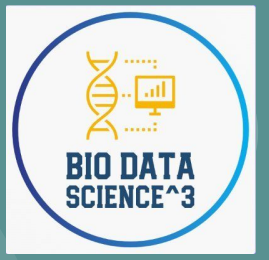


Convolutional Neural Networks

Guillem & Roderic, Summer 2025

Recap 1:

Modeling typically involves finding the values of the _____ that _____ the _____

Recap 1:

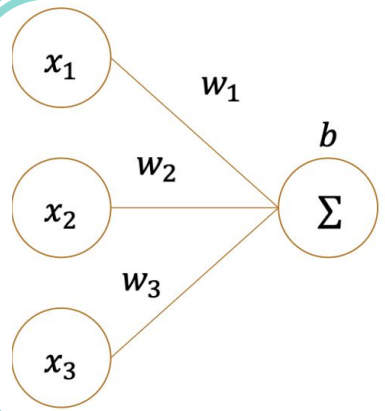
Modeling typically involves finding the values of the **parameters** that **optimize** the **objective function**

Recap 2:

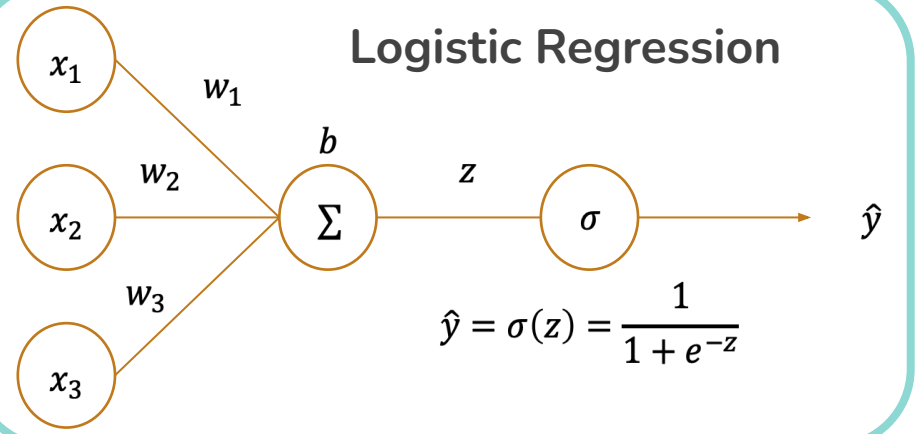
Models that we have discussed so far:

Algorithm	Objective Function	Goal	Optimization	Main Use
PCA	Variance	Maximize	Direct computation	Dimensionality Reduction
Linear Regression	RSS/MSE	Minimize	Direct computation	Regression
Logistic Regression	Cross-Entropy	Minimize	Gradient Descent (unique outcome)	Classification
Neural Network	MSE, Cross-Entropy, etc.	Minimize	Gradient Descent (variable outcome)	All

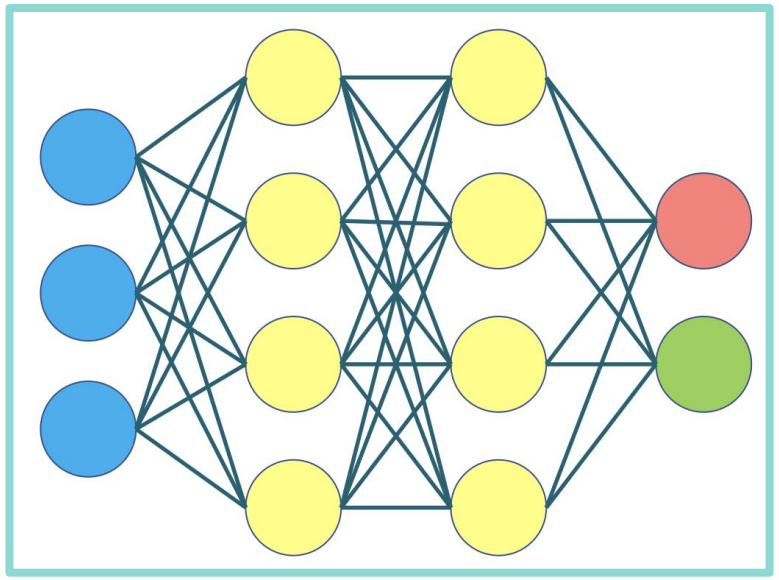
Recap 3:



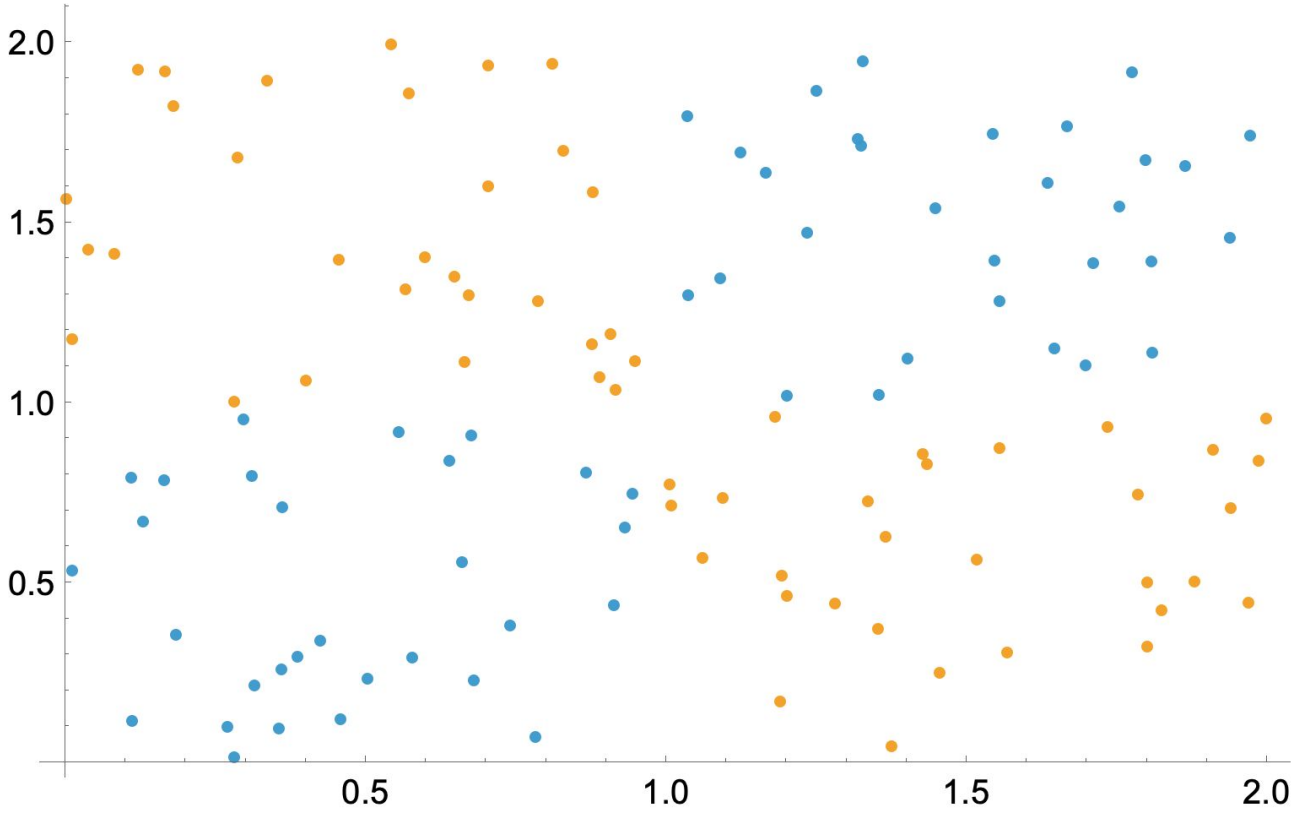
Linear Regression



Neural Networks

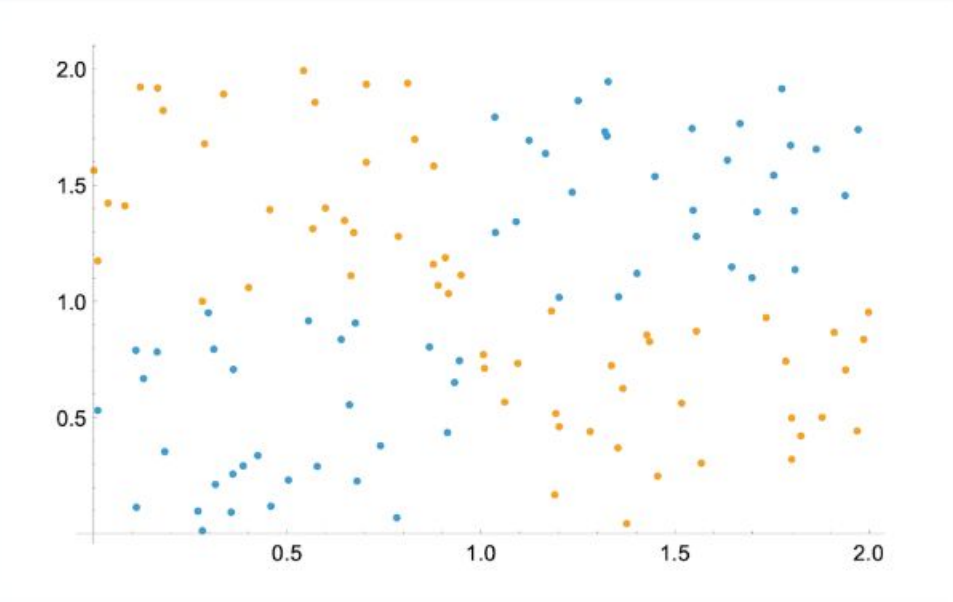


XOR Classification



Goal: Classify points into blue and orange labels according to their position.

Using logistic regression, what is the expected accuracy of the resulting classification?

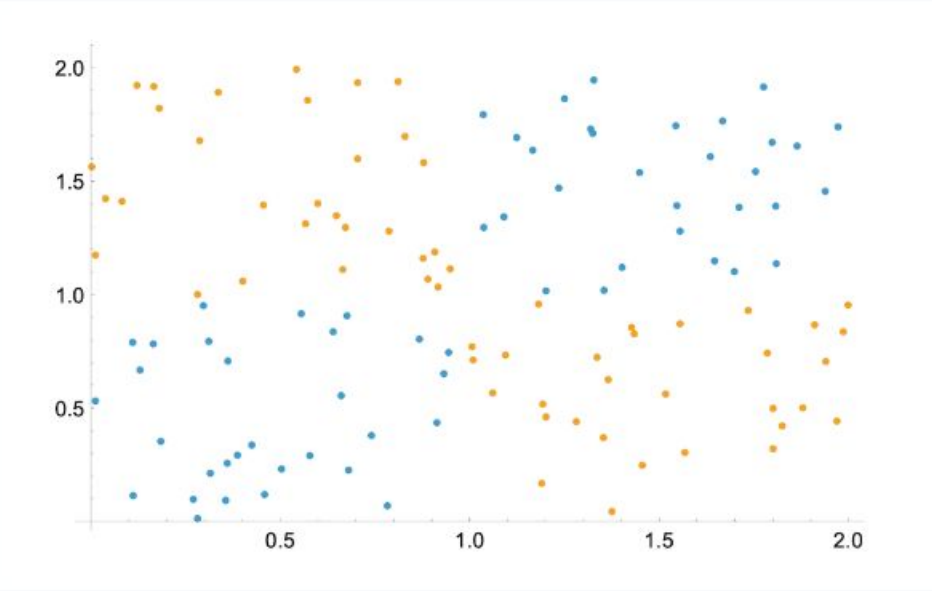


$\simeq 100\%$

$\simeq 75\%$

$\simeq 50\%$

Using logistic regression, what is the expected accuracy of the resulting classification?



