

Convolutional Neural Networks

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Pattern Recognition and Machine Learning

Lecture 16 Convolutional Neural Networks

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences



Autumn 2026

Lecture plan and time allocation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Topic	Minutes
Convolution	45
Feature maps and network layers	45
Vision architectures	45

One image patch, then a complete classifier

Reuse a local filter; track shapes and parameters; match the head to the task.

The numerical steps are shown on the slides.



Flattening a large image makes a dense layer expensive

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

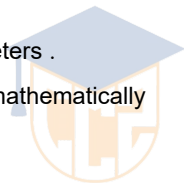
A spatial image contains positions and channels with defined meanings.

Many useful patterns are local and recur at different positions. A convolution encodes these assumptions.

$$224 \cdot 224 \cdot 3 = 150,528$$

Dense layer to 1,000 units: $150,528(1,000) + 1,000 = 150,529,000$ parameters .

Check. This prior is helpful when it matches the task; images do not make dense layers mathematically invalid.



One RGB position has three channel values

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Axis	Interpretation	Example
H	height, row coordinate	32 Rows
W	width, column coordinate	32 Columns
C	channels at each spatial position	RGB is 3
B	examples in a batch	8 images

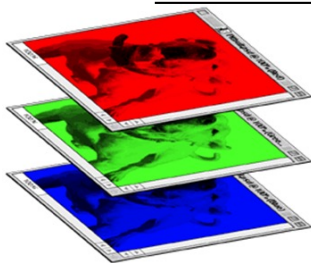


Image libraries may store $H \times W \times C$; permute axes explicitly.

One spatial location has a red, green and blue value. Together they describe one pixel. The channel dimension holds these measurements of the same image.

Preprocessing gives pixel values a defined meaning

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Scaling changes the numerical range; normalization uses a specified channel mean and standard deviation.

$$q \in \{0, \dots, 255\}$$

$$q = 128 \quad \mu = 0.5$$
$$\sigma = 0.25$$

$$x = q/255, \quad x'_c = (x_c - \mu_c)/\sigma_c.$$



$$q = 128 : \quad x = 128/255 \approx 0.50196, \quad x' = (0.50196 - 0.5)/0.25 \approx 0.00784.$$

Spatial arrangement distinguishes an image from a bag of pixels

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Representation choices must preserve information that matters for the target. Spatial order often matters for vision.



$$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad X' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Local patterns recur at different positions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

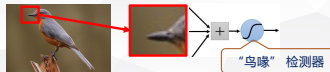
Local connectivity limits what each unit sees; weight sharing reuses the same test across positions.

对图像，为什么要用CNN？

- 有些模式比整个图像小得多

神经元不需要看整幅图像就能发现模式

与一个小区域连接所需的参数更少



Equivariance moves responses; invariance keeps a decision

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For a stride-one shared convolution away from boundary effects:

$$\text{Conv}(T_{\Delta}X) = T_{\Delta} \text{Conv}(X).$$



A filter makes one local weighted sum at each position

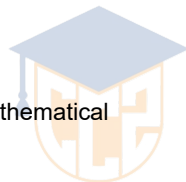
ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

For one channel, stride 1, no padding, and a 3×3 filter:

$$Y_{i,j} = b + \sum_{u=0}^2 \sum_{v=0}^2 K_{u,v} X_{i+u,j+v}.$$

Check. Most deep-learning libraries perform cross-correlation: the filter is not flipped. Mathematical convolution flips the kernel. Learned filters still serve the same modeling role.



A complete example: a 3x3 filter on a 4x4 input

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The right side is brighter than the left in this patch; this filter assigns a negative response.

