Modern CNN Pipelines and Transfer Learning

Blocks, Scaling, Residuals, and Reusing Backbones

Prof. Nipun Batra

ES 667 · Deep Learning · IIT Gandhinagar

L8 gave us the operator; today we spend it deliberately

V VISUAL

L8 · MECHANICS

Local/shared cross-correlation; output shapes, receptive fields, weights, and MACs.

→

TODAY · BLOCKS

Allocate context, channel mixing, depth, and compute inside one stage.

→

TRANSFER

Reuse a learned backbone; validate how much of it must adapt.

L8 gave us the operator; today we spend it deliberately

V VISUAL

L8 · MECHANICS

Local/shared cross-correlation; output shapes, receptive fields, weights, and MACs.

→

TODAY · BLOCKS

Allocate context, channel mixing, depth, and compute inside one stage.

→

TRANSFER

Reuse a learned backbone; validate how much of it must adapt.

L6 already established why residual routes help optimization. L7 fixed the validation/test protocol. We will retrieve both—then use them inside a deployable CNN pipeline.

Commit: which design can meet the stage budget?

Q QUESTION

HELD CASE · DEPLOYMENT CONTRACT $28 \times 28 \times 64 \to 28 \times 28 \times 128$, same spatial grid, receptive field at least 3×3, fewer than 10 million convolution MACs.

A · STANDARD	B · FACTORIZED	C · EITHER
one shared 3×3, $64 \to 128$	separate spatial filtering from channel mixing	sharing alone guarantees the budget

Commit: which design can meet the stage budget?

Q QUESTION

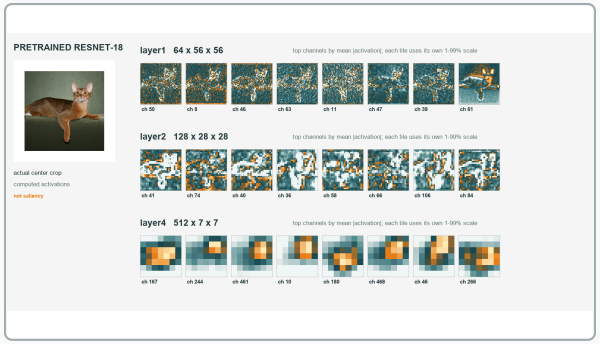
HELD CASE · DEPLOYMENT CONTRACT $28 \times 28 \times 64 \to 28 \times 28 \times 128$, same spatial grid, receptive field at least 3×3, fewer than 10 million convolution MACs.

A · STANDARD	B · FACTORIZED	C · EITHER
one shared 3×3, $64 \to 128$	separate spatial filtering from channel mixing	sharing alone guarantees the budget

Commit to A, B, or C. Write the count you would calculate first; keep the choice until the closing revisit.

Outline

V VISUAL



MEASURED actual pretrained ResNet-18 activations on the held photograph

01 · CONTEXT

Spend depth and kernel size to enlarge what a feature can see.

02 · BOTTLENECKS

Mix channels cheaply with 1×1 projections.

03 · RESIDUALS

Preserve an identity route while learning a correction.

04 · EFFICIENT CONV

Factor spatial filtering from channel mixing.

05 · TRANSFER

Reuse a pretrained backbone under its exact input recipe.

06 · VALIDATE

Probe, fine-tune, and diagnose on separated data roles.

held input · learned operation · context / shape · compute / adaptation · verified · failure

One ledger will carry every design decision

V VISUAL

HELD CASE · LEDGER CONVENTION All stage candidates use $H_{\text{out}} = W_{\text{out}} = 28$, stride 1, stated same padding, and no bias. One MAC means one multiply–accumulate per weight per output location.

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	?	?

Budget: fewer than $10,000,000$ MACs. Shape alone is not enough.

One ledger will carry every design decision

V VISUAL

HELD CASE · LEDGER CONVENTION All stage candidates use $H_{\text{out}} = W_{\text{out}} = 28$, stride 1, stated same padding, and no bias. One MAC means one multiply–accumulate per weight per output location.

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	?	?

Budget: fewer than $10,000,000$ MACs. Shape alone is not enough.

Every candidate must preserve the output contract; only context, factorization, route, and cost may change.

Baseline: count the standard 3×3 before optimizing

Q QUESTION

HELD CASE · CANDIDATE 0 One same-padded 3×3 convolution maps $64 \to 128$ channels at 28×28 output locations. Bias is omitted.

$$\text{weights} = 3 \cdot 3 \cdot 64 \cdot 128$$

Baseline: count the standard 3×3 before optimizing

Q QUESTION

HELD CASE · CANDIDATE 0 One same-padded 3×3 convolution maps $64 \to 128$ channels at 28×28 output locations. Bias is omitted.

$$\text{weights} = 3 \cdot 3 \cdot 64 \cdot 128$$

$$\text{MACs} = H_{\text{out}} W_{\text{out}} \cdot \text{weights}$$

Baseline: count the standard 3×3 before optimizing

Q QUESTION

HELD CASE · CANDIDATE 0 One same-padded 3×3 convolution maps $64 \to 128$ channels at 28×28 output locations. Bias is omitted.

$$\text{weights} = 3 \cdot 3 \cdot 64 \cdot 128$$

$$\text{MACs} = H_{\text{out}} W_{\text{out}} \cdot \text{weights}$$

Predict both exact counts and whether the design meets the 10M budget.

Baseline answer: sharing is necessary but not sufficient

A ANSWER

Baseline answer: sharing is necessary but not sufficient

A ANSWER

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	$57,802,752$

$57.803\text{M} > 10\text{M}$: the output contract passes; the deployment budget fails.

WHAT L8 BOUGHT

One kernel bank is reused at all 784 locations.

WHAT REMAINS

Reduce work inside the kernel bank without changing the output.

Baseline answer: sharing is necessary but not sufficient

A ANSWER

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	$57,802,752$

$57.803\text{M} > 10\text{M}$: the output contract passes; the deployment budget fails.

WHAT L8 BOUGHT

One kernel bank is reused at all 784 locations.

WHAT REMAINS

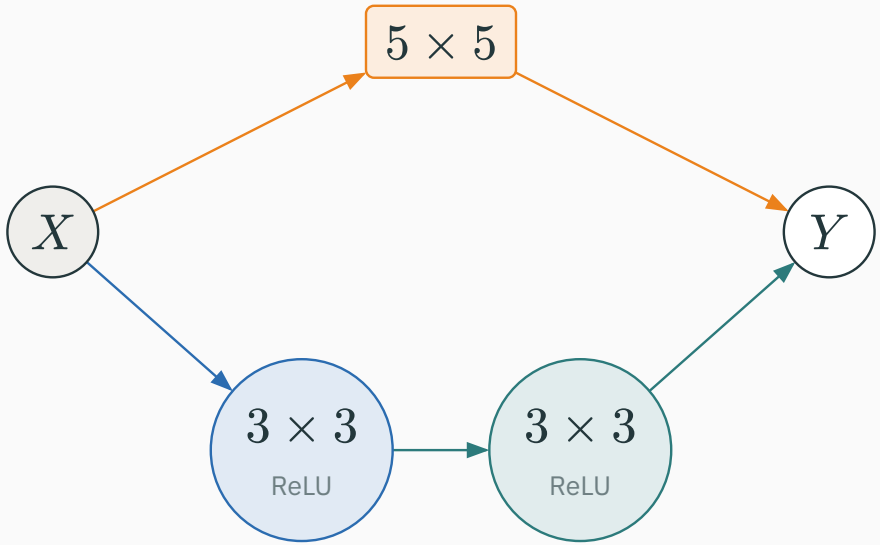
Reduce work inside the kernel bank without changing the output.

The lecture is now a controlled search for a cheaper stage—not a tour of architecture names.

Design move 1 — grow context with small kernels

HELD CASE · CONTEXT Keep the held $28 \times 28 \times 64 \to 28 \times 28 \times 128$ output. Now require a 5×5 receptive field.

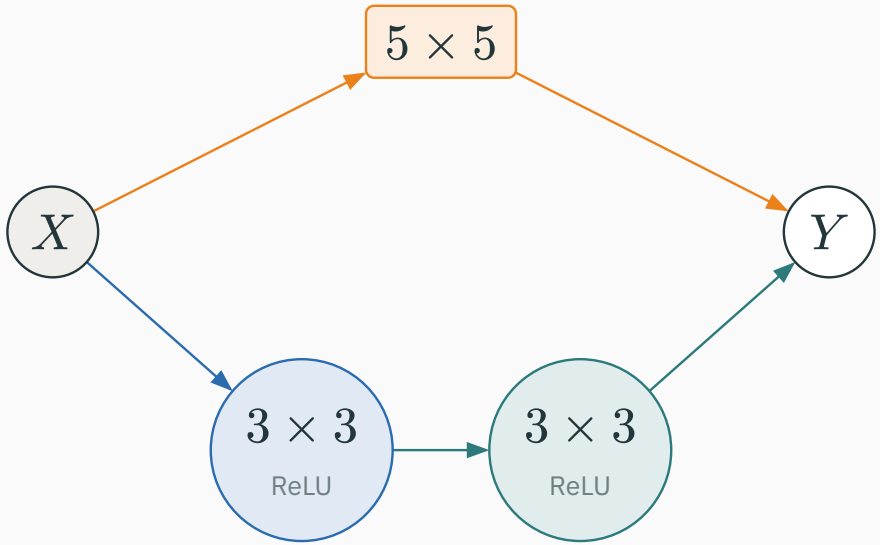
Should we use one 5×5 convolution—or compose two same-padded 3×3s?



Design move 1 — grow context with small kernels

HELD CASE · CONTEXT Keep the held $28 \times 28 \times 64 \to 28 \times 28 \times 128$ output. Now require a 5×5 receptive field.

Should we use one 5×5 convolution—or compose two same-padded 3×3s?



Both paths see the same spatial extent. Only the blue–teal path inserts an extra learned nonlinearity.

Receptive field: derive it one layer at a time

D DERIVATION

With stride 1, each $k \times k$ layer grows the receptive-field side length by $k - 1$:

Receptive field: derive it one layer at a time

D DERIVATION

With stride 1, each $k \times k$ layer grows the receptive-field side length by $k - 1$:

$$r_\ell = r_{\ell-1} + (k - 1)$$

Receptive field: derive it one layer at a time

D DERIVATION

With stride 1, each $k \times k$ layer grows the receptive-field side length by $k - 1$:

$$r_\ell = r_{\ell-1} + (k - 1)$$

$$1 \overset{3\times3}{\to} 3 \overset{3\times3}{\to} 5 \overset{3\times3}{\to} 7 \overset{3\times3}{\to} 9$$

Four stacked 3×3 layers match one 9×9 receptive field.

They also insert **four** transformations and **four** activations instead of one.

Depth grows context gradually—and learns what to keep after each step.