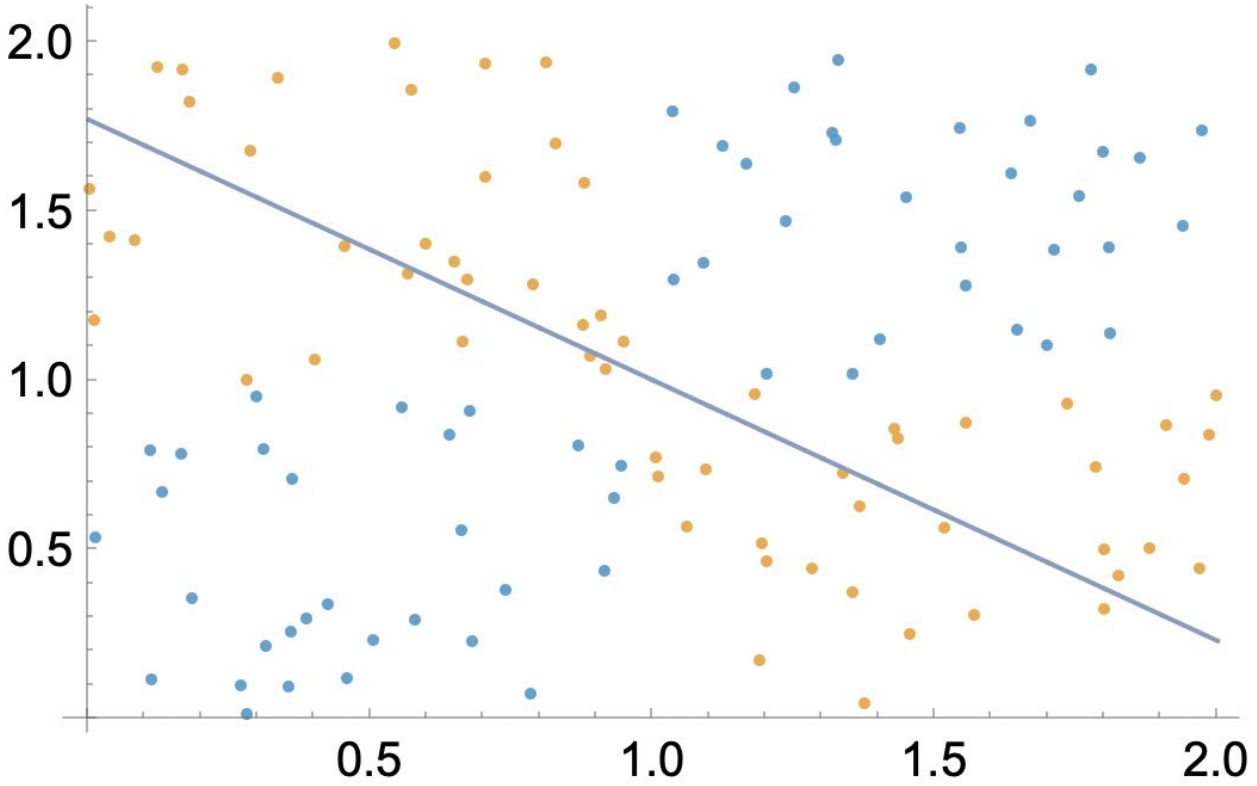
$\simeq 100\%$ 0%

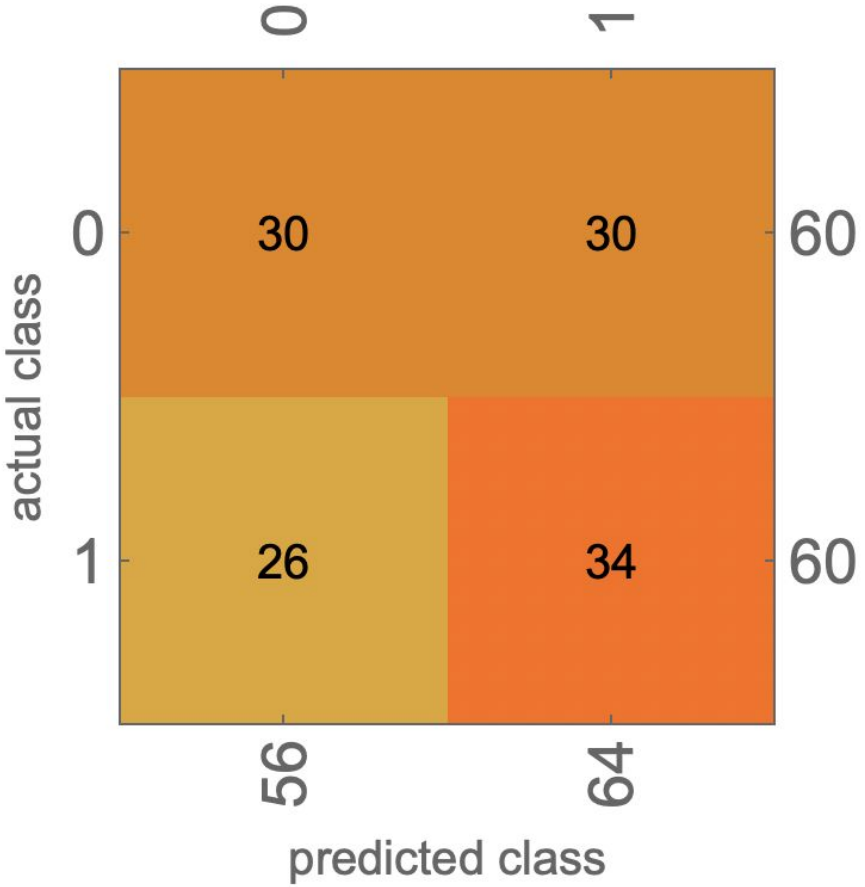
$\simeq 75\%$ 0%

$\simeq 50\%$ 0%

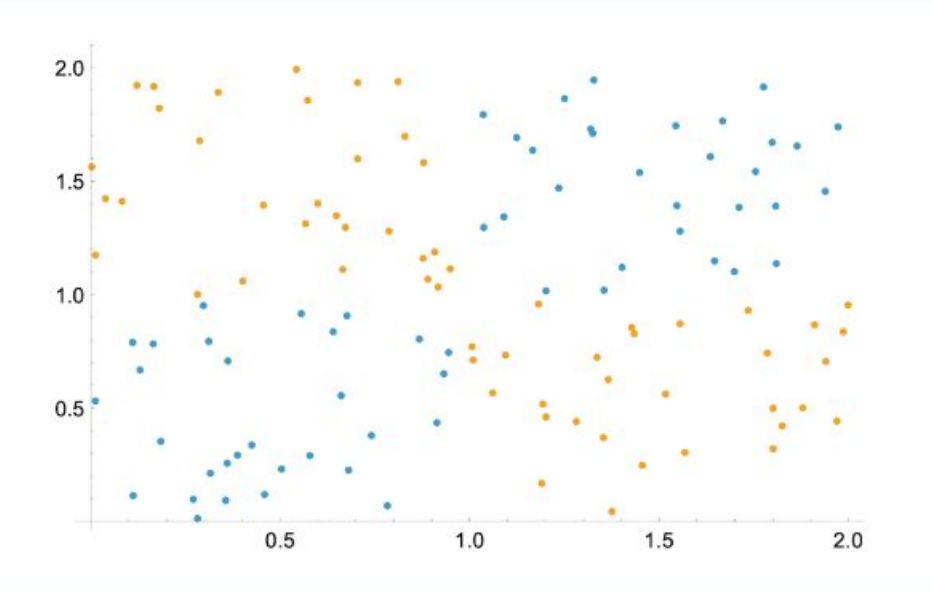
XOR Classification: Logistic Regression

53% of the points predicted correctly





Using logistic regression, what is the expected accuracy of the resulting classification?

