



CPH 100: Foundations for CPH

Vision Models

Irene Y. Chen

Slides adapted
from Adam Yala

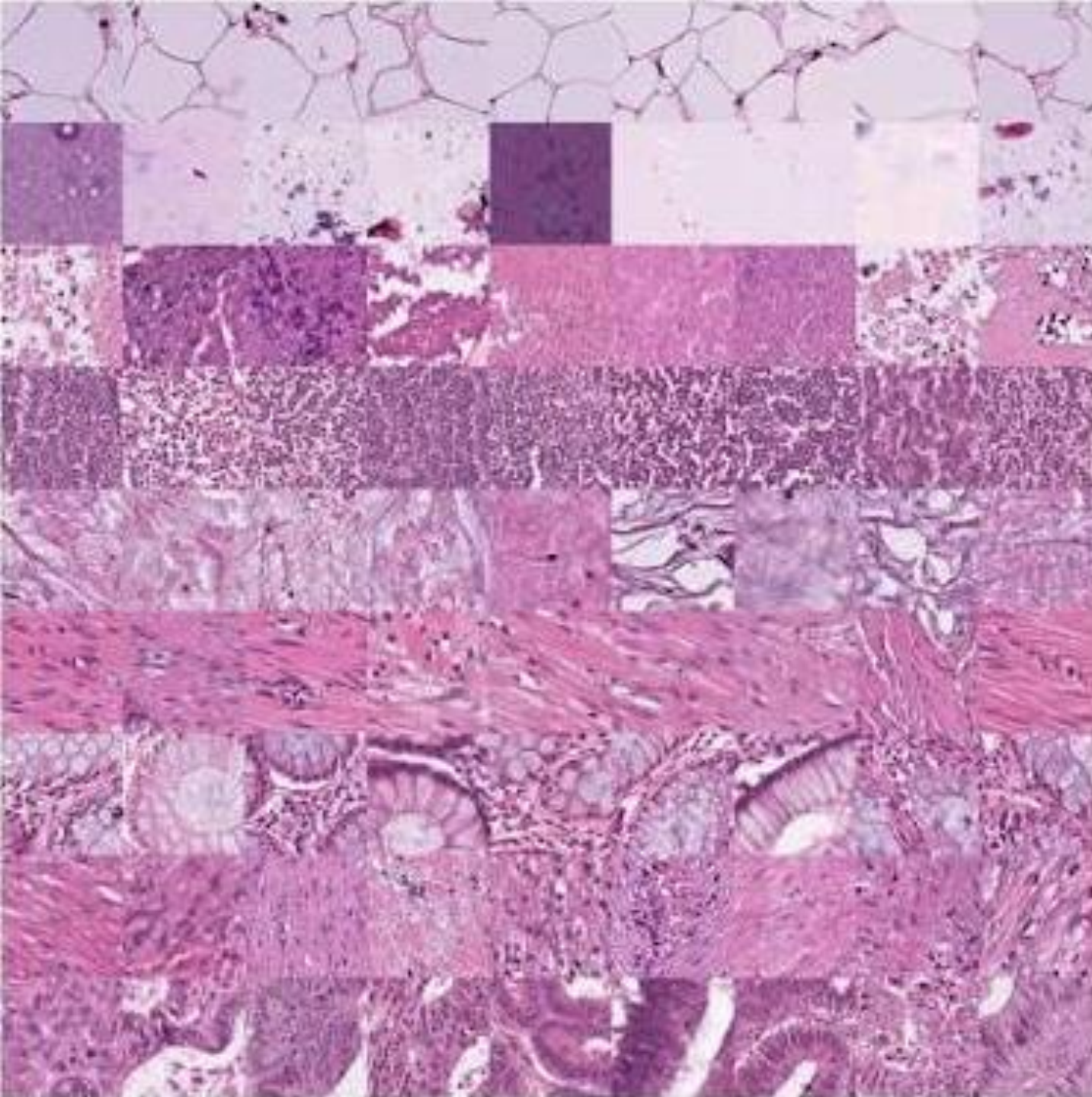
Outline

1. What is in an image? (15 mins)
2. Where is it? (20 mins)
3. Evaluation and beyond (15 mins)



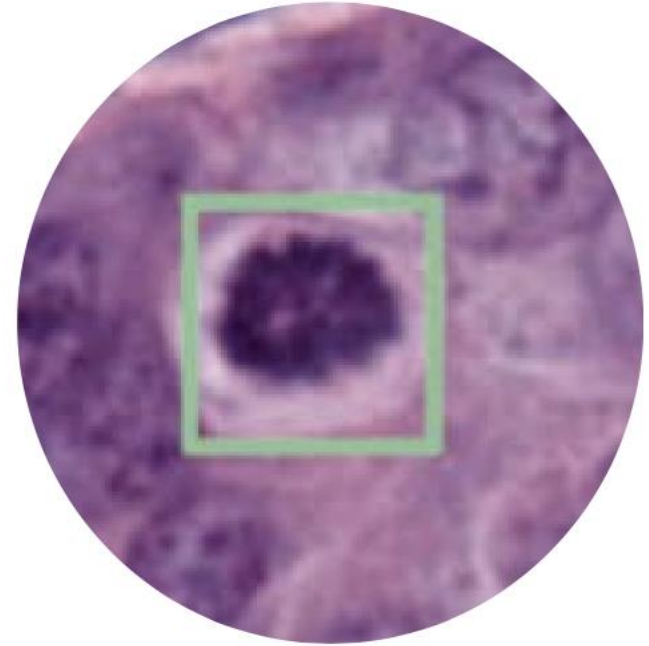
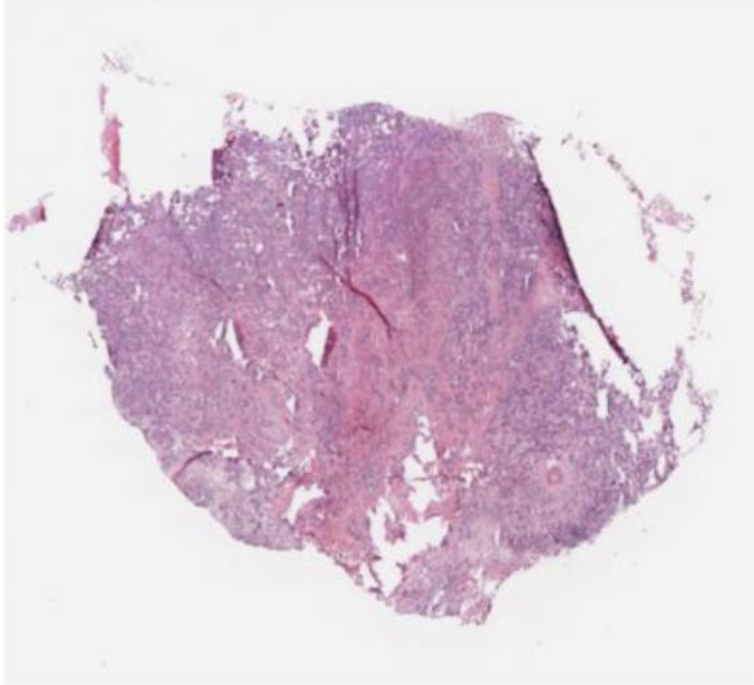
How can we make Data
146 better for you?

