Redes Neuronales Convolucionales

Siham Tabik

UNIVERSIDAD DE GRANADA

Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
siham@ugr.es

DaSCI

Outline

1. What are CNNs?
2. How CNNs work?
   - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. Overfitting/underfitting
4. Transfer learning
5. Data augmentation
6. Regulation techniques
7. Interpretability

By the way, what is image classification?

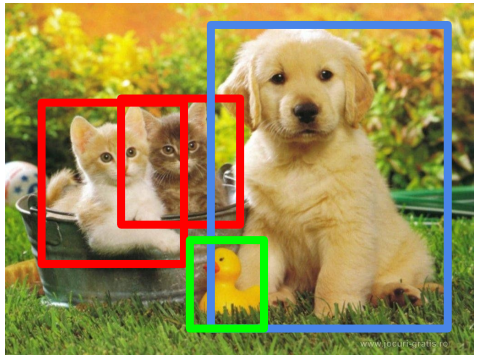
Classification

CAT

Classification + Localization

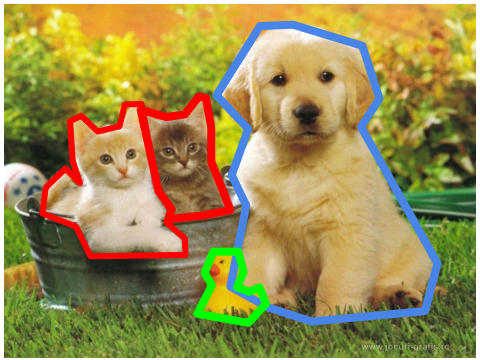
CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation

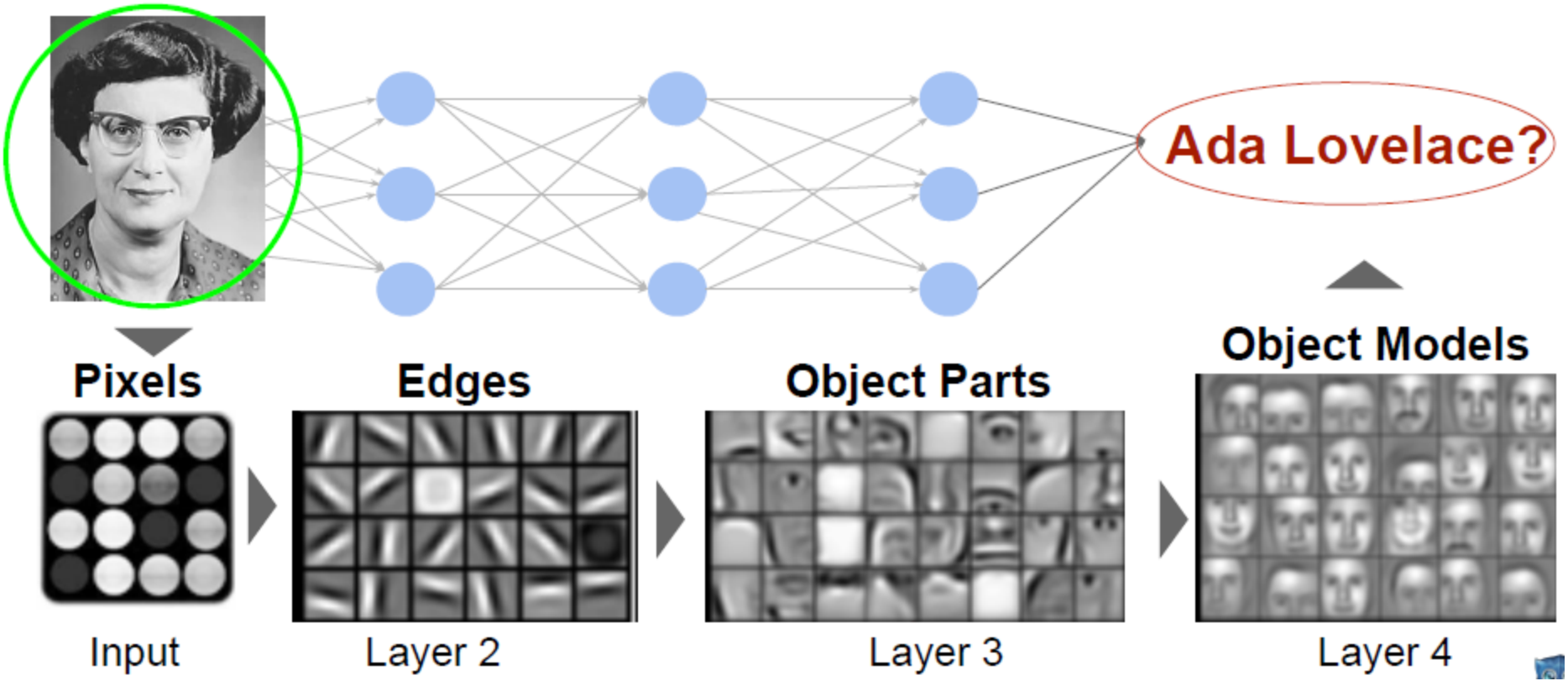


CAT, DOG, DUCK

Single object

Multiple objects

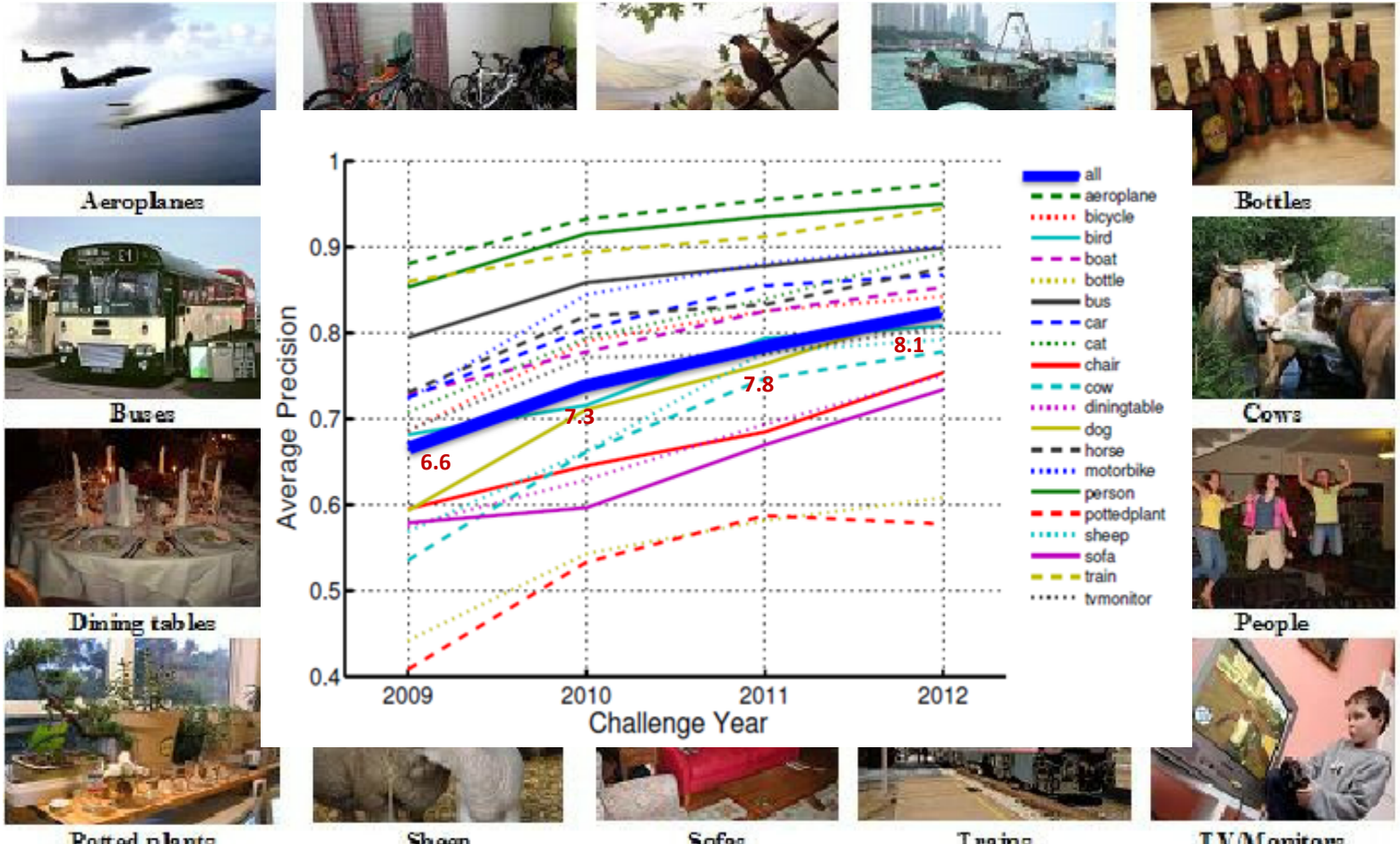
Convolutional Neural Networks



PASCAL Visual Object Classes Challenge (2005-2012)

- Database: Public dataset of 10,103 images& 20 object classes
- Annual Competition: PASCAL VOC Challenge
- Networks: The most accurate net wins the challenge

PASCAL Visual Object Classes Challenge

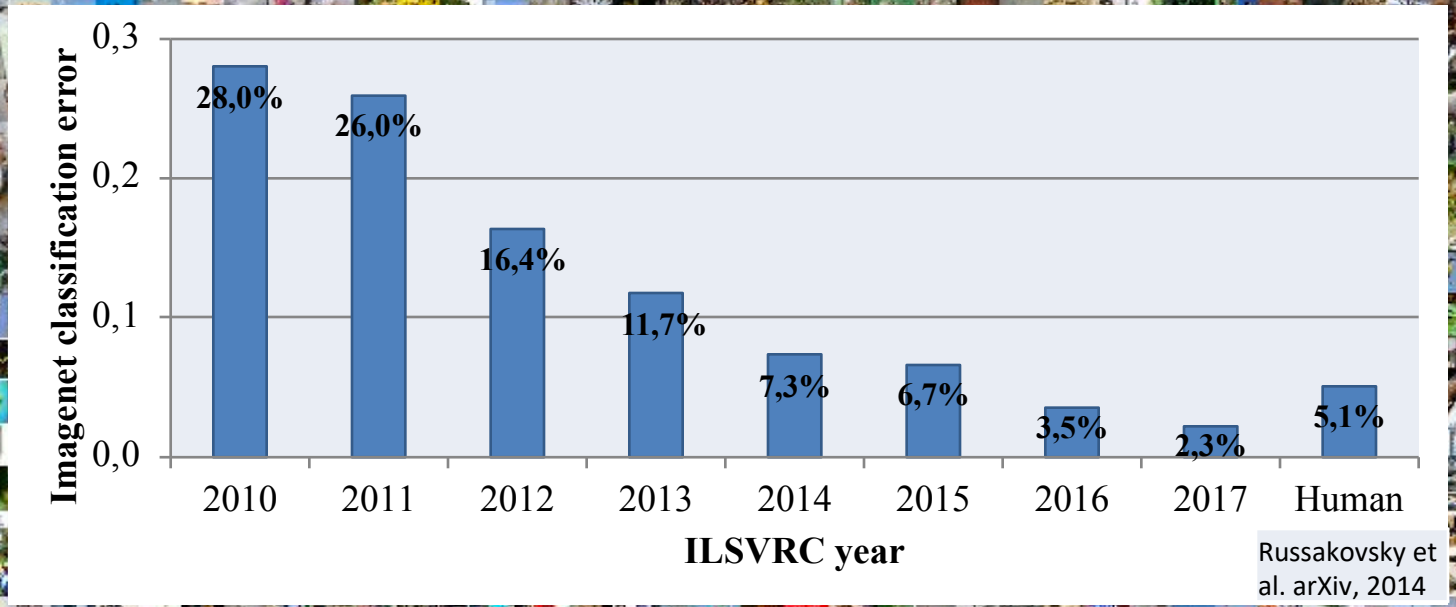


IMAGENET image classification challenge

- Database: Public dataset of 14,197,122 images & 21,841 classes
- Annual Competition: The Large Scale Visual Recognition Challenge (ILSVRC)
  - **Classification task**: 1,431,167 images & 1000 classes (1.2M train+100.000 test)
  - **Detection task, segmentation task**, …
- Networks: The most accurate net win the challenge

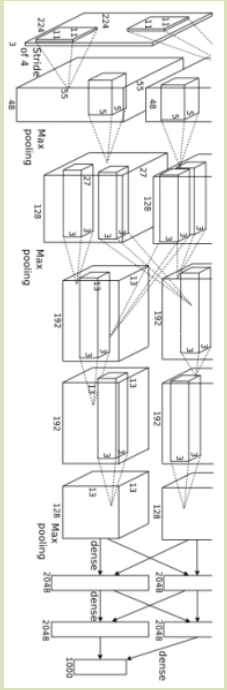
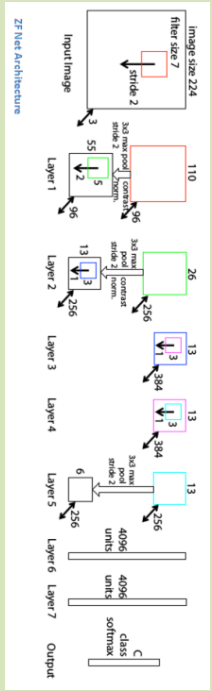
Imagenet: A large-scale hierarchical image database J Deng, W Dong, R Socher, LJ Li, K Li, L Fei-Fei (2009)

IMAGENET image classification challenge

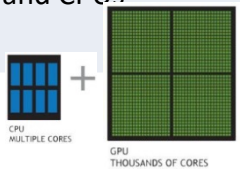


Russakovsky et al. arXiv, 2014

IMAGENET image classification challenge

Year	**2012**	**2013**	**2014**		**2015**	**2016**
	AlexNet	ZFnet	GoogLeNet	VGG	ResNet	Inception&ResNet
	8 layers	8 layers	19 layers	22 layers	152 layers	
						
	Krizhevsky et al. NIPS	Zeiler and Fergus	Frizhevsky et al. arxiv &	simonyan et al. arxiv	He et al CVPR	

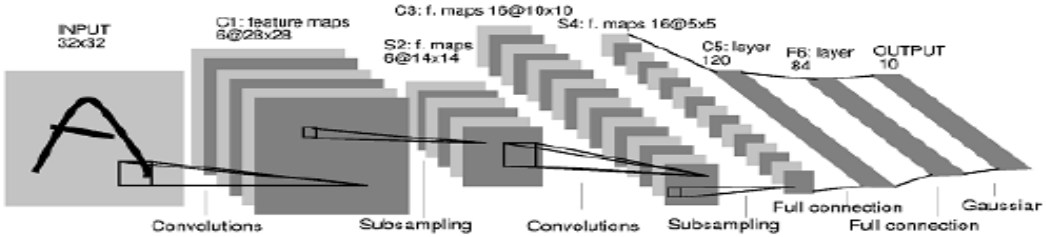
Evolution of CNNs

	1975	1985	1998	2009	2012
CNNs	**1st implementation of Conv. & pooling layers** by K. Fukushima	**1st CNN trained with Backpropagation**	**1st CNN trained with backp.** on MNIST: **LeNet** (0.8% error) by Y. LeCun		Won the ILSVRC classification challenge
Learning Algorithms	Costly	Backpropagation	Backpropagation	Backpropagation	Backpropagation
Database	Small	Small	MNIST	1st massive dataset **ImageNet** 	
Hardware	Limited capacity	Limited capacity	Limited capacity	Powerful GPUs and CPUs 	

CNNs architecture

1998

LeCun et al.



# of transistors

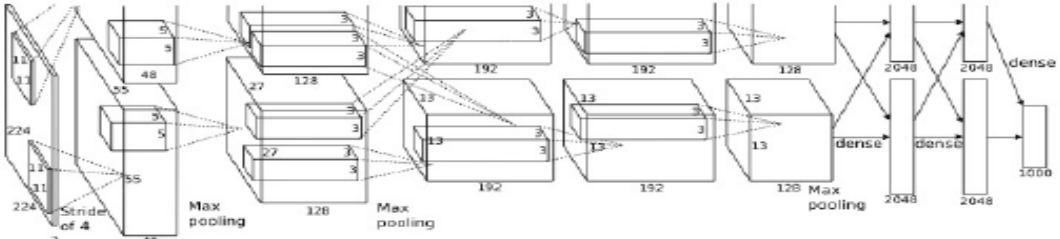
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors GPUs

10^9

# of pixels used in training

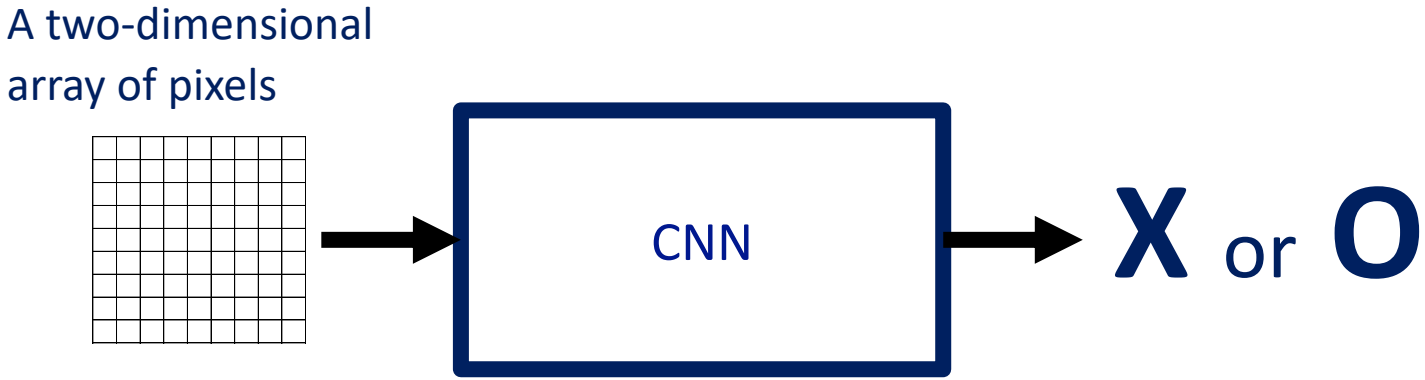
10^{14} IMAGENET

Outline

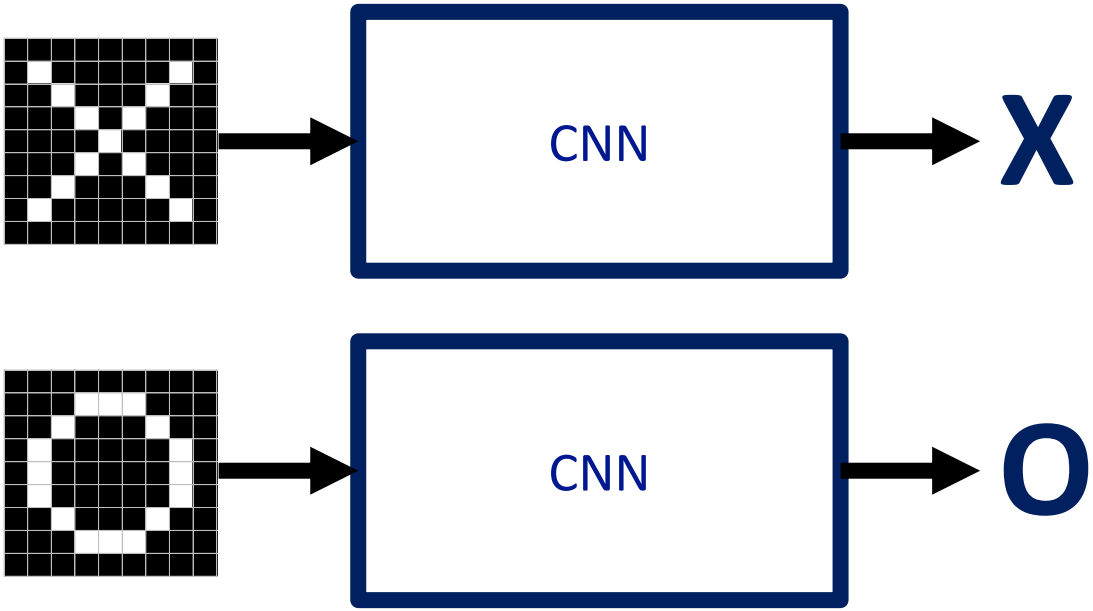
1. What are CNNs?
2. **How CNNs work?**
   - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. Overfitting/underfitting
4. Transfer learning
5. Data augmentation
6. Regulation techniques
7. Interpretability

How CNNs work?