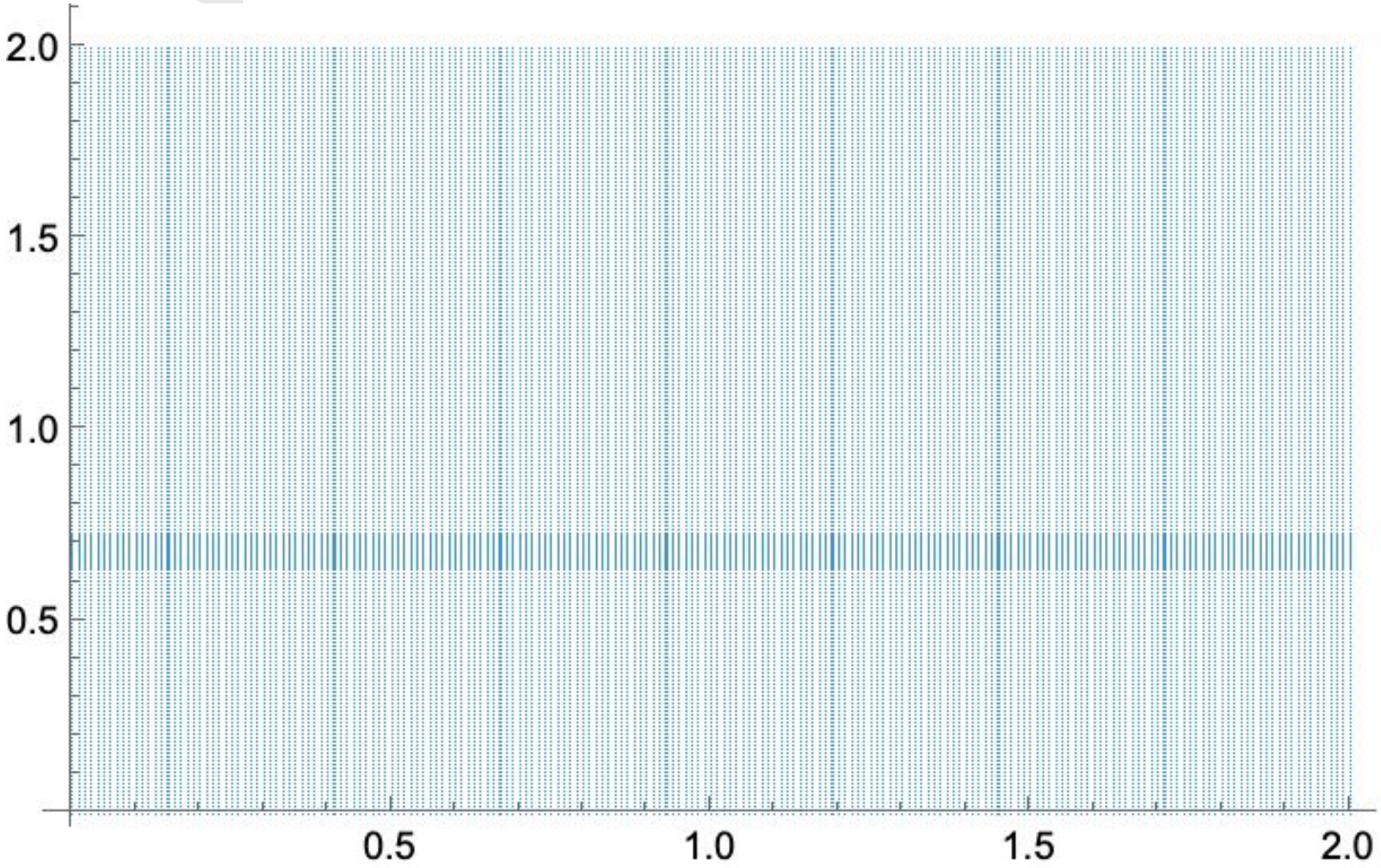


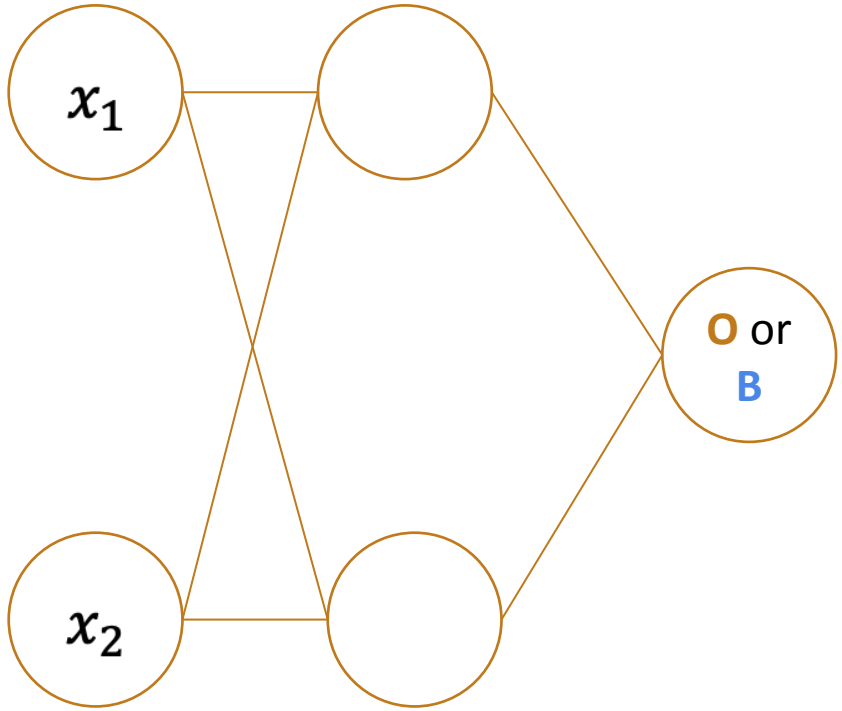
$\simeq 100\%$ 0%

$\simeq 75\%$ 0%

$\simeq 50\%$ 0%

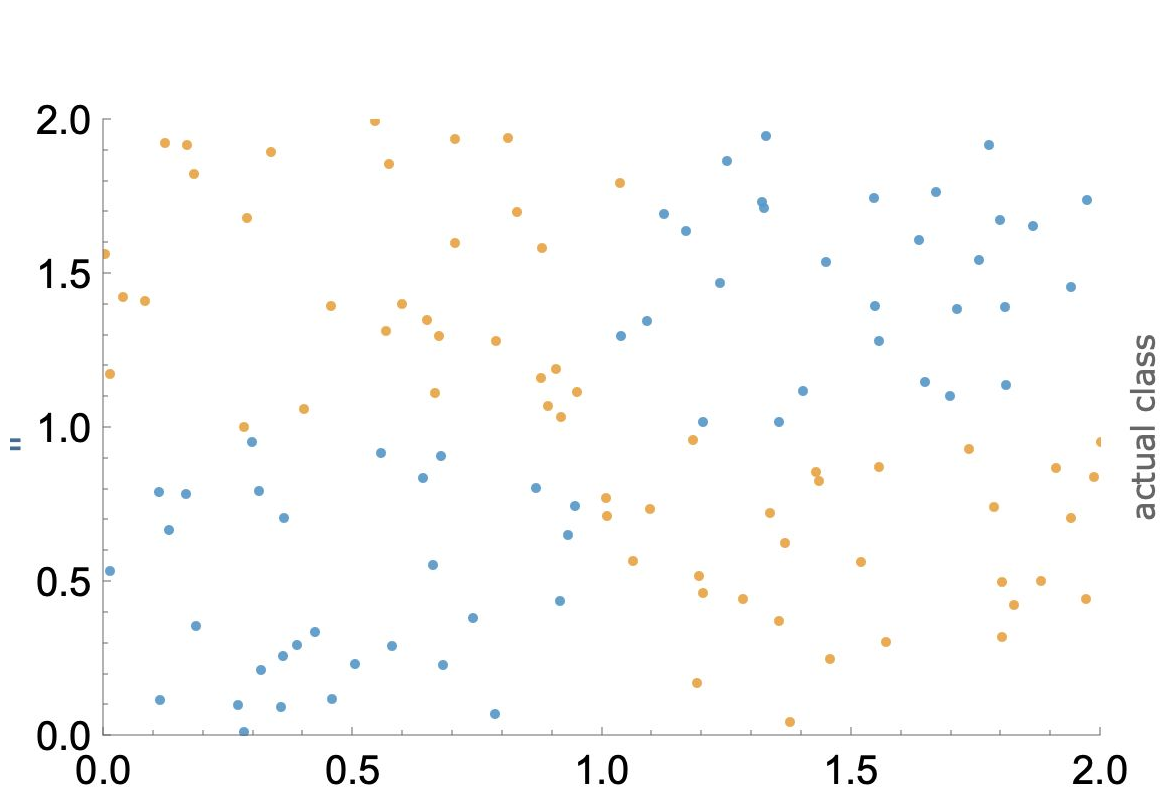
XOR Classification: Neural Network

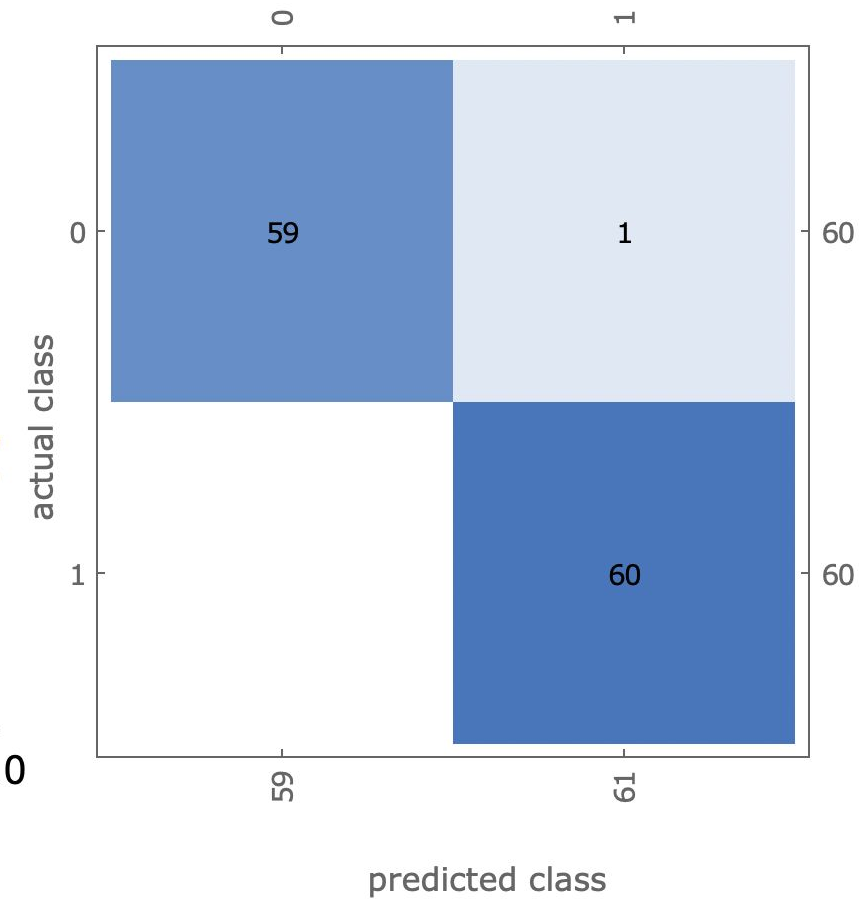




XOR Classification: Neural Network

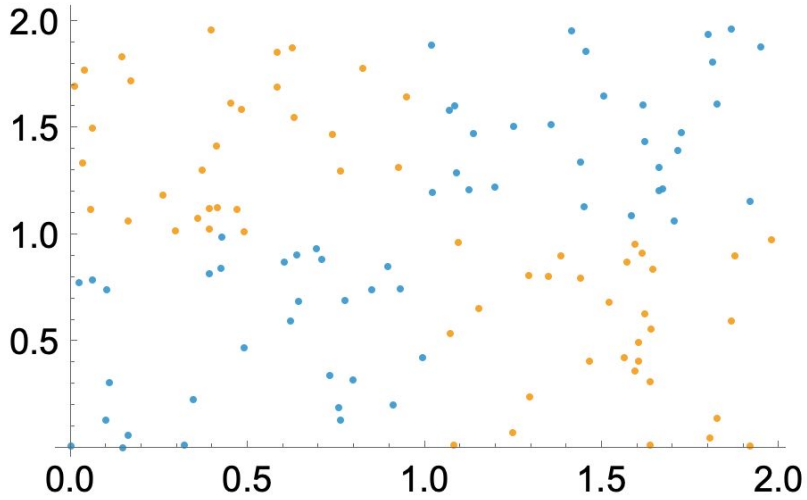
99% of the original points predicted correctly