Learning Objective: Prepare students for Problem set 2



What does a
model see when
it sees a
pathology slide?

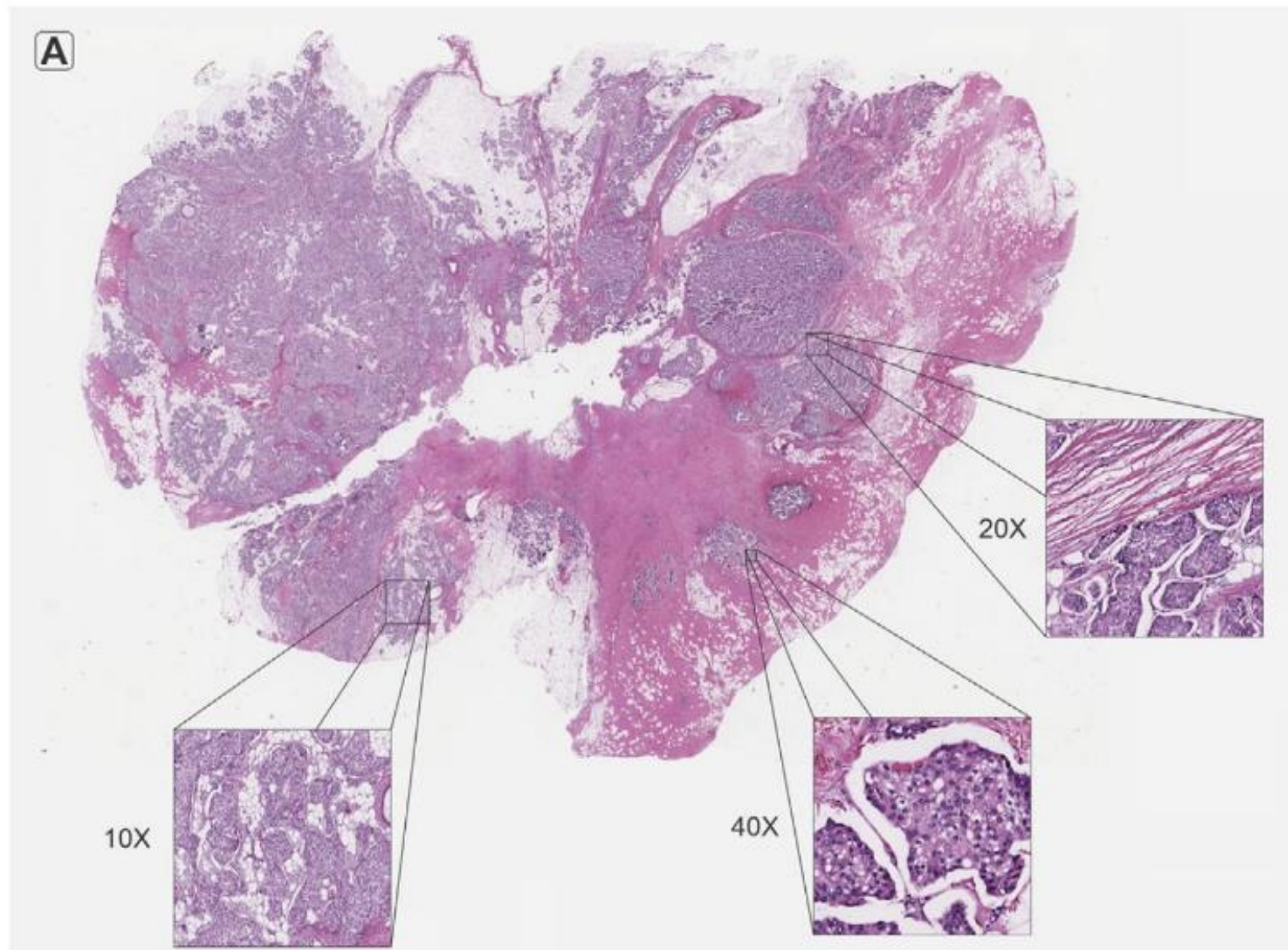
Pathology slides are huge



Pathology slide

- 30k pixels x 30k pixels
- 1.5 cm x 1.5 cm (0.6 inch)

1 cell = 15 pixels

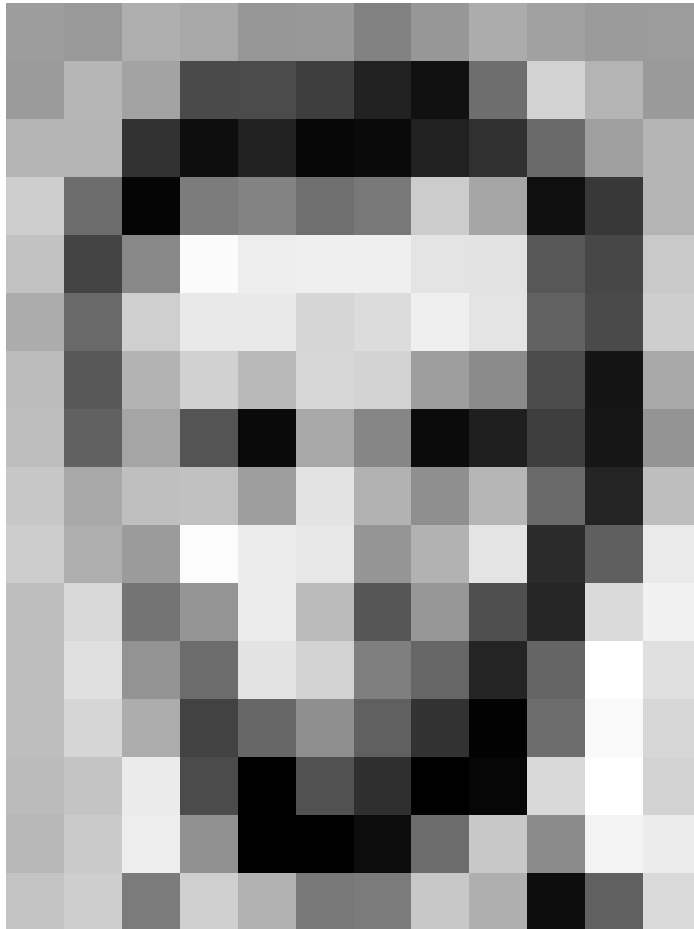


An example of breast H&E WSI from TCGA-BRCA dataset: (A) Illustration of 3 tile images in 10 $\times$, 20 $\times$, and 40 $\times$ magnifications in a WSI.