Same output and context: count both candidates

D DERIVATION

The stacked path uses $64 \to 64 \to 128$; the direct path uses $64 \to 128$. Biases are omitted.

	two 3×3s	**one 5×5**
weights	$9(64 \cdot 64 + 64 \cdot 128)$	$25 \cdot 64 \cdot 128$
count	$110,592$	$204,800$
ReLUs	2	1

Same output and context: count both candidates

D DERIVATION

The stacked path uses $64 \to 64 \to 128$; the direct path uses $64 \to 128$. Biases are omitted.

	two 3×3s	one 5×5
weights	$9(64 \cdot 64 + 64 \cdot 128)$	$25 \cdot 64 \cdot 128$
count	$110,592$	$204,800$
ReLUs	2	1

$204,800 - 110,592 = 94,208$ fewer weights: a 46% reduction.

Small kernels are not merely aesthetic: they buy equal context, fewer weights, and more nonlinear stages.

Ledger: more context still misses the deployment budget

D DERIVATION

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28 \times 28 \times 128$	5×5	$110,592$	86.704M
one 5×5	$28 \times 28 \times 128$	5×5	$204,800$	160.563 M

All three preserve shape with stated same padding. None meets 10M.

Ledger: more context still misses the deployment budget

D DERIVATION

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28 \times 28 \times 128$	5×5	$110,592$	86.704M
one 5×5	$28 \times 28 \times 128$	5×5	$204,800$	160.563 M

All three preserve shape with stated same padding. None meets 10M.

> Stacking small kernels is a context decision. It does not, by itself, solve the channel-pair cost.

Misconception check: does depth always grow spatial context?

Q QUESTION

Stack ten 1×1 convolutions. What is the receptive field?

A · 1×1

B · 10×10

C · depends on channels

Misconception check: does depth always grow spatial context?

Q QUESTION

Stack ten 1×1 convolutions. What is the receptive field?

A · 1×1

B · 10×10

C · depends on channels

Answer: A. Depth composes transformations; kernel size and stride determine spatial reach.

Design move 2 — mix channels without mixing neighbours

At one spatial location, collect the C_{in} channel values into a vector x_{hw}:

Design move 2 — mix channels without mixing neighbours

At one spatial location, collect the C_{in} channel values into a vector x_{hw}:

$$y_{hw} = W x_{hw} + b, \quad W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$$

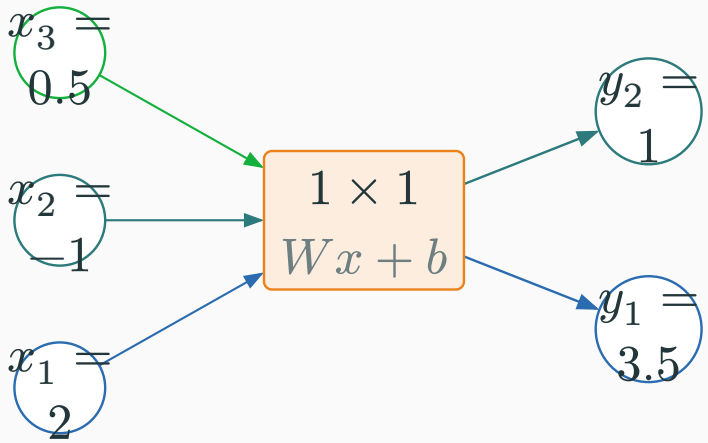
The **same** matrix W is used at every location.

No value from $(h+1, w)$ or $(h, w+1)$ is read.

A 1×1 convolution is a shared, per-pixel linear layer across channels.

One pixel, two learned channel combinations

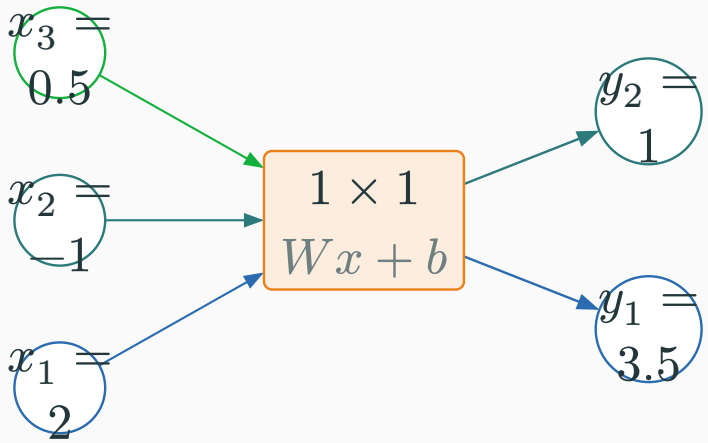
D DERIVATION



For $x = (2, -1, 0.5)^\top$, choose $W = \begin{pmatrix} 1 & -1 & 0 \\ 0.5 & 0 & 2 \end{pmatrix}$ and $b = (0.5, -1)^\top$.

One pixel, two learned channel combinations

D DERIVATION



For $x = (2, -1, 0.5)^\top$, choose $W = \begin{pmatrix} 1 & -1 & 0 \\ 0.5 & 0 & 2 \end{pmatrix}$ and $b = (0.5, -1)^\top$.

$$y = \begin{pmatrix} 1 & -1 & 0 \\ 0.5 & 0 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ 0.5 \end{pmatrix} + \begin{pmatrix} 0.5 \\ -1 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 1 \end{pmatrix}.$$

Two output channels = two different learned summaries of the same input channels.

The same channel mixer runs over the whole grid

V VISUAL

The same channel mixer runs over the whole grid

V VISUAL

x_{hw}	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
three channels	$\begin{pmatrix} 2 \\ -1 \\ 0.5 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 0 \\ 1 \end{pmatrix}$
$Wx + b$	$\begin{pmatrix} 3.5 \\ 1 \end{pmatrix}$	$\begin{pmatrix} -0.5 \\ 3 \end{pmatrix}$	$\begin{pmatrix} -1.5 \\ -1.5 \end{pmatrix}$	$\begin{pmatrix} 3.5 \\ 2.5 \end{pmatrix}$

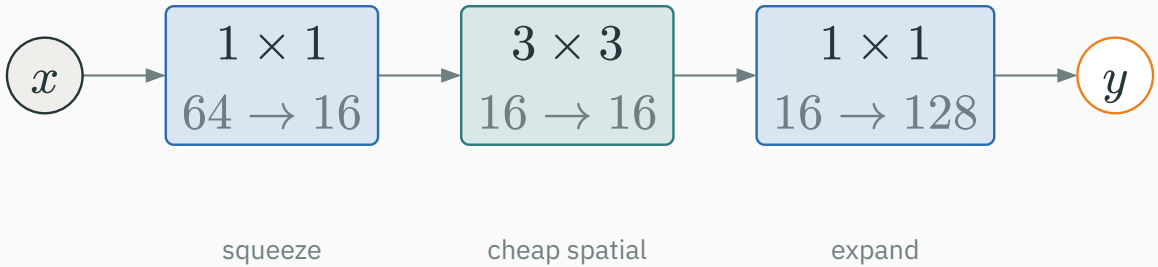
$$2 \times 2 \times 3 \xrightarrow{\text{same } 1\times1 \text{ weights}} 2 \times 2 \times 2$$

Spatial coordinates stay aligned; only the channel representation changes.

Bottleneck: spend the 3×3 in a narrow channel space

V VISUAL

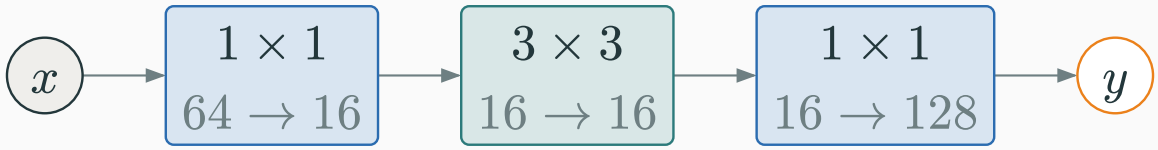
HELD CASE · CANDIDATE 2 Narrow $64 \to 16$, perform the spatial work at width 16, then expand $16 \to 128$. Same 28×28 grid throughout.



Bottleneck: spend the 3×3 in a narrow channel space

V VISUAL

HELD CASE · CANDIDATE 2 Narrow $64 \to 16$, perform the spatial work at width 16, then expand $16 \to 128$. Same 28×28 grid throughout.



squeeze cheap spatial expand

$$64 \xrightarrow{\text{choose 16 mixtures}} 16 \xrightarrow{\text{spatial work}} 16 \xrightarrow{\text{expand}} 128$$

The two 1×1s learn which channel combinations deserve expensive spatial processing.

Bottleneck: complete cost comparison

D DERIVATION

For the held 28×28 stage, compare the direct 3×3, $64 \to 128$, with $64 \to 16 \to 16 \to 128$:

$$\text{direct weights} = 3 \cdot 3 \cdot 64 \cdot 128 = 73,728.$$

With the bottleneck:

Bottleneck: complete cost comparison

D DERIVATION

For the held 28×28 stage, compare the direct 3×3, $64 \to 128$, with $64 \to 16 \to 16 \to 128$:

$$\text{direct weights} = 3 \cdot 3 \cdot 64 \cdot 128 = 73,728.$$

With the bottleneck:

$$64 \cdot 16 + 3 \cdot 3 \cdot 16 \cdot 16 + 16 \cdot 128 = 1,024 + 2,304 + 2,048 = 5,376.$$

Each weight is reused at $28^2 = 784$ locations: 57.803 million versus 4.215 million MACs.

$5,376 \cdot 784 = 4,214,784$**: the first candidate under the** 10**M budget.**

Misconception check: what did the 1×1 save?

A ANSWER

It did **not** make the final 3×3 kernel spatially smaller.

wrong: $3 \times 3 \to 1 \times 1$ spatial window

right: $64 \to 16$ channels **before** the 3×3

Misconception check: what did the 1×1 save?

A ANSWER

It did **not** make the final 3×3 kernel spatially smaller.

wrong: $3 \times 3 \to 1 \times 1$ spatial window

right: $64 \to 16$ channels **before** the 3×3

The saving comes from fewer input–output channel pairs inside the expensive spatial convolution.

Ledger: channel compression changes the decision

V VISUAL

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28 \times 28 \times 128$	5×5	$110,592$	86.704M
bottleneck 16	$28 \times 28 \times 128$	3×3	$5,376$	4.215M

GREEN means the candidate satisfies shape, receptive-field, and compute contracts.

Ledger: channel compression changes the decision

V VISUAL

candidate	output	RF	weights	MACs
standard 3×3	$28 \times 28 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28 \times 28 \times 128$	5×5	$110,592$	86.704M
bottleneck 16	$28 \times 28 \times 128$	3×3	$5,376$	4.215M

GREEN means the candidate satisfies shape, receptive-field, and compute contracts.

The bottleneck clears the budget with 5.785M MACs to spare.

Let one block inspect several scales

HELD CASE · CANDIDATE 3 The held stage may need channel evidence, 3×3 texture, 5×5 context, and a pooled summary at the same location.

Can four branches preserve the 28×28 grid, concatenate to 128 channels, and stay under budget?

Let one block inspect several scales

HELD CASE · CANDIDATE 3 The held stage may need channel evidence, 3×3 texture, 5×5 context, and a pooled summary at the same location.