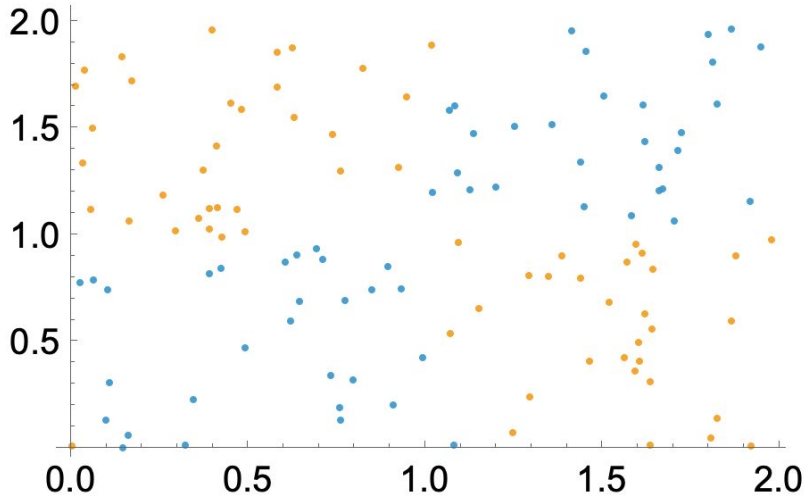


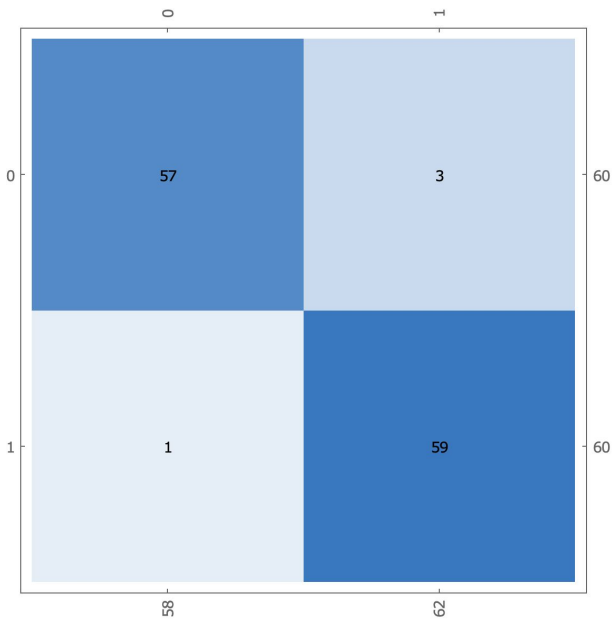


XOR: Test Data (new points)

96% of the original points predicted correctly







Neural Networks Solve Everything

Suppose you have to use a Neural Network to solve a given problem. What questions do you have?

Nobody has responded yet.

Hang tight! Responses are coming in.

Or Do They?

Main Paradigms:

Learning: your NN needs to adapt the needs of the problem you are trying to solve: regression/classification

Design: the structure of a neural network (the number of layers, type of layers, size of layers) should be informed by your data inputs/outputs

Performance: over-simplifying the network can lead to lack of accuracy, over-complicating it can lead to overfitting. **Bias-variance** tradeoff.

Today: Convolutional Neural Networks

🖼️ Computer Vision

- Image classification (medical imaging, face recognition)
- Object detection
- Image segmentation (identifying object boundaries)

🎞️ Video Analysis

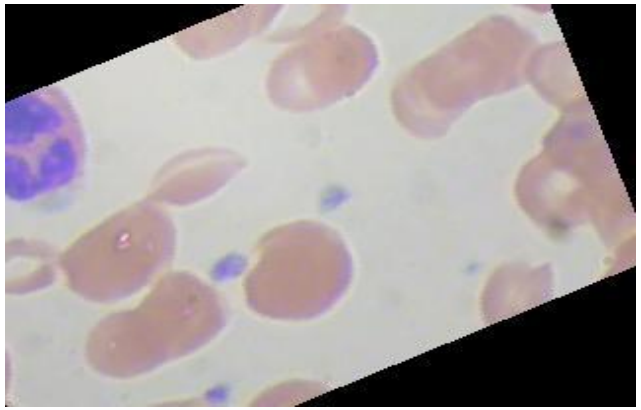
- Action recognition in video sequences
- Video classification or summarization

📝 Other Domains

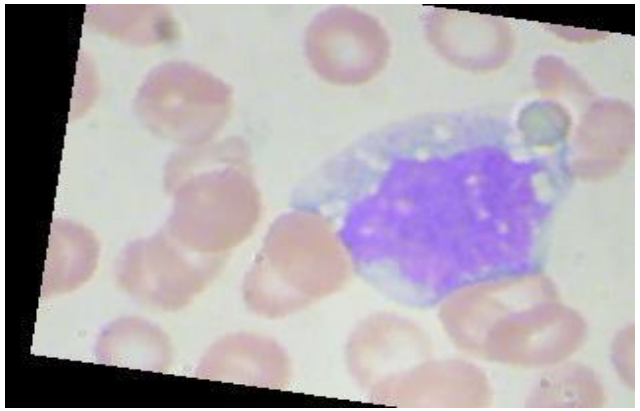
- Text classification (e.g., sentiment analysis using 1D CNNs)
- Audio signal processing (e.g., speech recognition using spectrograms)

Image Classification: Biology

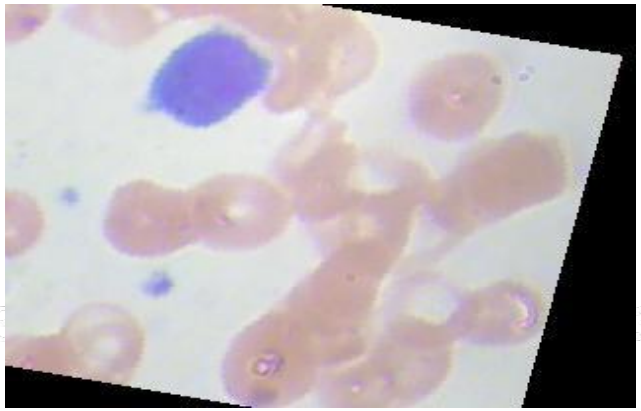
Neutrophil



Eosinophil



Monocyte

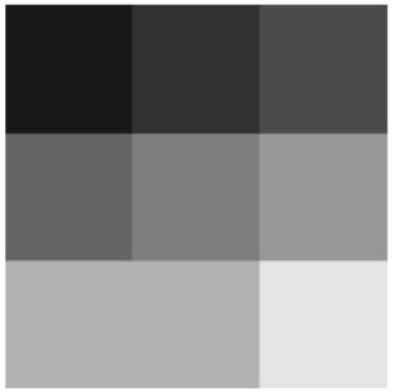


Lymphocyte

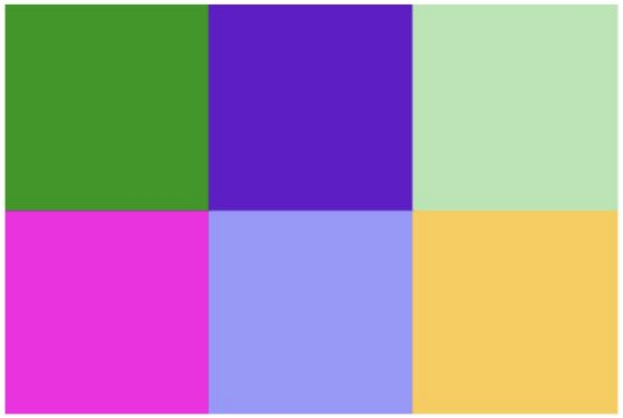
Main Idea:

Images and signals have strong **local structure**: nearby pixels are more related than distant ones.

Digital Images as Matrices



$$\begin{bmatrix} 25 & 51 & 76 \\ 102 & 127 & 153 \\ 178 & 178 & 229 \end{bmatrix}$$

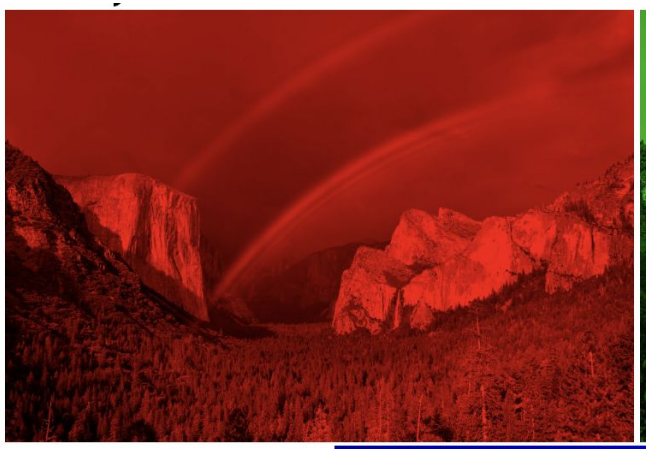


$$\begin{bmatrix} \begin{bmatrix} 0 \\ 153 \\ 0 \end{bmatrix} & \begin{bmatrix} 102 \\ 25 \\ 204 \end{bmatrix} & \begin{bmatrix} 178 \\ 229 \\ 178 \end{bmatrix} \\ \begin{bmatrix} 255 \\ 0 \\ 229 \end{bmatrix} & \begin{bmatrix} 153 \\ 153 \\ 255 \end{bmatrix} & \begin{bmatrix} 255 \\ 204 \\ 76 \end{bmatrix} \end{bmatrix}$$

Images as Matrices: Grayscale

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	8	38	137	146	232	254	255	255	197	109	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	87	197	253	253	253	253	253	253	253	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	120	237	253	253	253	248	209	139	139	230	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	112	229	210	128	96	0	0	0	0	117	253	188	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	0	45	241	245	82	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	253	125	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	32	134	148	98	127	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	7	201	253	200	59	12	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	122	246	253	223	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	233	253	253	236	55	11	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	233	253	253	253	253	210	48	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	68	193	185	243	253	253	173	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	253	253	226	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	253	253	226	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	93	253	253	123	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	14	180	253	253	228	42	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	95	181	253	253	253	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	6	33	33	100	178	253	253	253	241	88	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	32	236	253	253	253	253	253	228	122	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	143	253	253	253	154	76	24	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Images as Matrices: RGB



Images as Matrices: YCbCr (Adversarial)

