C1 / 18 MINUTES Week of 23 November 2026

Can one filter recognize a scratch after it moves?

Two inspection cameras see the same two orientations. Camera A records bright scratches on a dark surface. Camera B records dark scratches on a lighter, noisier surface. Build a small network and find out which parts of its response survive a change in position or camera.



A · horizontal A · vertical B · horizontal B · vertical

Each enlarged tile is an actual 12 × 12 input from the notebook. Horizontal is class 0. Vertical is class 1.

Trace one image prediction

01 Use the small image and kernel printed on page 2. At each valid position multiply the matching input and kernel entries, add them and add the bias. Keep the kernel in the shown orientation. This is the cross-correlation used by CNN libraries.

02 Complete **response_at**. At row 0, column 1, show the four products. Calculate which locations survive ReLU, then average the six hidden values and subtract 0.2 to get the logit. Apply the sigmoid to get P(vertical).

03 Identify a hidden activation greater than 1. Explain how it differs from the image probability, and what would happen if two affine layers had no nonlinearity between them.

Working file

Open **CS418_Lab11_Starter.ipynb** in Jupyter or choose Upload notebook from Colab's File menu. The cells generate the data. A CPU with NumPy 1.20+ and Matplotlib is sufficient. Complete the marked functions and keep the figures in the notebook.

Two-hour session. C1 forward pass 18 min · C2 shared gradient 22 min · break 10 min · C3 geometry 18 min · C4 shifted images 25 min · C5 transfer 27 min.

One patch · Four products, bias and ReLU output

C2 / 22 MINUTES CS418 Lab 11

How a shared filter learns from six windows

The shaded patch below is one use of the kernel. Every valid window uses the same four weights and the same bias. A weight update collects evidence from every location at which the weight was used.

Input X

0	1	0	0
1	1	0	1
0	0	1	0

Kernel K

1	-1
0	1

Bias = −0.5

The tinted patch starts at row 0, column 1. Indices begin at zero.

$$p = \sigma(\text{mean}(\text{ReLU}(a)) - 0.2), \qquad y = 1$$

$$\delta_{ij} = \frac{p - y}{6} \mathbf{1}[a_{ij} > 0], \qquad \frac{\partial \mathcal{L}}{\partial K_{uv}} = \sum_{i,j} \delta_{ij} X_{i+u, j+v}$$

Follow K[0, 0] from prediction to update

01 For sigmoid binary cross-entropy, the derivative of the loss with respect to the logit is p − y. The mean shares it across six positions. ReLU passes the derivative only at positive pre-activations. Write the six resulting values of δ.

02 For the top-left kernel weight, each position contributes δ times the top-left input in its patch. Circle the two positions with nonzero contributions. Sum them before updating. The bias derivative is the sum of all six δ values.

03 Complete **shared_gradient**. Update the four filter weights and the bias together using learning rate 0.2. Keep the head fixed. Report the old and new binary cross-entropy and give the direction of the K[0,0] update.

Check one gate by hand

For the top-left window, the products sum to zero and the bias gives a response of −0.5. ReLU outputs zero there. A zero derivative from this window can coexist with a nonzero total gradient for the shared filter.

The two contributions to K[0,0] and their sum

C3 / 18 MINUTES CS418 Lab 11

Count activations and learned parameters

Move from the toy crop to the 12 × 12 tile. The full network has six trainable 3 × 3 kernels, each followed by ReLU. Max pooling reduces each map locally. Global averaging produces one value per channel for a binary head.

$$H_{\text{out}} = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor + 1$$

Here H is the input side length, P the padding on each side, K the kernel width and S the stride. Use the same formula for width. All convolutions in this lab use dilation 1.

Operation	Geometry to use	Your output for one tile
Input	One grayscale channel	1 × 12 × 12
Convolution + ReLU	6 filters, K = 3, S = 1, P = 0	_____ × _____ × _____
Local max pool	K = 2, S = 2, P = 0	_____ × _____ × _____
Global spatial mean	Average each map	_____ features
Binary head	One weight per feature + bias	_____ score

01 Complete **output_width** and **pool2x2**. The code reshapes each map into blocks. Reduce over the two within-block axes to keep the largest entry per window. Check the provided 4 × 4 example.

02 Count weights and biases in the convolution and the head. If the camera instead records RGB, recalculate the size of the filter bank. If only the input side doubles, which counts change?

03 Check a second geometry: input side 15, kernel 3, stride 2, padding 1. Calculate the output width before running the notebook.

Pooling and receptive fields

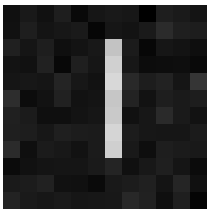
A convolution at one position sees a 3 × 3 input patch. A max over a 2 × 2 group of adjacent convolution responses can depend on a 4 × 4 region of input. ReLU, max pooling and global averaging add no learned weights.

My parameter count · Filter bank, head and total

C4 / 25 MINUTES CS418 Lab 11

Move the image by one pixel

Train the source network for 35 epochs using 96 training tiles and 48 separate validation tiles. The supplied routine performs mini-batch SGD with momentum. Your pooling function is part of its forward and backward passes.



Original validation tile



Shifted one pixel right

The added column is filled with zero. The mark remains fully visible. Both panels represent the same validation example.

01 Trace the forward pass from patches to one logit. Plot training and validation binary cross-entropy (BCE). Read validation accuracy and BCE at epochs 5 and 35. Explain how one can keep changing after the other is flat.

02 Inspect a learned feature map for the original and moved tile, using the same color scale. Compare original column j with moved column j + 1 where both exist. Record the largest aligned difference across all 48 pairs.

03 Report three separate quantities: paired agreement of the decisions, accuracy against the true labels, and the mean absolute change in the predicted probability. Use them to explain which outputs stayed identical in this experiment.

What counts as evidence?

A map is translation-equivariant if it moves with its input. An invariant output keeps the same value. Write whether your evidence concerns the feature map, the predicted class, or the predicted probability. One-pixel shifts can move responses between pooling windows.

Shift evidence · Aligned map difference, class agreement and probability change

C5 / 27 MINUTES CS418 Lab 11

Adapt the filters to a second camera

Camera B supplies 24 labeled training originals, 48 validation originals and 80 reserved test originals. Start both runs from the Camera A filters and the same new classification head.

Choose which weights can move

Run	Trainable parts	What to record
Frozen features	Head weights and head bias	Validation BCE, accuracy, filter change
Fine-tuned	Filters, filter biases and head	Validation BCE, accuracy, filter change

01 Use a left-right mirror to make one additional training view per original. State the label of a mirrored vertical scratch. What label would a 90° turn require? Explain why the split happens before augmentation.

02 Complete **trainable_names**. Train both runs for the same 30 epochs at learning rate 0.075. Hold the starting weights, examples and batch order fixed. Count how many parameters each run can update and verify the frozen kernels did not move.

03 Select the candidate with lower validation BCE. Resolve an exact tie in favor of the frozen run. Record the choice before opening the test cell. Report test BCE, accuracy and sample size. Name one real inspection condition missing from these tiles.

Optional preview · Put the scratch at the border

Shift the validation tiles by eight pixels. Compare paired label agreement and inspect the two views of one example. Is there enough visible evidence left to recover its orientation? Explain how cropping changes the interpretation of the result.

Keep in your notebook

One worked patch and shared-weight update, annotated shapes and parameter counts, the learning curves, the paired maps and shift statistics, and the frozen-versus-fine-tuned comparison with a final test result.

My selected run, validation evidence and test result