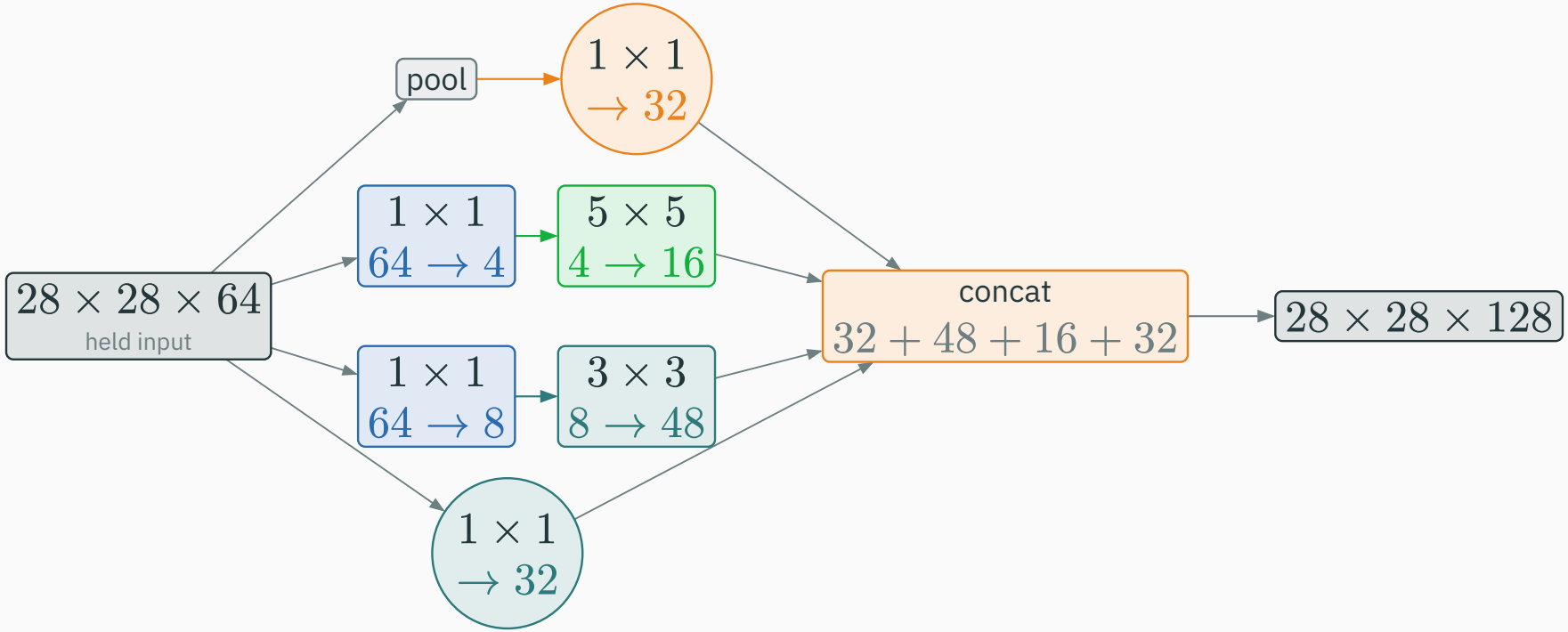
Can four branches preserve the 28×28 grid, concatenate to 128 channels, and stay under budget?

1×1 channel evidence 3×3 local pattern 5×5 wider context pooling summary

Inception runs alternatives in parallel; the stage output keeps every branch as a channel group.

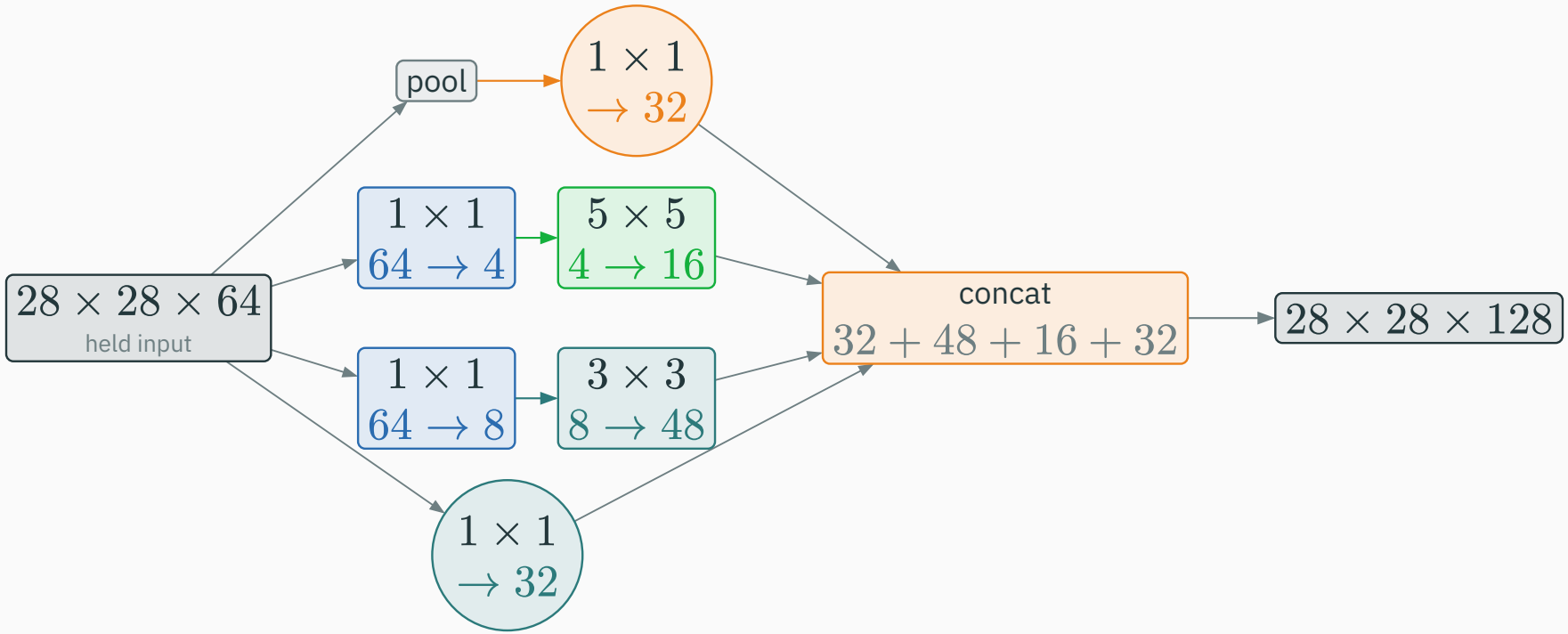
A shaped Inception candidate: every width is visible

V VISUAL



A shaped Inception candidate: every width is visible

V VISUAL



$32 + 48 + 16 + 32 = 128$ output channels; height and width remain 28×28.

Concatenation stacks channels; it does not average or add branch outputs.

Count the multi-scale candidate branch by branch

D DERIVATION

All convolutions use stride 1. The 3×3, 5×5, and 3×3 pool use padding 1, 2, and 1, so every branch remains 28×28.

Count the multi-scale candidate branch by branch

D DERIVATION

All convolutions use stride 1. The 3×3, 5×5, and 3×3 pool use padding 1, 2, and 1, so every branch remains 28×28.

branch	weight expression	weights
$1 \times 1 \to 32$	$64 \cdot 32$	$2,048$
$1 \times 1 \to 8 \xrightarrow{3\times3} 48$	$64 \cdot 8 + 9 \cdot 8 \cdot 48$	$3,968$
$1 \times 1 \to 4 \xrightarrow{5\times5} 16$	$64 \cdot 4 + 25 \cdot 4 \cdot 16$	$1,856$
pool $\xrightarrow{1\times1} 32$	$64 \cdot 32$	$2,048$

$9,920$ weights and $9,920 \cdot 784 = 7,777,280$ convolution MACs. Pooling comparisons are excluded from this ledger.

Concatenate only after every spatial shape agrees

D DERIVATION

$$28 \times 28 \times 32;\ \|;\ 28 \times 28 \times 48;\ \|;\ 28 \times 28 \times 16;\ \|;\ 28 \times 28 \times 32$$

Concatenate only after every spatial shape agrees

D DERIVATION

$$28 \times 28 \times 32;\ \|;\ 28 \times 28 \times 48;\ \|;\ 28 \times 28 \times 16;\ \|;\ 28 \times 28 \times 32$$

$$\text{concatenate} \rightarrow 28 \times 28 \times (32 + 48 + 16 + 32) = 28 \times 28 \times 128$$

The candidate meets shape and compute contracts; its widest branch sees a 5×5 neighbourhood.

Misconception check: branches are not extra sequential depth

Q QUESTION

Which statement is true?

A. Every signal crosses all four branches in sequence.

B. Each branch offers an alternative transform at one stage; concatenation preserves all outputs.

Misconception check: branches are not extra sequential depth

Q QUESTION

Which statement is true?

A. Every signal crosses all four branches in sequence.

B. Each branch offers an alternative transform at one stage; concatenation preserves all outputs.

Answer: B. Branching widens the choice of scale; stacking modules supplies depth.

L6 retrieval · learn a correction, not a replacement

Suppose a useful block should mostly preserve its input. A plain block must learn $H(x) \approx x$.

L6 retrieval · learn a correction, not a replacement

Suppose a useful block should mostly preserve its input. A plain block must learn $H(x) \approx x$.

A residual block instead writes

$$H(x) = x + F(x)$$

If “do nothing” is best, the learned branch only needs $F(x) \approx 0$.

The shortcut makes identity behaviour explicit; the residual path learns only the needed change.

Commit: residual arithmetic on one scalar

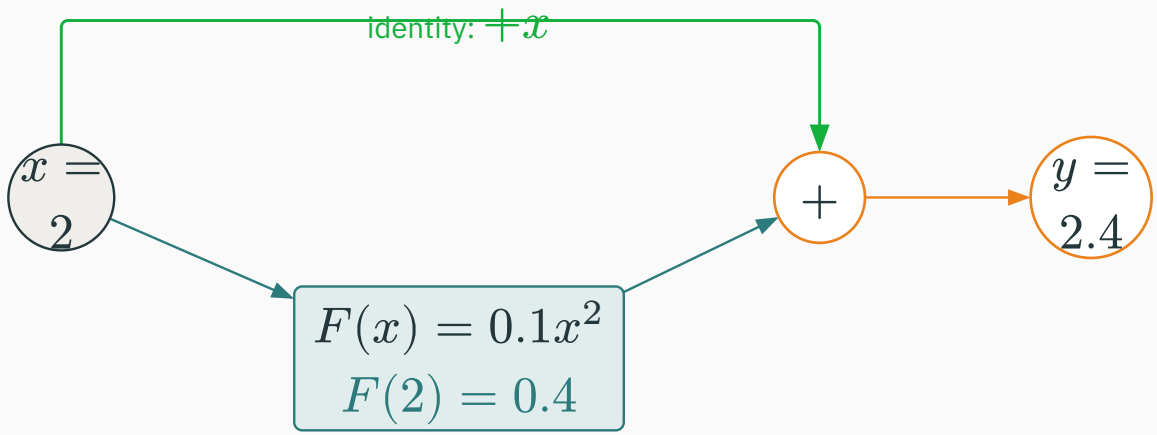
Q QUESTION

HELD CASE · ONE SCALAR BLOCK At $x = 2$, let $F(x) = 0.1x^2$. First analyze the addition $u = x + F(x)$; then apply $y = \mathrm{ReLU}(u)$.