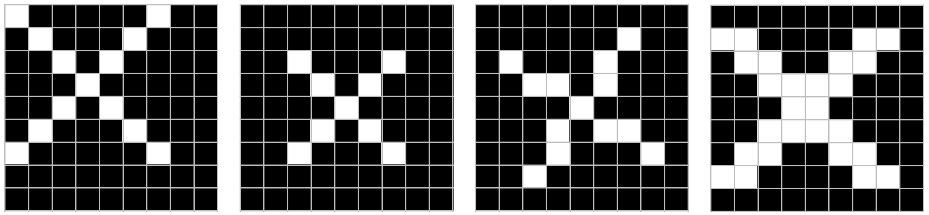
A toy CNN: says whether a picture is of an X or an O



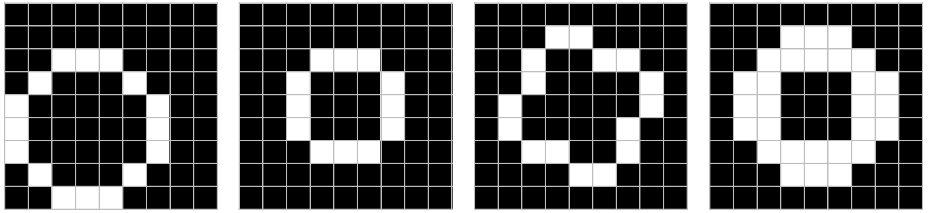
For example

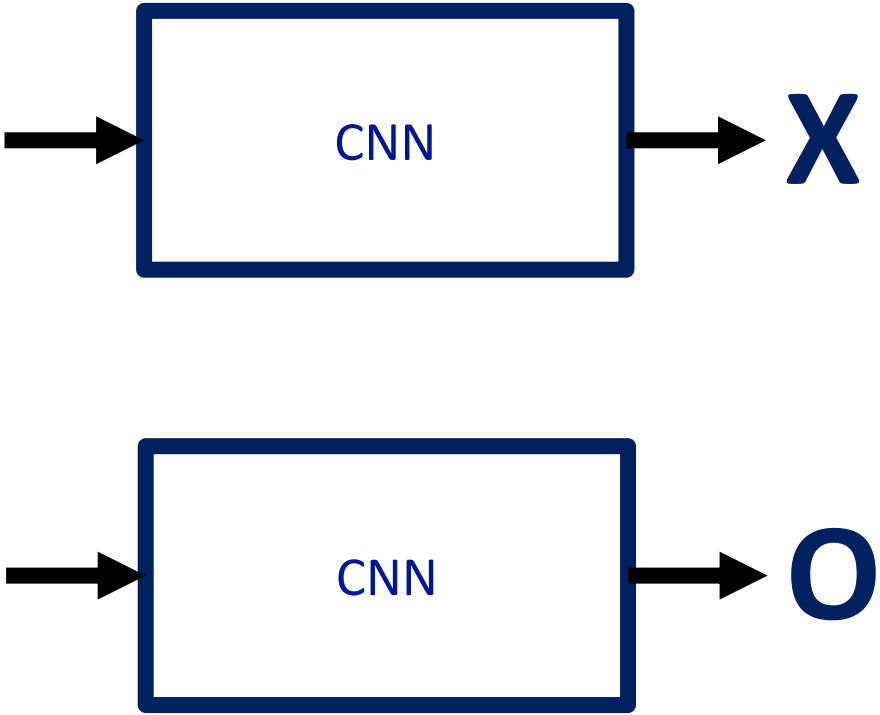


Trickier cases



translation scaling rotation weight





Deciding is hard

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

What computers see

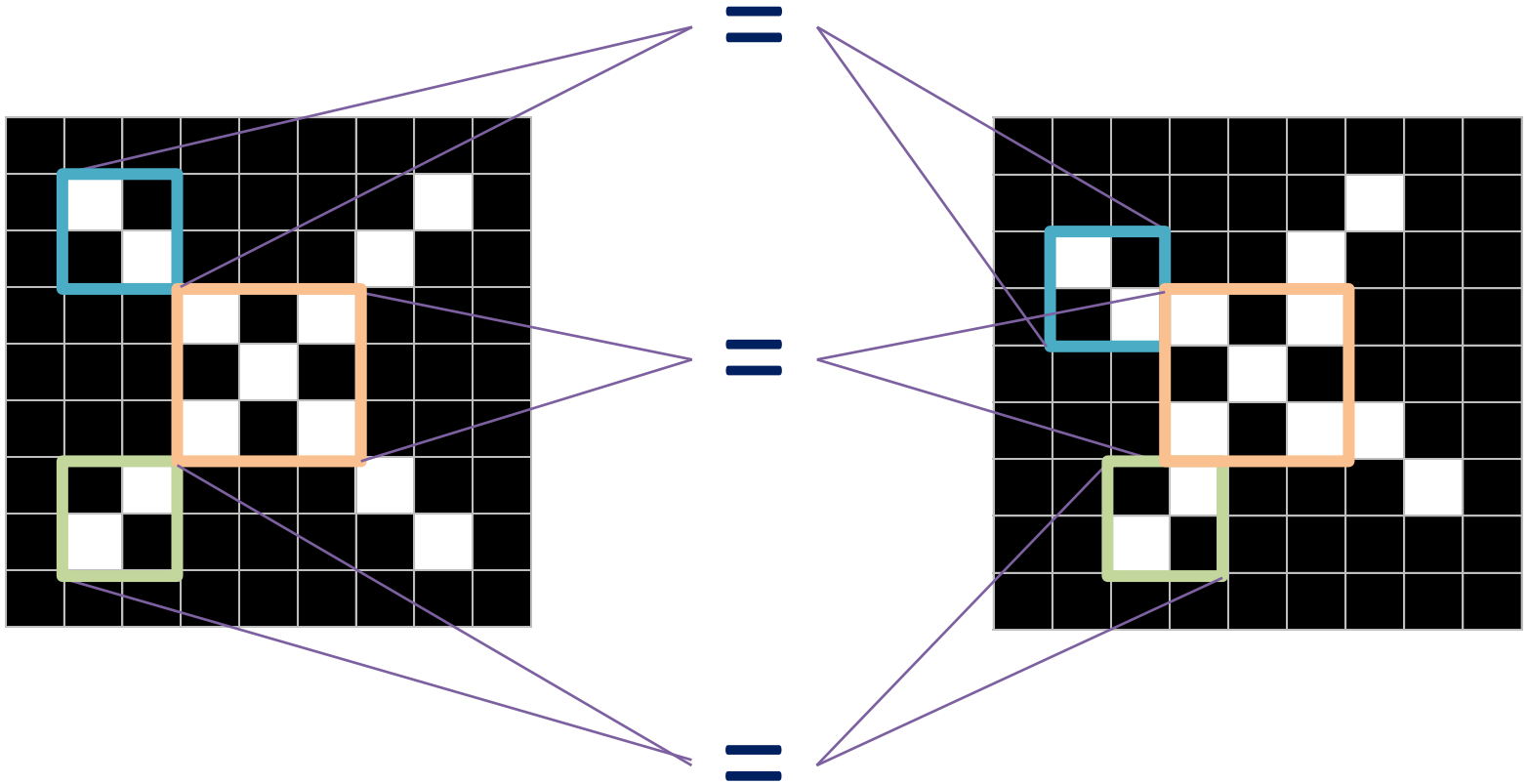
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	X	-1	-1	-1	-1	X	X	-1
-1	X	X	-1	-1	X	X	-1	-1
-1	-1	X	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	X	-1	-1
-1	-1	X	X	-1	-1	X	X	-1
-1	X	X	-1	-1	-1	-1	X	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

Computers are literal

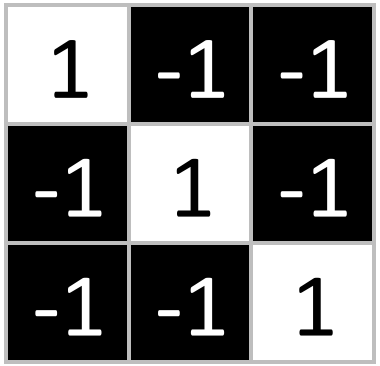
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

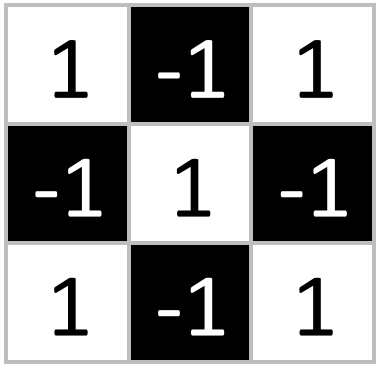
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	1	-1	-1
-1	-1	-1	1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

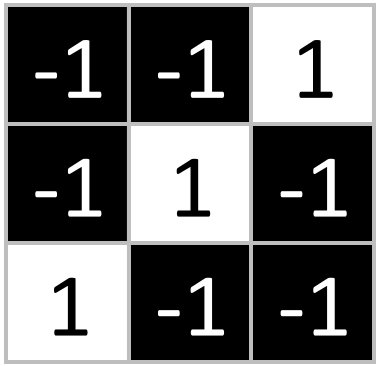
CNNs match pieces of the image

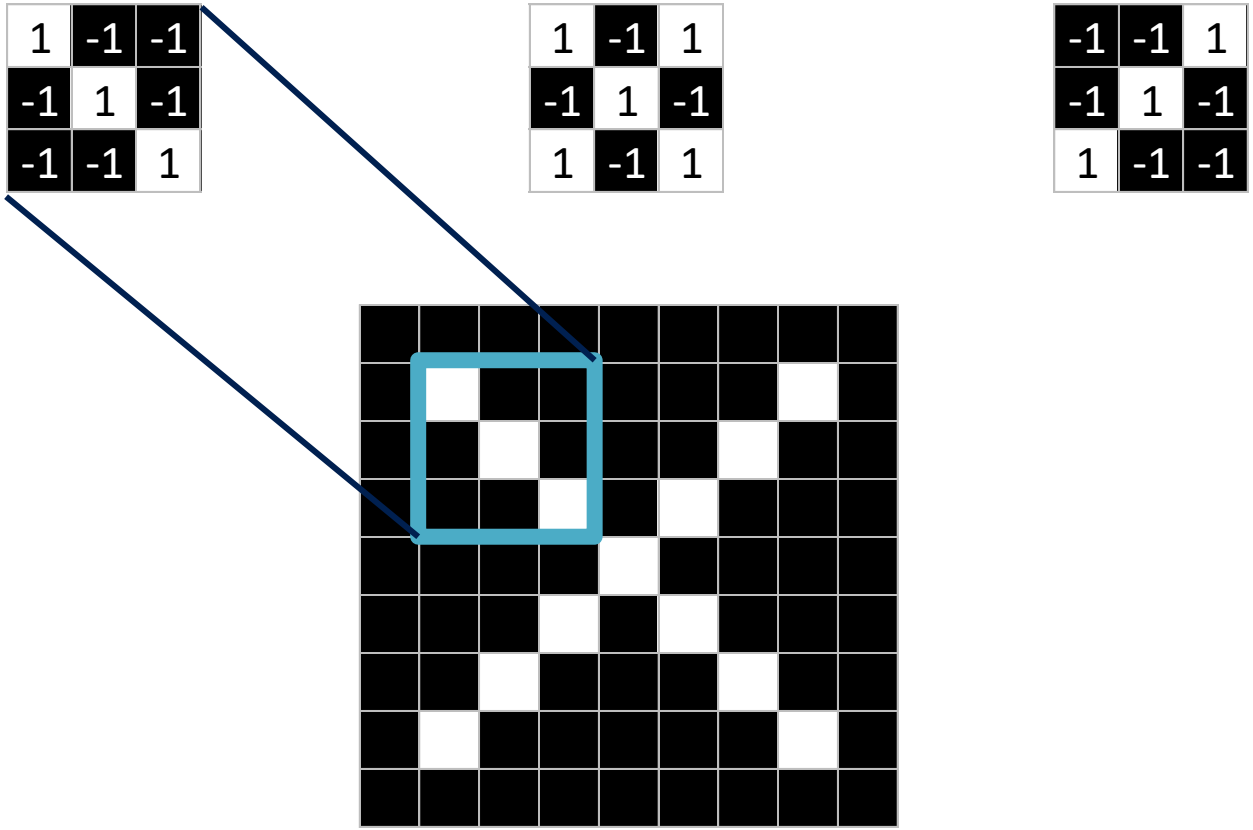


Features match pieces of the image





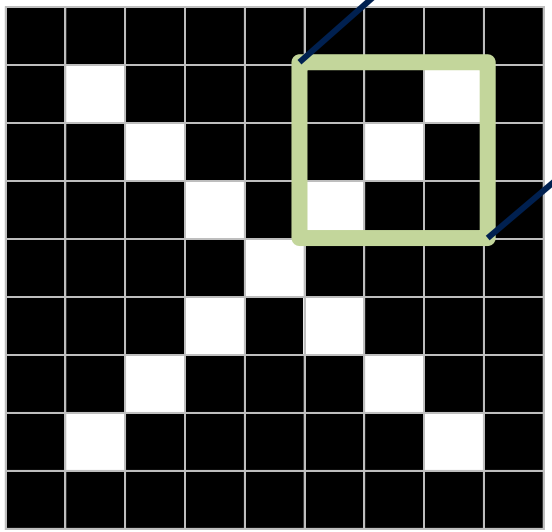



1 -1 -1
-1 1 -1
-1 -1 1
1 -1 1
-1 1 -1
1 -1 1
-1 -1 1
-1 1 -1
1 -1 -1

1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

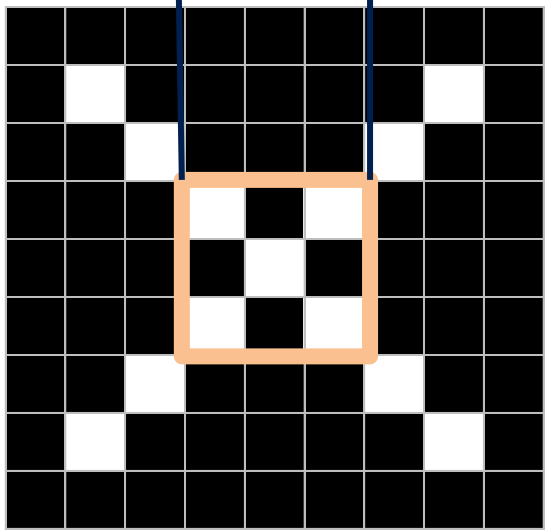
-1	-1	1
-1	1	-1
1	-1	-1

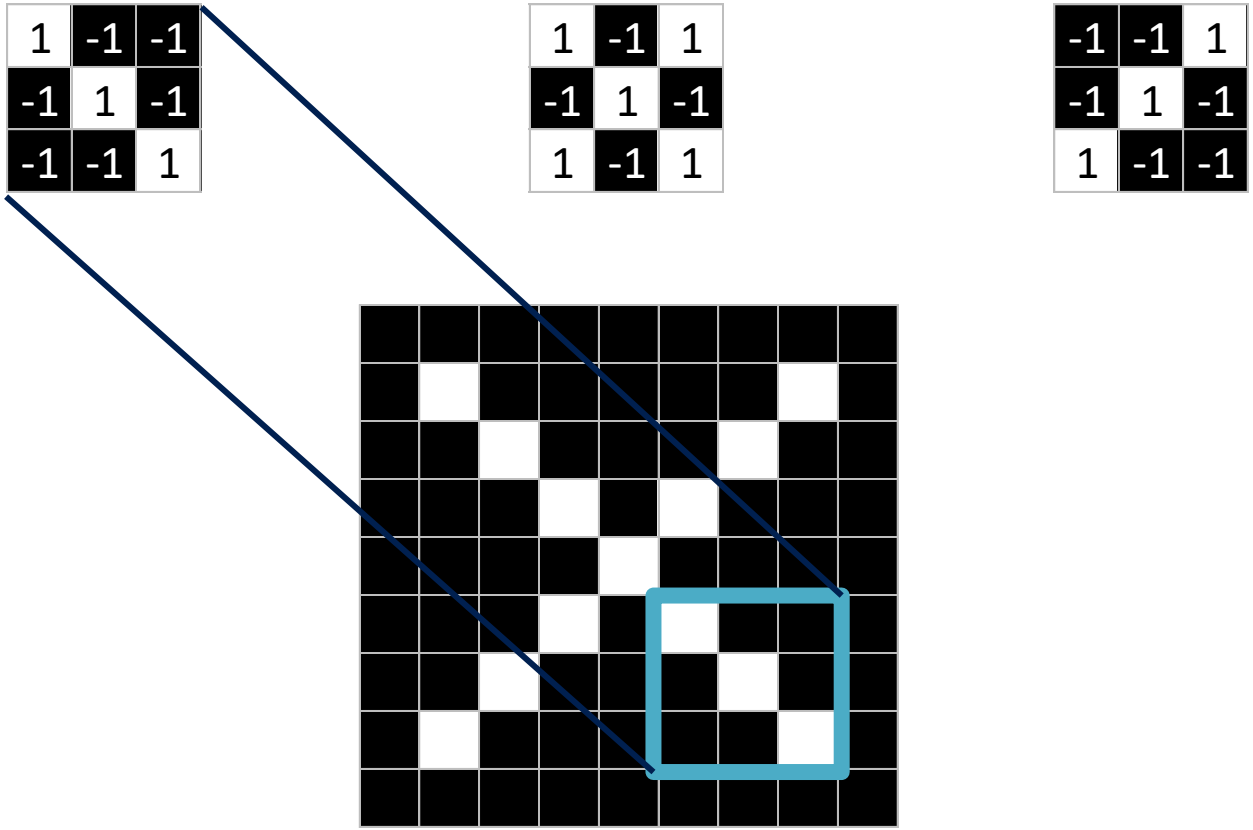


1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1

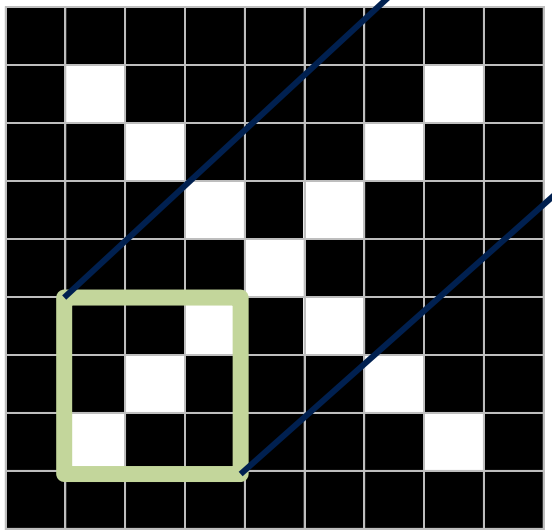



1 -1 -1
-1 1 -1
-1 -1 1
1 -1 1
-1 1 -1
1 -1 1
-1 -1 1
-1 1 -1
1 -1 -1

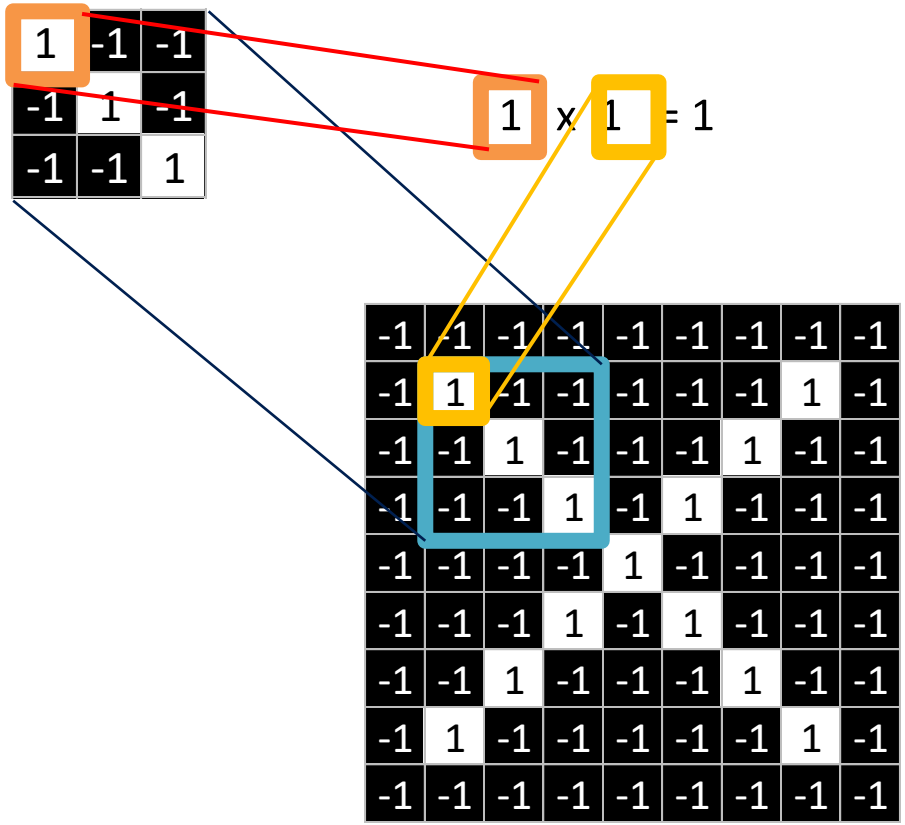
1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