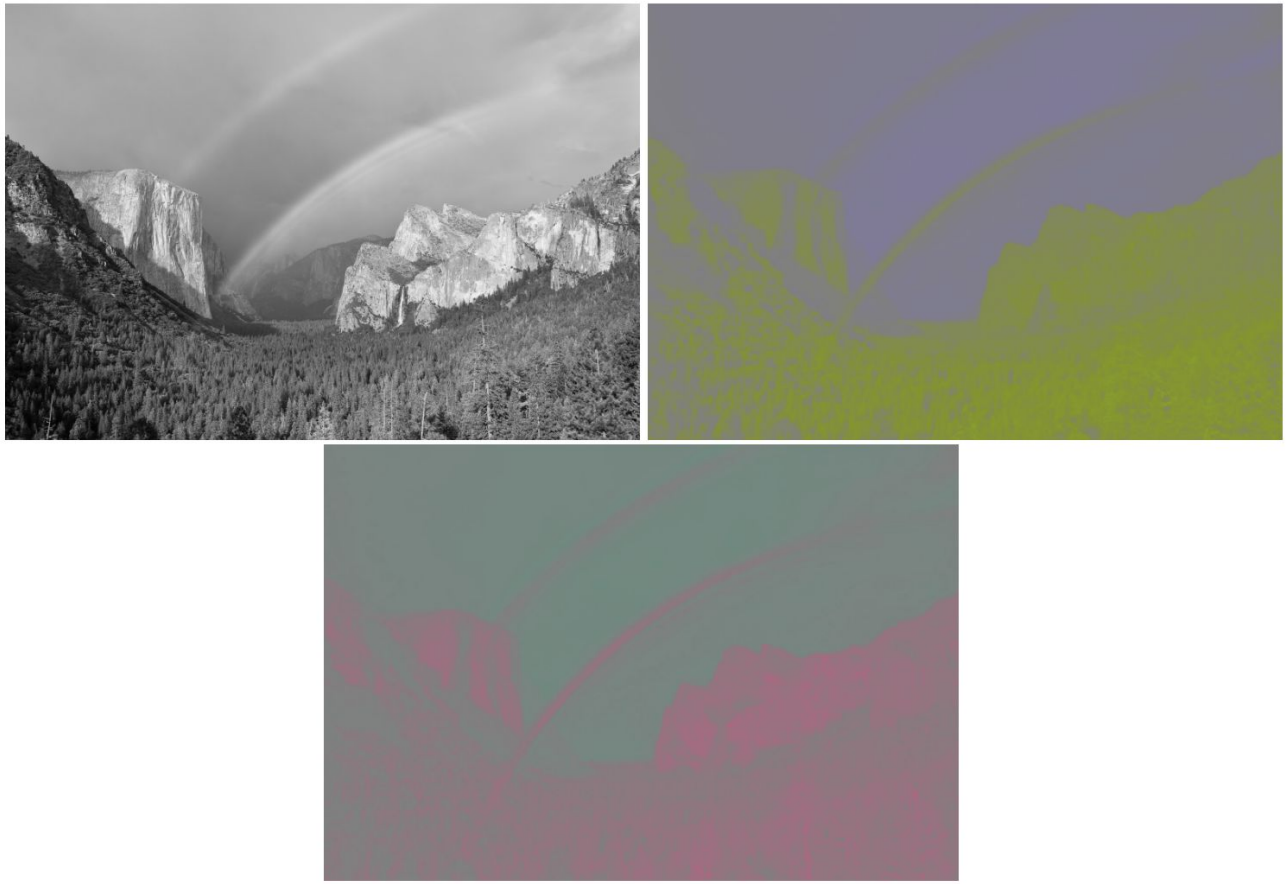


Image Processing: Filters



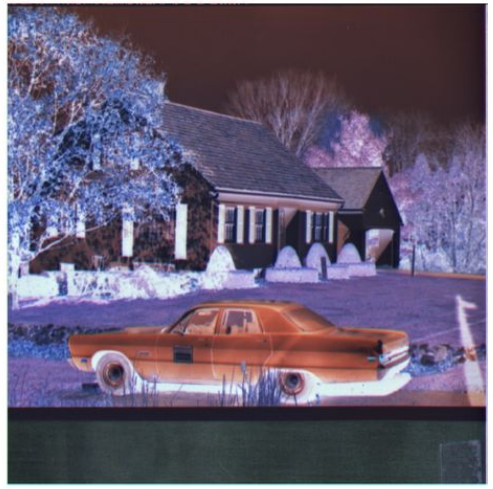



Image Processing

(R > G + B and B < 0.2)

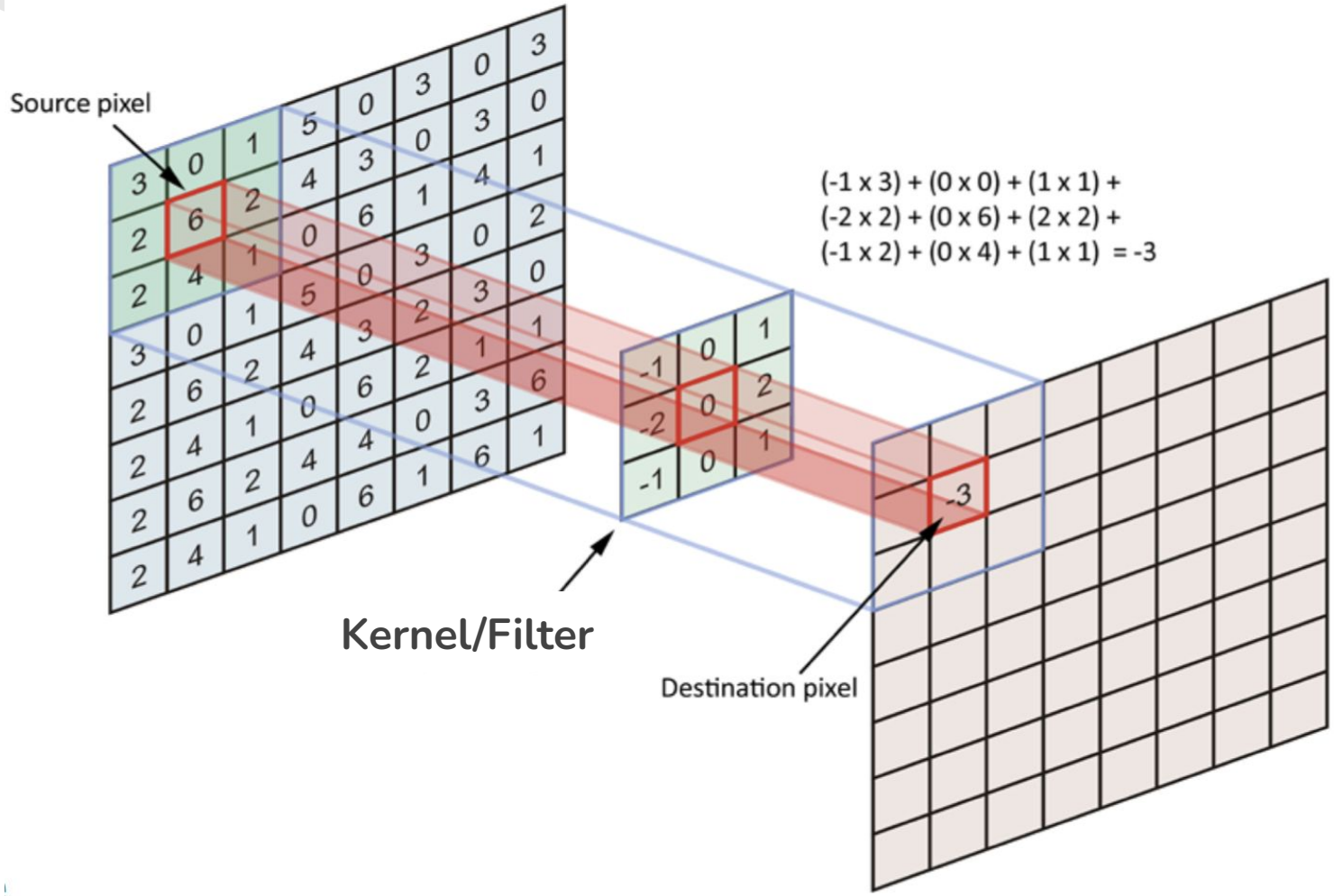
or

(B > R + G and B > 2G)

Image Processing

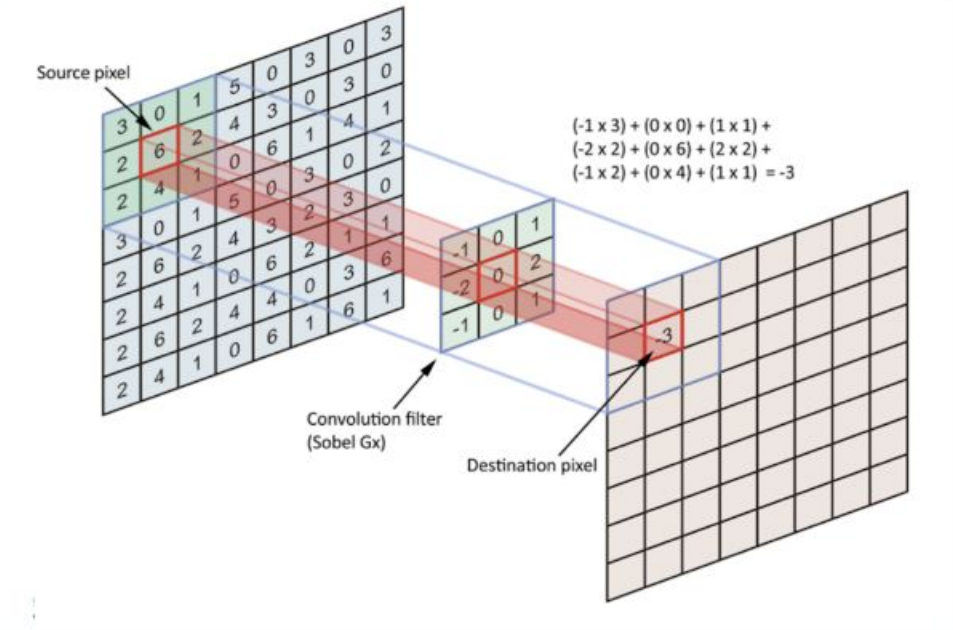
$$\begin{pmatrix} 0.393 & 0.769 & 0.189 \\ 0.349 & 0.686 & 0.168 \\ 0.272 & 0.534 & 0.131 \end{pmatrix}$$

Image Convolutions



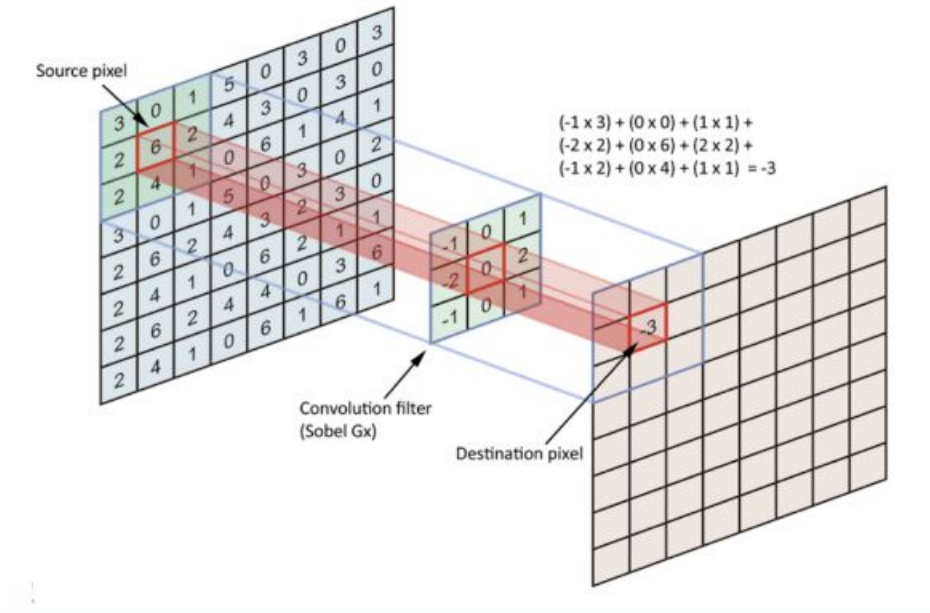
Convolutions treat pixels that are far apart as **independent**, focusing on the **local structure**.