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad b = 0.$$

$$Y_{1,1} = (1 - 3) + (5 - 7) + (9 - 11) = -6.$$



The top-left convolution response: all nine products

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Move the same kernel one column right. The weights remain unchanged while the covered input values change.

$$P = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}.$$



Worked calculation

$$\begin{aligned} Y_{1,1} &= 1(1) + 2(0) + 3(-1) \\ &\quad + 5(1) + 6(0) + 7(-1) \\ &\quad + 9(1) + 10(0) + 11(-1) + 0 \\ &= (1 - 3) + (5 - 7) + (9 - 11) = -6. \end{aligned}$$

A feature-map value is a response, not a label

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A feature map keeps one response for every valid filter placement.

$$Y_{1,1} = (1 - 3) + (5 - 7) + (9 - 11) = -6,$$

$$Y_{1,2} = (2 - 4) + (6 - 8) + (10 - 12) = -6,$$

$$Y_{2,1} = (5 - 7) + (9 - 11) + (13 - 15) = -6,$$

$$Y_{2,2} = (6 - 8) + (10 - 12) + (14 - 16) = -6.$$

$$Y = \begin{bmatrix} -6 & -6 \\ -6 & -6 \end{bmatrix}, \quad \text{ReLU}(Y) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$



Hidden activations are intermediate features. A CNN organizes them by both channel and spatial coordinate.

The activation determines which gradients reach the kernel

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

One kernel weight is reused at four output positions. Let G_{ij} be the loss derivative at the raw convolution output.

$$Y_{ij} = \sum_{u,v} K_{uv} X_{i+u,j+v} + b, \quad \frac{\partial L}{\partial K_{uv}} = \sum_{i,j} G_{ij} X_{i+u,j+v}.$$

Loss directly sums the raw outputs

Then $G_{ij} = 1$ at all four positions. For the top-left kernel weight,

$$\partial_{K_{00}} L = 1 + 2 + 5 + 6 = 14.$$

$$\partial_b L = 1 + 1 + 1 + 1 = 4.$$

Loss receives outputs through ReLU

In the running patch, all four raw responses are -6 . ReLU has derivative 0 there.

$$G_{ij} = \frac{\partial L}{\partial \text{ReLU}(Y_{ij})} 0 = 0.$$

Hence the kernel and bias gradients through these outputs are zero.

Different loss paths explain the different answers. Sharing means adding every use of the same parameter. In general, $\partial L / \partial b = \sum_{i,j} G_{ij}$ because the same bias is added at every output position.

Padding preserves size but introduces a boundary rule

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For each spatial axis, with input size n , kernel size k , stride s , and padding p on each side:

$$m = \left\lfloor \frac{n + 2p - k}{s} \right\rfloor + 1.$$

n	k	p	s	m
4	3	0	1	2
32	3	1	1	32
32	3	1	2	16
32	5	0	1	28



For odd k , stride 1, and dilation 1, using $p = (k - 1)/2$ on each side preserves spatial size.

For an odd kernel size, stride 1 and dilation 1, choose $p = (k - 1)/2$, so $k = 3$ gives $p = 1$, preserving output length 32.

Check. With stride greater than one, implementations may use asymmetric padding to achieve their documented output convention.

Worked review: one map, one filter, many uses

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The next filter reads several channels. Spatial size, channel count, parameter count, and compute cost must be tracked separately.

Quantity	Computation	Result
Trainable parameters	$3 \times 3 + 1$	10
Output Height	$10 - 3 + 1$	8
Output width	$12 - 3 + 1$	10
Output Value	8×10	80
Uses of each kernel weight	Once per output location	80 uses



All 80 outputs share 10 parameters. More output locations do not create a new set of weights at each location.

Stride changes the sampling grid

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A single-channel convolution already gives a complete spatial map.

Along one axis, take input length $n = 7$, kernel size $k = 3$ and no padding. Starting indices begin at 0. Stride changes the spacing of windows and therefore the number of outputs.

Stride	Window starts	Output length
$s = 1$	0, 1, 2, 3, 4	$\lfloor (7 - 3)/1 \rfloor + 1 = 5$
$s = 2$	0, 2, 4	$\lfloor (7 - 3)/2 \rfloor + 1 = 3$

Stride changes the spacing between window starts. A 3×3 single-input-channel filter still has nine weights, plus one bias per output channel.