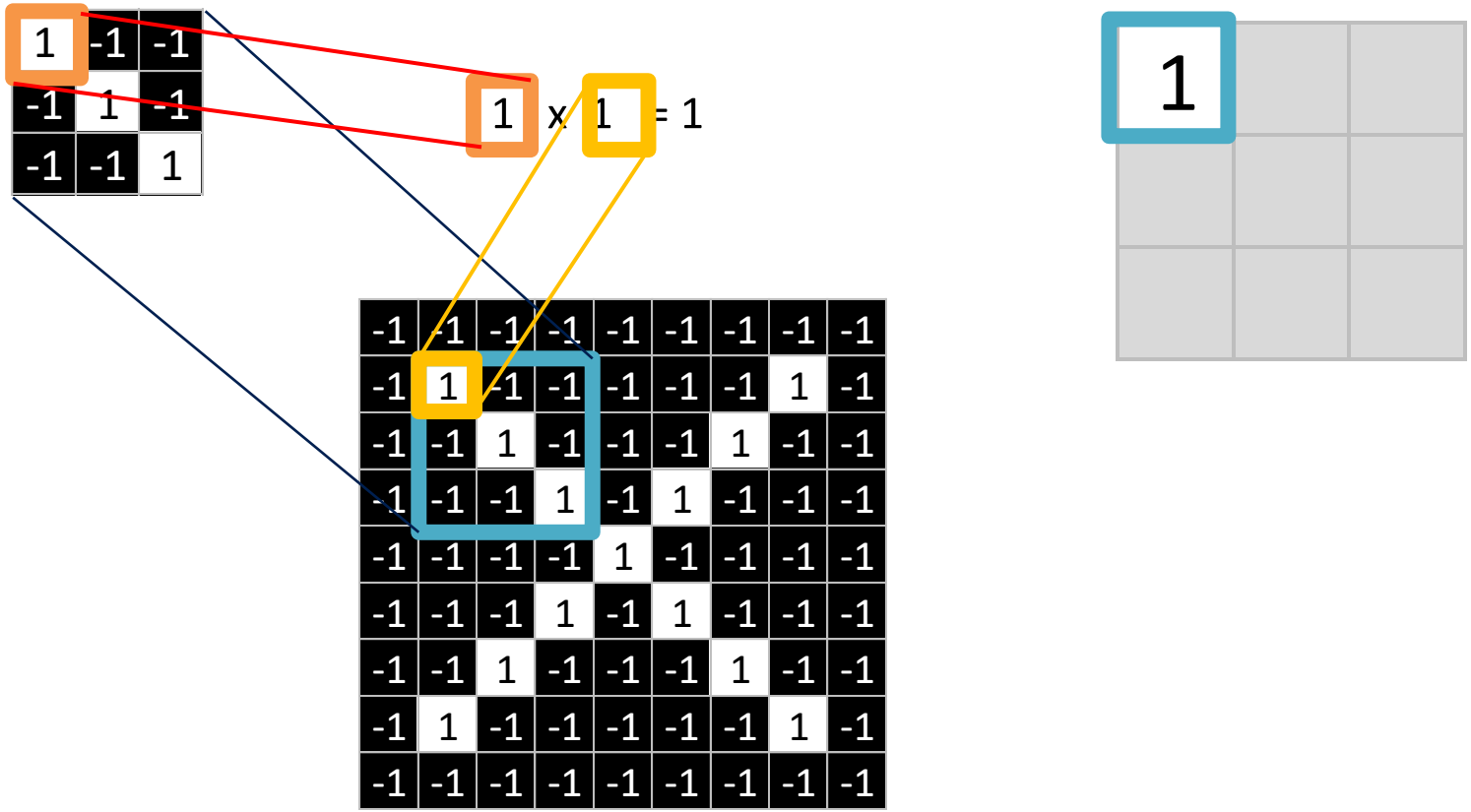
-1	-1	1
-1	1	-1
1	-1	-1



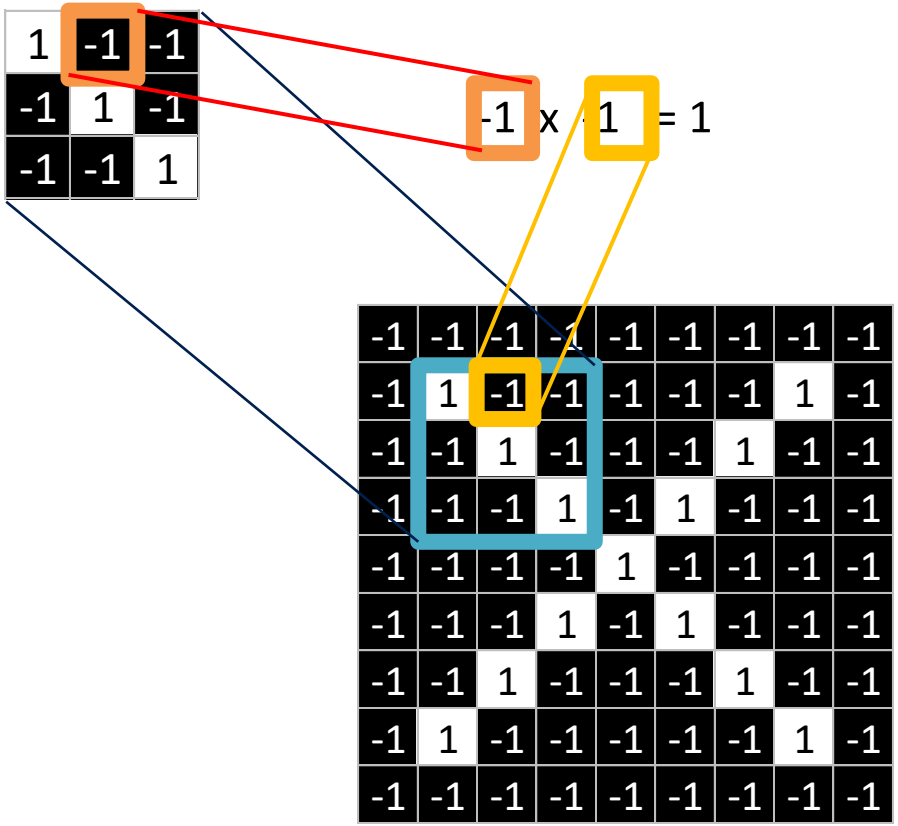
Filtering: The math behind the match

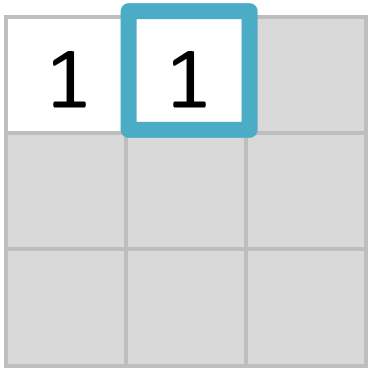


Filtering: The math behind the match

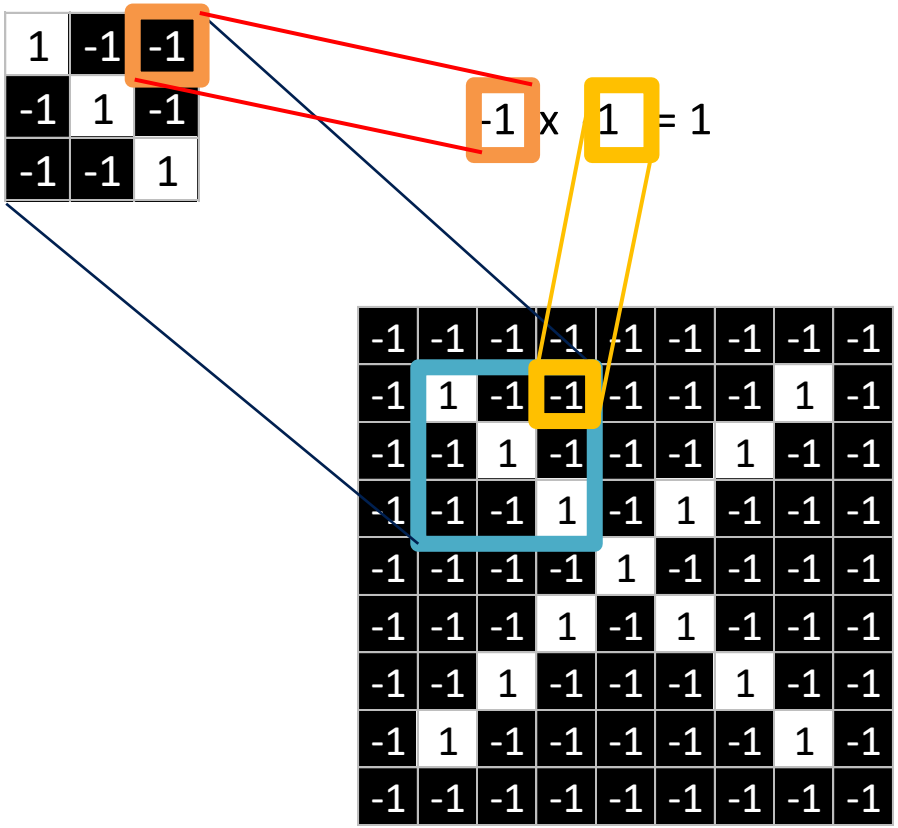


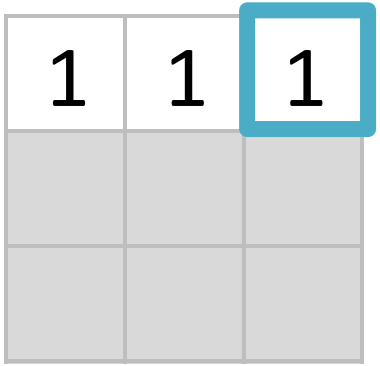
Filtering: The math behind the match



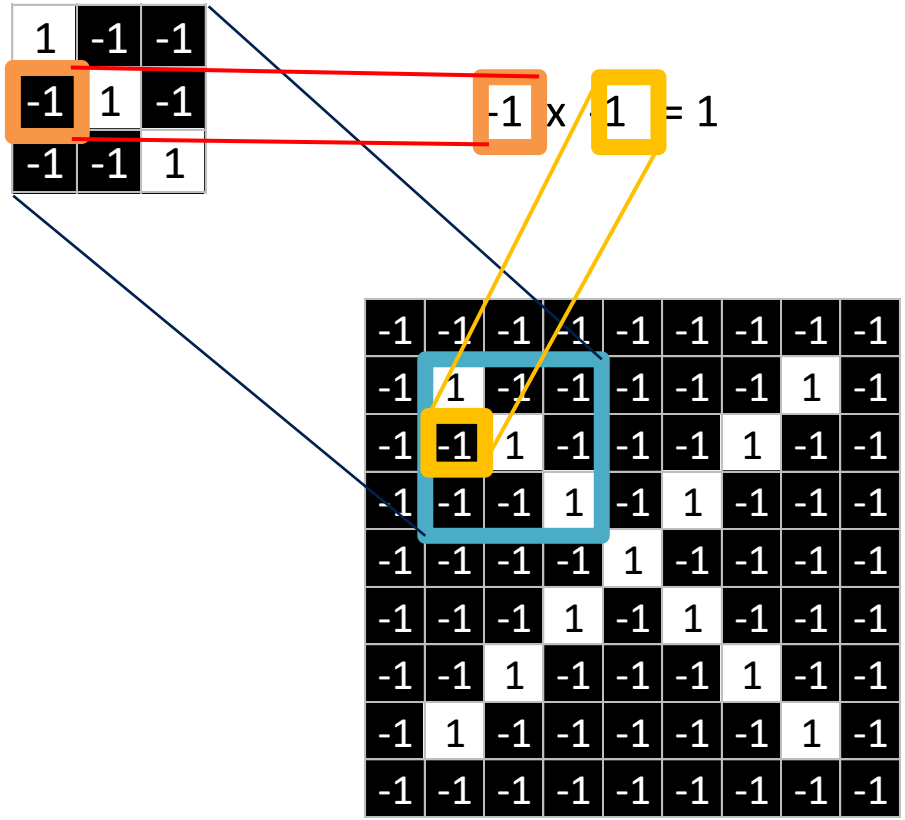


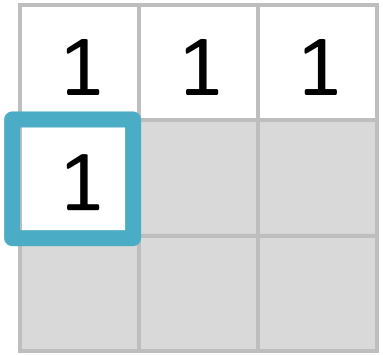
Filtering: The math behind the match



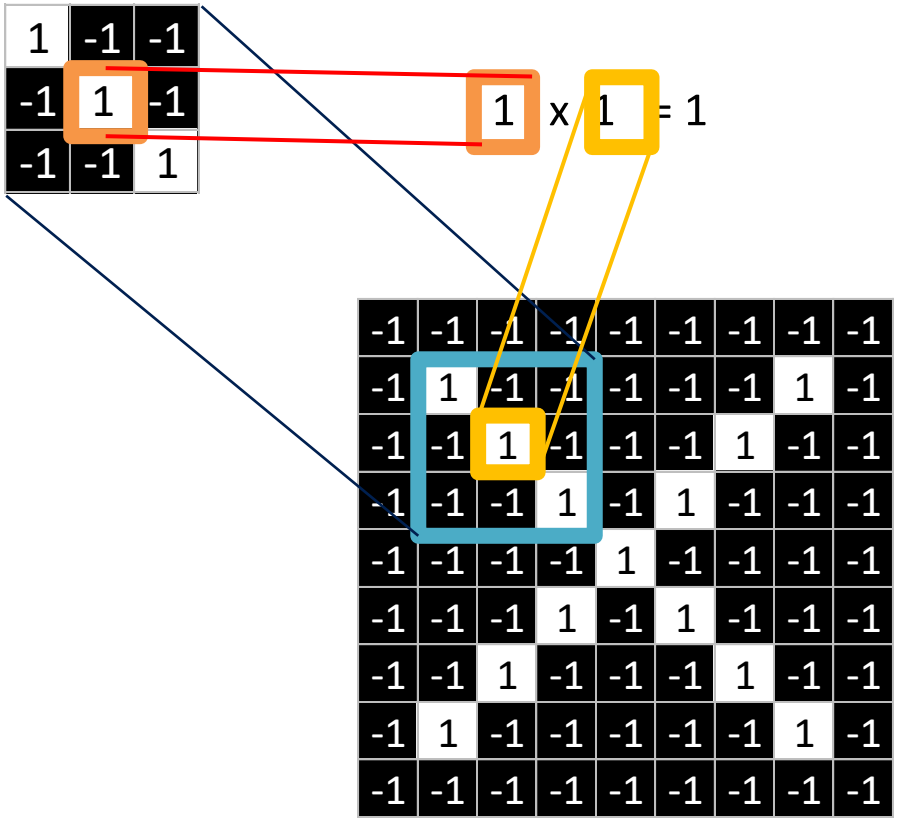


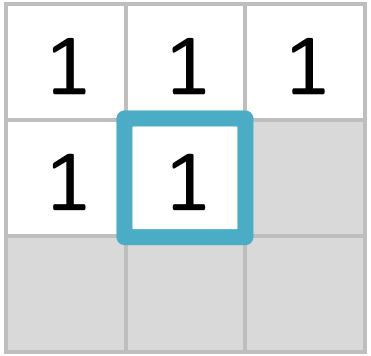
Filtering: The math behind the match



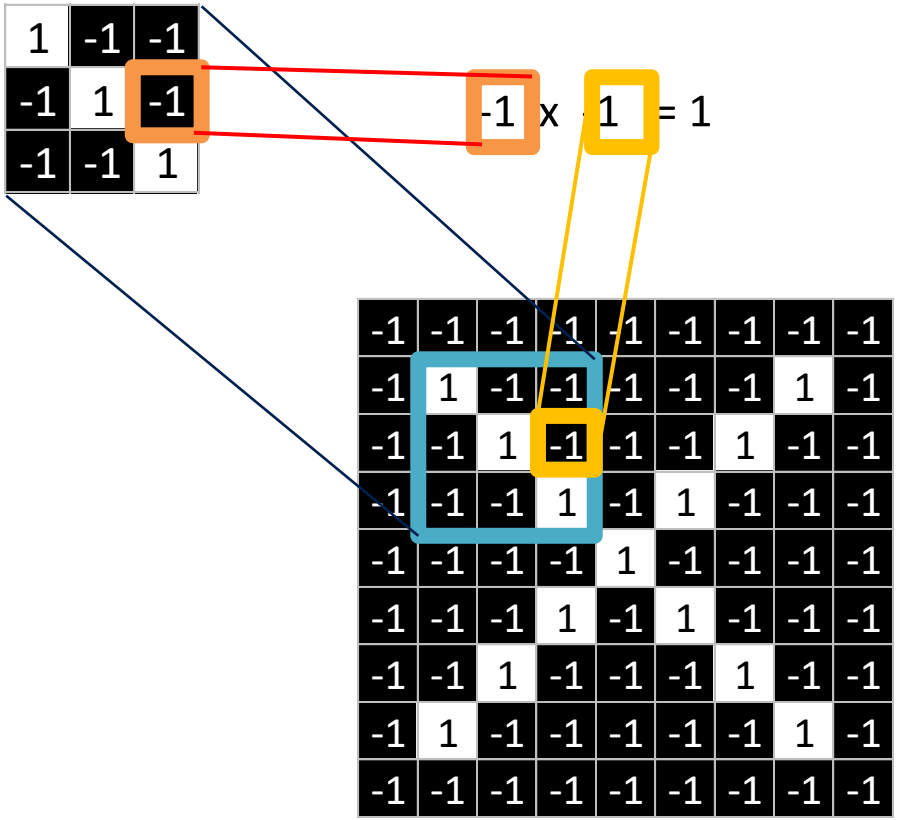


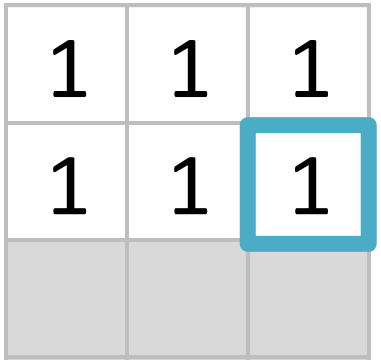
Filtering: The math behind the match



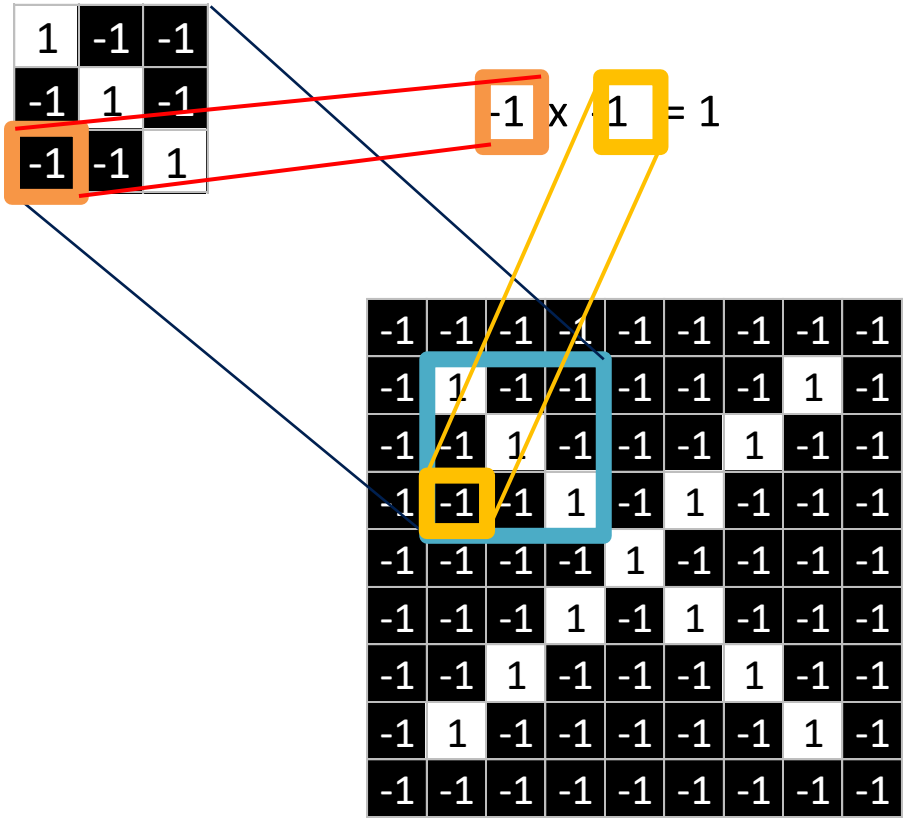


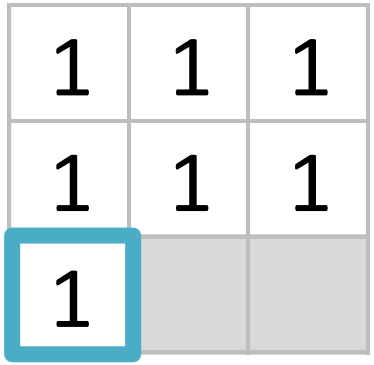