From: Publicly available datasets of breast histopathology H&E whole-slide images: A scoping review ⁵

Hand-crafted features have many problems

- Designing features is hard
- Potentially omits complex interactions
- Progress stagnates over time
- Errors compound



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

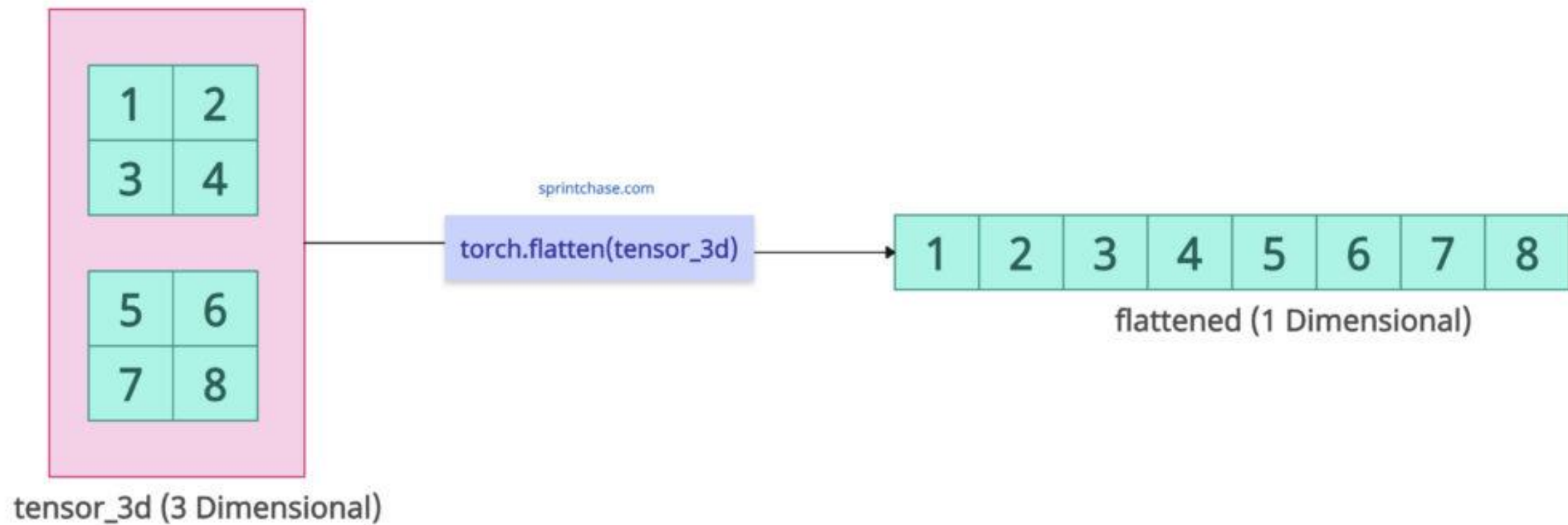
An image is a tensor

$$x \in \mathbb{R}^H \times W \times c$$

H = height, W = width, C = channels

224 x 224 RGB image

224 x 224 x 3 = 150,528 values



Baseline 1: Flatten Image

$$\mathbf{x} \in \mathbb{R}^H \times \mathbb{R}^W \times \mathbb{R}^c \rightarrow \mathbf{x_flat} \in \mathbb{R}^{HWc}$$

$$p(y \mid \mathbf{x}) = \text{softmax}(W\mathbf{x} + \mathbf{b})$$

Baseline 2: Add nonlinear hidden layers

$$\mathbf{h}_1 = \text{ReLU}(W_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{h}_2 = \text{ReLU}(W_2 \mathbf{h}_1 + \mathbf{b}_2)$$