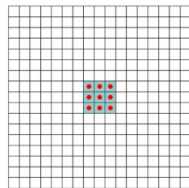


Dilation spreads the same kernel samples farther apart

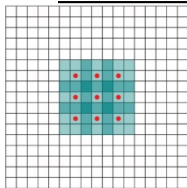
ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$$k_{\text{eff}} = d(k - 1) + 1, \quad m = \left\lfloor \frac{n + 2p - k_{\text{eff}}}{s} \right\rfloor + 1.$$

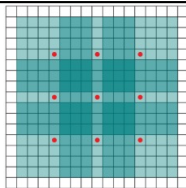
Kernel	Dilation	Effective extent	Weights per channel pair
3×3	1	3×3	9
3×3	2	5×5	9
3×3	3	7×7	9



(a)



(b)



(c)

The filter samples spaced input locations, not every point in the larger extent.

Each panel has nine sampled positions. Dilation increases their spacing and coverage without increasing the weight count. Stride instead changes the movement between neighboring output windows.



One output channel combines all input channels

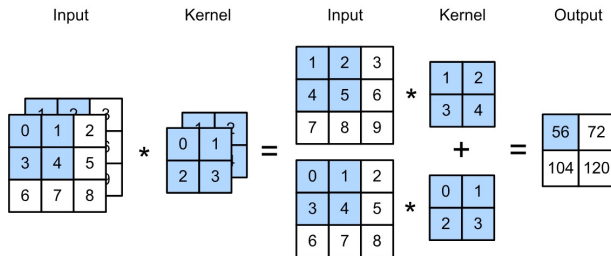
ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Channel 1: $1(1) + 2(2) + 4(3) + 5(4) = 37$,

Channel 2: $0(0) + 1(1) + 3(2) + 4(3) = 19$,

Combined output: $37 + 19 + 0 = 56$.

$$Y_{ij} = b + \sum_{c=1}^{C_{in}} \sum_{u,v} K_{cuv} X_{c,i+u,j+v}.$$



A filter has one spatial slice per input channel; adding those slice responses gives one output channel.

Convolution parameters depend on channels and kernel size

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

One complete filter covers every input channel and produces one output map.

$$X \in \mathbb{R}^{B \times C_{\text{in}} \times H \times W}, \quad K \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_H \times k_W}, \quad b \in \mathbb{R}^{C_{\text{out}}}, \\ Y \in \mathbb{R}^{B \times C_{\text{out}} \times H' \times W'}.$$

Key idea

For standard convolution with one bias per output channel:

$$\# \text{ parameters} = C_{\text{out}}(C_{\text{in}}k_Hk_W + 1).$$

Layer	Weights	Biases	Sum
$3 \rightarrow 16, 3 \times 3$	$16 \cdot 3 \cdot 9 = 432$	16	448
$16 \rightarrow 32, 3 \times 3$	$32 \cdot 16 \cdot 9 = 4,608$	32	4,640

Local connections and sharing remove different parameters

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Both restrictions encode image assumptions and reduce model freedom.

Connectivity	Parameter reasoning	Sum
Dense to every output	$(3072 + 1) \cdot 16384$	50,348,032
Local, no sharing	$16384(27 + 1)$	458,752
Shared 3×3 Conv	$16(27 + 1)$	448



A 1x1 convolution mixes channels at each position

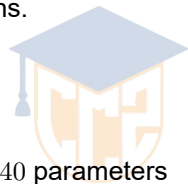
ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

It combines channels at one location; it does not directly mix neighboring locations.

$$Y_{o,i,j} = b_o + \sum_{c=1}^{C_{in}} K_{o,c} X_{c,i,j}.$$

$$0.5(2) - 0.2(5) + 1 = 1$$

Another 64 → 16 for this 1 × 1 Layer: 16(64 + 1) = 1,040 parameters



Pooling gradients follow the forward selection

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Downsampling can help classification efficiency but can hurt localization of small structures.