Commit: residual arithmetic on one scalar

Q QUESTION

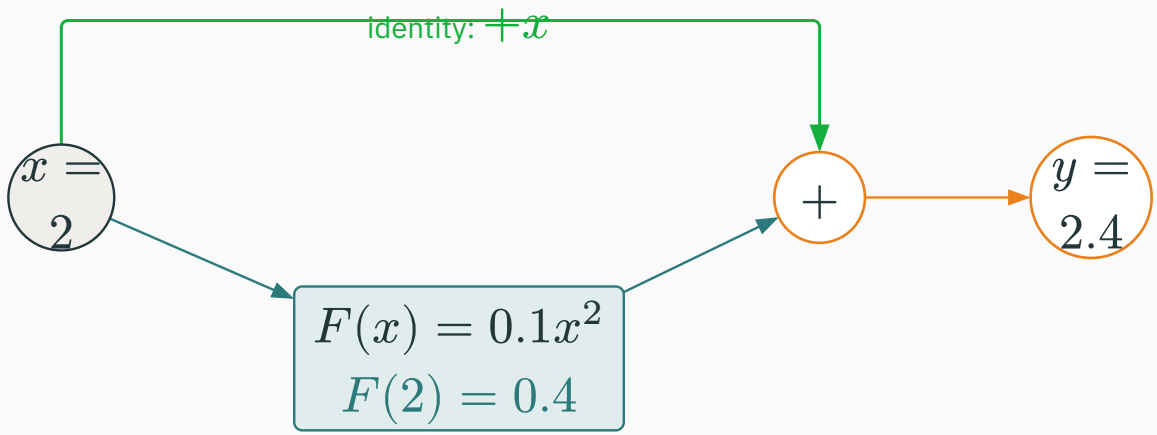
HELD CASE · ONE SCALAR BLOCK At $x = 2$, let $F(x) = 0.1x^2$. First analyze the addition $u = x + F(x)$; then apply $y = \mathrm{ReLU}(u)$.



Commit: residual arithmetic on one scalar

Q QUESTION

HELD CASE · ONE SCALAR BLOCK At $x = 2$, let $F(x) = 0.1x^2$. First analyze the addition $u = x + F(x)$; then apply $y = \mathrm{ReLU}(u)$.

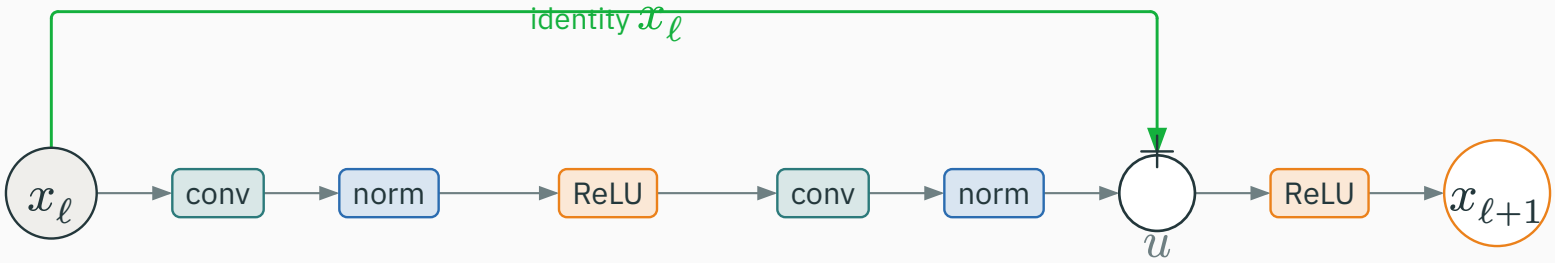


$$F(2) = 0.4, \quad u = x + F(x) = 2.4, \quad y = \mathrm{ReLU}(u) = 2.4.$$

The output keeps the original 2 and adds the learned correction 0.4.

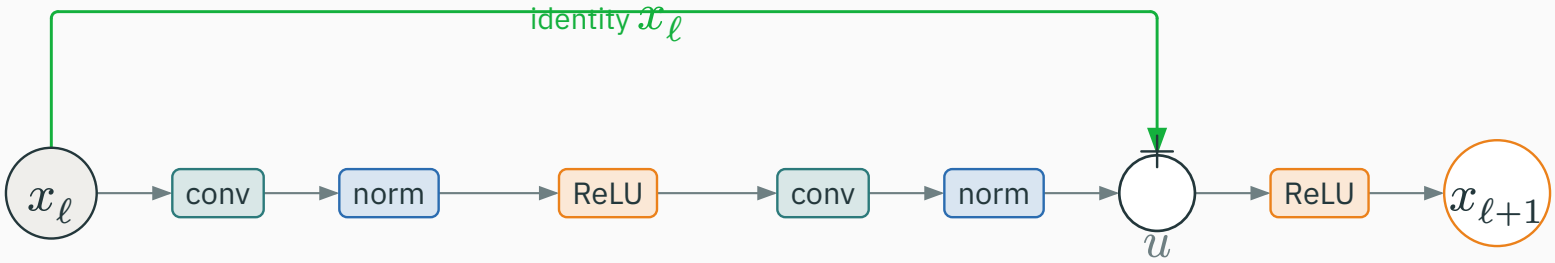
The residual block as a node diagram

V VISUAL



The residual block as a node diagram

V VISUAL



$u = x_\ell + F(x_\ell)$ at the addition; the drawn block then outputs $x_{\ell+1} = \mathrm{ReLU}(u)$.

The addition node is only legal when the two incoming tensors have identical shapes.

Why the gradient has a direct route

D DERIVATION

For the drawn post-activation block, $u = x + F(x)$ and $y = \mathrm{ReLU}(u)$:

Why the gradient has a direct route

D DERIVATION

For the drawn post-activation block, $u = x + F(x)$ and $y = \mathrm{ReLU}(u)$:

$$\frac{\partial y}{\partial x} = \mathbb{1}[u > 0] \cdot \left(1 + \frac{\partial F}{\partial x}\right).$$

Continue the scalar example $F(x) = 0.1x^2$ at $x = 2$:

$u = 2.4 > 0$, so the ReLU gate is open, $F'(2) = 0.4$, and

Why the gradient has a direct route

D DERIVATION

For the drawn post-activation block, $u = x + F(x)$ and $y = \mathrm{ReLU}(u)$:

$$\frac{\partial y}{\partial x} = \mathbb{1}[u > 0] \cdot \left(1 + \frac{\partial F}{\partial x}\right).$$

Continue the scalar example $F(x) = 0.1x^2$ at $x = 2$:

$u = 2.4 > 0$, so the ReLU gate is open, $F'(2) = 0.4$, and

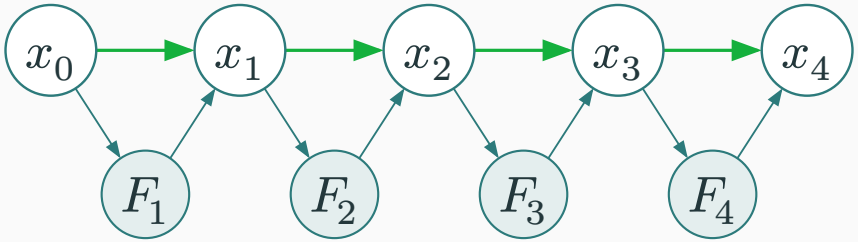
$$\frac{dy}{dx} = 1 + 0.4 = 1.4.$$

The addition contributes a direct 1 term; the post-add activation still gates the full gradient.

Residual depth: the identity path survives many blocks

V VISUAL

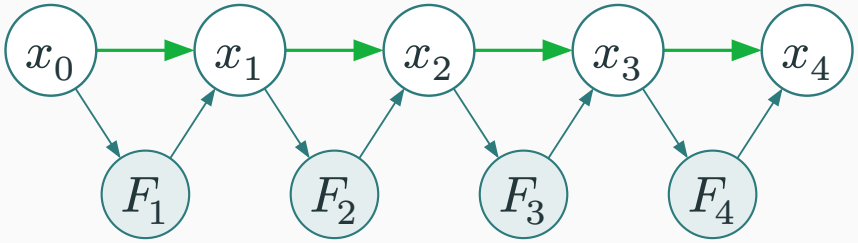
ADDITION-ONLY ABSTRACTION · nonlinear gates are suppressed to expose the shortcut algebra



Residual depth: the identity path survives many blocks

V VISUAL

ADDITION-ONLY ABSTRACTION · nonlinear gates are suppressed to expose the shortcut algebra



$$x_4 = x_0 + F_1(x_0) + F_2(x_1) + F_3(x_2) + F_4(x_3)$$

Depth becomes a sequence of refinements around a persistent representation.

The held channel change makes an identity shortcut illegal

Q QUESTION

HELD CASE · CANDIDATE 4 $x \in \mathbb{R}^{28\times28\times64}$ but the required stage output is $28 \times 28 \times 128$. Can $x + F(x)$ be added directly?

$$F(x) : 28 \times 28 \times 64 \to 28 \times 28 \times 128.$$

The held channel change makes an identity shortcut illegal

Q QUESTION

HELD CASE · CANDIDATE 4 $x \in \mathbb{R}^{28\times28\times64}$ but the required stage output is $28 \times 28 \times 128$. Can $x + F(x)$ be added directly?

$$F(x) : 28 \times 28 \times 64 \to 28 \times 28 \times 128.$$

The identity tensor cannot be added as-is. Apply a 1×1, stride-1 projection:

$$P(x) : 28 \times 28 \times 64 \to 28 \times 28 \times 128.$$

Now the addition is well-defined:

$$y = F(x) + P(x).$$

Projection shortcuts change shape; identity shortcuts preserve it.

A projected residual candidate still clears the budget

D DERIVATION

Use $F : 64 \to 8 \to 8 \to 128$ with 1×1, 3×3, 1×1; use $P : 64 \to 128$ with 1×1. Biases are omitted.

A projected residual candidate still clears the budget

D DERIVATION

Use $F : 64 \to 8 \to 8 \to 128$ with 1×1, 3×3, 1×1; use $P : 64 \to 128$ with 1×1. Biases are omitted.

$$\text{weights}_F = 64 \cdot 8 + 9 \cdot 8 \cdot 8 + 8 \cdot 128 = 2,112,$$

$$\text{weights}_P = 64 \cdot 128 = 8,192.$$

A projected residual candidate still clears the budget

D DERIVATION

Use $F : 64 \to 8 \to 8 \to 128$ with 1×1, 3×3, 1×1; use $P : 64 \to 128$ with 1×1. Biases are omitted.

$$\text{weights}_F = 64 \cdot 8 + 9 \cdot 8 \cdot 8 + 8 \cdot 128 = 2,112,$$

$$\text{weights}_P = 64 \cdot 128 = 8,192.$$

candidate	output	RF	weights	MACs
projected residual	$28 \times 28 \times 128$	3×3	$10,304$	$8,078,336$

$y = F(x) + P(x)$ is legal; both routes end at the same shape.