$$\hat{y} = \text{softmax}(W_3 \mathbf{h}_2 + \mathbf{b}_3)$$

Baseline 2: Add nonlinear hidden layers

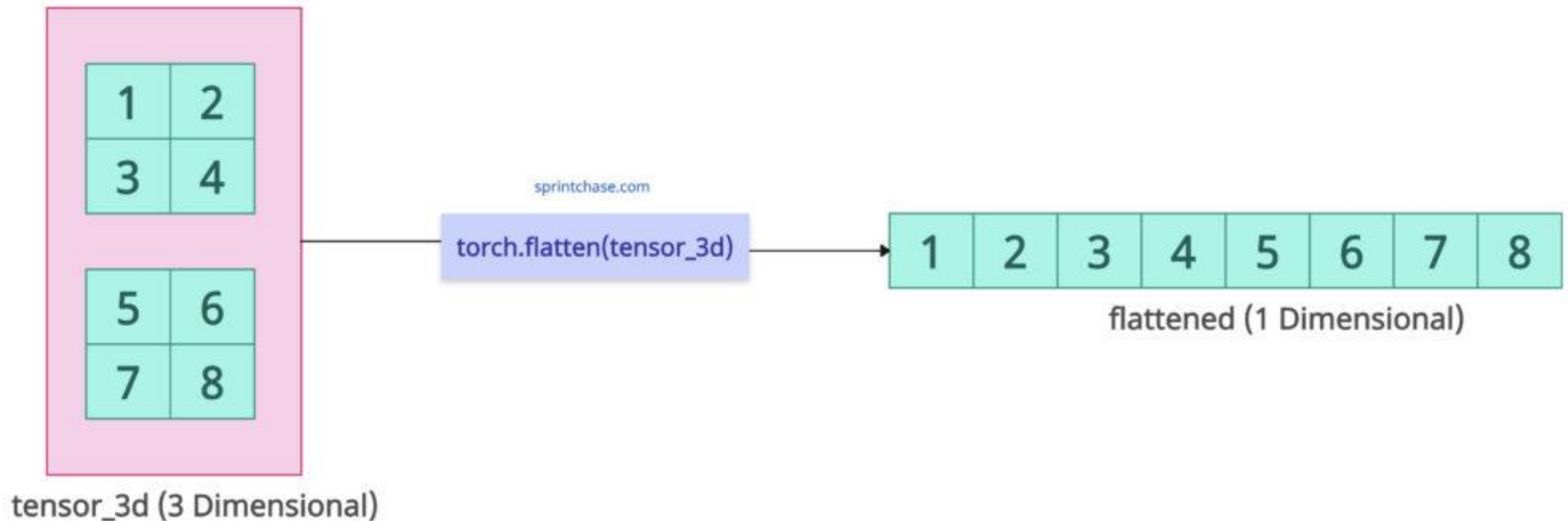
$$h_1 = \text{ReLU}(W_1x + b_1)$$

$$h_2 = \text{ReLU}(W_2h_1 + b_2)$$

$$\hat{y} = \text{softmax}(W_3h_2 + b_3)$$

What's missing here?

What does flattening throw away?



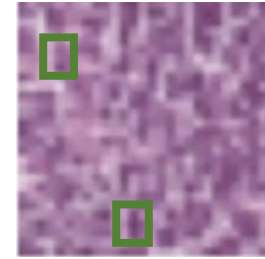
What does flattening throw away?

No locality



- Nearby pixels are related
- Edges and textures are spatially local

No translation structures



- Ignores repeating patterns
- E.g., vertical edge

Source Layer

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

Convolutional
Kernel

-1	0	1
2	1	2
1	-2	0

Destination Layer

	5						

$$\begin{aligned}
 &(-1 \times 5) + (0 \times 2) + (1 \times 6) + \\
 &(2 \times 4) + (1 \times 3) + (2 \times 4) + \\
 &(1 \times 3) + (-2 \times 9) + (0 \times 2) = 5
 \end{aligned}$$

$$z_{ij} = \sum K_{a\beta} c_{x_{i+a}, y_{j+\beta}} + b$$

One filter produces one feature map

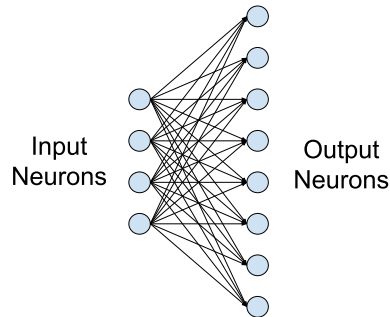
input: $H \times W \times C$ kernel: $k \times k \times C$ output: $H' \times W' \times 1$

With F filters:

$$H' \times W' \times F$$

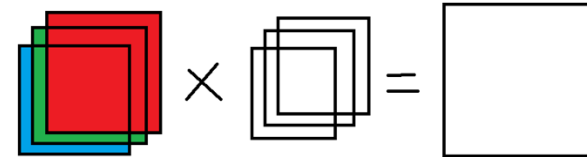
Convolution uses fewer parameters

Fully connected



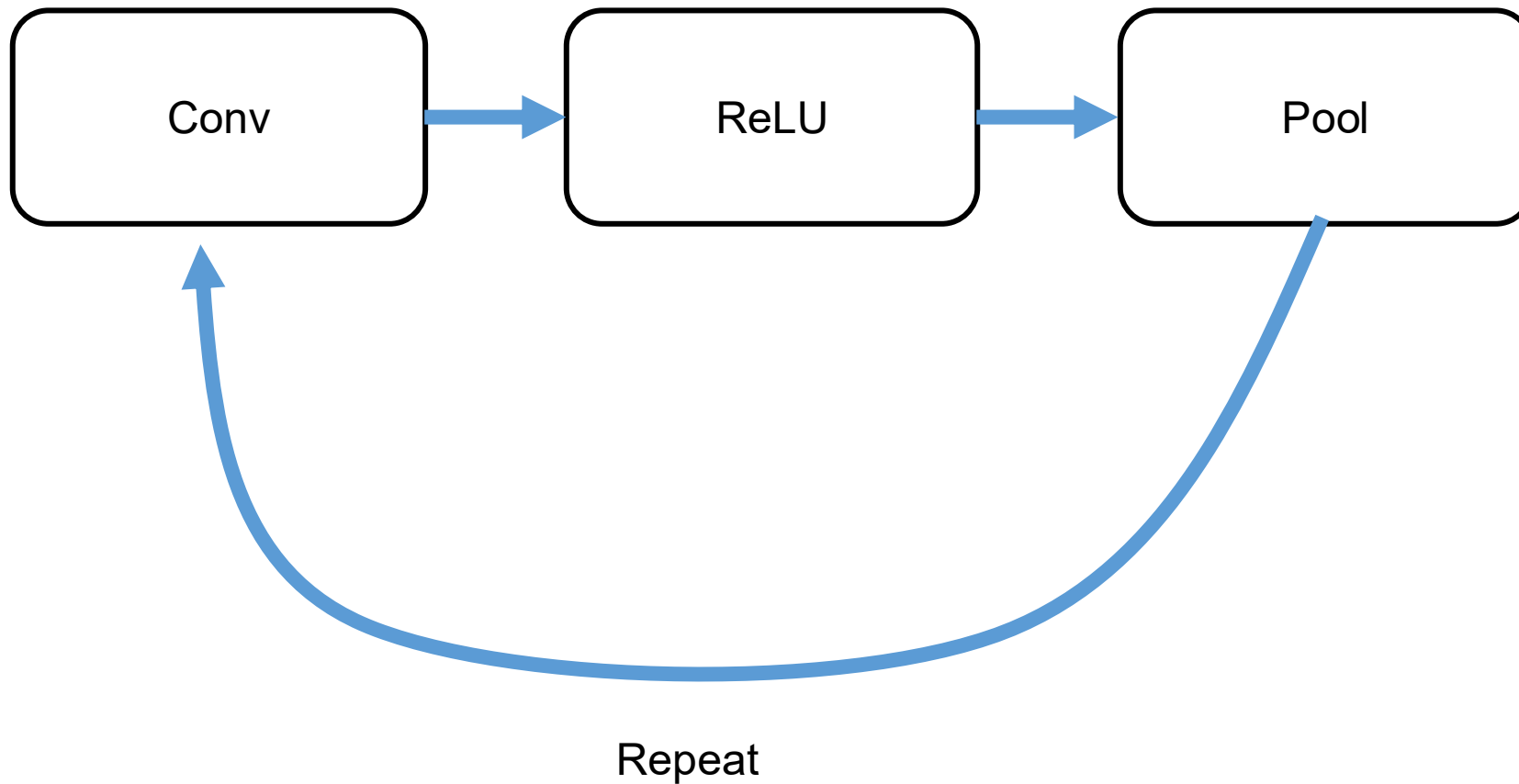
- 28 x 28 x 3 input with 64 units
- 150,000 weights!