Filtering: The math behind the match



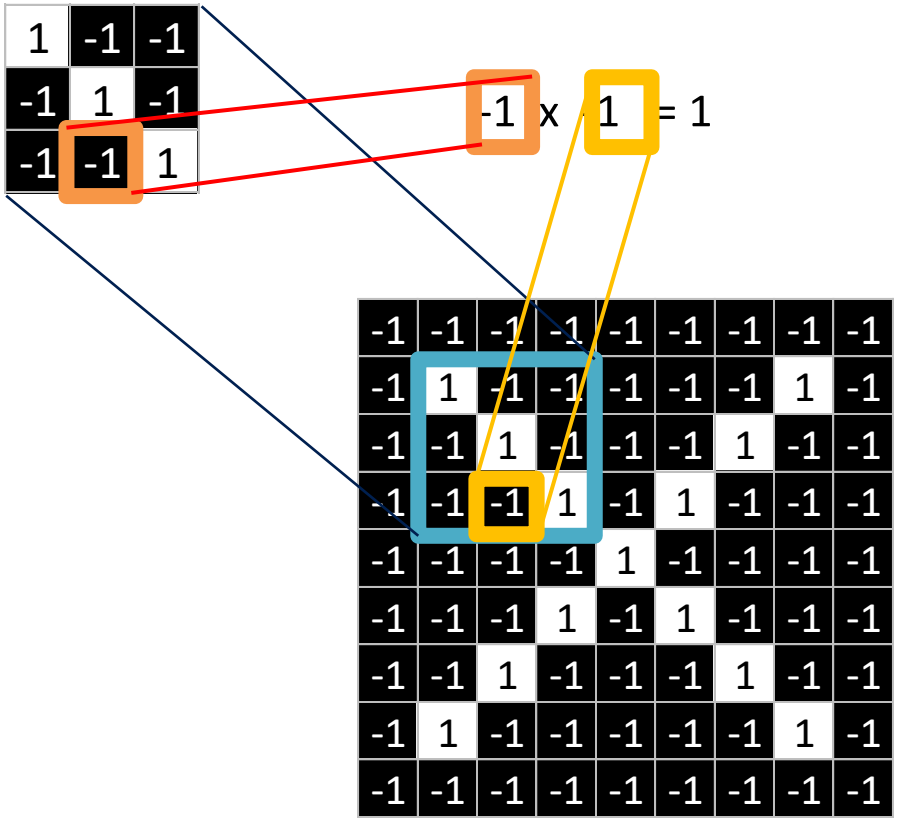


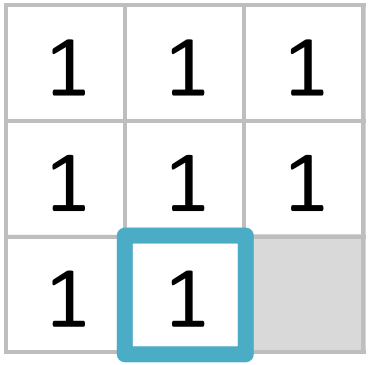
Filtering: The math behind the match



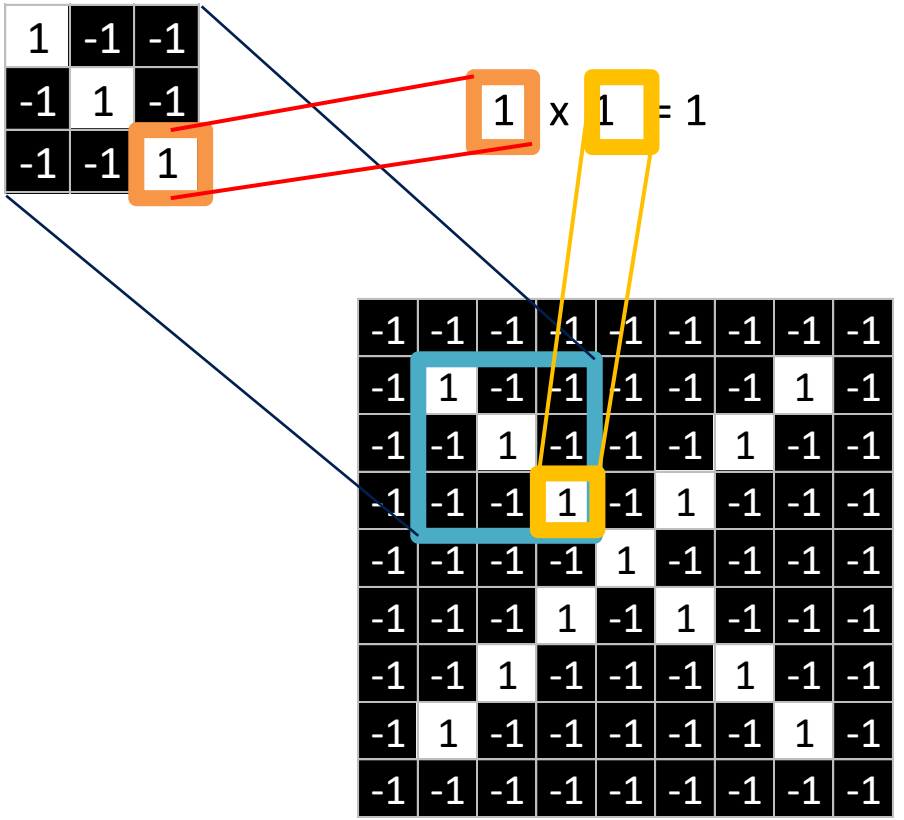


Filtering: The math behind the match



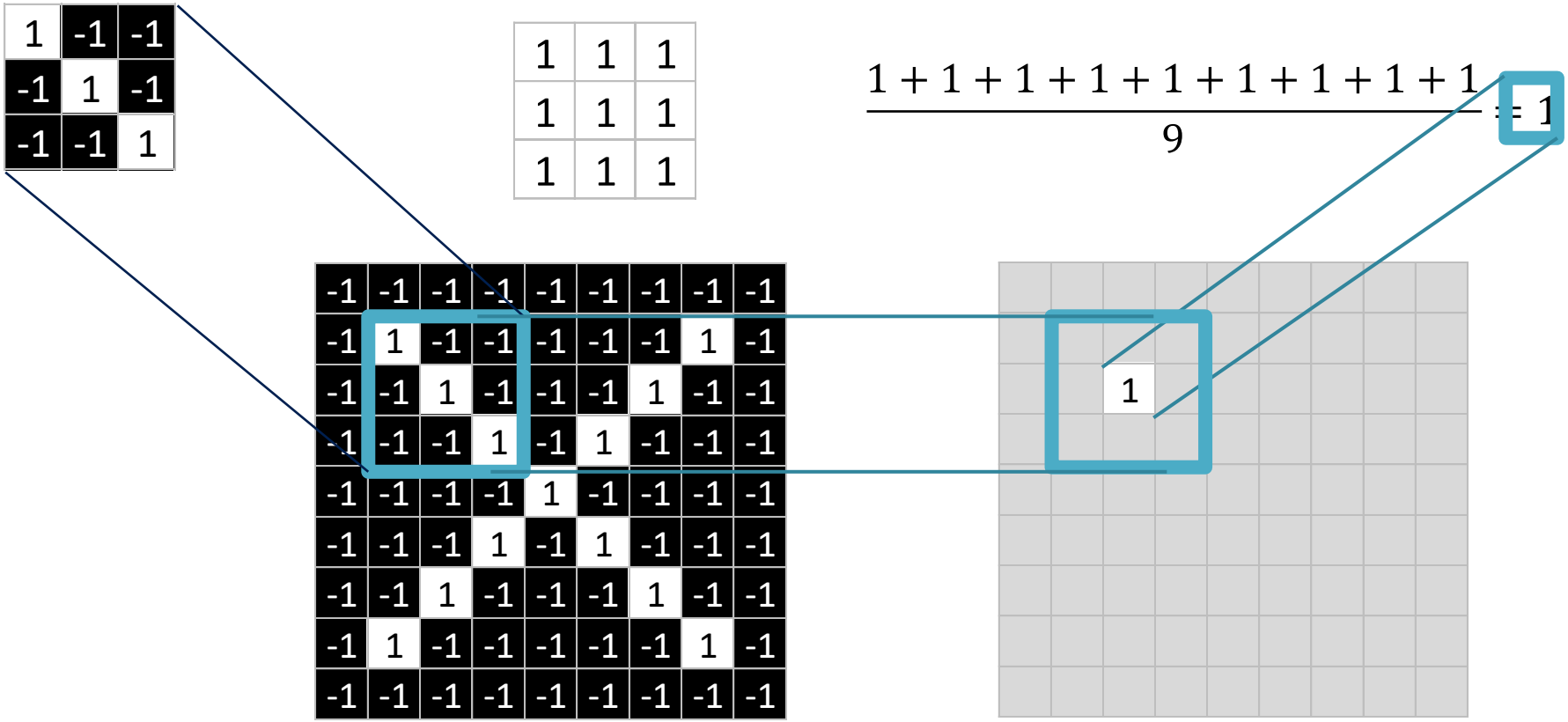


Filtering: The math behind the match

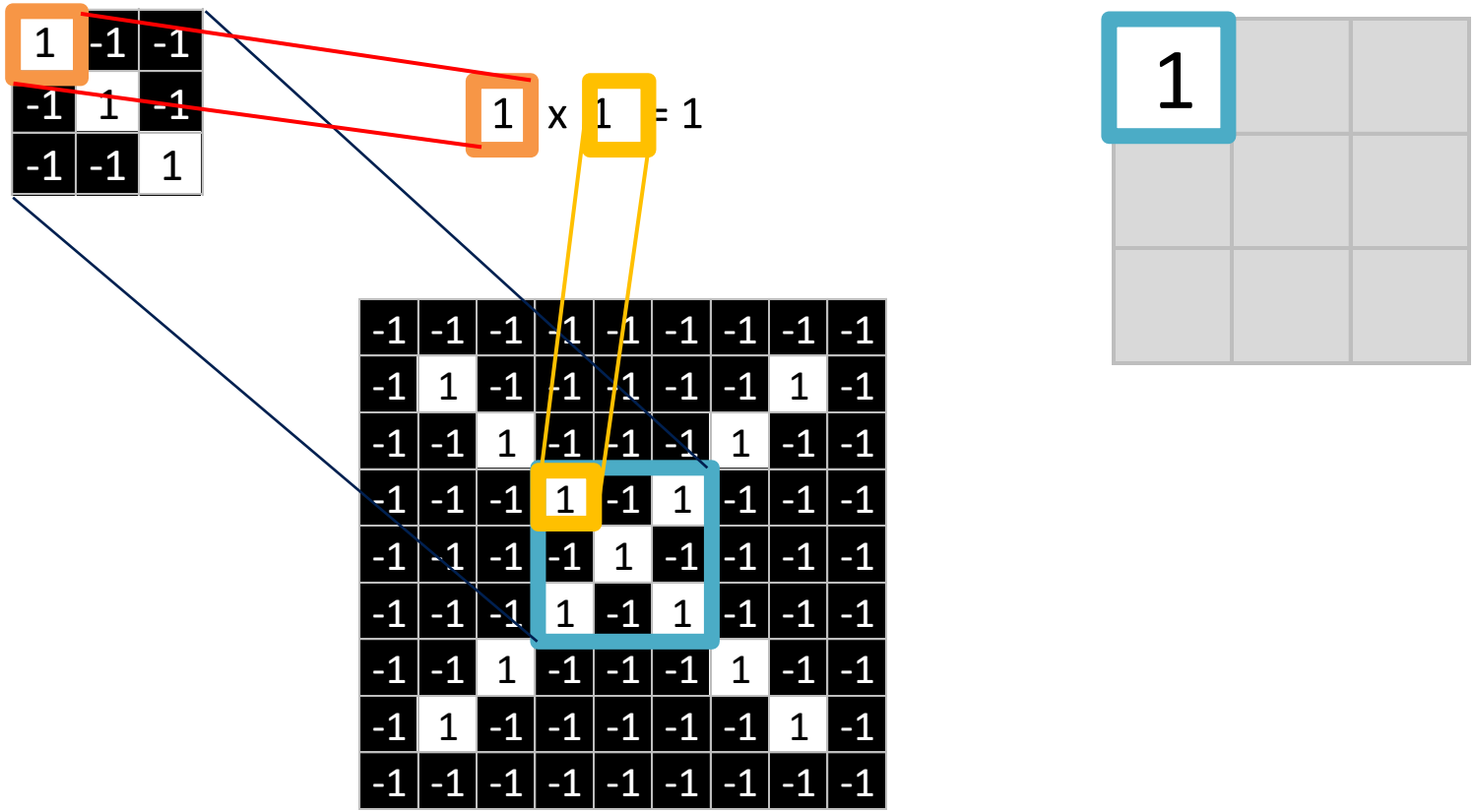


1	1	1
1	1	1
1	1	1

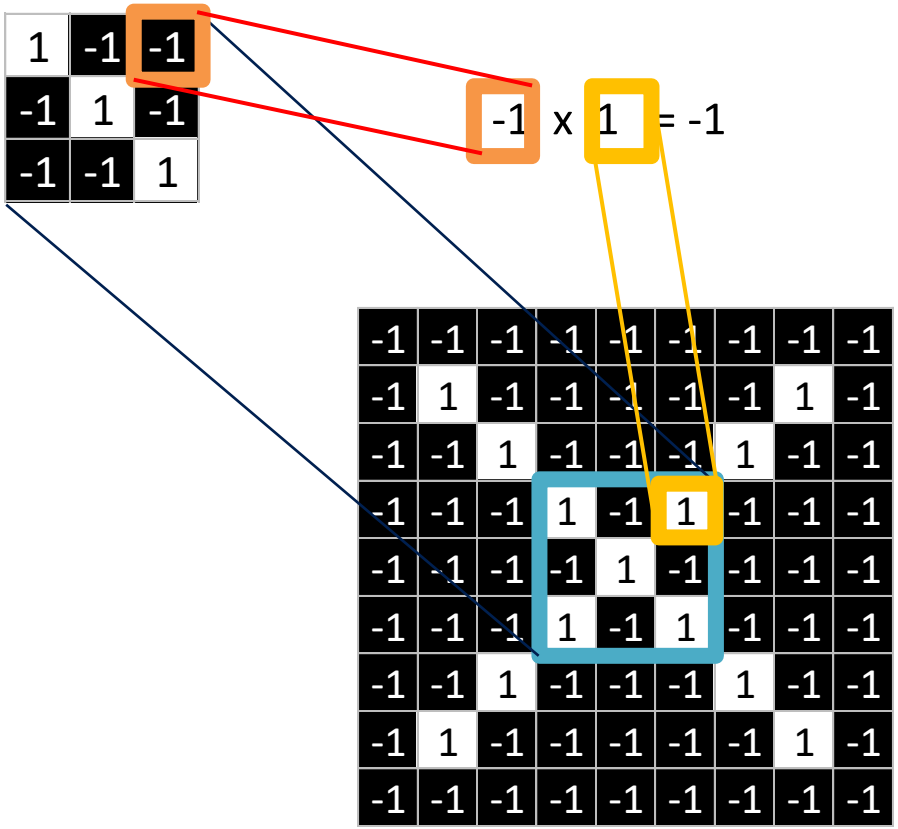
Filtering: The math behind the match

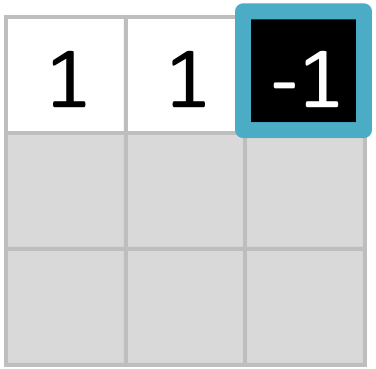


Filtering: The math behind the match

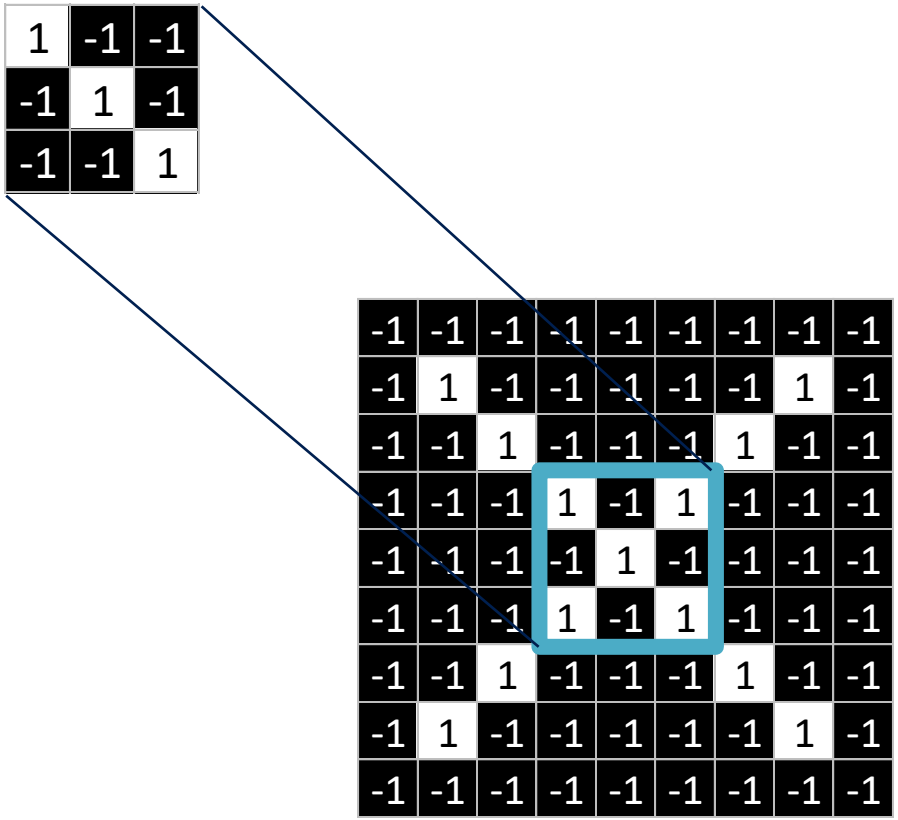


Filtering: The math behind the match



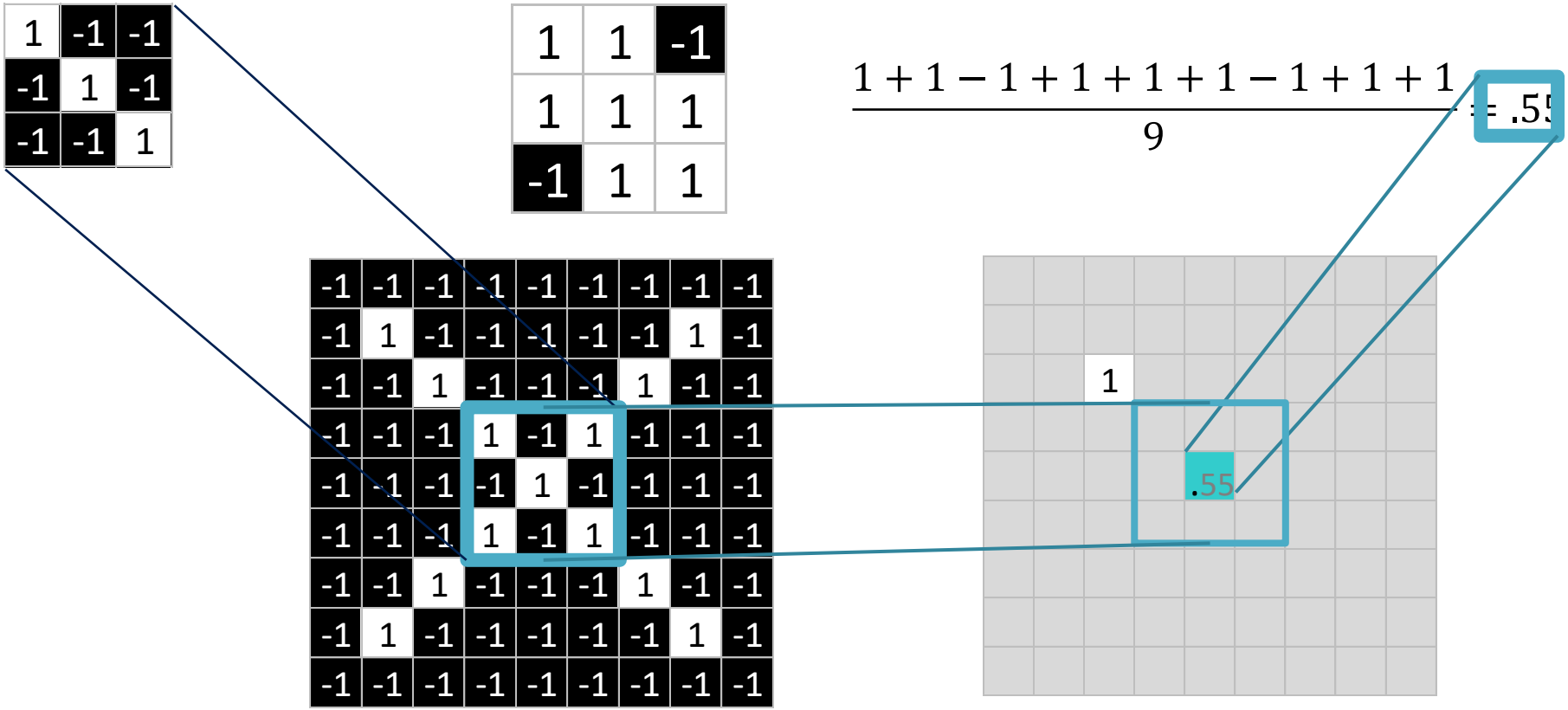


Filtering: The math behind the match

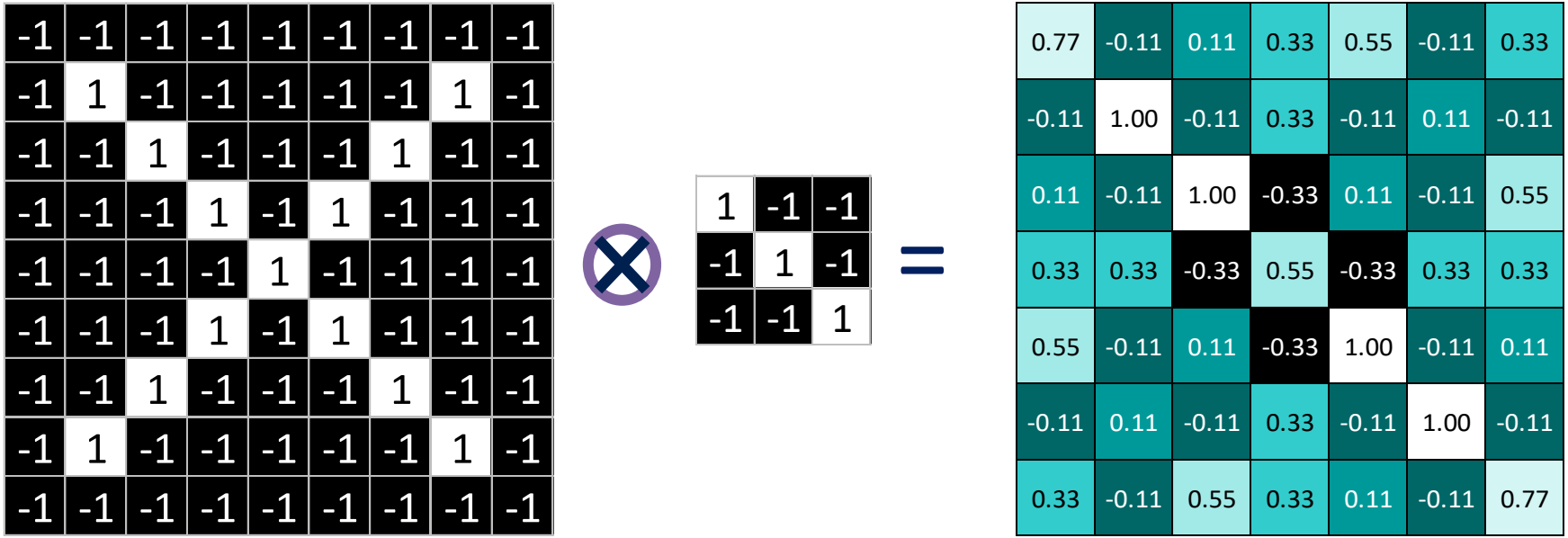


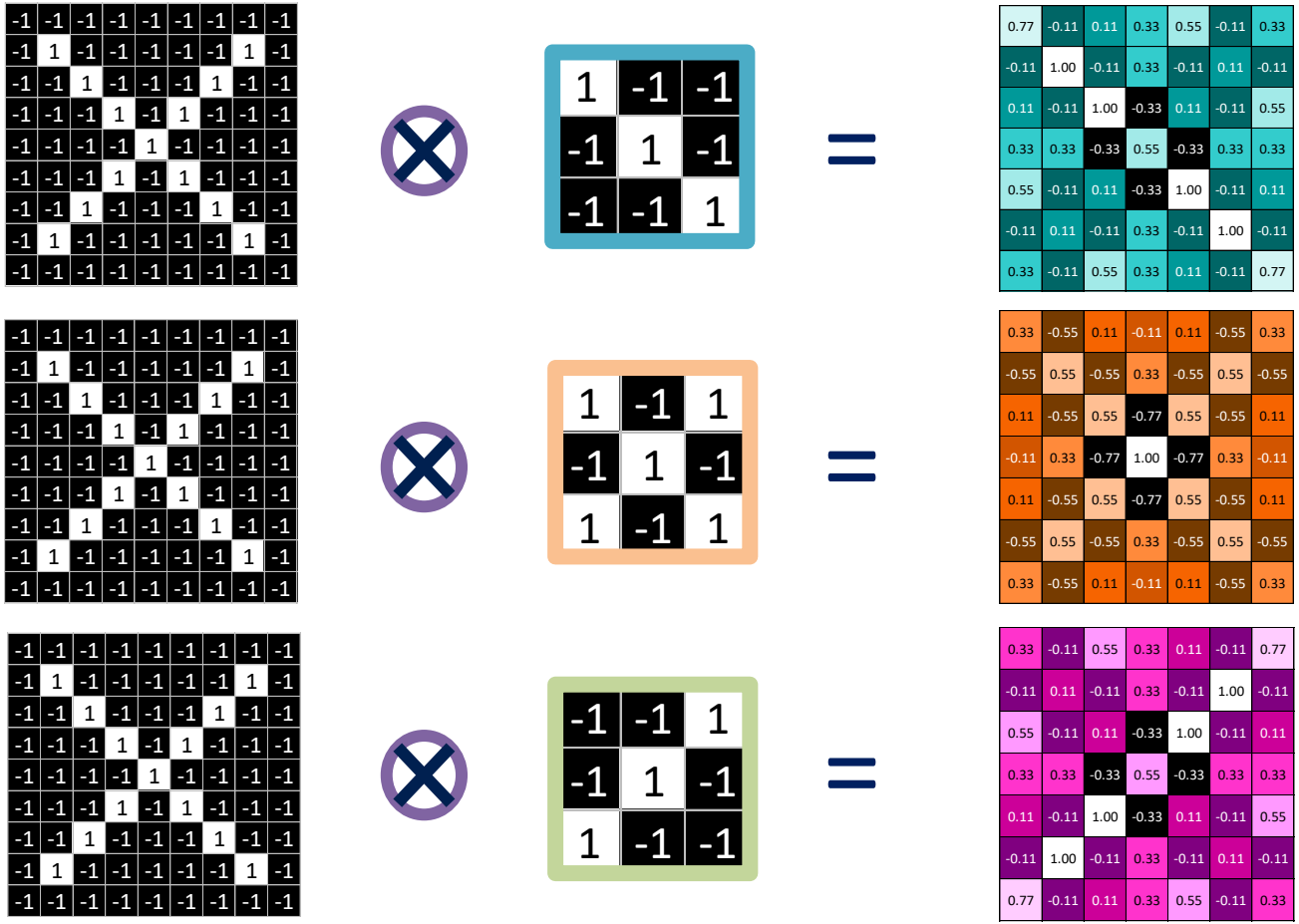
1	1	-1
