_2
(26, 26, 10)
max_pool_2
(13, 13, 10)
output
(10)
Red channel
Green
Blue
lifeboat
ladybug
pizza
bell pepper
school bus
koala
espresso
red panda
orange
sport car
0.0 0.5 1.0
-1.49 0.00 1.49
-2.22 0.00 2.22
-4.64 0.00 4.64
-5.84 0.00 5.84
0.0 0.8