3x3 Convolution



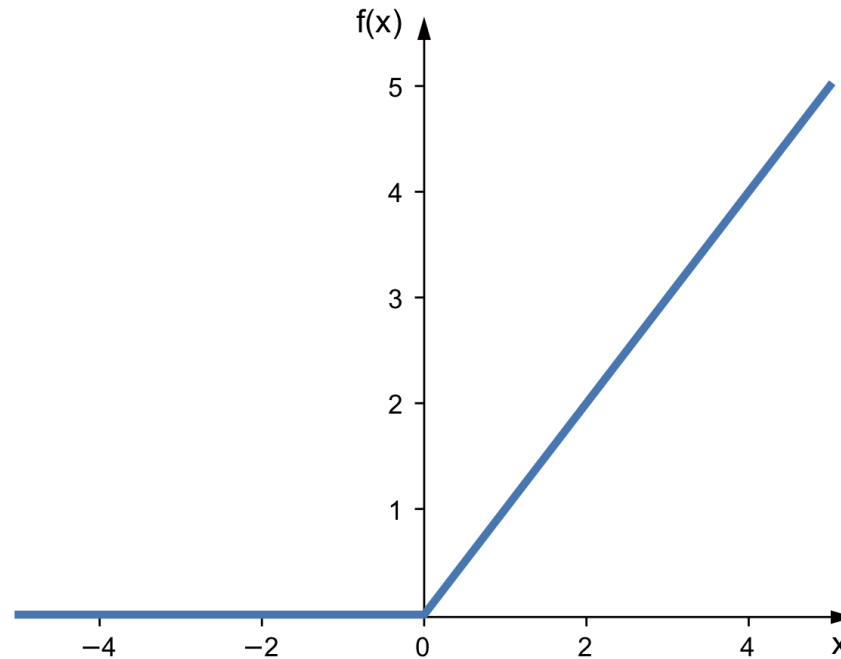
- 64 filters
- 3 x 3 x 3 x 64
- 1728 weights