1	1	1
-1	1	1

Filtering: The math behind the match



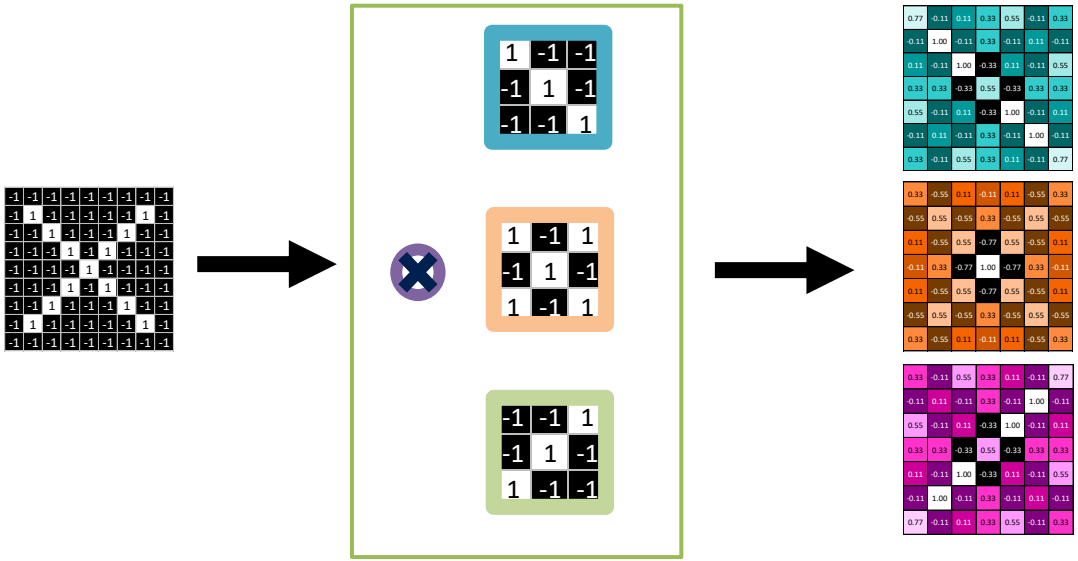
Convolution: Trying every possible match




-1 -1 -1 -1 -1 -1 -1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 -1 -1 1 -1 -1 -1 -1
-1 -1 -1 1 -1 1 -1 -1 -1
-1 -1 1 -1 -1 -1 1 -1 -1
-1 1 -1 -1 -1 -1 -1 1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1
1 -1 -1
-1 1 -1
-1 -1 1
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1 -1 1
-1 1 -1
1 -1 1
0.33 -0.55 0.11 -0.11 0.11 -0.55 0.33
-0.55 0.55 -0.55 0.33 -0.55 0.55 -0.55
0.11 -0.55 0.55 -0.77 0.55 -0.55 0.11
-0.11 0.33 -0.77 1.00 -0.77 0.33 -0.11
0.11 -0.55 0.55 -0.77 0.55 -0.55 0.11
-0.55 0.55 -0.55 0.33 -0.55 0.55 -0.55
0.33 -0.55 0.11 -0.11 0.11 -0.55 0.33
-1 -1 1
-1 1 -1
1 -1 -1
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33