What is the value of the resulting pixel to the RIGHT of -3?



- -3
- 3
- -8
- 4

What is the value of the resulting pixel to the RIGHT of -3?



-3

0%

3

0%

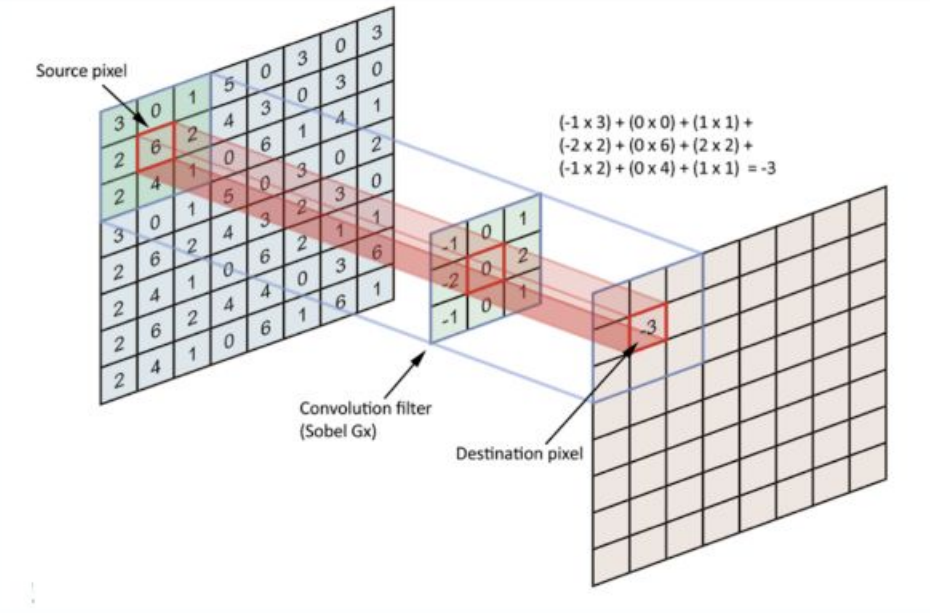
-8

0%

4

0%

What is the value of the resulting pixel to the RIGHT of -3?



-3

0%

3

0%

-8

0%

4

0%

Image Convolutions

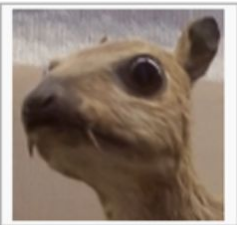
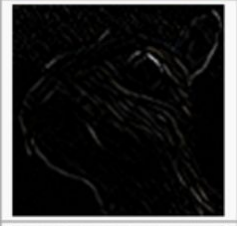
Kernel/Filter

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

Kernel/Filter

	0	1	0	
	1	-4	1	
	0	1	0	

Image Convolutions

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

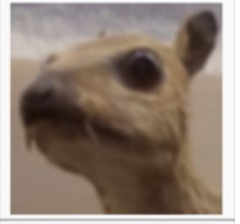
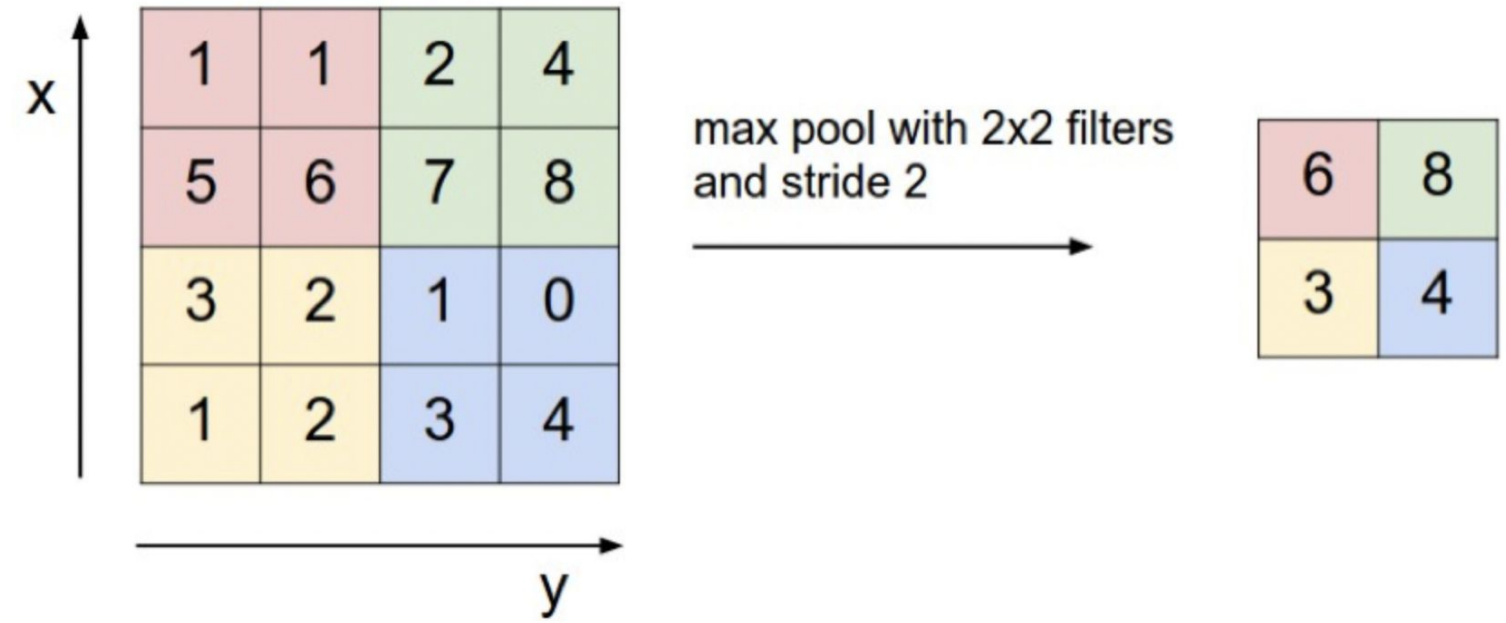
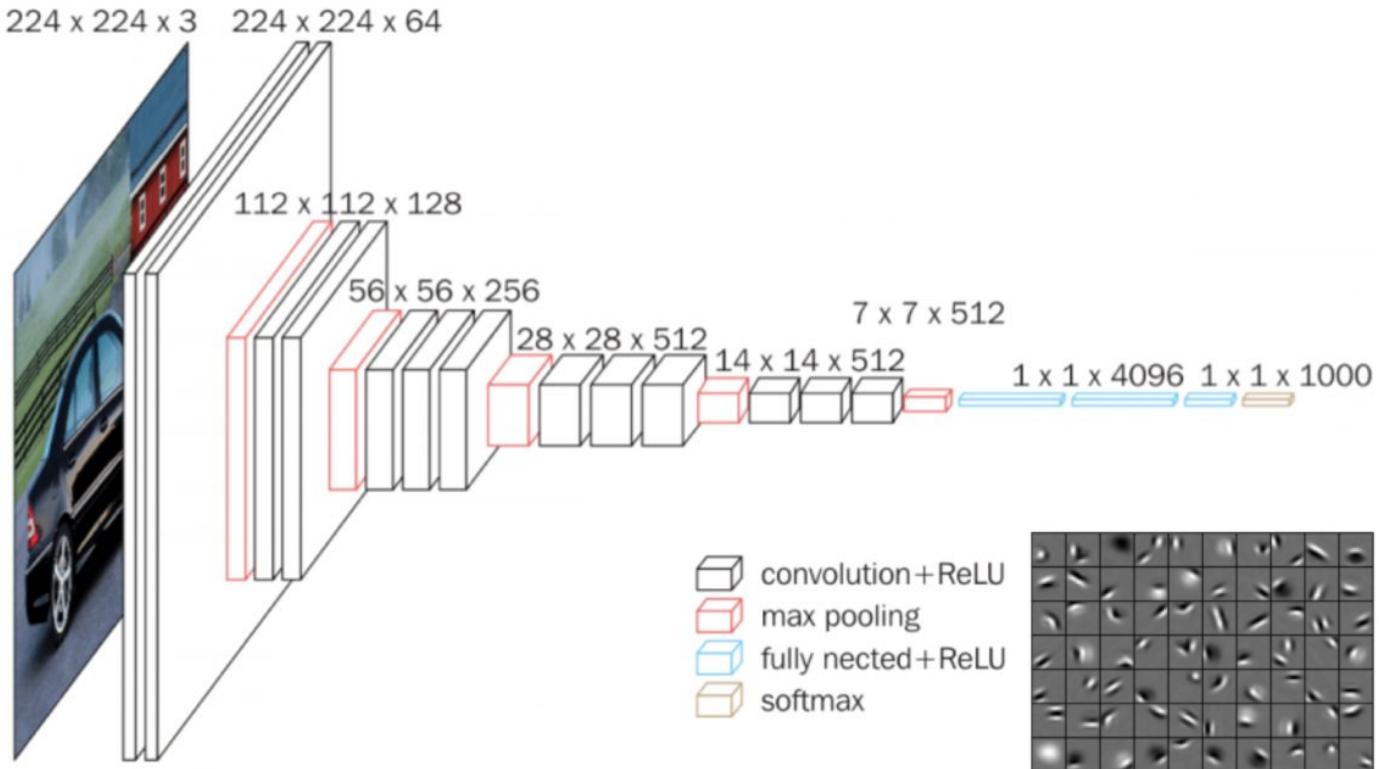
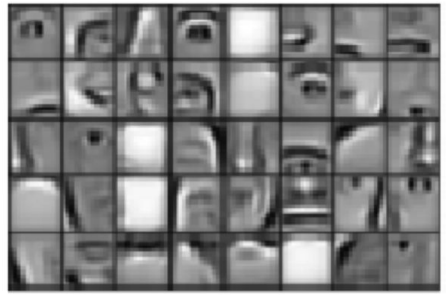
Operation	Filter	Convolved Image
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Image Pooling



Convolutional neural networks (CNNs)





Güngör, Cengiz, and Kenan Zengin. "GE-International Journal of Engineering Research."