Basic CNN block

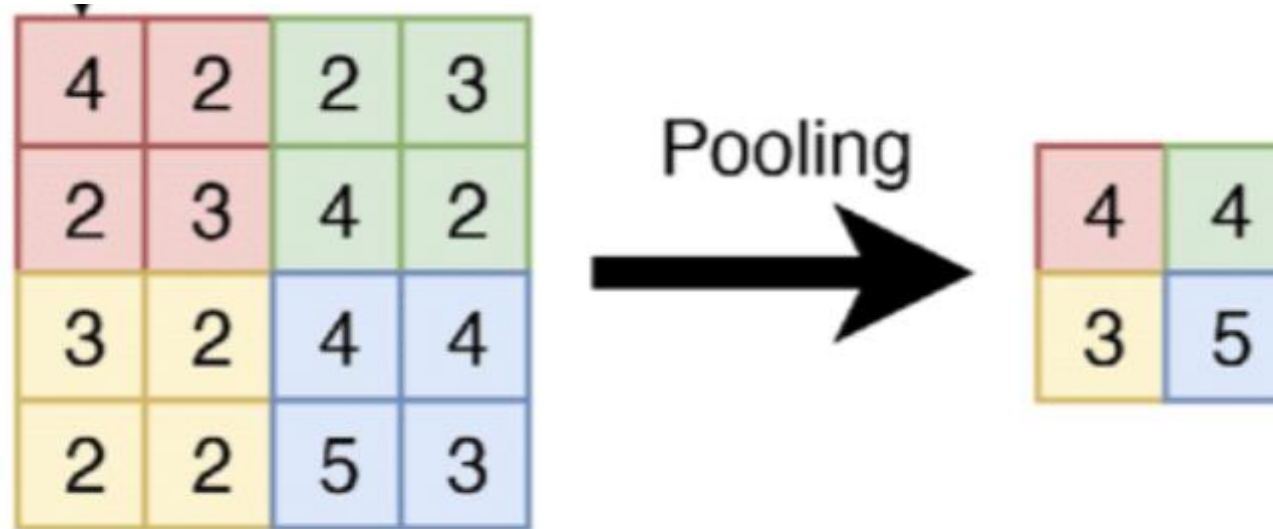


ReLU makes depth possible

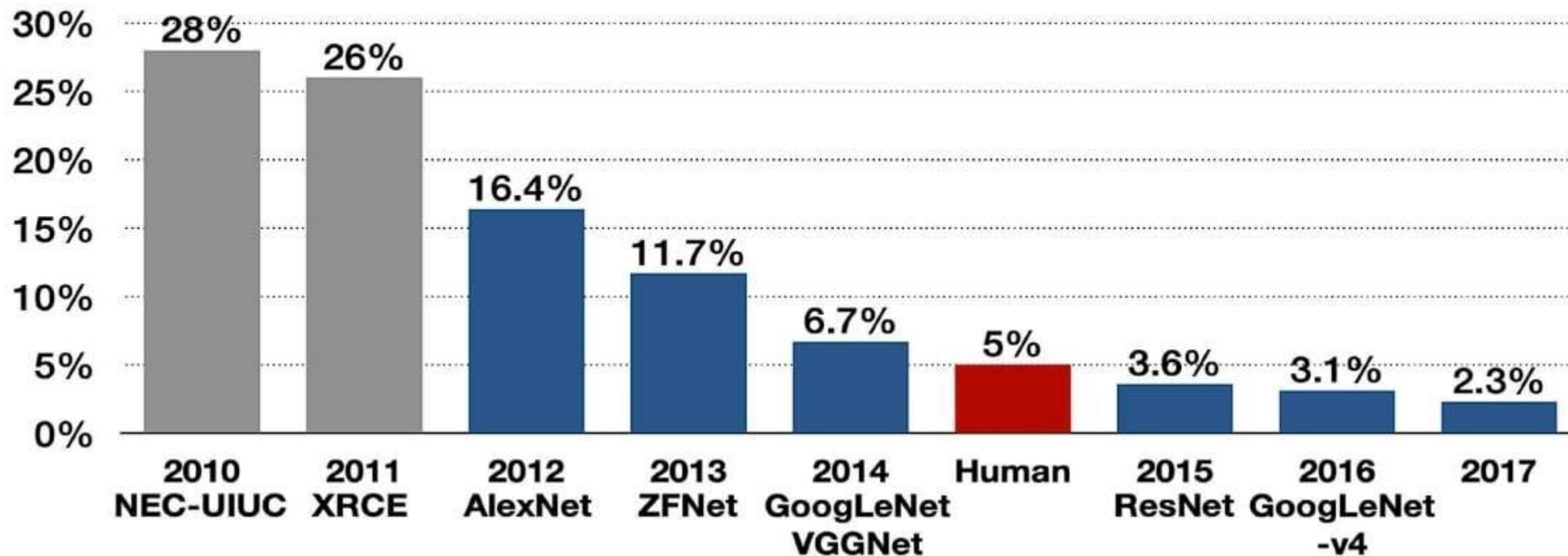
$$\text{ReLU}(x) = \max(0, x)$$



Pooling reduces spatial resolution



Top-5 error



Why did AlexNet succeed?

1. Architecture innovation
2. Data change
3. GPUs

Data augmentation can encode invariance

$x \rightarrow T(x)$ while y stays fixed

- Cropping
- Flipping
- Intensity / color

Part 1: CNN Implementation for Medical Image Classification [25 pts]

Implement and compare different architectures for PathMNIST tissue classification:

0. **Simple Linear classifier:** Flatten input images and re-implement logistic regression in PyTorch.
1. **MLP:** Flatten input images and use a multi-layer perceptron (MLP) model
2. **Your CNN:** Design and implement your own convolutional architecture. Explore different architecture layouts.

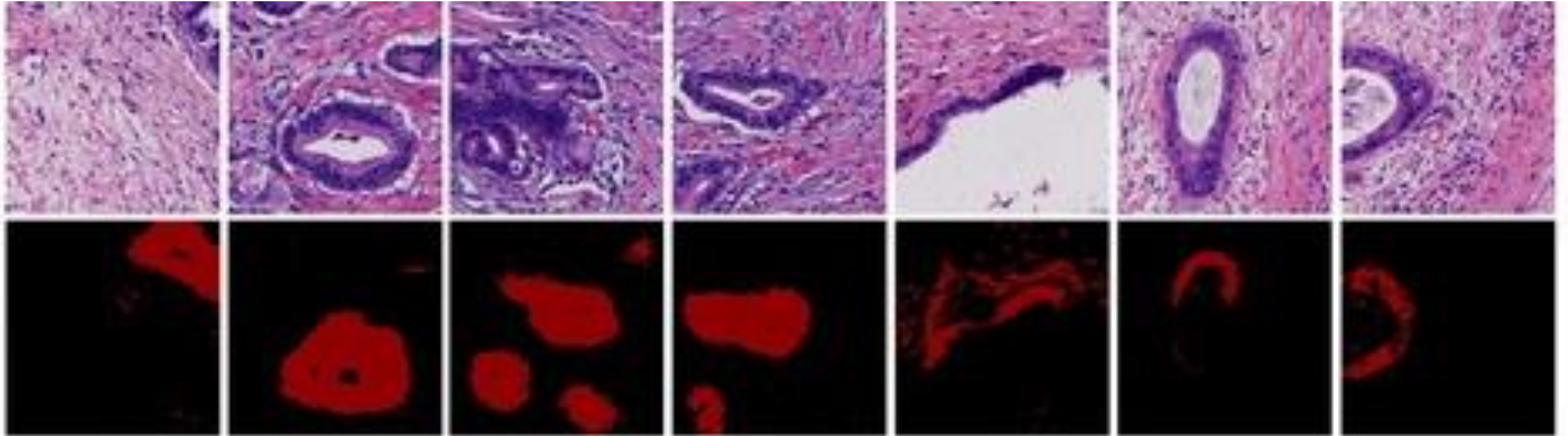
Performance Expectations (1 epoch, 100 steps):

- MLP Model: ~32% validation accuracy
- Your CNN: >54% validation accuracy

Final Model Requirements:

- Train your best model on a longer training schedule.
- **Target:** >90% validation accuracy