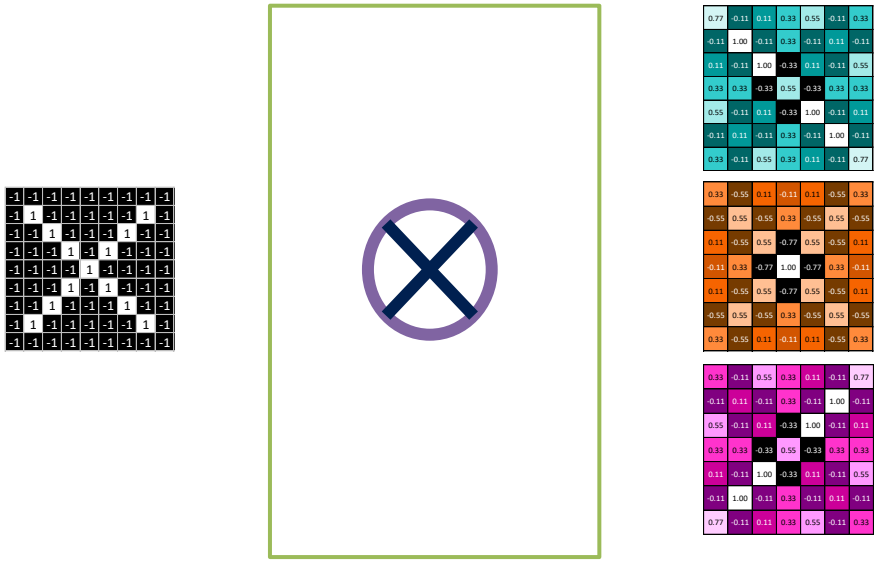
Convolution layer

One image becomes a stack of filtered images



Convolution layer

One image becomes a stack of filtered images



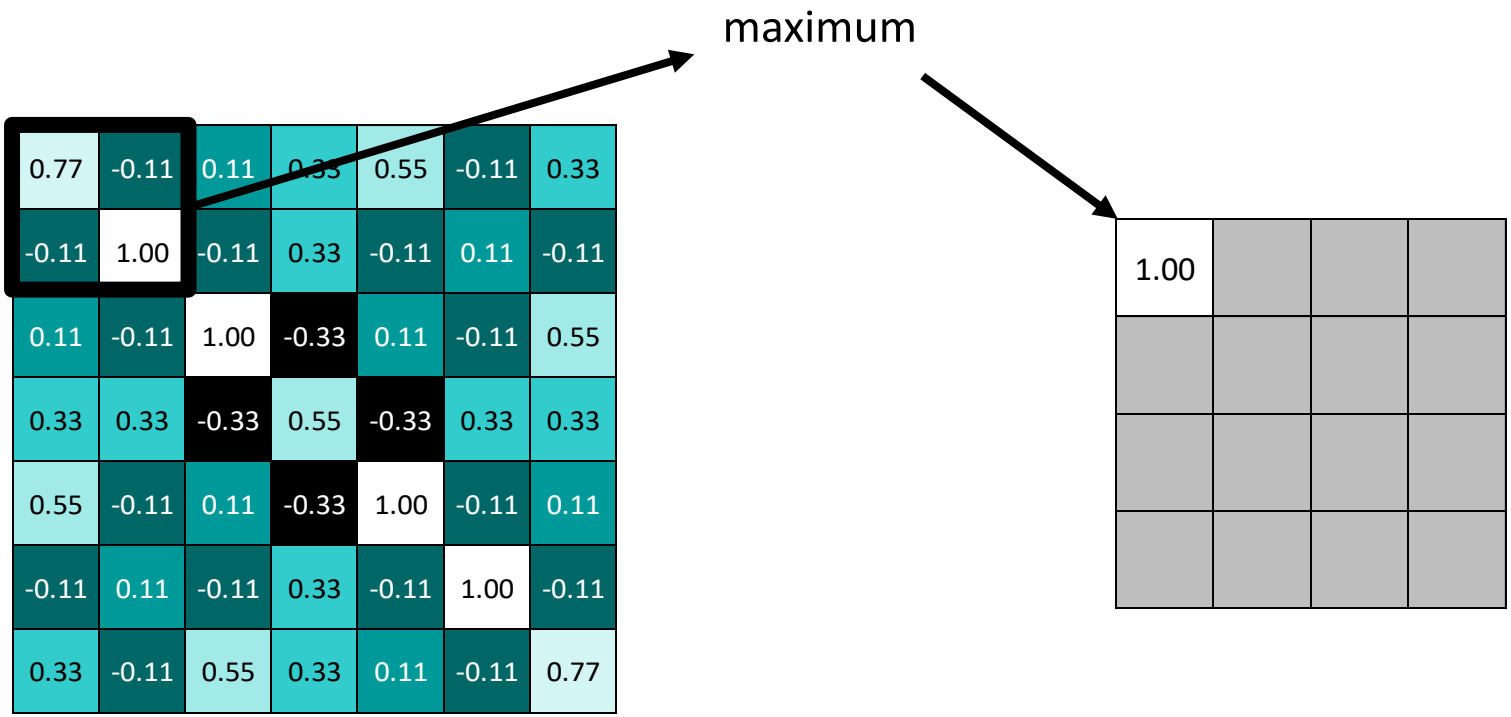
Pooling: Shrinking the image stack

Achieves a more abstract representation

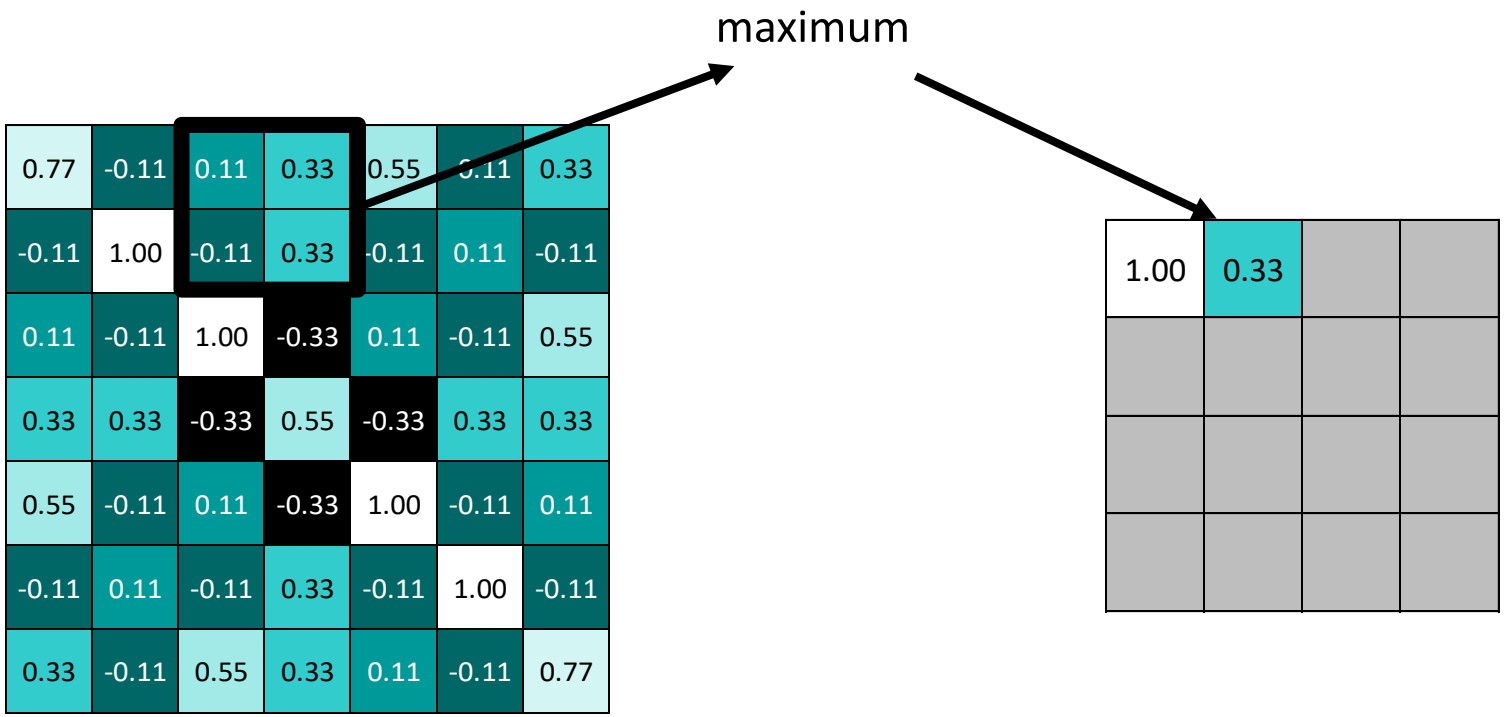
Improves invariance to geometric transformation

1. Pick a window size (usually 2 or 3).
2. Pick a stride (usually 2).
3. Walk your window across your filtered images.
4. From each window, take the maximum value.

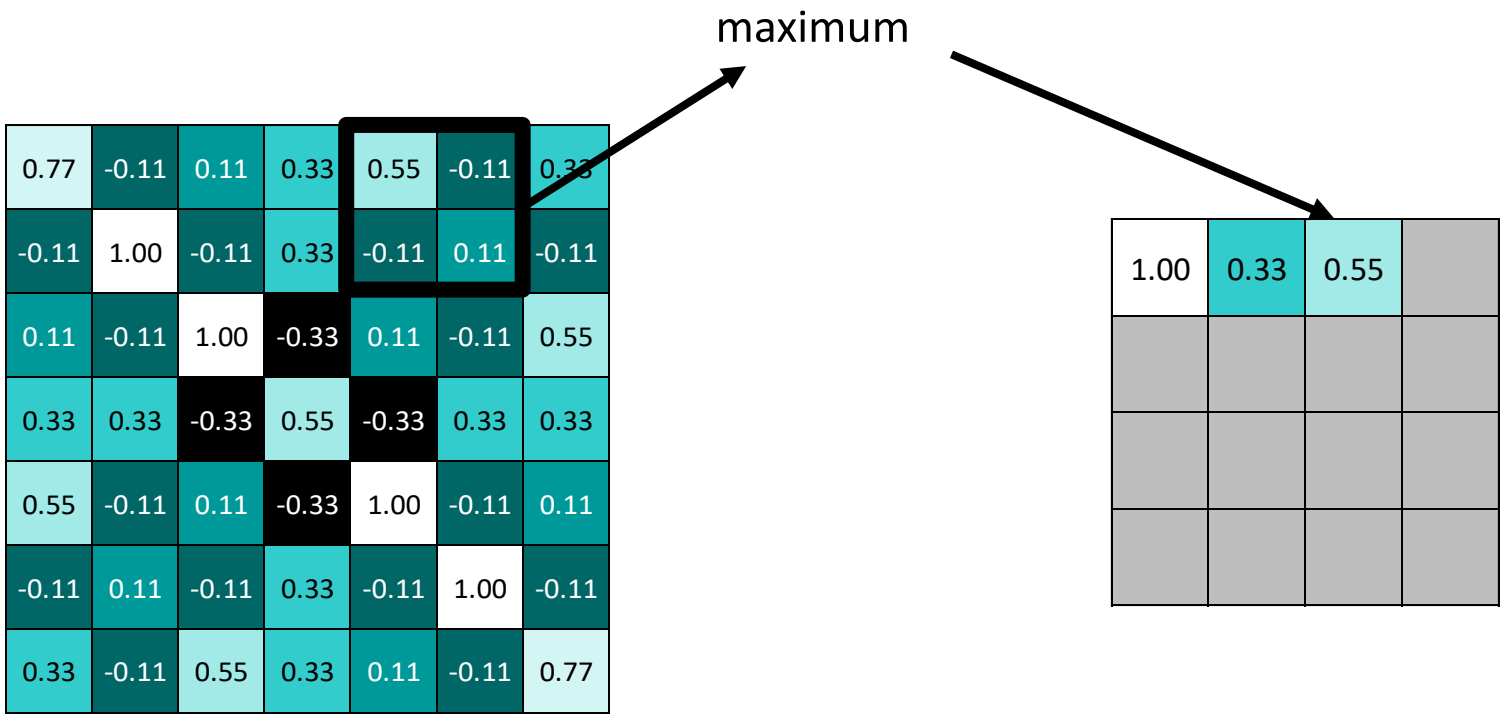
Pooling


maximum
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1.00

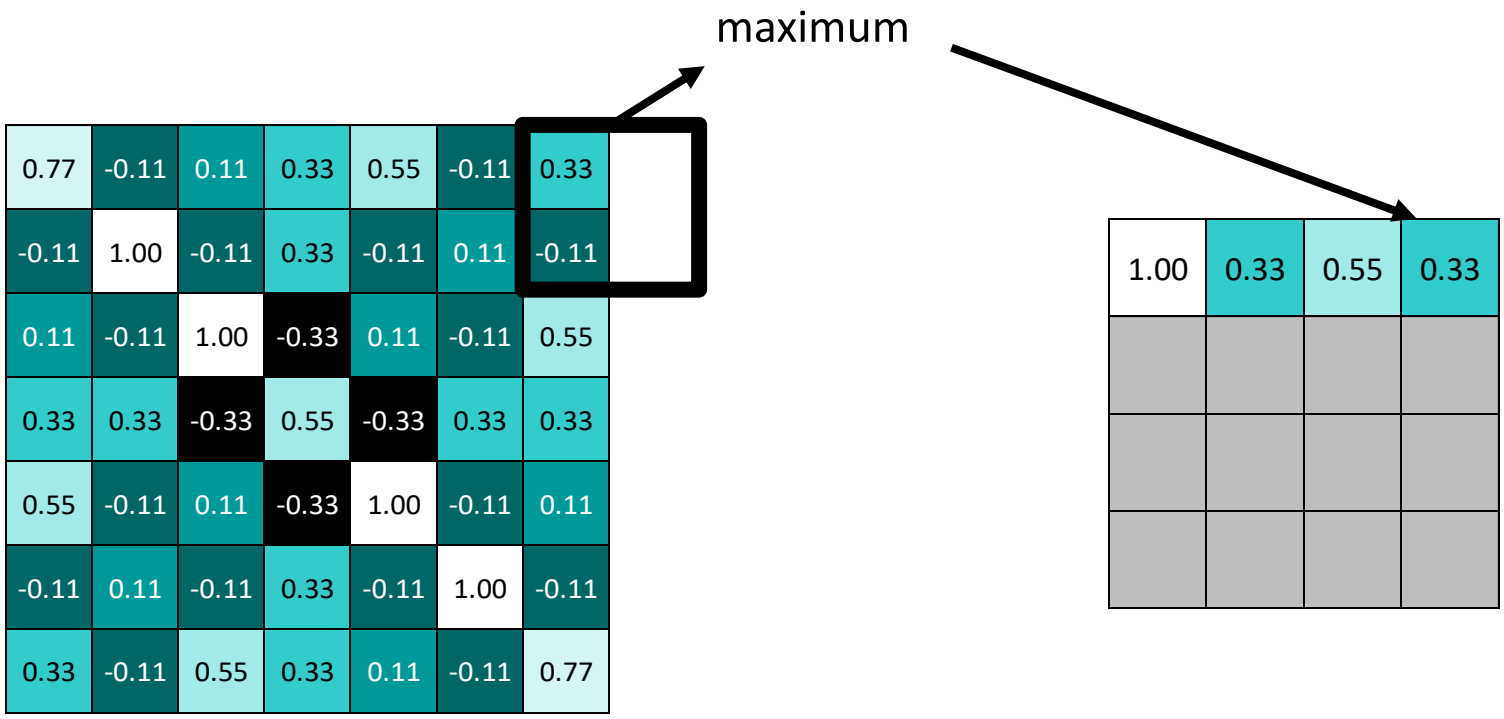
Pooling


maximum
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1.00 0.33

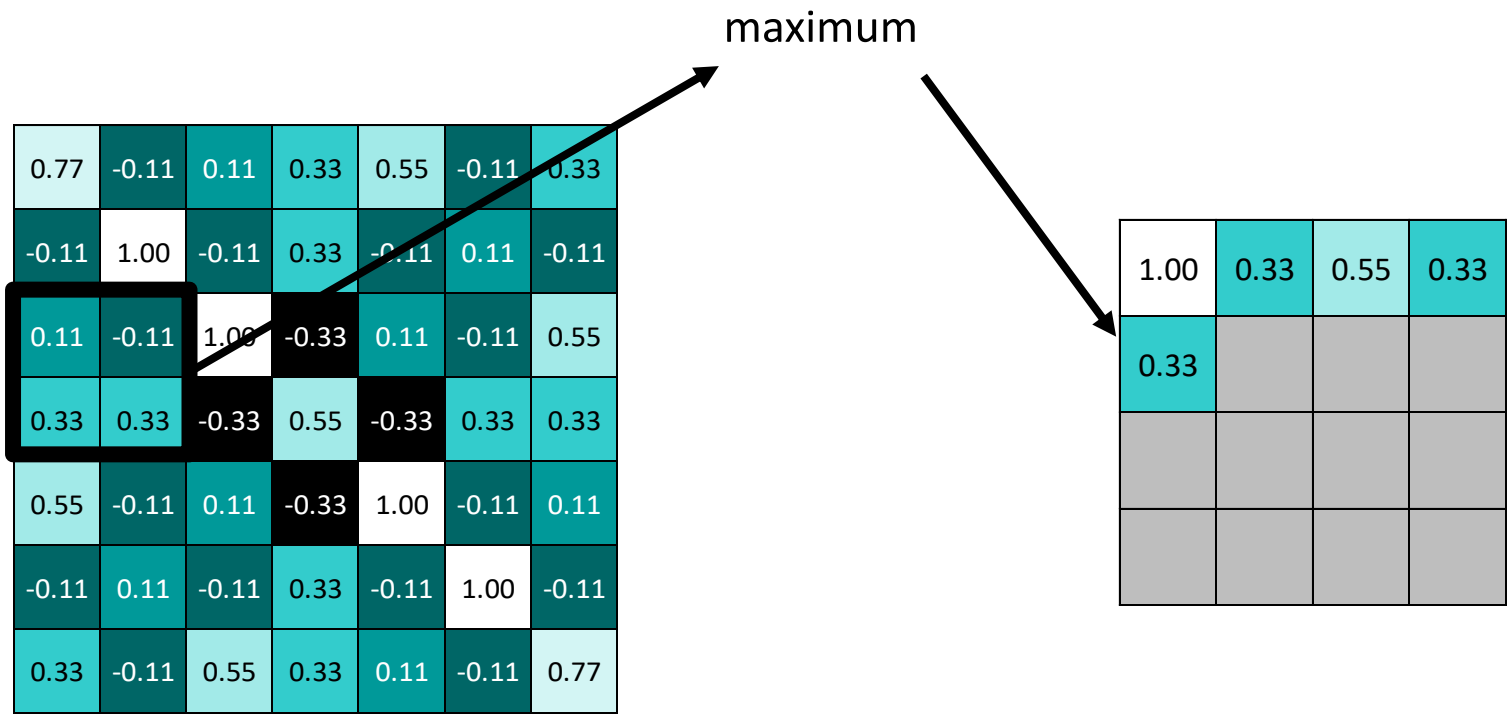
Pooling


maximum
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1.00 0.33 0.55

Pooling


maximum
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1.00 0.33 0.55 0.33

Pooling


maximum
0.77 -0.11 0.11 0.33 0.55 -0.11 0.33
-0.11 1.00 -0.11 0.33 -0.11 0.11 -0.11
0.11 -0.11 1.00 -0.33 0.11 -0.11 0.55
0.33 0.33 -0.33 0.55 -0.33 0.33 0.33
0.55 -0.11 0.11 -0.33 1.00 -0.11 0.11
-0.11 0.11 -0.11 0.33 -0.11 1.00 -0.11
0.33 -0.11 0.55 0.33 0.11 -0.11 0.77
1.00 0.33 0.55 0.33
0.33

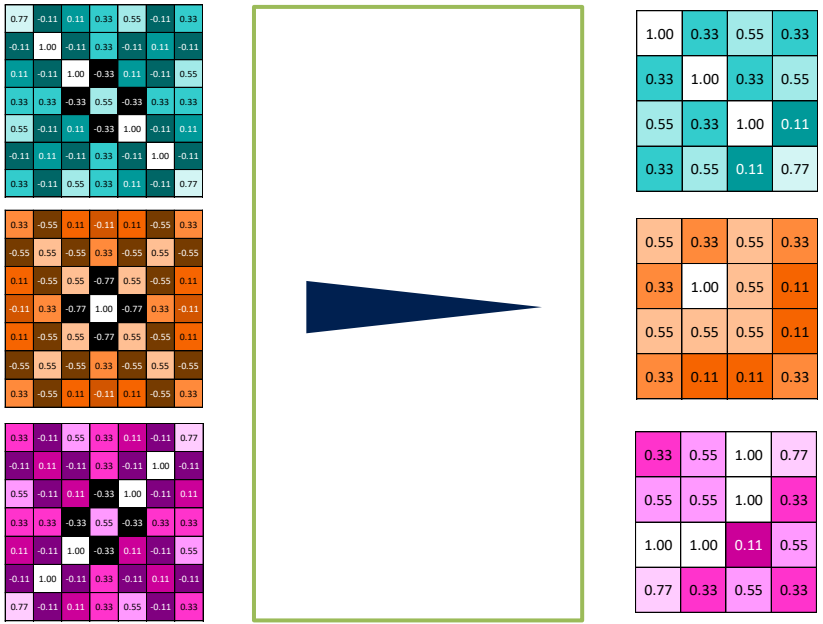
Pooling

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	0.33
0.33	1.00	0.33	0.55
0.55	0.33	1.00	0.11
0.33	0.55	0.11	0.77

Pooling layer

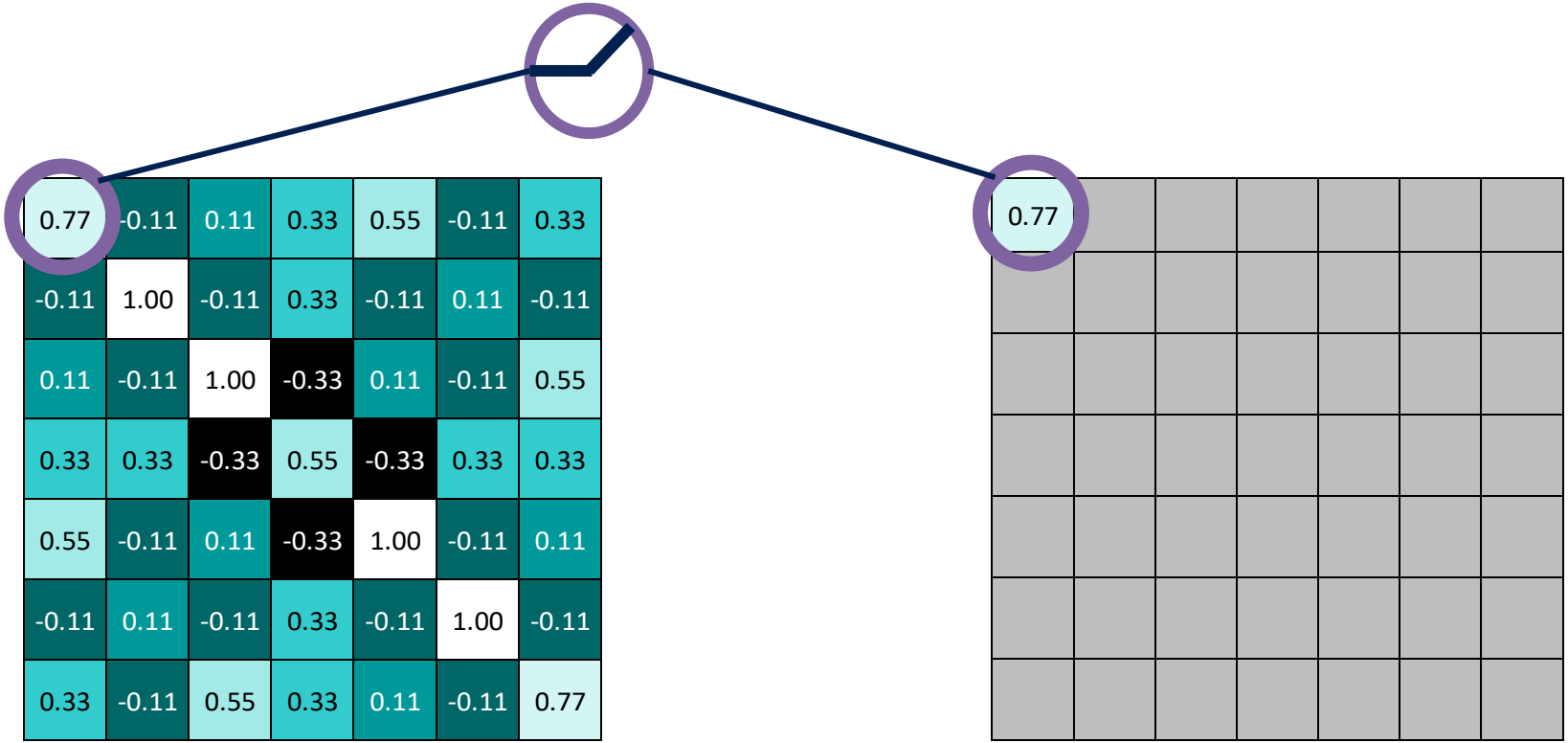
A stack of images becomes a stack of smaller images.