The shortcut is not free when shape changes—but this narrow residual branch remains within budget.

Misconception check: is the shortcut “skipping learning”?

A ANSWER

No. The output still depends on the learned residual $F(x)$.

$$x + F(x) = y$$

preserved representation $+$ learned update

The shortcut changes the **parameterization** of the target function. It makes preservation cheap while leaving the branch free to learn large or small changes.

Misconception check: is the shortcut “skipping learning”?

A ANSWER

No. The output still depends on the learned residual $F(x)$.

$$x + F(x) = y$$

preserved representation $+$ learned update

The shortcut changes the **parameterization** of the target function. It makes preservation cheap while leaving the branch free to learn large or small changes.

Residual learning is controlled refinement—not a bypass around the model.

Core move · factor spatial filtering from channel mixing

A standard $k \times k$ convolution does both jobs at once:

Core move · factor spatial filtering from channel mixing

A standard $k \times k$ convolution does both jobs at once:

$$k^2 \cdot C_{\text{in}} C_{\text{out}}$$

k^2 spatial positions $\times$ $C_{\text{in}} C_{\text{out}}$ channel pairs

Ignoring bias, its weight count is $k^2 C_{\text{in}} C_{\text{out}}$; its MAC count is

$$H_{\text{out}} W_{\text{out}} k^2 C_{\text{in}} C_{\text{out}}.$$

Depthwise separable convolution performs the two jobs in two explicit steps.

Standard convolution: establish the baseline

D DERIVATION

Let $H = W = 28$, $k = 3$, $C_{\text{in}} = 64$, and $C_{\text{out}} = 128$.

Standard convolution: establish the baseline

D DERIVATION

Let $H = W = 28$, $k = 3$, $C_{\text{in}} = 64$, and $C_{\text{out}} = 128$.

$$\text{weights} = 3 \cdot 3 \cdot 64 \cdot 128 = 73,728.$$

There are $28 \cdot 28 = 784$ output locations:

$$\text{MACs} = 784 \cdot 73,728 = 57,802,752.$$

This is the cost the factorized version must match in output shape—not in arithmetic.

Depthwise step: one spatial filter per input channel

D DERIVATION

$$28 \times 28 \times 64 \xrightarrow{\text{64 separate } 3\times3 \text{ filters}} 28 \times 28 \times 64$$

Depthwise step: one spatial filter per input channel

D DERIVATION

$$28 \times 28 \times 64 \xrightarrow{\text{64 separate } 3\times3 \text{ filters}} 28 \times 28 \times 64$$

$$\text{weights} = 3 \cdot 3 \cdot 64 = 576,$$

$$\text{MACs} = 784 \cdot 576 = 451,584.$$

No channel is mixed with another yet.

Depthwise convolution answers only: “what spatial pattern occurs within this channel?”

Pointwise step: mix the filtered channels

D DERIVATION

$$28 \times 28 \times 64 \xrightarrow{1\times1,\ 128\ \text{outputs}} 28 \times 28 \times 128$$

Pointwise step: mix the filtered channels

D DERIVATION

$$28 \times 28 \times 64 \xrightarrow{1\times1,\ 128\ \text{outputs}} 28 \times 28 \times 128$$

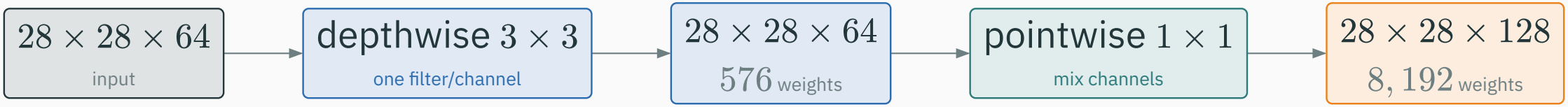
$$\text{weights} = 64 \cdot 128 = 8,192,$$

$$\text{MACs} = 784 \cdot 8,192 = 6,422,528.$$

The pointwise layer answers: “which channel combinations should form the 128 outputs?”

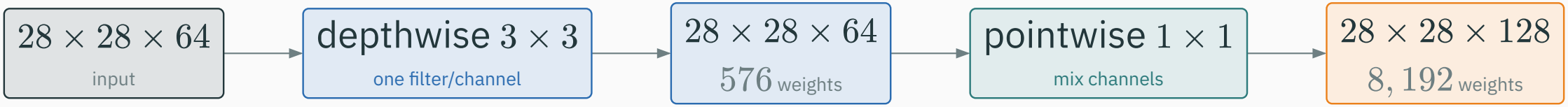
Put the factorized block back together

V VISUAL



Put the factorized block back together

V VISUAL



$$\text{separable weights} = 576 + 8,192 = 8,768,$$

$$\text{separable MACs} = 451,584 + 6,422,528 = 6,874,112.$$

$\frac{57,802,752}{6,874,112} \approx 8.41 \times$ **fewer MACs than the standard convolution.**

Grouped convolution connects the extremes

V VISUAL

one group $\rightarrow$ a standard convolution mixes **all** input channels

g groups $\rightarrow$ each output sees only $\frac{C_{\text{in}}}{g}$ input channels

$g = C_{\text{in}} \rightarrow$ depthwise connectivity: each output sees one input channel

Grouped convolution connects the extremes

V VISUAL

one group $\rightarrow$ a standard convolution mixes **all** input channels

g groups $\rightarrow$ each output sees only $\frac{C_{\text{in}}}{g}$ input channels

$g = C_{\text{in}} \rightarrow$ depthwise connectivity: each output sees one input channel

With depth multiplier 1, $C_{\text{out}} = C_{\text{in}}$ and there is one spatial filter per channel; a later 1×1 restores full mixing.

Three scaling knobs have different costs

D DERIVATION

For a representative convolution, MACs scale approximately as

$$\text{MACs} \propto r^2 d w^2,$$

where r scales resolution, d depth, and w channel width.

Double	What changes	rough MAC multiplier
resolution r	$H, W \to 2H, 2W$	$4\times$
depth d	twice as many blocks	$2\times$
width w	$C_{\text{in}}, C_{\text{out}} \to 2\times$	$4\times$

Three scaling knobs have different costs

D DERIVATION

For a representative convolution, MACs scale approximately as

$$\text{MACs} \propto r^2 d w^2,$$

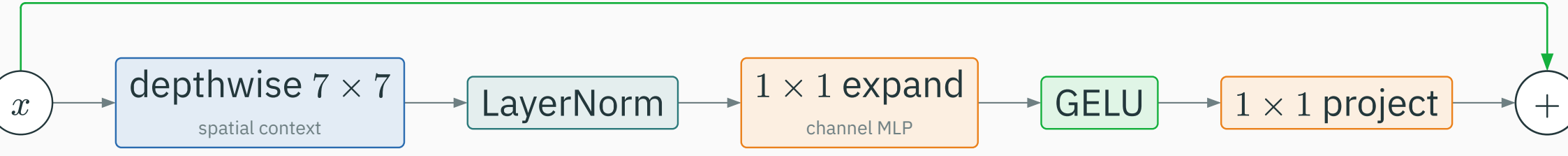
where r scales resolution, d depth, and w channel width.

Double	What changes	rough MAC multiplier
resolution r	$H, W \to 2H, 2W$	$4\times$
depth d	twice as many blocks	$2\times$
width w	$C_{\text{in}}, C_{\text{out}} \to 2\times$	$4\times$

EfficientNet's compound scaling grows all three gradually instead of overspending on one axis.

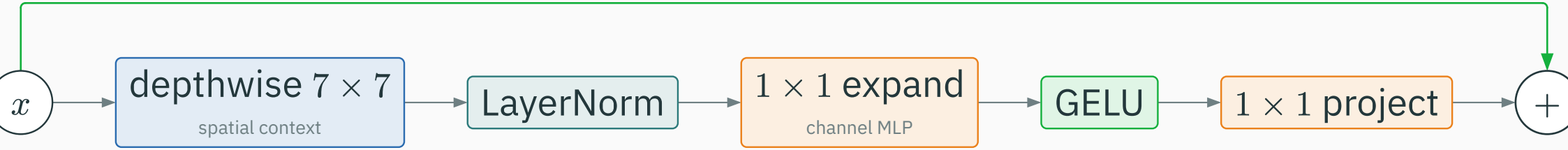
ConvNeXt recombines familiar moves

V VISUAL



ConvNeXt recombines familiar moves

V VISUAL



Modern CNNs recombine the same ideas: large-context depthwise filtering, cheap channel mixing, normalization, and a residual path.

Diagnose: fewer MACs need not mean lower latency

Q QUESTION

Does $8.41\times$ fewer MACs guarantee $8.41\times$ lower wall-clock latency?

Diagnose: fewer MACs need not mean lower latency

Q QUESTION

Does $8.41\times$ fewer MACs guarantee $8.41\times$ lower wall-clock latency?

Not necessarily.

Wall-clock speed also depends on memory traffic, kernel launch overhead, tensor layout, batch size, and hardware support.

$$\text{algorithmic cost} \neq \text{measured latency}$$

Use MACs and parameters for reasoning; benchmark latency on the deployment device before making the final choice.

The stage ledger now contains architectural choices

V VISUAL

candidate	output	RF	weights	MACs
standard 3×3	$28^2 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28^2 \times 128$	5×5	$110,592$	86.704M
bottleneck 16	$28^2 \times 128$	3×3	$5,376$	4.215M
Inception	$28^2 \times 128$	up to 5×5	$9,920$	7.777M
projected residual	$28^2 \times 128$	3×3	$10,304$	8.078M
depthwise + pointwise	$28^2 \times 128$	3×3	$8,768$	6.874M

The stage ledger now contains architectural choices

V VISUAL

candidate	output	RF	weights	MACs
standard 3×3	$28^2 \times 128$	3×3	$73,728$	57.803M
two 3×3s	$28^2 \times 128$	5×5	$110,592$	86.704M
bottleneck 16	$28^2 \times 128$	3×3	$5,376$	4.215M
Inception	$28^2 \times 128$	up to 5×5	$9,920$	7.777M
projected residual	$28^2 \times 128$	3×3	$10,304$	8.078M
depthwise + pointwise	$28^2 \times 128$	3×3	$8,768$	6.874M

The best block is conditional: satisfy the task contract, then measure the resource that matters on the target hardware.