$$A = \begin{bmatrix} 1 & 3 & 2 & 0 \\ 4 & 2 & 1 & 5 \\ 0 & 1 & 6 & 2 \\ 3 & 2 & 4 & 1 \end{bmatrix}.$$

$$\text{MaxPool}(A) = \begin{bmatrix} 4 & 5 \\ 3 & 6 \end{bmatrix},$$

$$\text{AvgPool}(A) = \begin{bmatrix} 2.5 & 2 \\ 1.5 & 3.25 \end{bmatrix}.$$

For the window $\begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix}$ with upstream gradient 2, max pooling sends the gradient to the entry 4. Average pooling divides it equally among all four entries.

$$g = 2$$

$\nabla_A L$ through max pooling

$$= \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix},$$

$\nabla_A L$ through average pooling

$$= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}.$$

Track receptive-field size and spacing through a stack

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Let r_l be receptive-field width in original pixels and j_l the spacing between adjacent output centers. Start with $r_0 = j_0 = 1$.

$$r_l = r_{l-1} + (k_{\text{eff}} - 1)j_{l-1}, \quad j_l = j_{l-1}s_l.$$

Stage	Kernel	Stride	r	j
Input	—	—	1	1
Convolution	3	1	$1 + 2(1) = 3$	1
Pooling	2	2	$3 + 1(1) = 4$	2
Convolution	3	1	$4 + 2(2) = 8$	2
Pooling	2	2	$8 + 1(2) = 10$	4



Without downsampling, three stride-one 3×3 layers give $3 \rightarrow 5 \rightarrow 7$. After downsampling, use the old spacing j when enlarging the field.

Channel count does not by itself change the spatial receptive field. Padding affects boundary coordinates; this table tracks interior extents.

A small classifier: calculate every stage

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$32 \times 8 \times 8$ to ten classes: flatten then dense uses $2048(10) + 10 = 20,490$ parameters; global average then dense uses $32(10) + 10 = 330$.

$$2048(10) + 10 = 20,490 \quad 32(10) + 10 = 330$$

Use multiclass cross-entropy on the ten logits; backpropagation trains both filters and the output head.

Stage	Operation	Output per image	Parameters
Input	RGB image	$3 \times 32 \times 32$	0
1	Conv $3 \rightarrow 16$, $k = 3$, $p = 1$; ReLU	$16 \times 32 \times 32$	448
2	Max pooling $k = 2$, $s = 2$	$16 \times 16 \times 16$	0
3	Conv $16 \rightarrow 32$, $k = 3$, $p = 1$; ReLU	$32 \times 16 \times 16$	4,640
4	Max pooling $k = 2$, $s = 2$	$32 \times 8 \times 8$	0
5	Global average pooling	32	0
6	Dense layer $32 \rightarrow 10$	10 logits	330

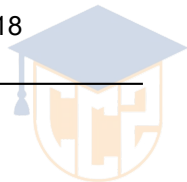
$$448 + 4,640 + 330 = 5,418$$

Head design can dominate the parameter count

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The convolutional feature maps are identical up to the last pooling stage.

Head	Trunk parameters	Head parameters	Sum
After flattening $2048 \rightarrow 10$	5,088	20,490	25,578
After global average pooling $32 \rightarrow 10$	5,088	330	5,418



The code follows the shape table

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

```
model = torch.nn.Sequential(  
    torch.nn.Conv2d(3, 16, 3, padding=1),  
    torch.nn.ReLU(),  
    torch.nn.MaxPool2d(2),  
    torch.nn.Conv2d(16, 32, 3, padding=1),  
    torch.nn.ReLU(),  
    torch.nn.MaxPool2d(2),  
    torch.nn.AdaptiveAvgPool2d(1),  
    torch.nn.Flatten(),  
    torch.nn.Linear(32, 10))  
logits = model(images) # images: B x 3 x 32 x 32  
loss = torch.nn.functional.cross_entropy(logits, labels)
```

Logits have shape $B \times 10$;
labels are integer class indices
with shape B . Count
parameters and compare with
5,418.