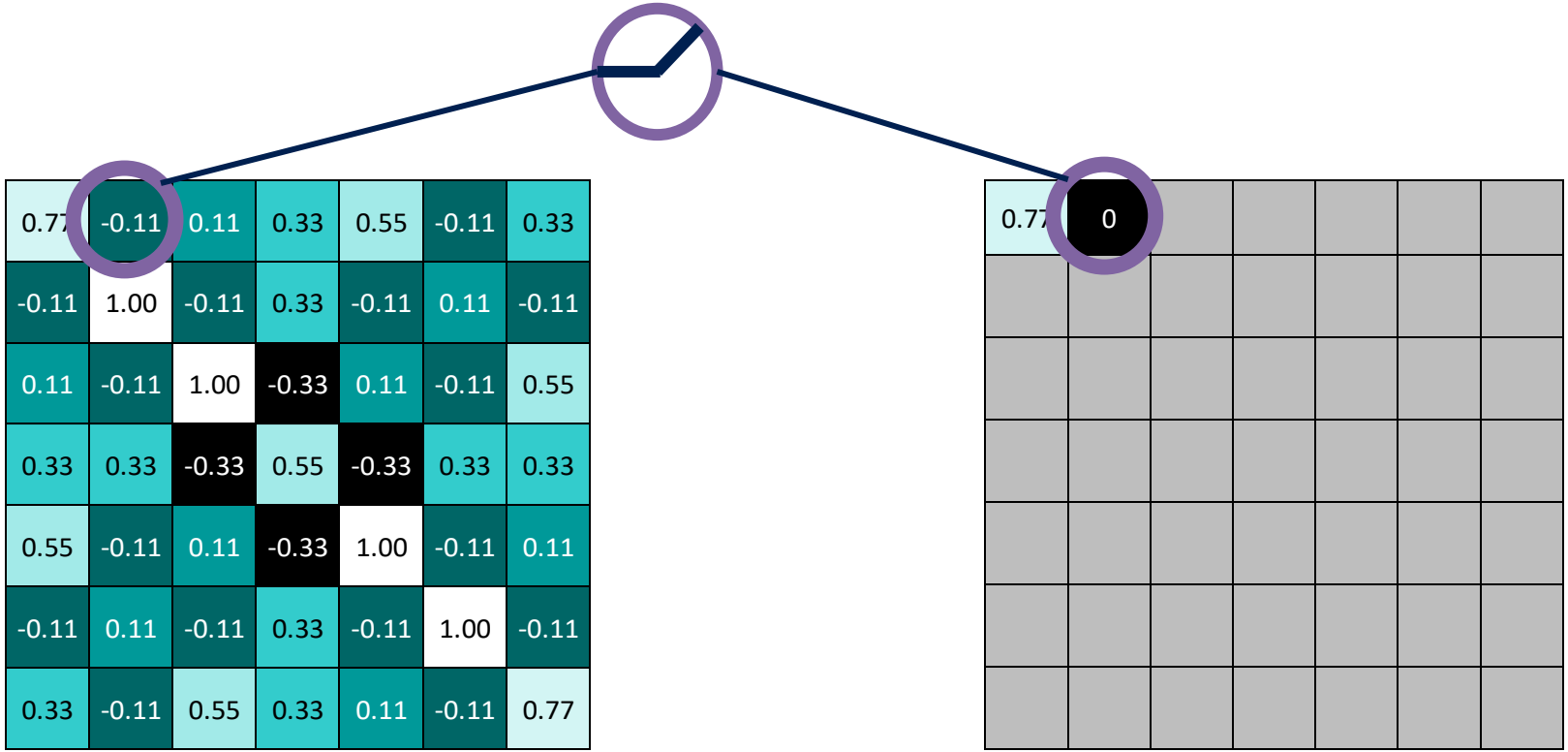
Normalization

- Keep the math from breaking by tweaking each of the values just a bit.
- Change everything negative to zero.

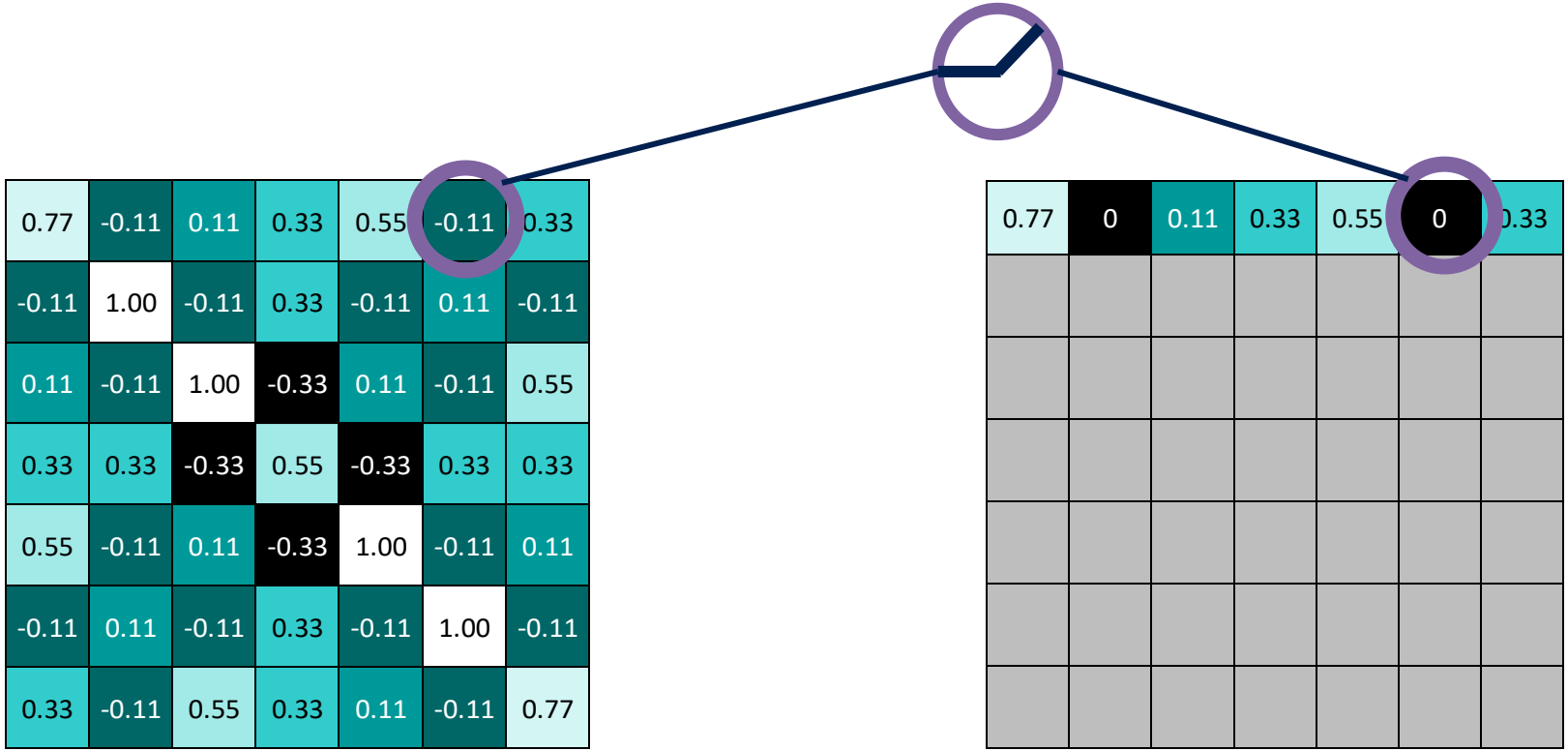
Rectified Linear Units (ReLUs)



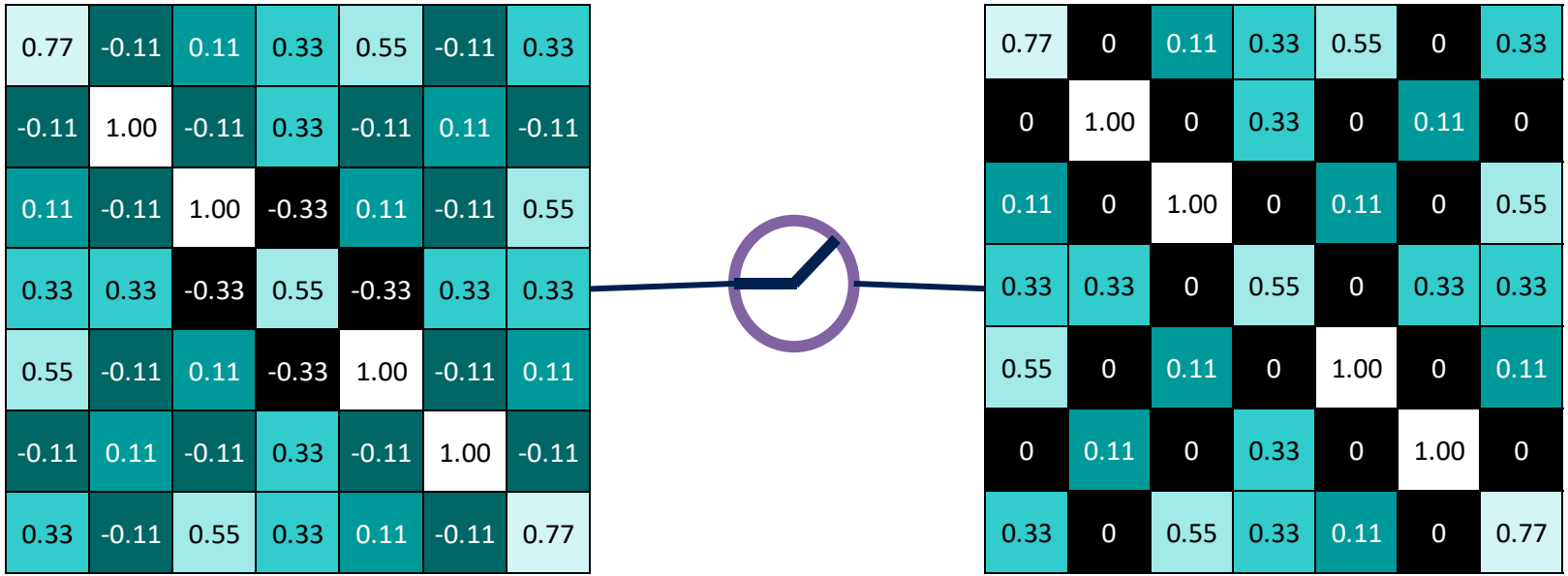
Rectified Linear Units (ReLUs)



Rectified Linear Units (ReLUs)

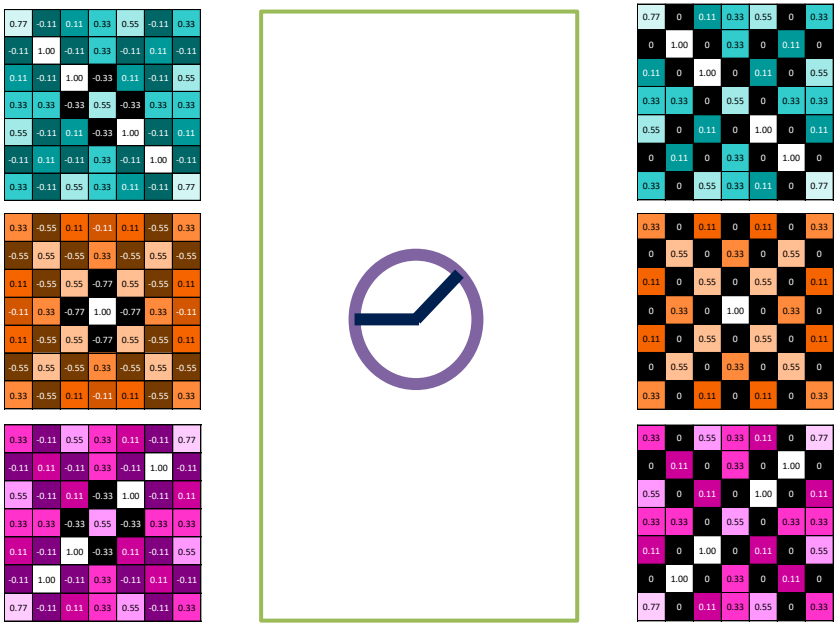


Rectified Linear Units (ReLUs)



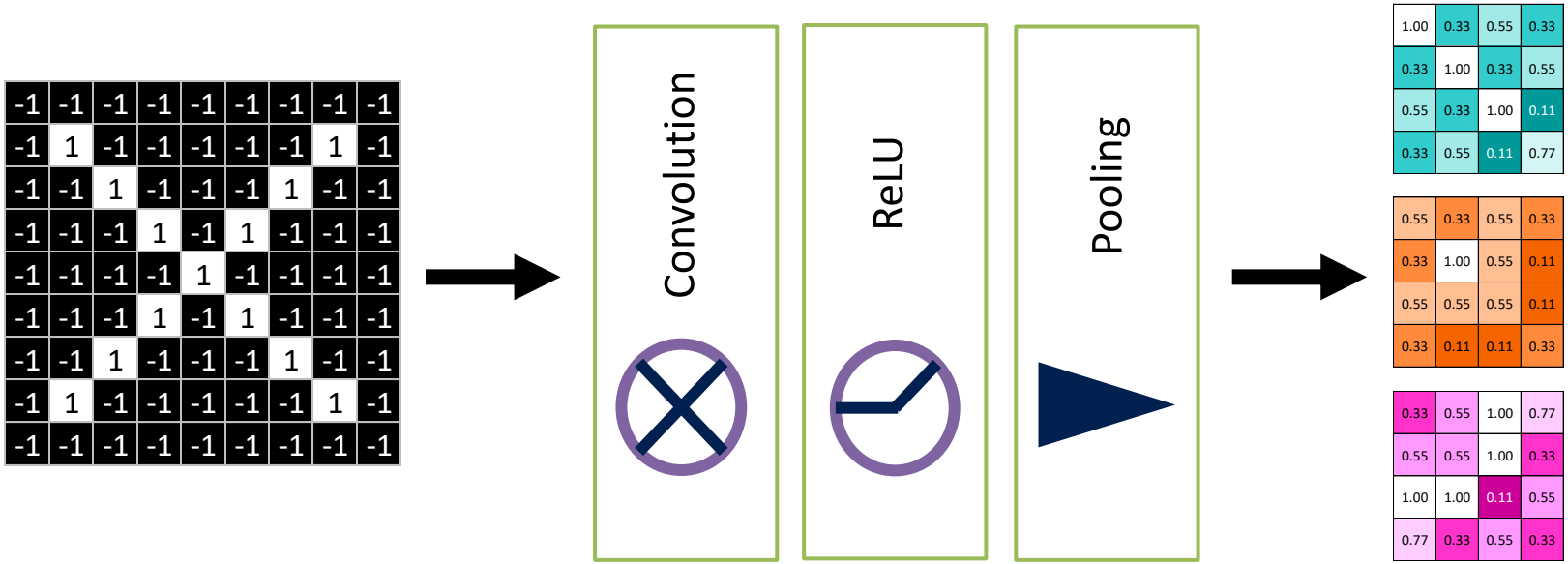
ReLU layer

A stack of images becomes a stack of images with no negative values.



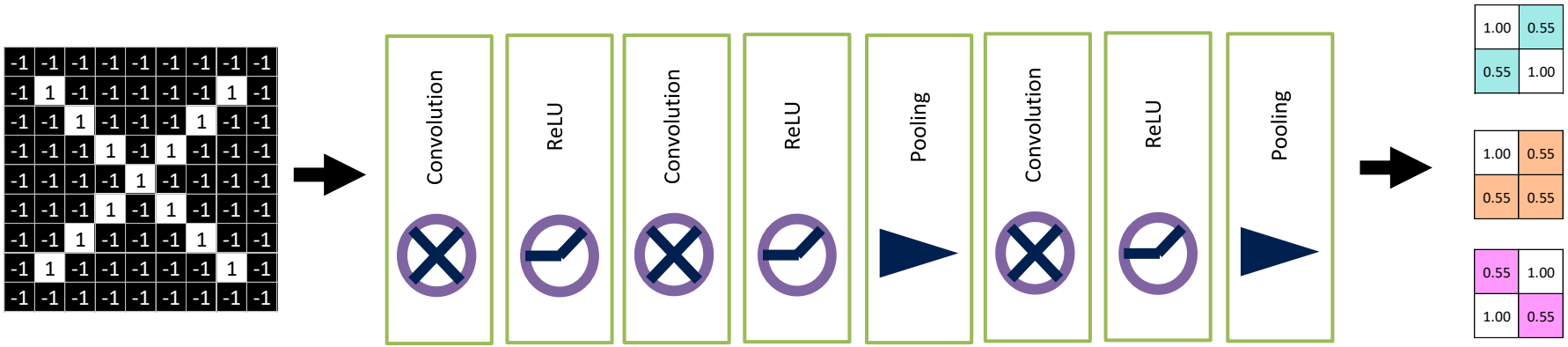
Layers get stacked

The output of one becomes the input of the next.



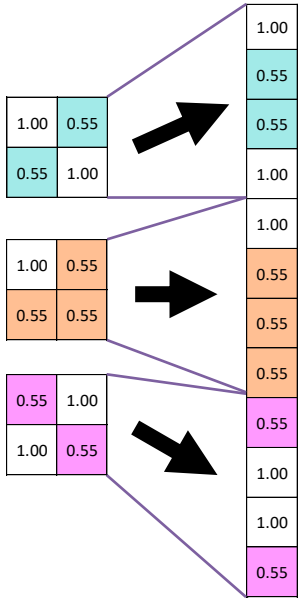
Deep stacking

Layers can be repeated several (or many) times.