A stage becomes one repeated part of a backbone

V VISUAL

A stage becomes one repeated part of a backbone

V VISUAL

STAGES

spatial transforms build a hierarchy

POOL

$7 \times 7 \times 512 \to h \in \mathbb{R}^{512}$

HEAD

$h \to z \in \mathbb{R}^{K}$

A stage becomes one repeated part of a backbone

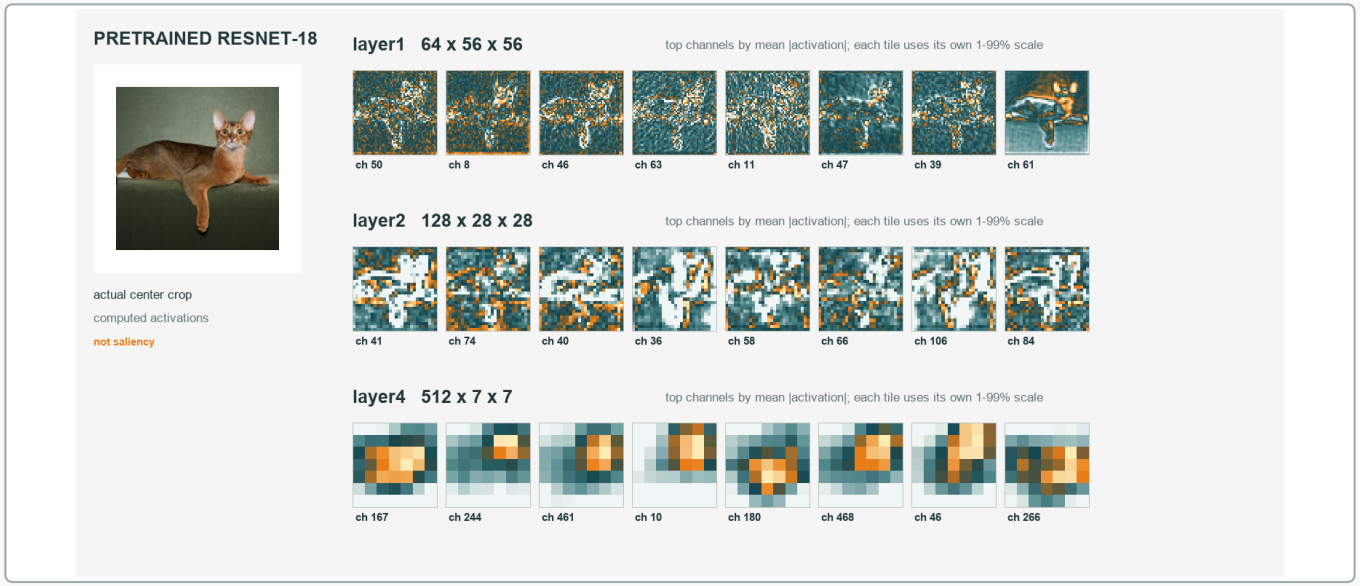
V VISUAL

STAGES	POOL	HEAD
spatial transforms build a hierarchy	$7 \times 7 \times 512 \to h \in \mathbb{R}^{512}$	$h \to z \in \mathbb{R}^{K}$

The block ledger governs the backbone; transfer asks how much of that learned hierarchy to reuse.

One real crop produces a hierarchy of pretrained model activations

V VISUAL



OBSERVED + COMPUTED actual center crop $\longrightarrow$ executed ResNet-18 `IMAGENET1K_V1` activations; top channels by mean $|a|$; each tile uses its own 1–99% scale; **not saliency**

Spatial grids contract $56 \to 28 \to 7$ while channels expand $64 \to 128 \to 512$: transfer reuses a computed representation.

Commit before the curves: how much should this backbone adapt?

Q QUESTION

GT + OBSERVED Oxford-IIIT Pet labels · six cat breeds · fixed SHA-ranked subset: 72 train / 36 validation / 36 sealed test

A pretrained backbone emits $h \in \mathbb{R}^{512}$. The exact split, head initialization, optimizer family, minibatch order, and 15-epoch budget are held fixed.

A · PROBE	B · LATE BLOCKS	C · ALL
freeze the backbone; train only $512 \to 6$	train the head and final backbone stage	fine-tune every parameter immediately

Commit before the curves: how much should this backbone adapt?

Q QUESTION

GT + OBSERVED Oxford-IIIT Pet labels · six cat breeds · fixed SHA-ranked subset: 72 train / 36 validation / 36 sealed test

A pretrained backbone emits $h \in \mathbb{R}^{512}$. The exact split, head initialization, optimizer family, minibatch order, and 15-epoch budget are held fixed.

A · PROBE	B · LATE BLOCKS	C · ALL
freeze the backbone; train only $512 \to 6$	train the head and final backbone stage	fine-tune every parameter immediately

> Commit to A, B, or C now. Validation selects the regime and epoch; only that frozen checkpoint may open the 36-image test subset once.

Count the new six-class head exactly

D DERIVATION

$$z = Wh + b, \quad W \in \mathbb{R}^{6\times 512}, \quad b \in \mathbb{R}^{6}.$$

Count the new six-class head exactly

D DERIVATION

$$z = Wh + b, \quad W \in \mathbb{R}^{6\times 512}, \quad b \in \mathbb{R}^{6}.$$

$$\text{trainable head parameters} = 6 \cdot 512 + 6 = 3,078.$$

Count the new six-class head exactly

D DERIVATION

$$z = Wh + b, \quad W \in \mathbb{R}^{6\times 512}, \quad b \in \mathbb{R}^{6}.$$

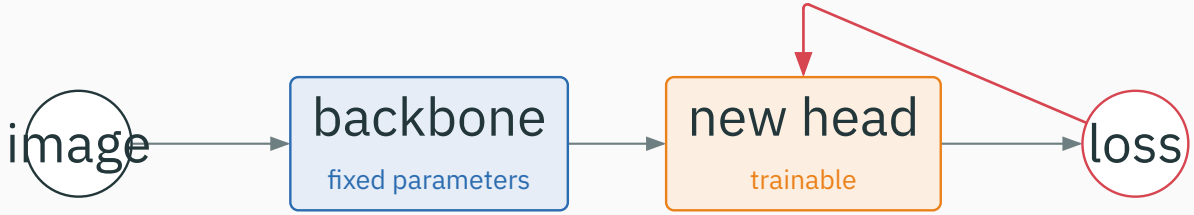
$$\text{trainable head parameters} = 6 \cdot 512 + 6 = 3,078.$$

If the backbone has 11 million parameters, a linear probe trains $\frac{3,078}{11,000,000+3,078} \approx 0.028\%$ of the model.

The first transfer experiment is intentionally tiny and inspectable.

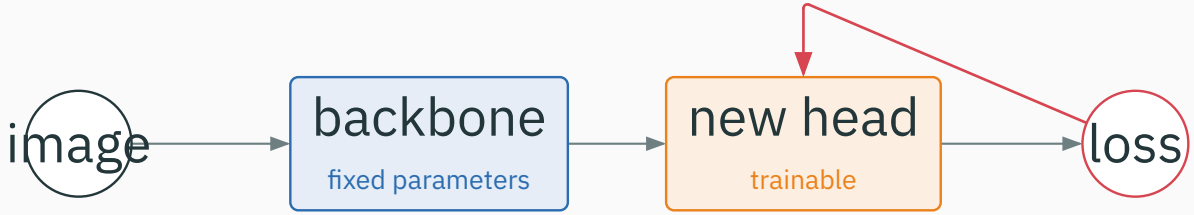
A linear probe computes through the backbone but updates only the head

V VISUAL



A linear probe computes through the backbone but updates only the head

V VISUAL



A probe measures linear separability in pretrained feature space; it does not make backbone computation disappear.

Freezing parameters does not freeze model state

D DERIVATION

PARAMETERS

`requires_grad=False`; exclude them from the optimizer.

BUFFERS

BatchNorm running means/variances can still change in `train()` mode.

STOCHASTIC LAYERS

Dropout also changes behaviour between `train()` and `eval()`.

Freezing parameters does not freeze model state

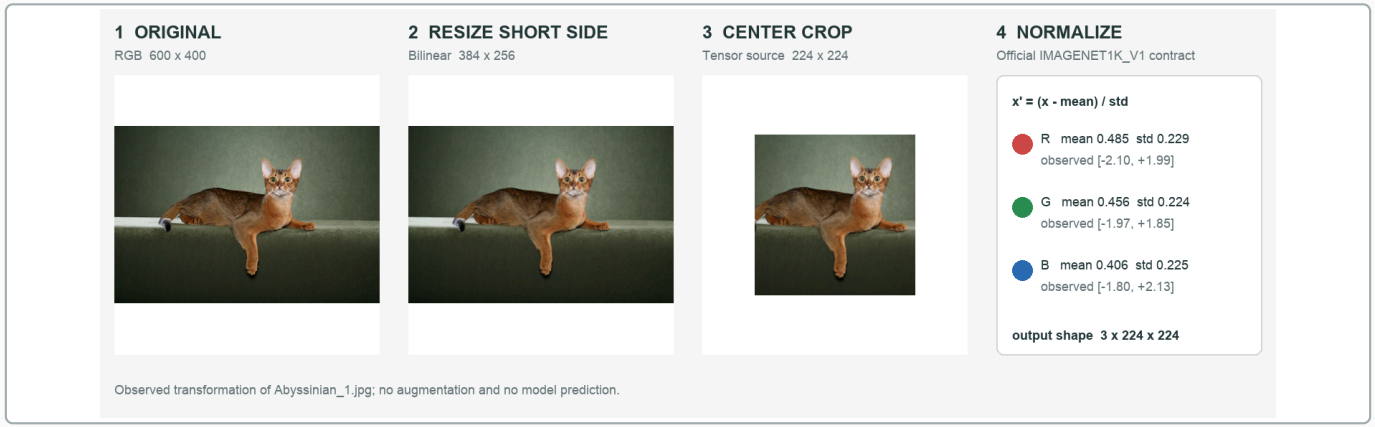
D DERIVATION

PARAMETERS	BUFFERS	STOCHASTIC LAYERS
`requires_grad=False`; exclude them from the optimizer.	BatchNorm running means/variances can still change in `train()` mode.	Dropout also changes behaviour between `train()` and `eval()`.

For a truly fixed probe, keep the frozen backbone in `eval()` while the new head trains. Updating BatchNorm statistics deliberately is a different intervention—name and validate it.

The weight recipe includes the input transformation

V VISUAL



OBSERVED Abyssinian_1.jpg under the official `IMAGENET1K_V1` resize ⟶ center-crop ⟶ normalize contract; no augmentation and no prediction

BOARD

$224 \times 224 \times 3$ keeps the spatial story readable.

EXECUTED TENSOR

`float32` $1 \times 3 \times 224 \times 224$; $x'_c = \frac{x_c - \mu_c}{\sigma_c}$.

Wrong channel order, range, resize/crop, or normalization gives the pretrained model a different problem.