Convolutional neural networks (CNNs)

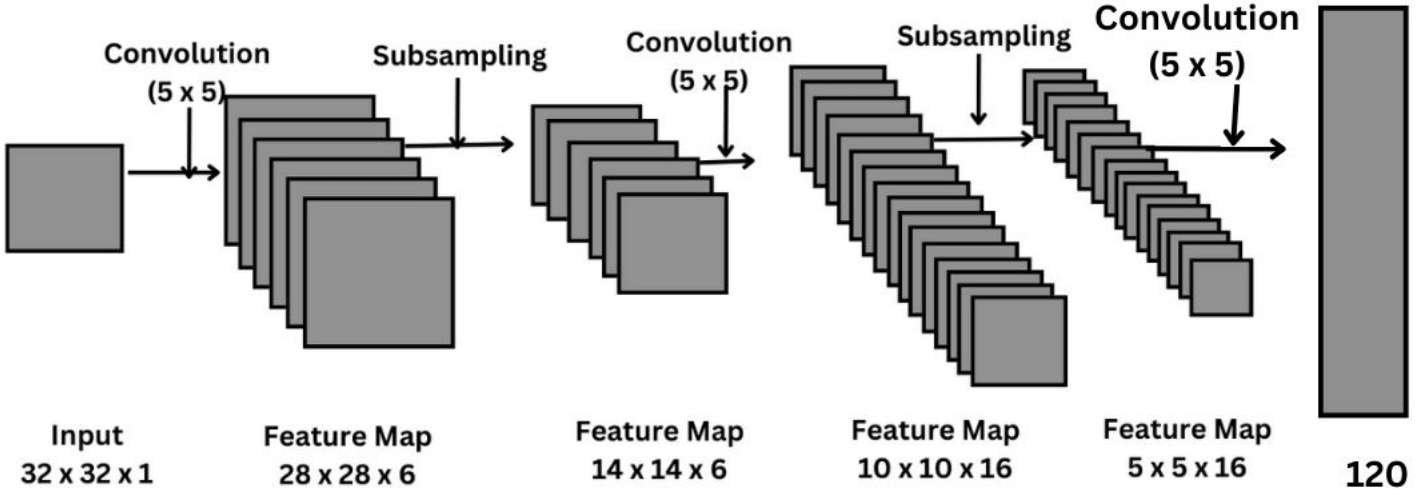
- Stack of convolutional layers, pooling layers and fully connected layers.
- The values of the kernels **are learnt**

Purposes

- **Convolutional layers** create feature maps that summarize the presence of those features in the input
- The output is typically **lower resolution** yet still contains the large or important structural elements. Typically convolutional layers have **several channels**
- **Pooling layers** down sample feature maps by summarizing the presence of features in patches

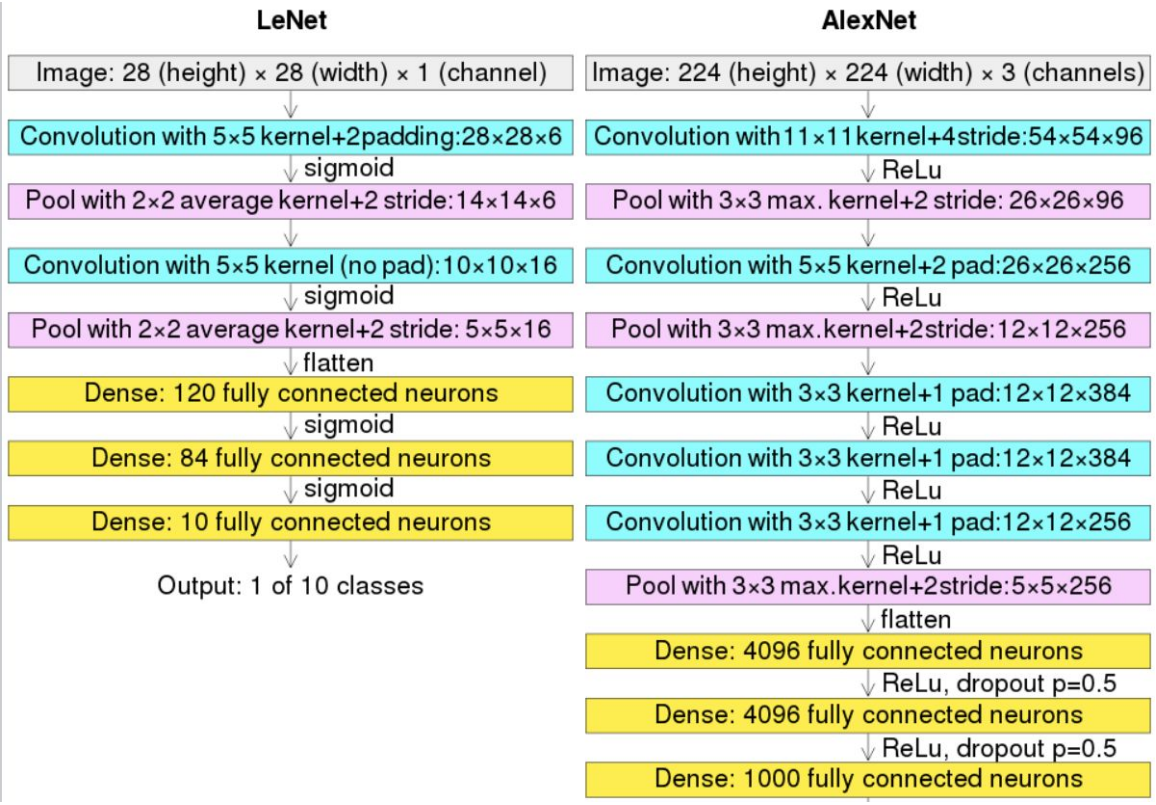
A Famoud CNN: LeNet 5

- CNN structure proposed by Yann Lecun in 1998
- LeNet-5 network has 5 layers (hence its name)
- 3 sets of convolution layers with a combination of average pooling, followed by 2 fully connected layers with a Softmax classifier



Samat, Nurul Ashikin, Mohd Najib Mohd Salleh, and Haseeb Ali. "The comparison of pooling functions in convolutional neural network for sentiment analysis task." *Recent Advances on Soft Computing and Data Mining: Proceedings of the Fourth International Conference on Soft Computing and Data Mining (SCDM 2020), Melaka, Malaysia, January 22– 23, 2020*. Springer International Publishing, 2020.

LeNet-5



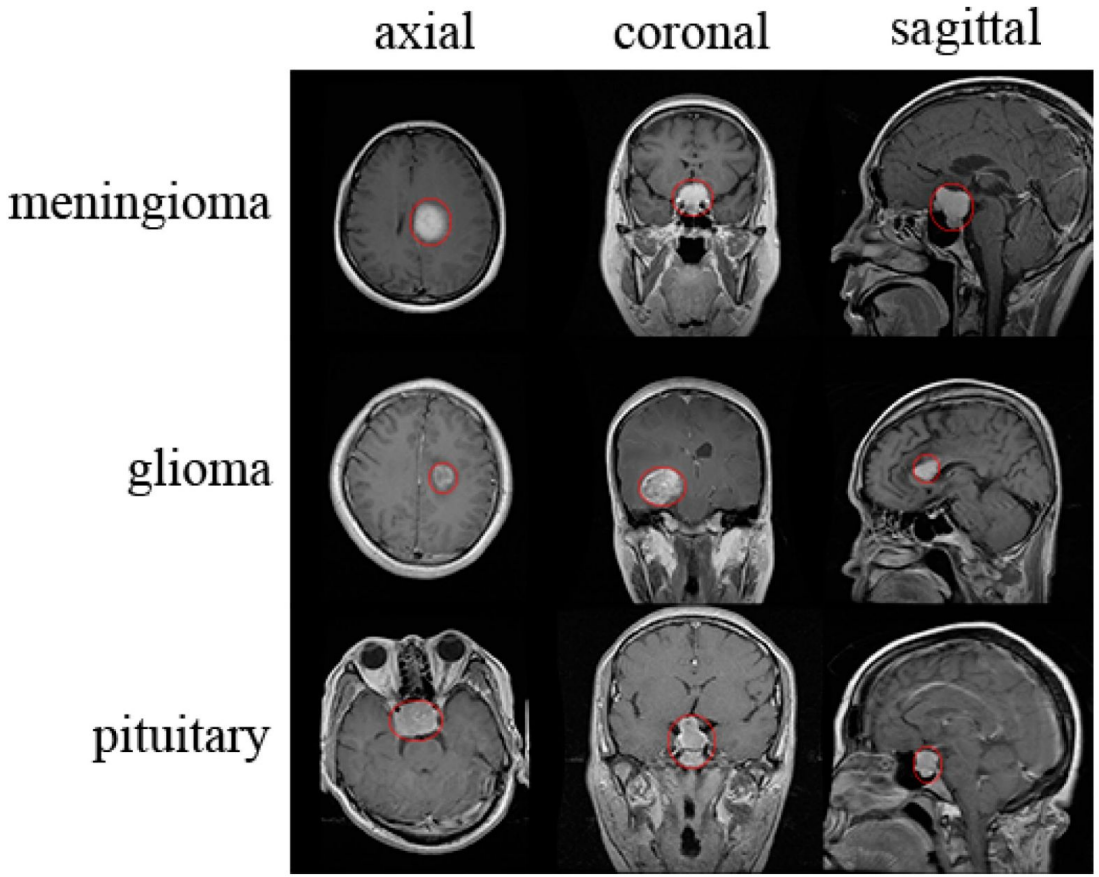
Layer (type)	Output Shape	Param #
conv2d_15 (Conv2D)	(None, 28, 28, 6)	456
max_pooling2d_8 (MaxPooling 2D)	(None, 14, 14, 6)	0
conv2d_16 (Conv2D)	(None, 10, 10, 16)	2416
max_pooling2d_9 (MaxPooling 2D)	(None, 5, 5, 16)	0
conv2d_17 (Conv2D)	(None, 1, 1, 120)	48120
flatten_2 (Flatten)	(None, 120)	0
dense_3 (Dense)	(None, 84)	10164
dense_4 (Dense)	(None, 10)	850

Total params: 62,006
Trainable params: 62,006
Non-trainable params: 0

Summary: Why use CNNs?

- **Local connectivity:** exploit the spatial structure of data through the local patterns
- **Translational invariance:** can detect patterns independently of their location in the input
- **Parameter efficient:** less parameters than fully connected
- **Hierarchical feature learning:** initial layers of the network learn simple and low-level features such as edges and textures, while subsequent layers learn more complex and high-level features

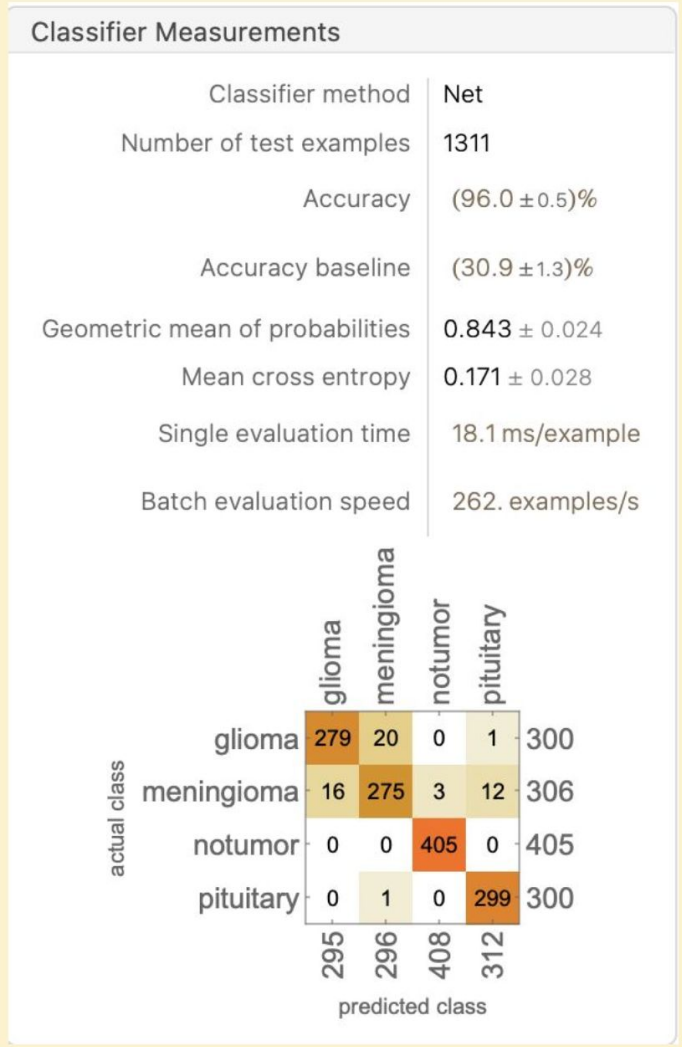
Simple Example: Brain Tumor Classification



Question: Can we identify different types of tumor from brain scans?