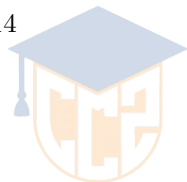


Worked review: four quantities, four different questions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Take input shape $3 \times 28 \times 28$, eight output channels, a 3×3 kernel, padding 1, stride 2 and dilation 1. Use the same configuration to compute shape, parameter count and receptive field.

Question	Calculation and result
Spatial output shape	$\lfloor (28 + 2 - 3)/2 \rfloor + 1 = 14$
Output shape	$8 \times 14 \times 14$
Number of parameters	$8(3 \times 3 \times 3 + 1) = 224$
Receptive field in input pixels	3×3
Spacing between outputs in input pixels	2 pixels



Increasing output channels to 16 adds response types and parameters. By itself, it does not enlarge the spatial receptive field.

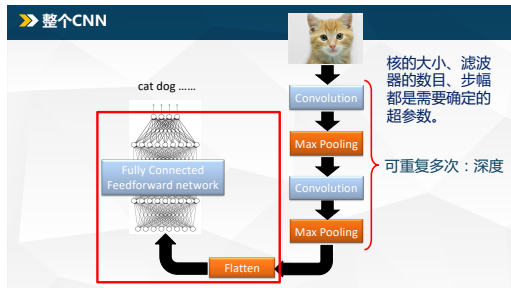
A hierarchy combines local evidence into larger context

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

We can now calculate the cost and geometry of individual layers. Early filters can respond to local patterns that later layers combine over larger regions. The pictured hierarchy is an illustration; a learned channel need not correspond to one named human concept.

Condition to check

A hierarchy enables feature composition; it does not guarantee that every channel has one named human concept.



Two small kernels can replace one larger spatial extent

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For stride-one layers with C input and output channels and biases omitted:

Option	Receptive-field extent	Weights	Nonlinear stages
One layer 5×5 Conv	5×5	$25C^2$	one
Two layers 3×3 Conv	5×5	$18C^2$	two if activated

Inception combines branches with matching spatial grids

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

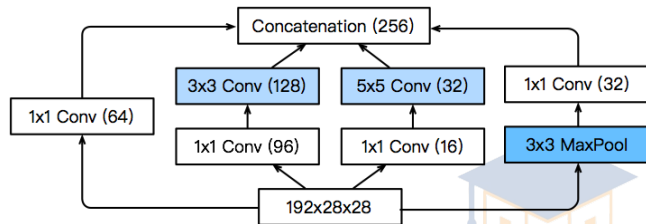
Branches with 64, 128, 32, 32 channels on the same $H \times W$ grid concatenate to $256 \times H \times W$.



Parallel branches concatenate their output channels

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Read the four branches upward. Their common spatial grid supports concatenation of 64, 128, 32 and 32 output channels into 256 channels. Concatenation stacks channels; addition would combine matching entries. The 1-by-1 convolutions also control intermediate channel counts.



A bottleneck reduces expensive channel mixing

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Only the spatial middle layer uses the small channel width.

Operation	Channel change	Weights
Direct 3×3	$128 \rightarrow 128$	$9 \cdot 128^2 = 147,456$
Use 1×1 Compression	$128 \rightarrow 32$	4,096
Use 3×3 for spatial processing	$32 \rightarrow 32$	9,216
Use 1×1 Extension	$32 \rightarrow 128$	4,096
Bottleneck total		17,408

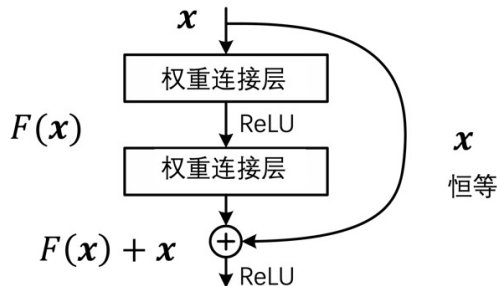


The identity path contributes an explicit gradient term

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

$$u = x + F(x; \theta), \quad x = 2, F(x) = 0.3 \Rightarrow u = 2.3.$$

$$\frac{\partial u}{\partial x} = I + J_F(x), \quad \frac{\partial L}{\partial x} = (I + J_F(x))^\top \frac{\partial L}{\partial u}.$$



Setting F close to zero preserves the input before any following activation.