L7 protocol: validation chooses how much to unfreeze

V VISUAL

TRAIN	VALIDATION	SEALED TEST
fit head/backbone parameters for one declared regime	choose probe vs late/all, learning rates, stopping epoch, and checkpoint	run once after the full decision rule and checkpoint are frozen

L7 protocol: validation chooses how much to unfreeze

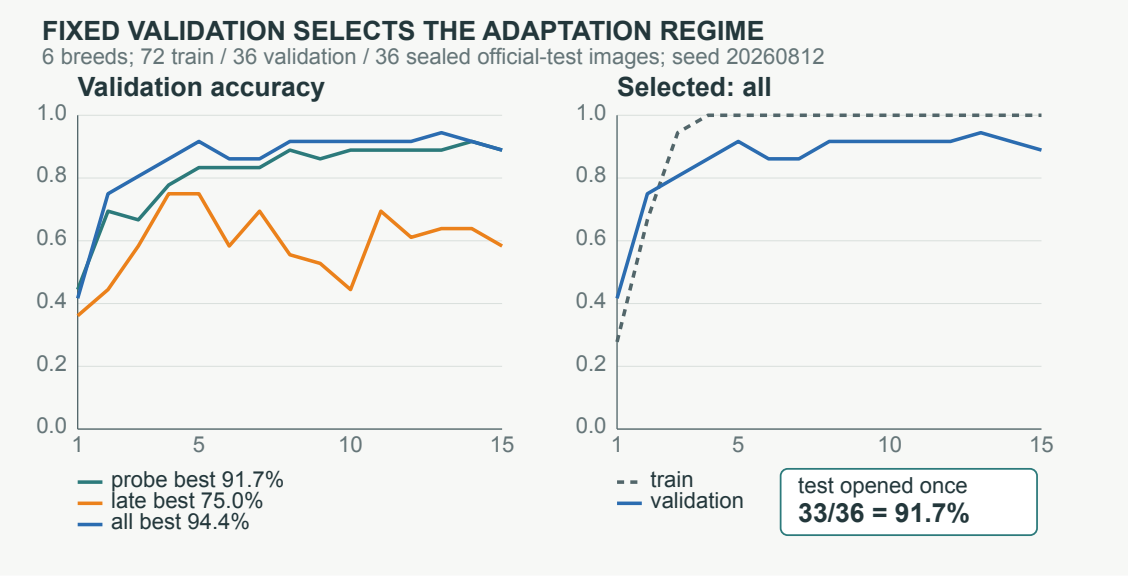
V VISUAL

TRAIN	VALIDATION	SEALED TEST
fit head/backbone parameters for one declared regime	choose probe vs late/all, learning rates, stopping epoch, and checkpoint	run once after the full decision rule and checkpoint are frozen

Repeatedly checking test performance while choosing unfreeze depth turns the test set into validation data.

Validation selected full fine-tuning on this fixed teaching subset

V VISUAL



MEASURED one manifest-bound run · seed 20260812 · 15 epochs · validation picks regime + epoch · test opened once after selection

Best validation: probe $\frac{33}{36}$ @ 14 · late $\frac{27}{36}$ @ 4 · all $\frac{34}{36}$ @ 13 $\longrightarrow$ sealed test $\frac{33}{36}$.

Read the curves before turning them into a prescription

V VISUAL

WHAT WE OBSERVED

The fixed probe reached 91.7% validation: pretrained 512-D features were strongly separable. Full adaptation gained one validation image and won the declared rule.

WHAT FAILED

Late unfreezing reached 100% training but only 75.0% best validation. More trainable parameters did not guarantee better generalization.

WHAT WE MAY CONCLUDE

Validation selected full adaptation for these 144 manifest-bound images, fixed optimizer settings, and one seed.

WHAT WE MAY NOT CONCLUDE

This is not an Oxford Pets benchmark; it cannot establish that full fine-tuning generally beats probing or late unfreezing.

Use a progressive ladder as an experiment design; let held-out evidence—not parameter count—choose the rung.

Parameter groups make the fine-tuning hypothesis explicit

D DERIVATION

$$\theta_h \xrightarrow{\eta_h} \theta_h - \eta_h \nabla_{\theta_h} \mathcal{L}, \quad \theta_b \xrightarrow{\eta_b} \theta_b - \eta_b \nabla_{\theta_b} \mathcal{L}$$

Parameter groups make the fine-tuning hypothesis explicit

D DERIVATION

$$\theta_h \xrightarrow{\eta_h} \theta_h - \eta_h \nabla_{\theta_h} \mathcal{L}, \quad \theta_b \xrightarrow{\eta_b} \theta_b - \eta_b \nabla_{\theta_b} \mathcal{L}$$

$$0 < \eta_b < \eta_h$$

NEW HEAD

larger step: it starts random and must learn the task

REUSED BACKBONE

smaller step: preserve useful features while adapting

Parameter groups make the fine-tuning hypothesis explicit

D DERIVATION

$$\theta_h \xrightarrow{\eta_h} \theta_h - \eta_h \nabla_{\theta_h} \mathcal{L}, \quad \theta_b \xrightarrow{\eta_b} \theta_b - \eta_b \nabla_{\theta_b} \mathcal{L}$$

$$0 < \eta_b < \eta_h$$

NEW HEAD

larger step: it starts random and must learn the task

REUSED BACKBONE

smaller step: preserve useful features while adapting

The ratio is a hyperparameter selected on validation—not a universal constant.

Diagnose transfer failures as symptom → suspect → test

V VISUAL

symptom	suspect	smallest discriminating test
loss stays near random	input contract mismatch	log batch shape/range; compare expected transform
frozen features drift	BN buffers or dropout active	diff parameters **and buffers** across one step
validation drops after unfreeze	backbone rate too large	reduce η_b; log layer-wise update norms
MACs fall; latency does not	memory/kernel overhead	benchmark the target device and batch size

Diagnose transfer failures as symptom → suspect → test

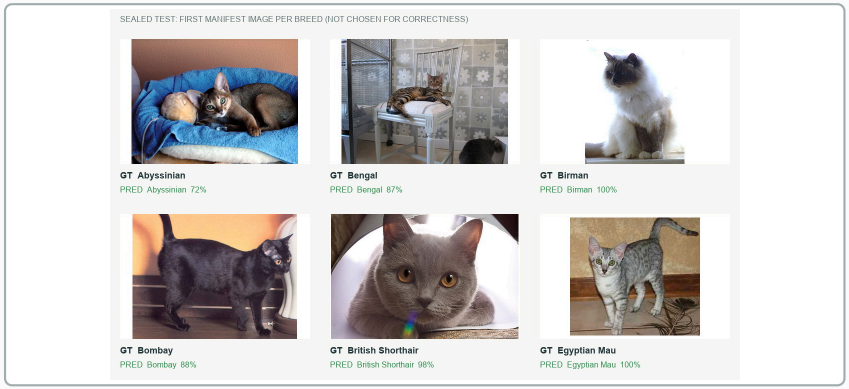
V VISUAL

symptom	suspect	smallest discriminating test
loss stays near random	input contract mismatch	log batch shape/range; compare expected transform
frozen features drift	BN buffers or dropout active	diff parameters **and buffers** across one step
validation drops after unfreeze	backbone rate too large	reduce η_b; log layer-wise update norms
MACs fall; latency does not	memory/kernel overhead	benchmark the target device and batch size

Change the experiment only after the test separates competing causes.

The selected checkpoint opened the sealed images once

I INTERACTIVE



GT + COMPUTED MODEL OUTPUT one test example per true breed; green line is the selected model's predicted class and probability

INTERACTIVE

↗ colab.research.google.com/github/nipunbatra/dl-teaching/blob/master/notebooks/L08B/01_depthwise_separable_and_transfer_head.ipynb

Reproduce the exact arithmetic, freezing checks, official preprocessing, cached 512-D features, and manifest-bound linear probe in Colab.

SEALED RESULT

full fine-tuning · epoch 13 · $\frac{33}{36} = 91.7\%$

AUDIT TRAIL

`selection-manifest.csv` locks every filename, split, and JPEG SHA-256; `results.json` records the single test pass.

Six shown successes are a contact sheet, not the score calculation; $\frac{33}{36}$ uses all sealed examples and includes three errors.

The transfer decision is small, but the protocol is complete

V VISUAL

START	SELECT	REPORT
$512 \to 6$ head = $3,078$ trainable parameters	validation chooses frozen/late/all and the checkpoint	sealed test once; include trainable count and measured latency

The transfer decision is small, but the protocol is complete

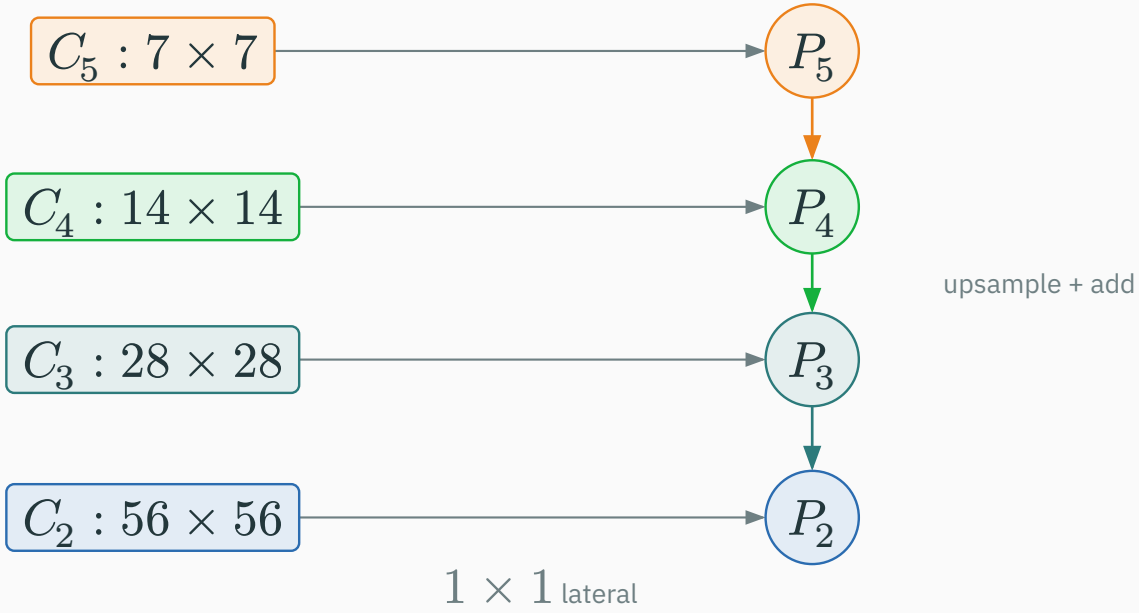
V VISUAL

START	SELECT	REPORT
$512 \to 6$ head = $3,078$ trainable parameters	validation chooses frozen/late/all and the checkpoint	sealed test once; include trainable count and measured latency

“Transfer learning worked” means a controlled candidate won on validation and survived the one-time test—not merely that training ran.

