ARTIFICIAL INTELLIGENCE AND NEURAL NETWORKS

Lecture 5b – Convolutional Neural Networks for Vision

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn
Advanced Computing and Digital Technology Research Center
University of Chinese Academy of Sciences

Spring 2026

Today's Question

What we are trying to answer

Why do images need a different kind of neural network, and what exactly does a CNN change compared with an ordinary dense network?

Why this lecture matters

If we only say that CNNs are good for images, the topic feels like vocabulary. If we understand what problem they solve, the architecture starts to feel natural.

What changes from NN4

NN4 explained how neural networks compute and learn. NN5 keeps the same training logic, but changes the architecture so it respects visual structure.

From NN4 to NN5

Last time

We studied neurons, activations, forward pass, backpropagation, and the choices that make deep learning trainable.

Today

We ask what happens when the input is not a short feature table, but a picture with local patterns spread across space.

One sentence

NN4 was about how neural networks learn. NN5 is about how network design changes when the data is visual.

What Stayed the Same

The learning logic did not change

We still make a forward pass, compute a loss, send gradients backward, and update weights.

What actually changed

The new idea is architectural: instead of connecting everything to everything immediately, we use local filters and shared weights.

CNN is not a new learning rule. It is a new way to organize the network.

Why Images Are Not Just Tables

Three facts about images

Nearby pixels matter together, the same pattern can appear in many positions, and local structure such as edges and corners is often more useful than raw isolated numbers.

Why this matters

If we flatten an image too early, we throw away the spatial relationships that made the image meaningful in the first place.

CNNs begin with a simple promise: respect the geometry of the data.

Why Dense Layers Struggle With Pictures

The parameter problem

A $224 \times 224 \times 3$ image already has more than 150,000 input numbers. If every hidden unit connects to all of them immediately, the parameter count explodes.

A classroom analogy

Recognizing a friend in a lecture hall should not require one detector for the front row, another for the middle, and another for the back. Dense layers behave too much like that.

What gets lost

Dense layers do not naturally encode locality or the idea that the same visual clue can appear in many positions.

Why CNNs help

Convolutions say: look locally, reuse what works, and build larger patterns gradually.

CNNs Start With Local Clues

The core shift

Instead of asking one huge layer to understand the whole image at once, CNNs ask small filters to look for local patterns repeatedly.

A human analogy

We also tend to notice edges, corners, textures, and parts before recognizing the whole object.

Why Translation Matters

An everyday fact

A cat in the upper-left corner is still a cat. A handwritten digit is still the same digit even if it shifts slightly on the page.

What CNNs assume

Useful visual patterns should still be detectable when they move across the image.

That is why reusing the same detector across many positions is such a powerful idea.

Convolution as Pattern Matching

The filter view

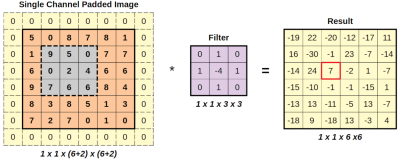
A convolution filter is a small grid of weights that reacts strongly when a particular local pattern is present.

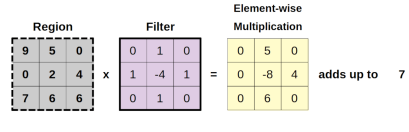
One useful formula

$$y_{i,j} = \sum_{u,v} K_{u,v}\, x_{i+u,j+v}$$

You do not need to memorize the indices. The picture is enough: slide a small detector, multiply, add, and record the response.

One Convolution Example





What the picture shows

A small patch meets the filter, the products are added, and one output value appears.

Why this one step matters

The whole CNN story is just this local computation repeated across positions and layers.

Receptive Fields and Weight Sharing

Receptive field

Each unit looks only at a local neighborhood rather than the whole image.

Weight sharing

The same filter is reused across the image, so the model learns one detector that can work in many places.

Why this combination matters

Local focus controls complexity. Shared weights capture repeating patterns.

Stride Changes What We Notice

Small stride

Move carefully, preserve more spatial detail, and compute more outputs.

Large stride

Move faster, shrink the output more aggressively, and keep a coarser view.

Stride is a design tradeoff between detail and efficiency.

Padding Protects the Borders

Replication Padding

5	5	0	8	7	8	1	1
5	**5**	**0**	**8**	**7**	**8**	**1**	1
1	**1**	**9**	**5**	**0**	**7**	**7**	7
6	**6**	**0**	**2**	**4**	**6**	**6**	6
9	**9**	**7**	**6**	**6**	**8**	**4**	4
8	**8**	**3**	**8**	**5**	**1**	**3**	3
7	**7**	**2**	**7**	**0**	**1**	**0**	0
7	7	2	7	0	1	0	0

Reflection Padding

9	1	9	5	0	7	7	7
0	**5**	**0**	**8**	**7**	**8**	**1**	8
9	**1**	**9**	**5**	**0**	**7**	**7**	7
0	**6**	**0**	**2**	**4**	**6**	**6**	6
7	**9**	**7**	**6**	**6**	**8**	**4**	8
3	**8**	**3**	**8**	**5**	**1**	**3**	1
2	**7**	**2**	**7**	**0**	**1**	**0**	1
3	8	3	8	5	1	3	1

