
Testing Architectural Assumptions

ECE 57000 — Fall 2026

David I. Inouye

| Architectural assumptions need **empirical evidence**

Arc 5: CNN fundamentals → **empirical evidence** → ResNet classification → U-Net segmentation.

1. **Compare:** what changes under the same training recipe?
2. **Stress:** what happens when image position changes?
3. **Decide:** which conclusions does the evidence support?

Locality and sharing motivate the CNN. Now test what they buy us.

| Compare architectures under **one stated recipe**

Keep fixed	Change
Same 12,000 training / 3,000 validation images	MLP: 784 → 64 → 10
Same batches, five epochs, Adam at 10^{-3}	CNN: the two-convolution model just shown
Three seeds; use the final epoch of every run	Connections, parameter count, and computation

Equal epochs give equal data passes—not equal computation.

No augmentation, tuning, or test-set access in this comparison.

| This CNN improves validation accuracy **under this recipe**

Model	Mean accuracy	Range across three seeds
MLP	83.4%	83.0–83.7%
CNN	86.0%	85.7–86.3%

Same 3,000 validation images; five training epochs per run.

Supported: this CNN recipe performed better on this split.

Not established: every CNN beats every MLP, or the gap persists under equal compute.

| Fewer parameters can still require **more computation**

Complete model	Learned parameters	Multiply-accumulates per image
MLP	50,890	50,816
CNN	32,618	313,600

A convolution **stores a filter once but applies it at many positions.**

This CNN has about **36% fewer parameters** and **6.2× as many multiply-accumulates.**

| A better average can hide **different mistakes**

Label: T-shirt/top



MLP: Trouser
CNN: T-shirt/top

Label: Sneaker



MLP: Sneaker
CNN: Ankle boot

The CNN fixes one MLP error; the MLP gets another example right.

Inspect disagreements against the label before deciding which errors matter.

Stress the assumption about **position**

Architecture evaluation · 2/3 Stress · Compare → **Stress** → Decide

| Predict what happens when the image moves

A CNN uses convolution, then flattening and a linear classifier.

Must shifting an image preserve its predicted class?

A. Yes, filters are shared

B. Only for one-pixel shifts

C. No, the classifier can use position

C. A pattern can move in the feature map; the classifier can weight positions differently.

| A shared filter can feed a position-sensitive decision

Use a size-one filter with weight 1 and bias 0. It copies each input value.

	Original	Shift right
Input and feature map	[0, 1, 0]	[0, 0, 1]
Class A score: middle entry	1	0
Class B score: right entry	0	1
Prediction	A	B

Same filter at every position; different class after a shift.

| Define the shift **before** interpreting its score

Original



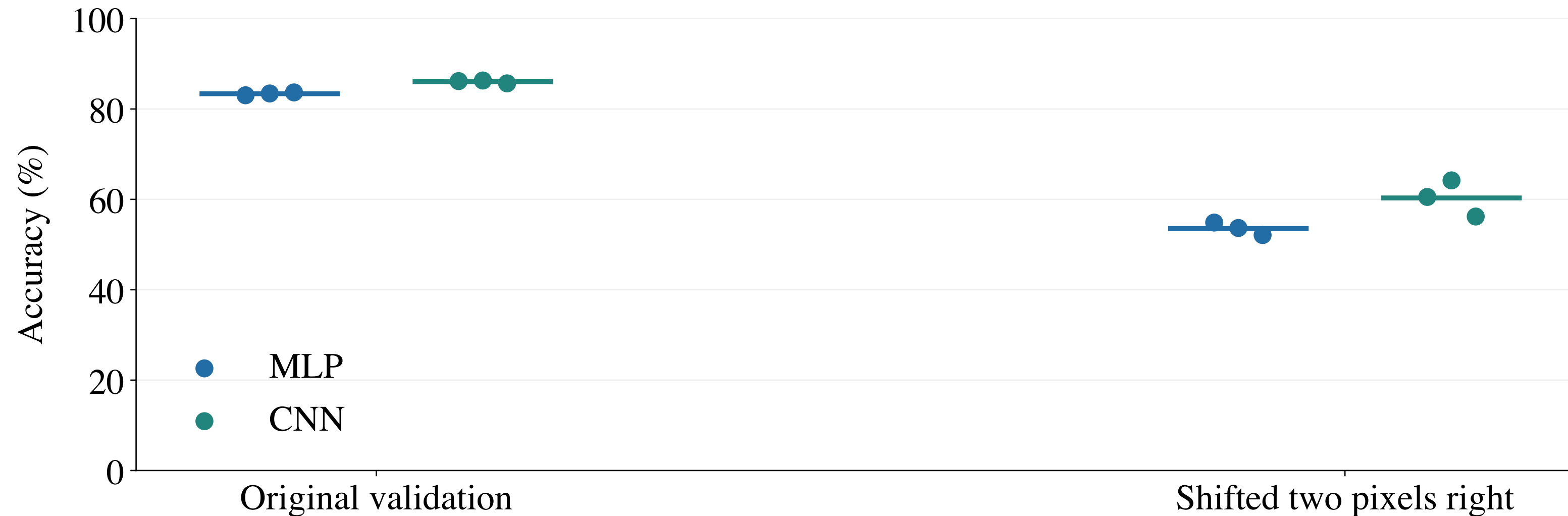
Two pixels right; zero fill



Move each validation image **two pixels right**. Fill the left edge with zeros; discard the rightmost two columns. No wrapping.

Keep the original labels. **This assumes the remaining image still supports the same category.**

| Both models lose accuracy on shifted images



Mean accuracy: MLP 83.4% → 53.5%; CNN 86.0% → 60.3%.

The CNN does better here, but its original validation accuracy **does not establish shift robustness**.

| Choose a model for the conditions it must handle

Architecture evaluation · 3/3 Decide

Evidence so far	Decision it informs
Higher original validation accuracy	Candidate for similarly positioned images
Fewer parameters, more arithmetic	Check storage and runtime budgets separately
Large shifted-image accuracy loss	Investigate position variation before deployment

Compare → **Stress** → **Decide**: a single accuracy number cannot settle all three.

| Architectural bias helps **when its assumptions fit**

- **Assumption:** local patterns recur; sharing is useful when that is true.
- **Evidence:** these two recipes favor the CNN on one split; both degrade under one shift.
- **Next test:** fix the recipe and selection rule, then evaluate on held-out data matching the intended conditions.

More expressive features, efficient gradients, and a working loop are necessary steps. **They do not replace evidence about where the model will be used.**