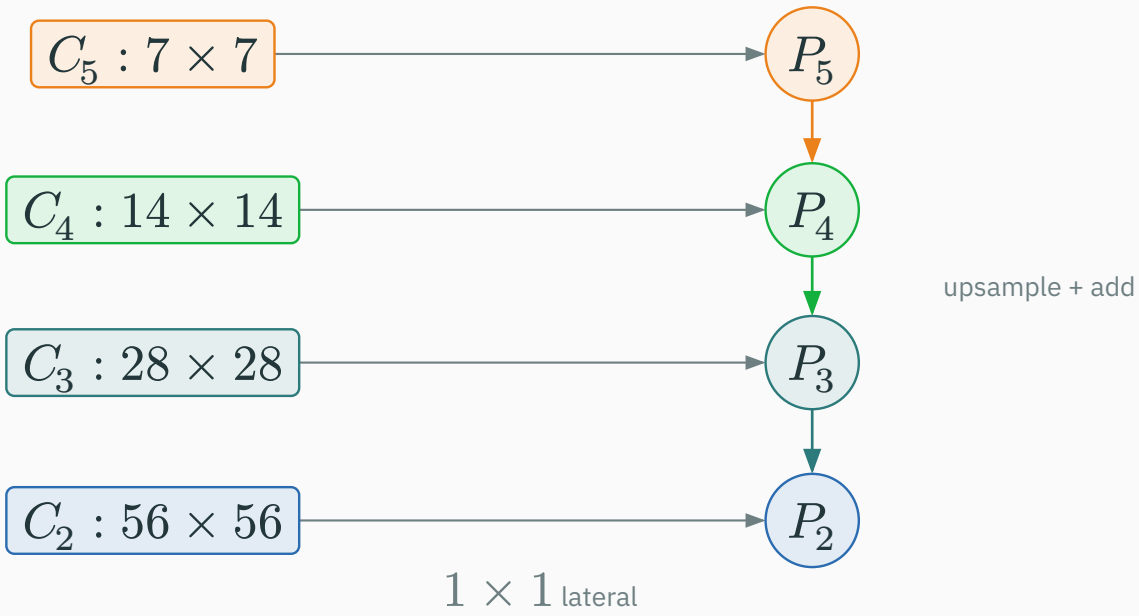
Keep spatial features when the next task asks where

V VISUAL



Keep spatial features when the next task asks where

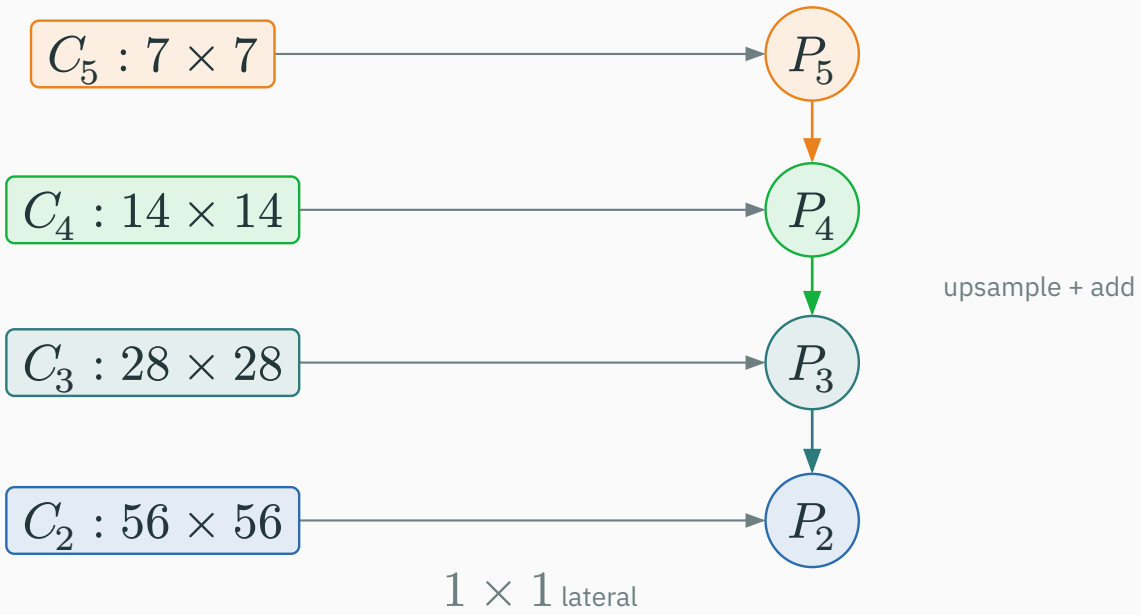
V VISUAL



A feature pyramid laterally projects backbone stages, upsamples coarse features, and adds only tensors with matching height, width, and channels.

Keep spatial features when the next task asks where

V VISUAL



A feature pyramid laterally projects backbone stages, upsamples coarse features, and adds only tensors with matching height, width, and channels.

Today the backbone is the reusable object. L10 will derive the localization heads and pyramid operations needed for detection.

Architecture names collapse into six reusable moves

V VISUAL

need more context $\rightarrow$ stack small kernels **(VGG)**

need cheaper channel work $\rightarrow$ squeeze with 1×1 **(bottleneck)**

need several scales $\rightarrow$ branch then concatenate **(Inception)**

need trainable depth $\rightarrow$ add identity or projected shortcuts **(ResNet)**

need mobile efficiency $\rightarrow$ factor spatial and channel work **(MobileNet)**

need a new task $\rightarrow$ reuse backbone, replace head, validate adaptation

Architecture names collapse into six reusable moves

V VISUAL

need more context $\rightarrow$ stack small kernels **(VGG)**

need cheaper channel work $\rightarrow$ squeeze with 1×1 **(bottleneck)**

need several scales $\rightarrow$ branch then concatenate **(Inception)**

need trainable depth $\rightarrow$ add identity or projected shortcuts **(ResNet)**

need mobile efficiency $\rightarrow$ factor spatial and channel work **(MobileNet)**

need a new task $\rightarrow$ reuse backbone, replace head, validate adaptation

Names provide provenance; shape, compute, route, and validation provide the reasoning.

Revisit the opening commitment before revealing it

Q QUESTION

HELD CASE · SAME DEPLOYMENT CONTRACT $28 \times 28 \times 64 \to 28 \times 28 \times 128$, at least 3×3 context, fewer than 10M convolution MACs.

A · STANDARD	B · FACTORIZED	C · EITHER
$73,728$ weights; 57.803M MACs	$8,768$ weights; 6.874M MACs	sharing alone guarantees budget

Revisit the opening commitment before revealing it

Q QUESTION

HELD CASE · SAME DEPLOYMENT CONTRACT $28 \times 28 \times 64 \to 28 \times 28 \times 128$, at least 3×3 context, fewer than 10M convolution MACs.

A · STANDARD	B · FACTORIZED	C · EITHER
$73,728$ weights; 57.803M MACs	$8,768$ weights; 6.874M MACs	sharing alone guarantees budget

Defend your original choice using the ledger—not the architecture name.

Answer: B meets every stated contract

A ANSWER

$$\frac{57{,}802{,}752}{6{,}874{,}112} \approx 8.41 \times \textbf{fewer MACs}$$

Answer: B meets every stated contract

A ANSWER

$$\frac{57{,}802{,}752}{6{,}874{,}112} \approx 8.41 \times \textbf{fewer MACs}$$

SHAPE	CONTEXT	BUDGET
$28 \times 28 \times 128$ on both paths	depthwise 3×3 supplies the required RF	$6.874\text{M} < 10\text{M}$

Answer: B meets every stated contract

A ANSWER

$$\frac{57{,}802{,}752}{6{,}874{,}112} \approx 8.41 \times \textbf{fewer MACs}$$

SHAPE	CONTEXT	BUDGET
$28 \times 28 \times 128$ on both paths	depthwise 3×3 supplies the required RF	$6.874\text{M} < 10\text{M}$

Factorization won this contract. A bottleneck, Inception, or projected residual candidate may win when context, branching, or optimization is the dominant need.

Final design checklist

Q QUESTION

Given an image task, ask in order:

1. What output shape and receptive field must the evidence satisfy?
2. Which operation mixes space; which mixes channels?
3. Are additions and concatenations shape-legal?
4. Which weights and output positions dominate MACs?
5. Does measured device latency agree with the ledger?
6. Can a pretrained backbone be probed before broader adaptation?
7. Which validation rule chooses the regime before the test is opened?

Final design checklist

Q QUESTION

Given an image task, ask in order:

1. What output shape and receptive field must the evidence satisfy?
2. Which operation mixes space; which mixes channels?
3. Are additions and concatenations shape-legal?
4. Which weights and output positions dominate MACs?
5. Does measured device latency agree with the ledger?
6. Can a pretrained backbone be probed before broader adaptation?
7. Which validation rule chooses the regime before the test is opened?

A strong CNN pipeline is a sequence of declared contracts and measured decisions—not a pile of named blocks.

L10 handoff: the backbone answers what; detection must retain where

V VISUAL

CLASSIFICATION

$$7 \times 7 \times 512 \xrightarrow{\text{global average}} 512 \to 6$$

one image-level answer

$\rightarrow$

DETECTION

$C_2, C_3, C_4, C_5 \rightarrow$ spatial heads

classes **and** boxes at many locations

L10 handoff: the backbone answers what; detection must retain where

V VISUAL

CLASSIFICATION

$$7 \times 7 \times 512 \xrightarrow{\text{global average}} 512 \to 6$$

one image-level answer

$\rightarrow$

DETECTION

$C_2, C_3, C_4, C_5 \rightarrow$ spatial heads

classes **and** boxes at many locations

The next lecture keeps the reusable backbone but changes the head, targets, geometry, and evaluation.

Primary provenance and executable evidence

V VISUAL

BLOCKS

Simonyan & Zisserman · VGG
Szegedy et al. · Inception
He et al. · ResNet
Howard et al. · MobileNet

SCALING + DATA

Tan & Le · EfficientNet
Liu et al. · ConvNeXt
Lin et al. · FPN
Oxford-IIIT Pet · dataset + CC BY-SA 4.0

Primary provenance and executable evidence

V VISUAL

BLOCKS

Simonyan & Zisserman · VGG
Szegedy et al. · Inception
He et al. · ResNet
Howard et al. · MobileNet

SCALING + DATA

Tan & Le · EfficientNet
Liu et al. · ConvNeXt
Lin et al. · FPN
Oxford-IIIT Pet · dataset + CC BY-SA 4.0

MODEL ARTIFACT

torchvision ResNet-18 `IMAGENET1K_V1`
weights SHA-256 `f37072fd…560e07ec`

EXECUTABLE CASE

Exact L08B Colab notebook
manifest + builder + histories + predictions + `results.json`

Primary provenance and executable evidence

V VISUAL

BLOCKS

Simonyan & Zisserman · VGG
Szegedy et al. · Inception
He et al. · ResNet
Howard et al. · MobileNet

SCALING + DATA

Tan & Le · EfficientNet
Liu et al. · ConvNeXt
Lin et al. · FPN
Oxford-IIIT Pet · dataset + CC BY-SA 4.0

MODEL ARTIFACT

torchvision ResNet-18 `IMAGENET1K_V1`
weights SHA-256 `f37072fd…560e07ec`

EXECUTABLE CASE

Exact L08B Colab notebook
manifest + builder + histories + predictions + `results.json`

Photographs remain CC BY-SA 4.0; every numeric claim on the transfer pages is bound to the fixed manifest and recorded run.