Circular Padding

0	7	2	7	0	1	0	7
1	**5**	**0**	**8**	**7**	**8**	**1**	5
7	**1**	**9**	**5**	**0**	**7**	**7**	1
6	**6**	**0**	**2**	**4**	**6**	**6**	6
4	**9**	**7**	**6**	**6**	**8**	**4**	9
3	**8**	**3**	**8**	**5**	**1**	**3**	8
0	**7**	**2**	**7**	**0**	**1**	**0**	7
1	5	0	8	7	8	1	5

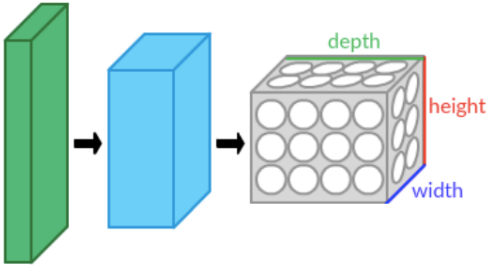
Why padding exists

Without it, border information shrinks away quickly and edge regions become underrepresented.

The practical lesson

Padding is not a cosmetic detail. It changes what the network is allowed to keep seeing across layers.

Feature Maps Are Clue Maps



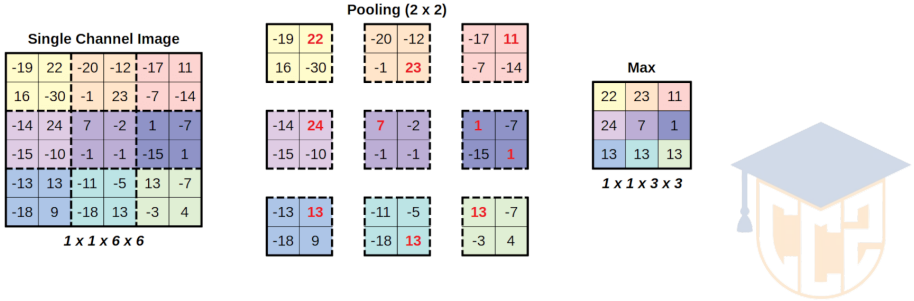
How to read a feature map

Bright or strong regions mean the filter responded strongly there.

Why this is helpful

Instead of one hidden vector with no visual meaning, we get a map of where certain clues seem to appear.

Why Pooling Was Introduced



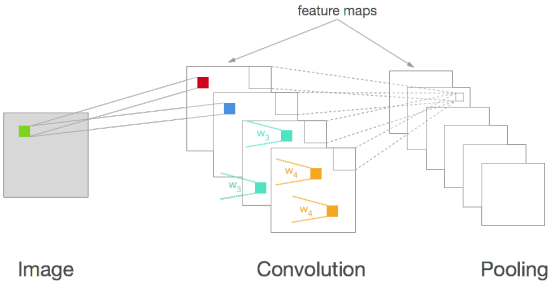
Pooling idea

Summarize a small neighborhood so the representation becomes smaller and somewhat less sensitive to tiny shifts.

Why it helped historically

It reduced computation and gave the model a little more robustness to local movement.

Convolution and Pooling as a Pipeline



The flow

Convolution detects clues. Activation reshapes the signal. Pooling compresses and stabilizes what remains.

Why this became a standard recipe

It gives the network a disciplined way to move from raw pixels toward more meaningful representations.

Activation Still Matters in CNNs

One compact rule

$$\mathrm{ReLU}(z) = \max(0, z)$$

What that means in a feature map

Weak negative responses get clipped away. Strong positive matches remain visible for later layers that are trying to assemble bigger patterns.

Why CNNs still need it

Without a nonlinearity, stacking convolutions would behave too much like one larger linear filter.

Why ReLU stayed popular

It is simple and often trains well, but dead units and brittle optimization can still appear if the rest of training is poorly set.

CNNs Build a Visual Hierarchy

Early layers

Often respond to edges, directions, and small textures.

Later layers

Can respond to parts, combinations of parts, and eventually patterns associated with whole objects.

The network is not jumping from pixels straight to “cat.” It is building the idea step by step.

From Edges to Parts to Objects

Why hierarchy matters

Low-level clues are reusable. A corner can help describe a window, an eye, or a letter. Higher layers combine these clues into richer concepts.

Why this feels natural

Human perception also seems to move from simple local structure toward more meaningful wholes.

Training Logic Stays Familiar

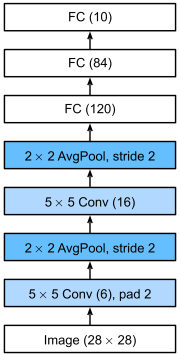
What CNN training still does

Forward pass, loss, backpropagation, and parameter update all remain the same basic cycle from NN4.

What changed instead

The gradients now flow through filters, feature maps, and pooling operations rather than only through dense layers.