Check. The original post-activation diagram applies ReLU after the sum; other residual designs place activations differently.

If $F(x) = 0.2x$ and the incoming gradient is $g = 3$, then $dL/dx = (1 + 0.2)3 = 3.6$. The two paths contribute 3 and 0.6.

$$u = x + F(x)$$

$$F(x) = 0.2x \quad g = 3$$

$$dL/dx = (1 + 0.2)3 = 3.6$$

Use a projection when residual shapes differ

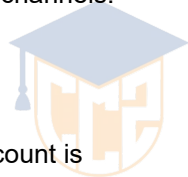
ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Adding 128-channel maps returns 128 channels; concatenating them returns 256 channels.

$$u = P(x) + F(x),$$

$$128 \cdot 64 = 8,192$$

A 1×1 projection with stride 2 maps $64 \times 32 \times 32$ to $128 \times 16 \times 16$. The weight count is $128 \times 64 = 8192$; add 128 if biases are included.



Depthwise separable convolution restricts channel connections

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The pointwise stage mixes them to produce 64 output channels.

$$C_{\text{in}} = 32 \quad C_{\text{out}} = 64$$

$$k = 3$$

Operation	Connection pattern	Weights
Standard convolution	spatial and channel mixing together	$9 \cdot 32 \cdot 64 = 18,432$
Depthwise stage	one spatial filter per input channel	$9 \cdot 32 = 288$
Pointwise stage	Use 1×1 channel mixing	$32 \cdot 64 = 2,048$
Separable total	depthwise then pointwise	2,336

Classification aggregates evidence for one image label

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Detection and segmentation must retain or recover spatial information.

The backbone and head are trained together using the image label.

feature map $C \times H \times W \longrightarrow$ pooled vector $C \longrightarrow K$ logits .

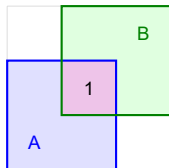
$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}, \quad L = -\log p_y.$$



Detection predicts both a category and a box

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Object detection needs predicted boxes as well as class scores. Overlap is measured by intersection over union. For these two area-4 boxes, the intersection is 1 and the union is 7.



$$|A| = 2 \times 2 = 4, \quad |B| = 4.$$

$$|A \cap B| = 1 \times 1 = 1.$$

$$|A \cup B| = 4 + 4 - 1 = 7.$$

$$\text{IoU} = \frac{1}{7} \approx 0.143.$$



Segmentation predicts classes on a spatial grid

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

and the target commonly has shape $B \times H \times W$ with one class index per pixel.

$$B \times K \times H \times W,$$

$$TP = 60 \quad FP = 20$$

$$FN = 20 \quad \text{IoU} = 60 / (60 + 20 + 20) = 0.6$$



Worked review: design and calculate a five-class CNN

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Earlier training provides useful visual features when the new labeled dataset is small.

For the displayed five-class network, verify output shapes, 5,253 parameters and a 7×7 second-layer receptive field. Then ask whether the task needs one image label or spatial outputs; global pooling would discard coordinates needed for detection.

$$k = 3, p = 1, s = 2 \quad k = 3, p = 1, s = 1$$

Result	Value	Reason
feature map	$16 \times 32 \times 32 \rightarrow 32 \times 32 \times 32$	stride then channel change
Parameters	$448 + 4,640 + 165 = 5,253$	two convolutions and one head
Receptive field	7×7	$r_1 = 3, j_1 = 2; r_2 = 3 + 2(2) = 7$