Segmentation requires spatial precision

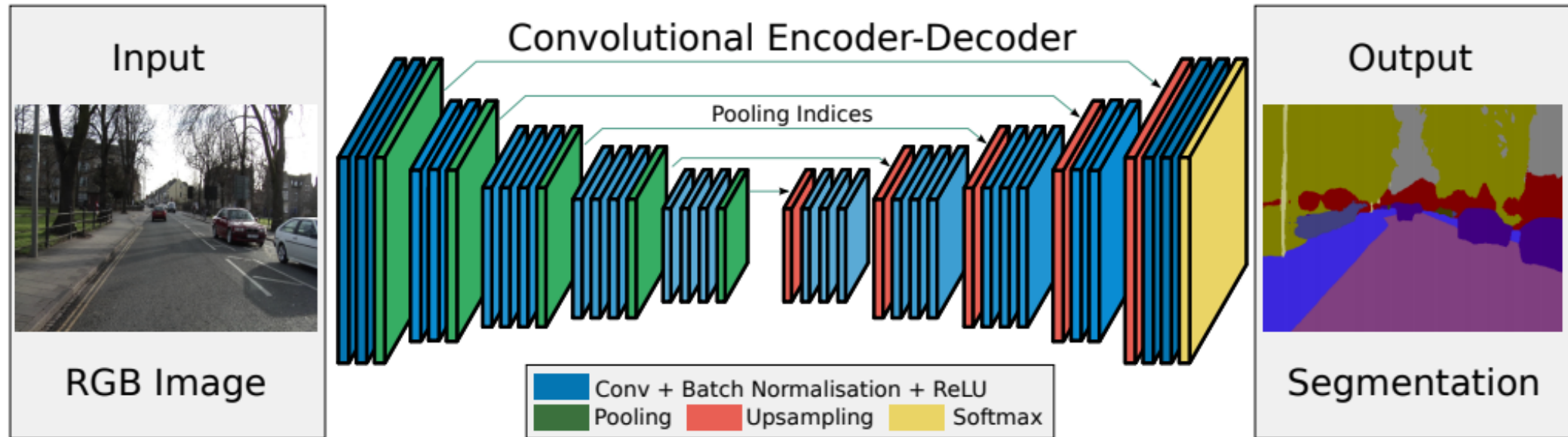


Segmentation predicts a class label for each pixel

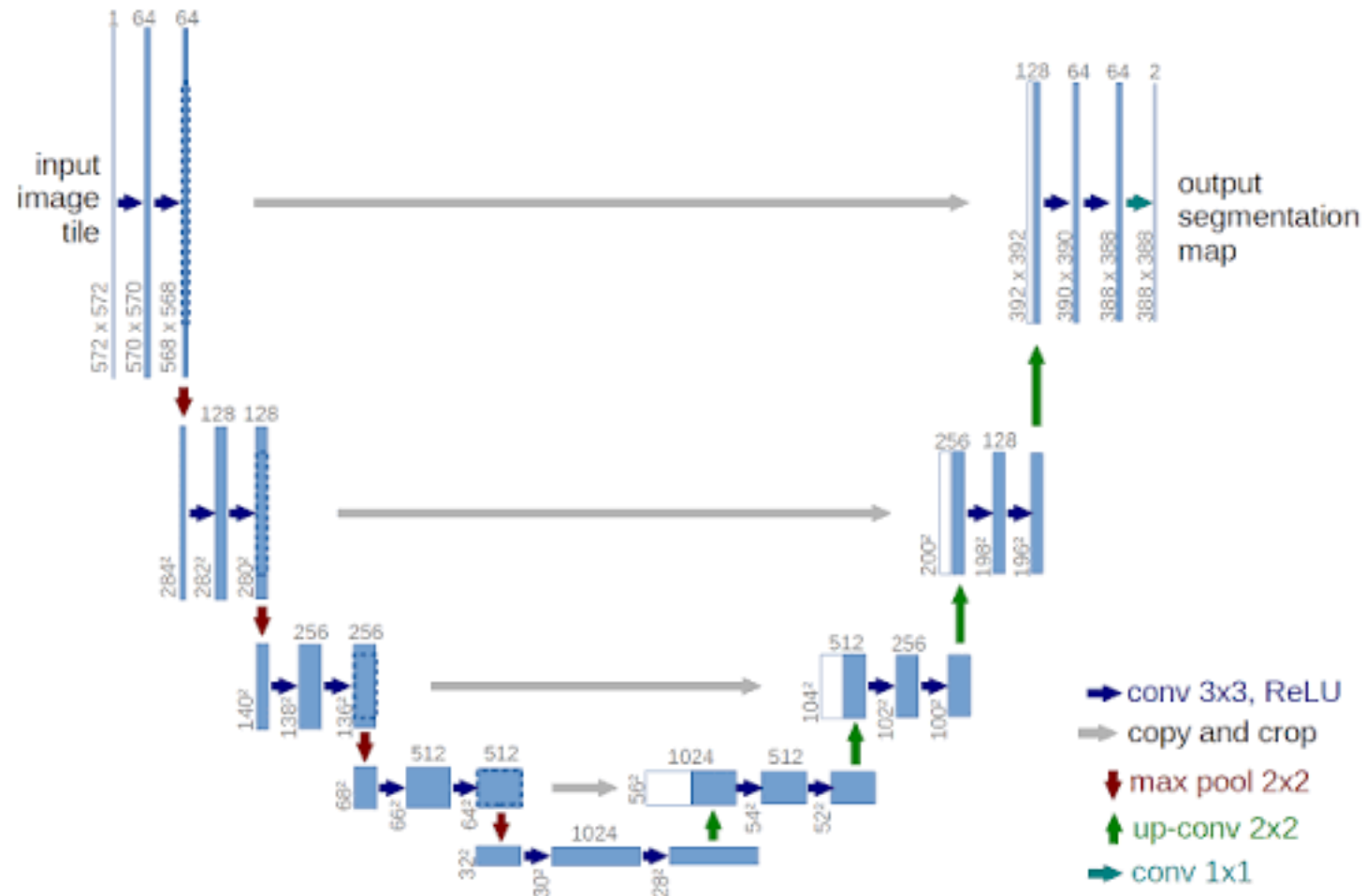
$$f : \mathbb{R}^{H \times W \times C} \rightarrow \{1, \dots, K\}^{H \times W}$$

H = height, W = width, C = channels, K = number of class labels

Encoder-decoder means compress then reconstruct



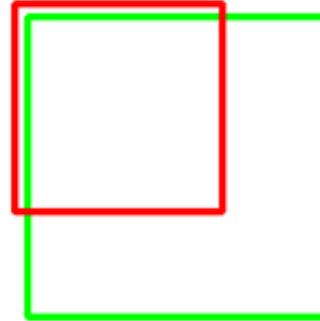
U-net is powerful tool for segmentation



Segmentation is evaluated with IoU

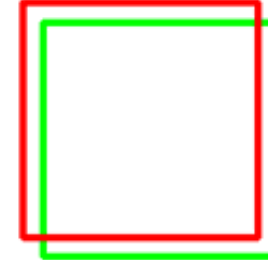
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