Simple Example: Brain Tumor Classification

```
[1]:= tumor = NetChain[{
        ConvolutionLayer[20, {3, 3}],
        Ramp,
        PoolingLayer[2, 2],
        ConvolutionLayer[10, {3, 3}],
        Ramp,
        PoolingLayer[2, 2],
        FlattenLayer[],
        LinearLayer[300],
        Ramp,
        LinearLayer[100],
        Ramp,
        LinearLayer[4],
        SoftmaxLayer[],
      },
      "Input" → encode,
      "Output" → decode]
```



96% accuracy, good or bad?

(Project done by high-school students in Harvard Summer School)

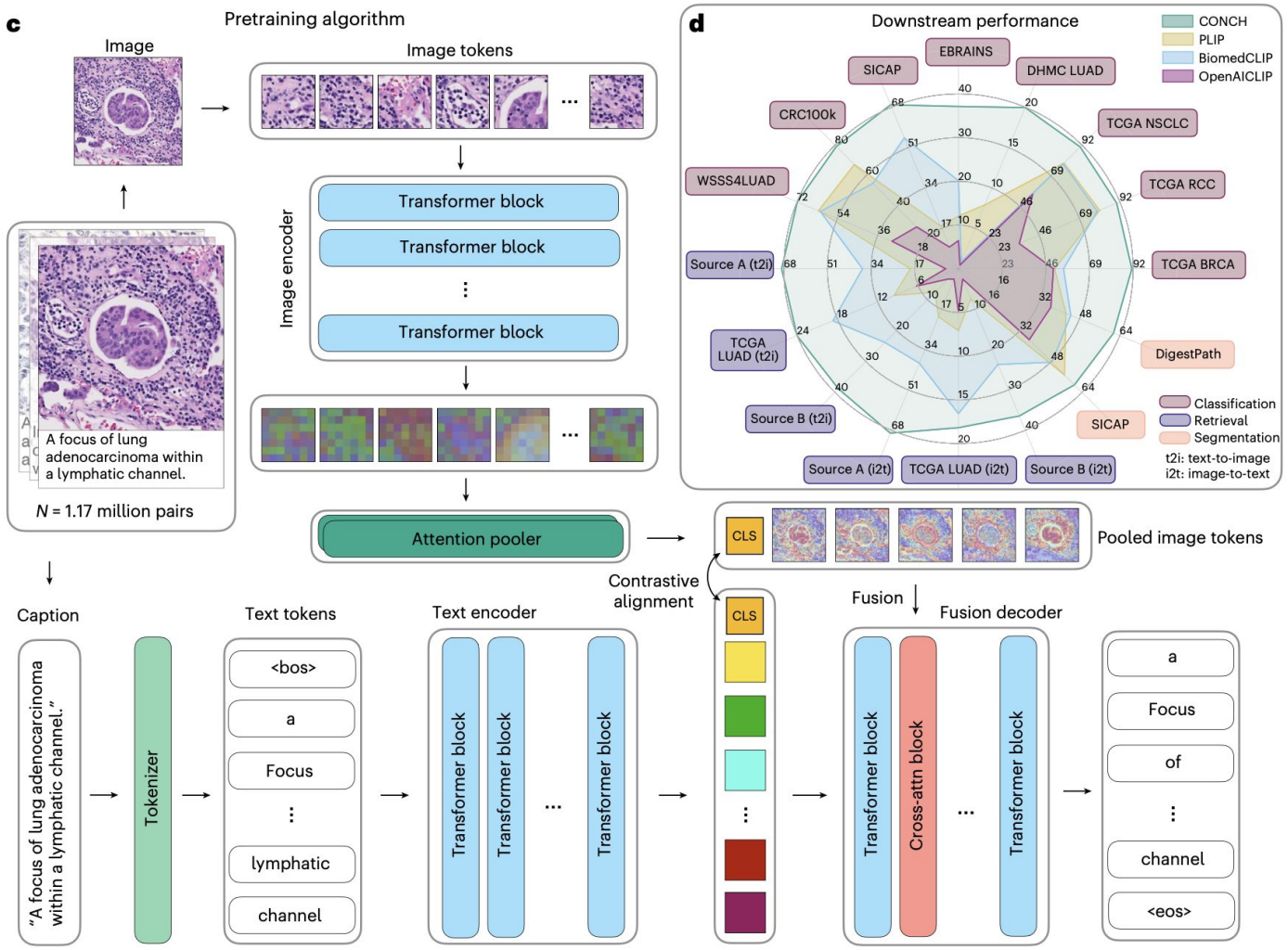
(Dataset available from Kaggle)

Research Example: UNI/CONCH

*"We introduce CONtrastive learning from Captions for Histopathology (CONCH), a **visual-language foundation model** developed using diverse sources of histopathology images, biomedical text and, notably, **over 1.17 million image–caption pairs** through task-agnostic pretraining. Evaluated on a suite of 14 diverse benchmarks, CONCH can be transferred to a wide range of downstream tasks involving histopathology images and/or text, achieving state-of-the-art performance on histology **image classification, segmentation, captioning, and text-to-image and image-to-text retrieval**."*

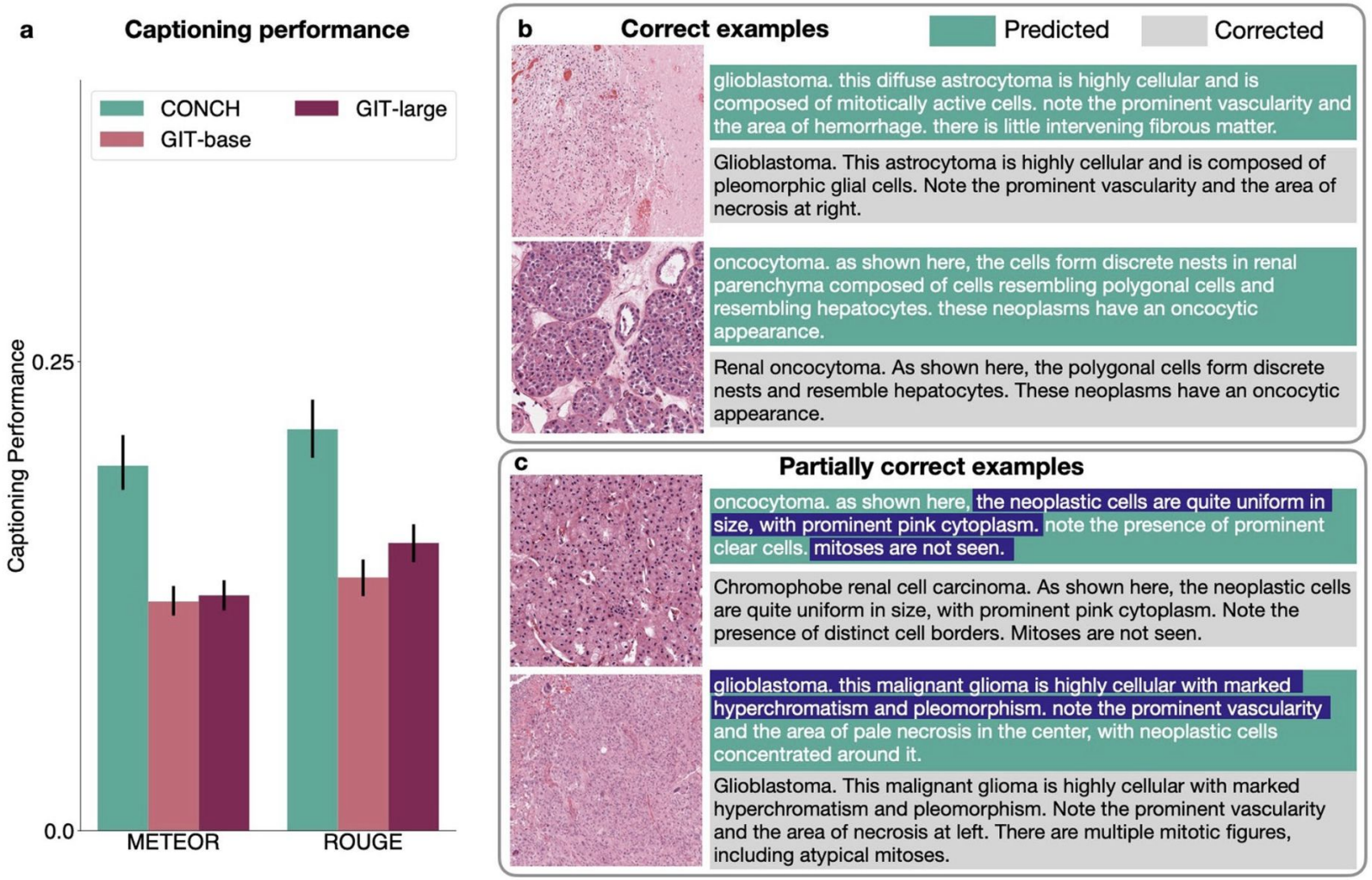
Lu, M.Y., Chen, B., Zhang, A., Williamson, D.F., Chen, R.J., Ding, T., Le, L.P., Chuang, Y.S. and Mahmood, F., 2023. Visual Language Pretrained Multiple Instance Zero-Shot Transfer for Histopathology Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19764-19775).

Research Example: UNI/CONCH



Structure of their model

Research Example: UNI/CONCH



Example of performance

Images sources

- Cat: https://www.goodhousekeeping.com/life/pets/a43276342/cat-instagram-captions/
- Turtle: https://en.wikipedia.org/wiki/Sea_turtle
- VGG-16 achitecture: https://pytorch.org/TensorRT/_notebooks/vgg-qat.html#1
- LeNet/Alexnet architectures: https://upload.wikimedia.org/wikipedia/commons/c/cc/Comparison_image_neural_networks.svg