LeNet Is a Clean Starting Story



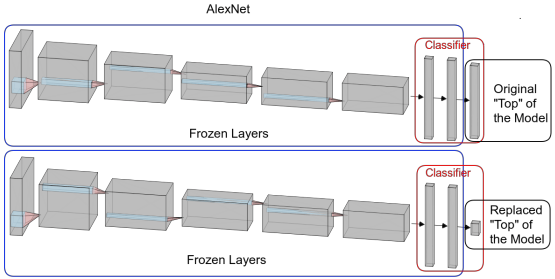
Why LeNet matters

It is one of the classic examples showing that convolutional structure can work for handwritten digit recognition.

Why teachers like it

The architecture is simple enough to explain without losing the main CNN ideas.

AlexNet Marked a Turning Point



Why it was important

AlexNet showed that deeper CNNs, large datasets, and strong computing hardware could dramatically improve image recognition performance.

The broader lesson

Architecture mattered, but so did data scale and computation.

Why Data and Hardware Mattered Too

The myth to avoid

CNN success was not only about one clever layer type.

What actually aligned

Better GPUs, bigger labeled datasets, improved regularization, and practical training tricks all helped CNNs become dominant in vision.

In AI, breakthroughs are often a combination of ideas, data, and compute.

Where CNNs Helped the Wider World

Common applications

Medical imaging, face grouping, document scanning, manufacturing inspection, traffic monitoring, and photo organization all benefited from CNN-based visual recognition.

Why this changed AI

CNNs gave AI a practical way to work with images at scale instead of relying only on hand-built visual features.

When CNNs Are Not Enough

The limitation

Some tasks need stronger long-range relationships, cross-image context, or multimodal reasoning than classic CNNs handle naturally.

What happens next

That is one reason later vision systems brought in attention mechanisms, sequence-style models, and larger multimodal designs.

The balanced view

CNNs were a major step, not the final word on vision.

A Good CNN Still Needs a Good Pipeline

The network is only one part

Data cleaning, labeling quality, train-validation split, normalization, and deployment checks all affect whether the system is trustworthy.

What students often miss

A strong architecture cannot rescue a weak data pipeline forever.

Data Augmentation Changes the Story

What augmentation does

It creates modified training examples such as flips, crops, brightness changes, or slight rotations so the model learns a broader version of the task.

Why this helps

The model becomes less dependent on one narrow visual presentation of each class.

Good augmentation teaches the right variation. Bad augmentation teaches the wrong invariance.

Learning Curves Still Tell a Story

What to watch

Training accuracy, validation accuracy, training loss, and validation loss still tell us whether the model is learning, memorizing, or simply stuck.

What has not changed

CNNs may look visually different from dense networks, but the discipline of validation remains exactly as important.

Common CNN Failure Modes

What can go wrong

Overfitting, weak labels, poor coverage of real-world conditions, sensitivity to background cues, and shortcuts based on spurious visual patterns.

A classic mistake

A model may seem to recognize an animal, but really rely on snow, grass, or camera style in the background.

Why this matters

Visual success on a benchmark can hide shallow reasoning.

Bias in Vision Data Is Easy to Hide

Why vision bias is tricky

If the dataset underrepresents certain people, conditions, environments, or devices, the model can fail unevenly while still looking strong overall.

Why this is socially important

Visual AI may affect policing, hiring, access control, healthcare, and public services. Uneven error is not a small side issue.

Interpretability Still Matters

What we want to know

Which part of the image influenced the prediction, and whether the model is using meaningful evidence or cheap shortcuts.

What interpretability can do

Heat maps, saliency tools, and feature inspection can help us ask better questions about model behavior.

A Simple Checklist Before Trust

Questions to ask

Is the data representative? Are the errors acceptable? Does validation reflect the real setting? Are the model's cues sensible? What happens when lighting, angle, or device changes?

Why this page matters

Trust in vision AI should come from disciplined checking, not from impressive pictures alone.

Why Sequence Models Come Next

Why CNNs are not the end of the neural story

Images have spatial structure, but language, speech, and time series add sequential structure. Those tasks push us toward different neural designs.

What carries forward

The big lesson stays the same: architecture should match the shape of the data.

Why AI5 and NN5 Belong Together

What AI5 showed

Good AI methods take problem structure seriously when they search through possible answers.

What NN5 showed

Good neural architectures also take structure seriously, but here the structure lives in the data itself, especially local visual patterns.

Summary

- CNNs were designed because images have local, repeated, spatial structure that dense layers handle poorly.
- Convolution uses small shared filters to detect patterns across many positions.
- Padding, stride, feature maps, and pooling all shape what information the network keeps or discards.
- CNN success came from architecture plus data, compute, and practical training choices together.
- Visual AI still needs careful judgment about bias, failure modes, validation, and interpretability.

Next Time

Where this story goes next

Later neural lectures move beyond images and ask how sequence structure changes the design of the model.

What to keep in mind

NN4 taught the learning machinery. NN5 showed how architecture changes when the data has geometry. The next step asks what happens when the data unfolds over time.

Thank You