IoU: 0.4034



Poor

IoU: 0.7330



Good

IoU: 0.9264



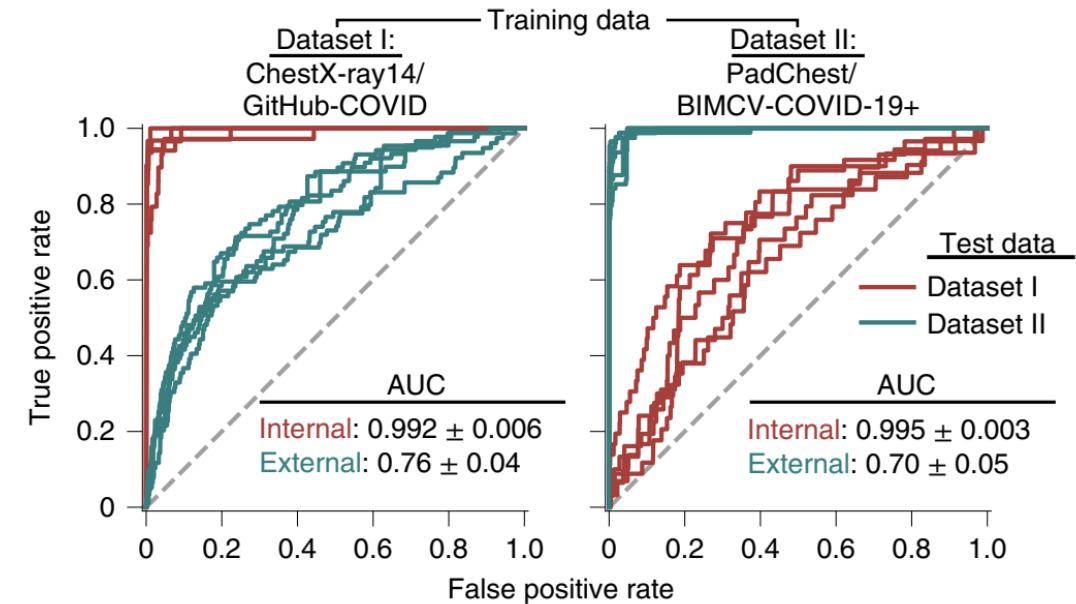
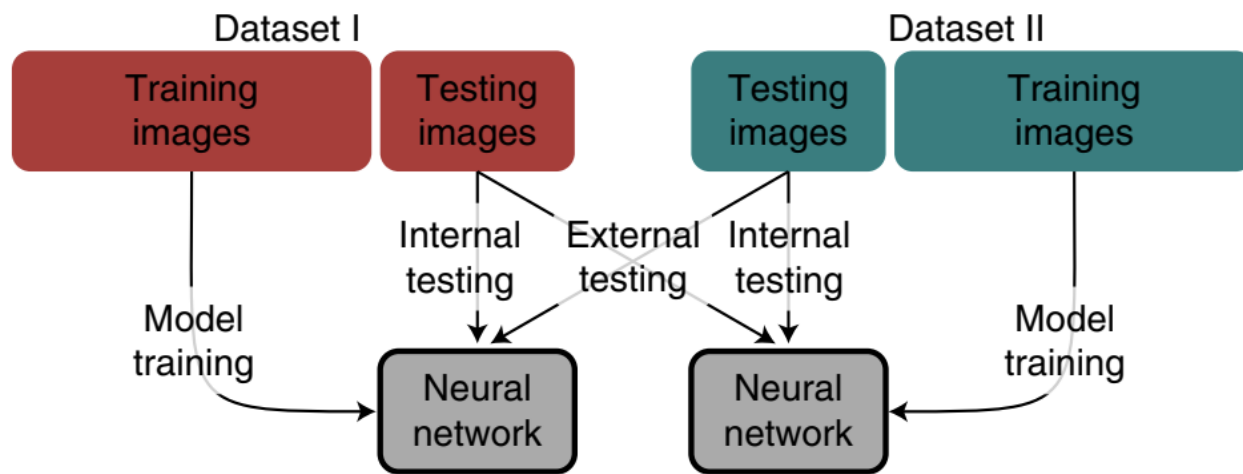
Excellent

What if your model gets 99%
accuracy?

Can we detect COVID-19 from chest x-rays?

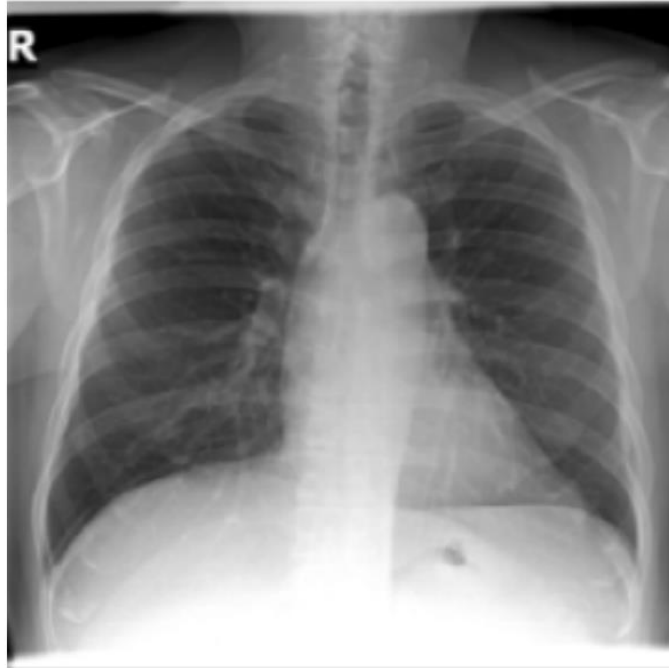
1. External vs internal test performance
2. Interpretability via saliency maps
3. Shortcut learning

External validation shows drop in performance

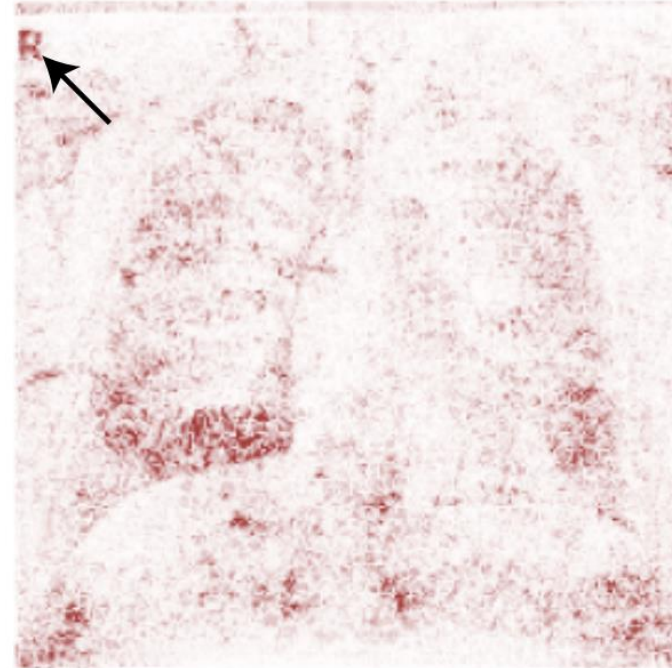


Saliency maps reveal shortcut learning

Negative image



Important pixels



Model output (log odds): -13.86

Shortcut learning tests

1. External validation drops AUC to 0.7 from 0.9
2. Saliency maps focused on laterality markers, text annotations, and other data-source features.
3. Trained a CycleGAN to generate COVID-positive / negative images and showed it added laterality markers
4. Laterality marker swapping flipped predictions
5. Changing the patient positioning also flipped predictions
6. Cropping did NOT resolve generalization problem

Summary

- ✓ What is in an image?
- ✓ Where is it?
- ✓ Evaluation and beyond



How can we make Data
146 better for you?

Next class: Vision transformer models with guest lecturer Zhongyuan Liang