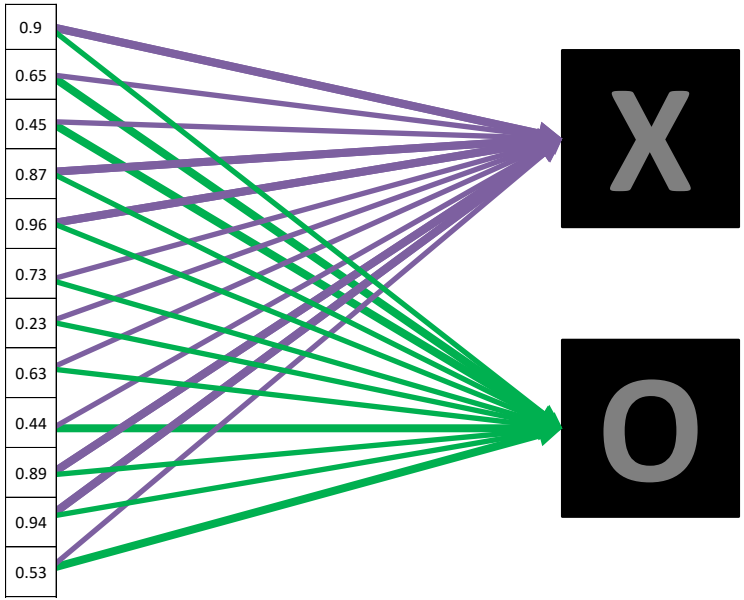
Fully connected layer

Every value gets a vote



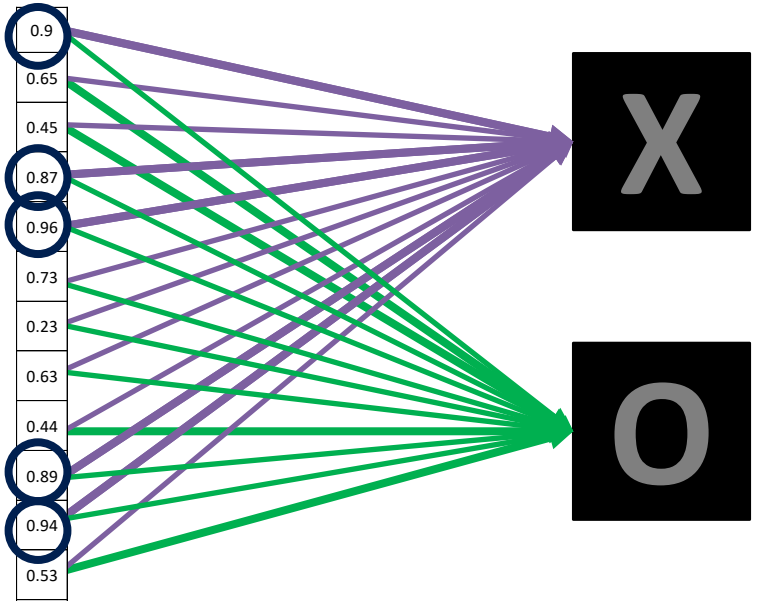
Fully connected layer

Future values vote on X or O



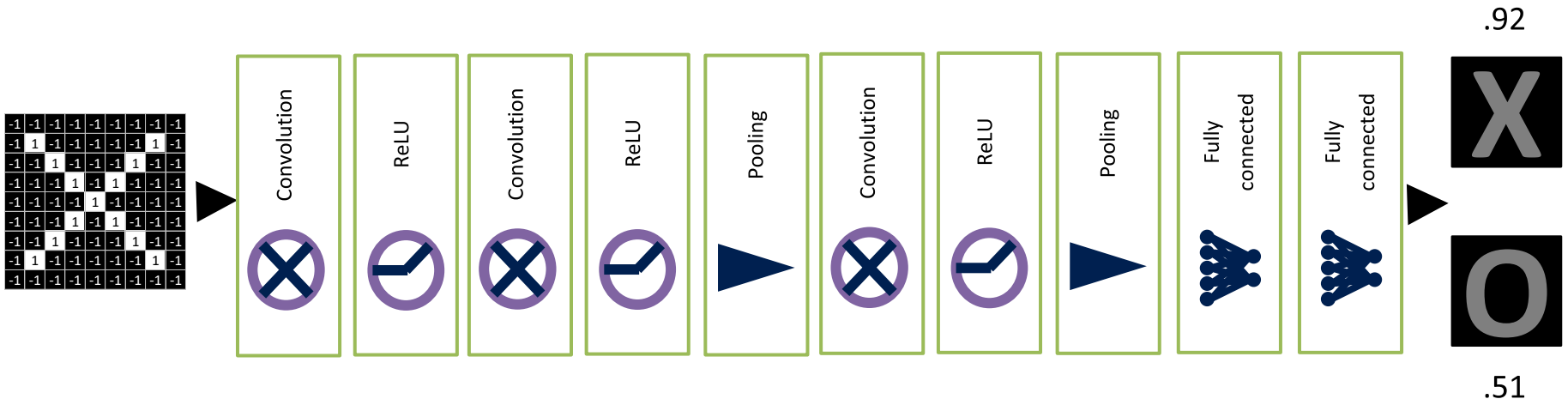
Fully connected layer

Future values vote on X or O



Putting it all together

A set of pixels becomes a set of votes.



Learning

Q: Where do all the magic numbers come from?

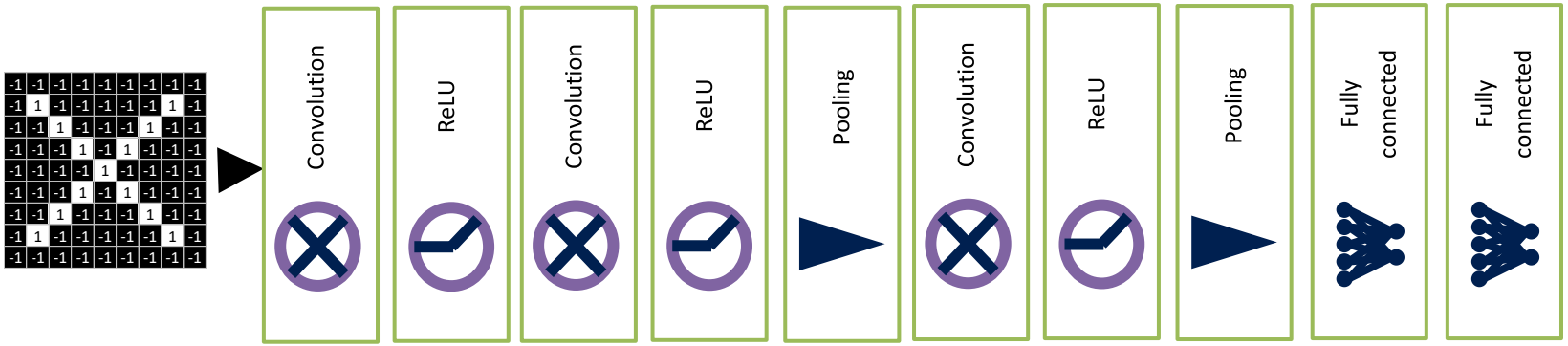
Features in convolutional layers

Voting weights in fully connected layers

A: Backpropagation

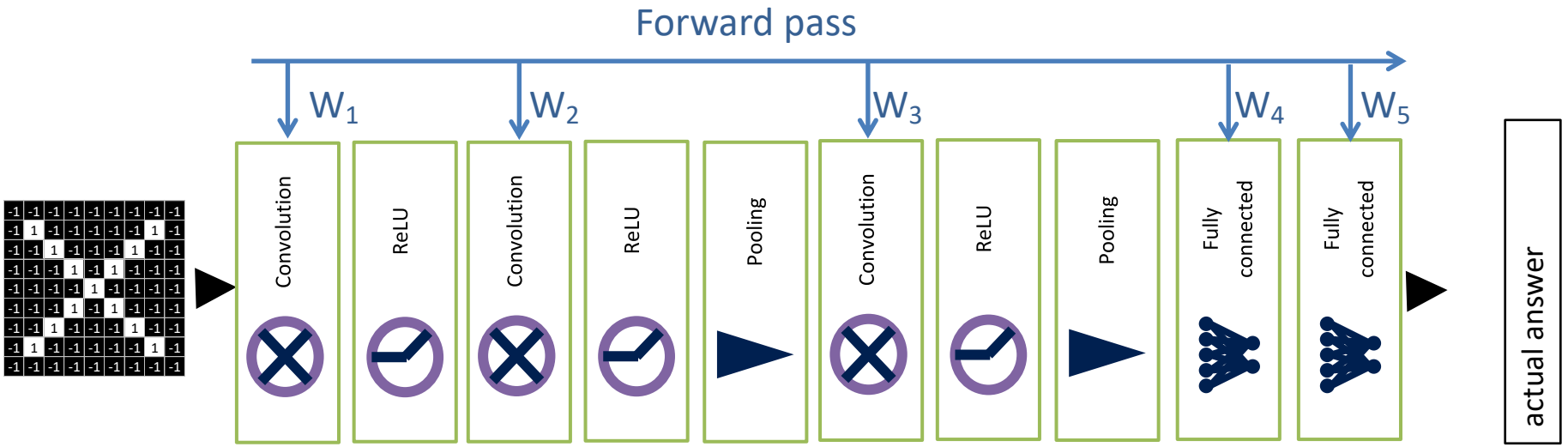
Backpropagation

In the first iteration Wi are set randomly



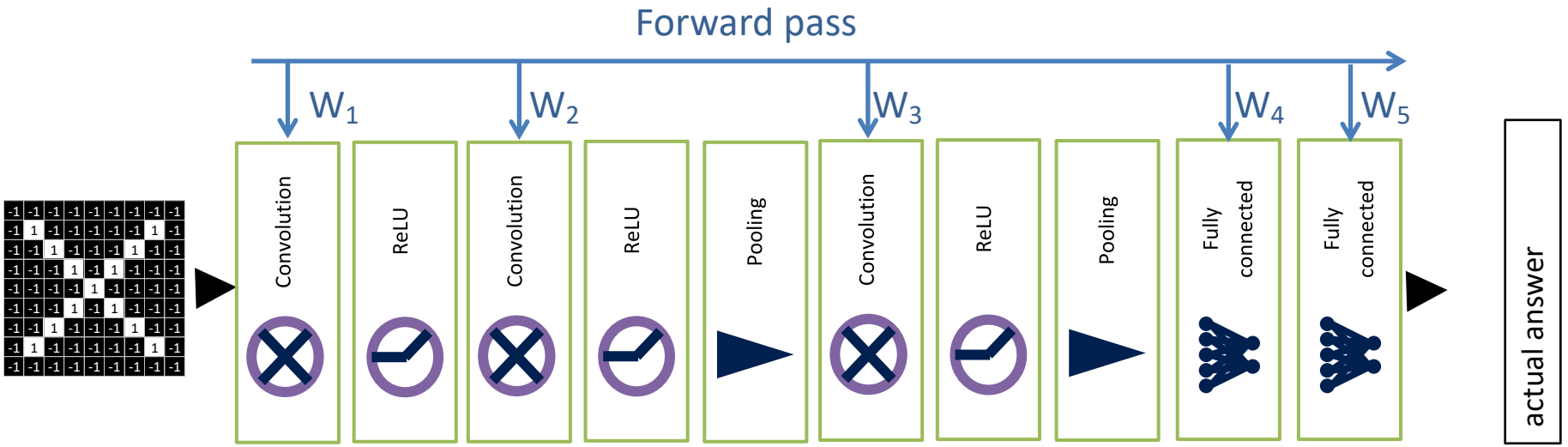
Backpropagation

In the first iteration Wi are set randomly



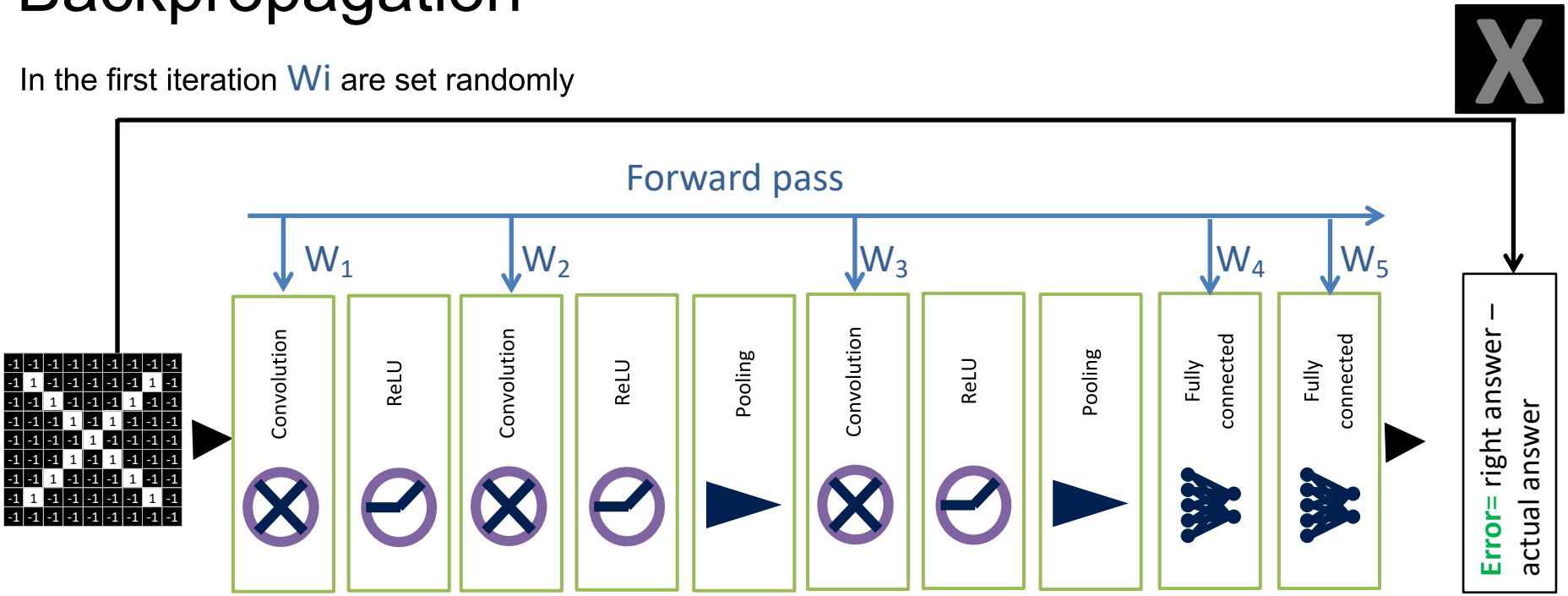
Backpropagation

In the first iteration Wi are set randomly



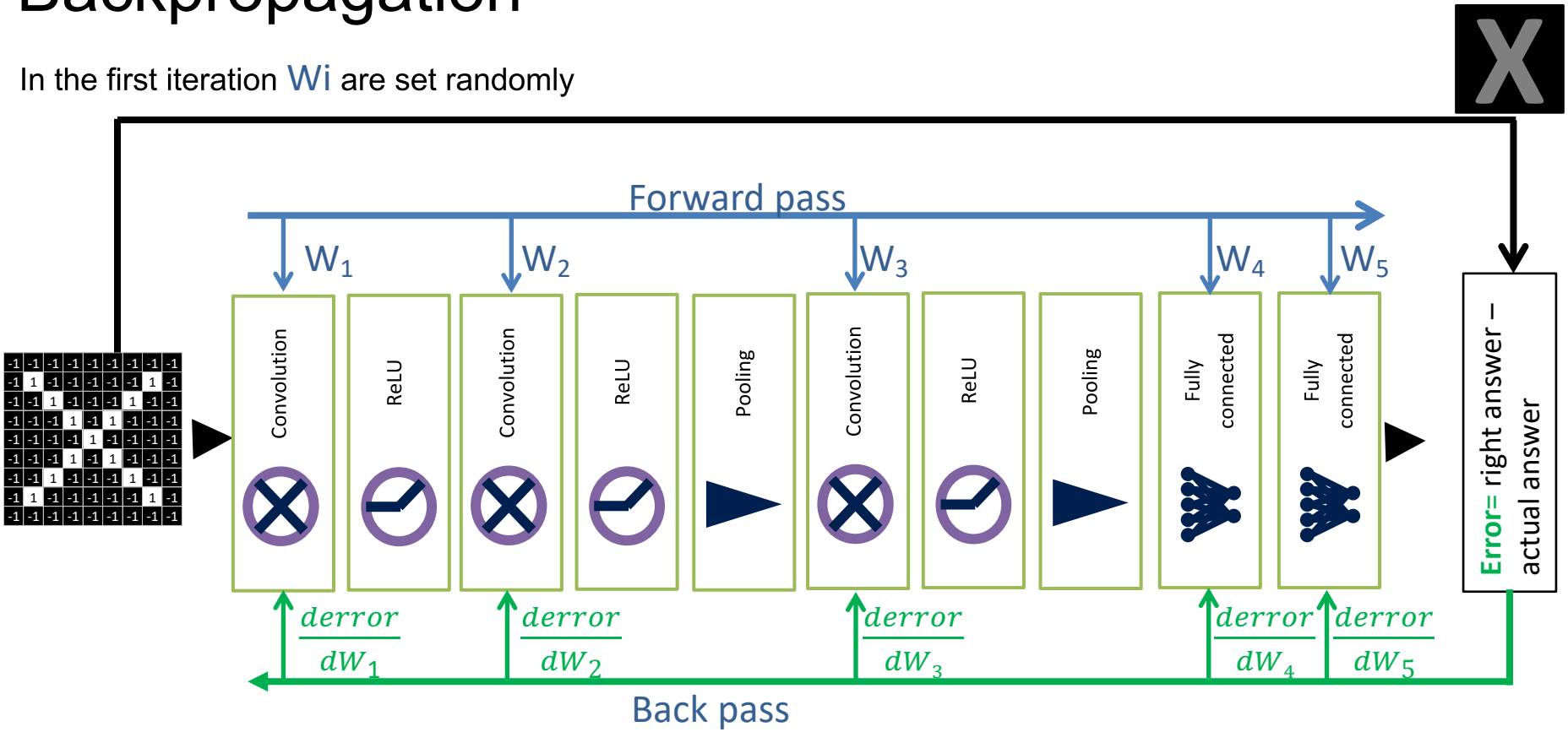
Backpropagation

In the first iteration Wi are set randomly



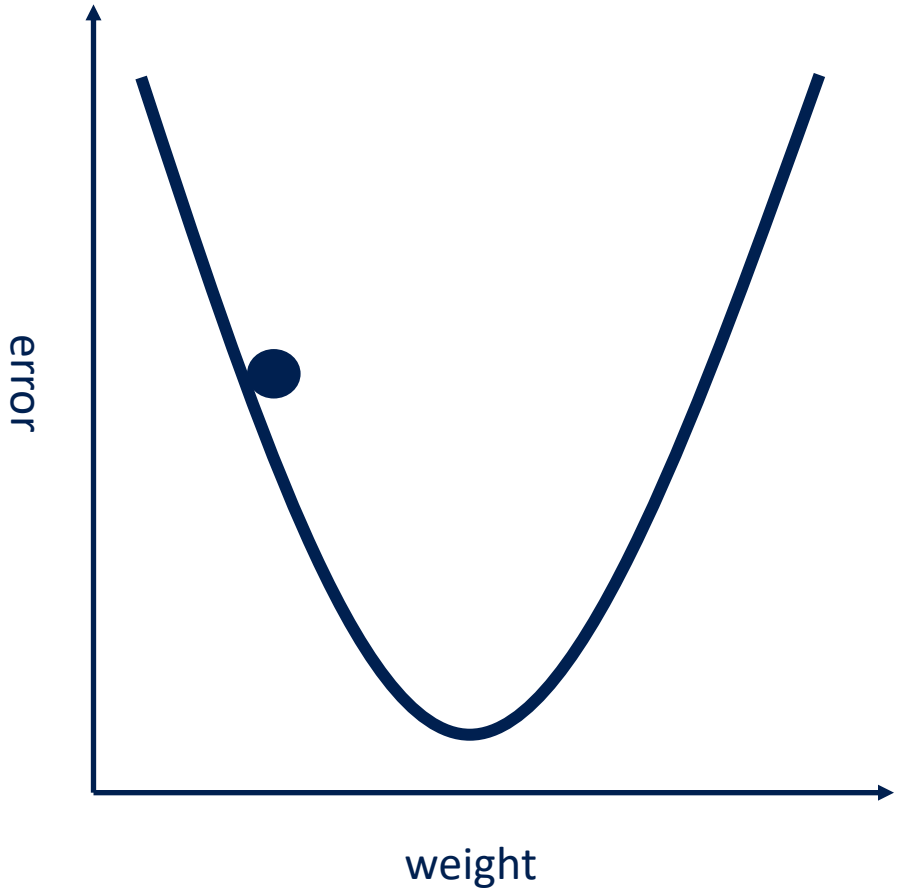
Backpropagation

In the first iteration Wi are set randomly



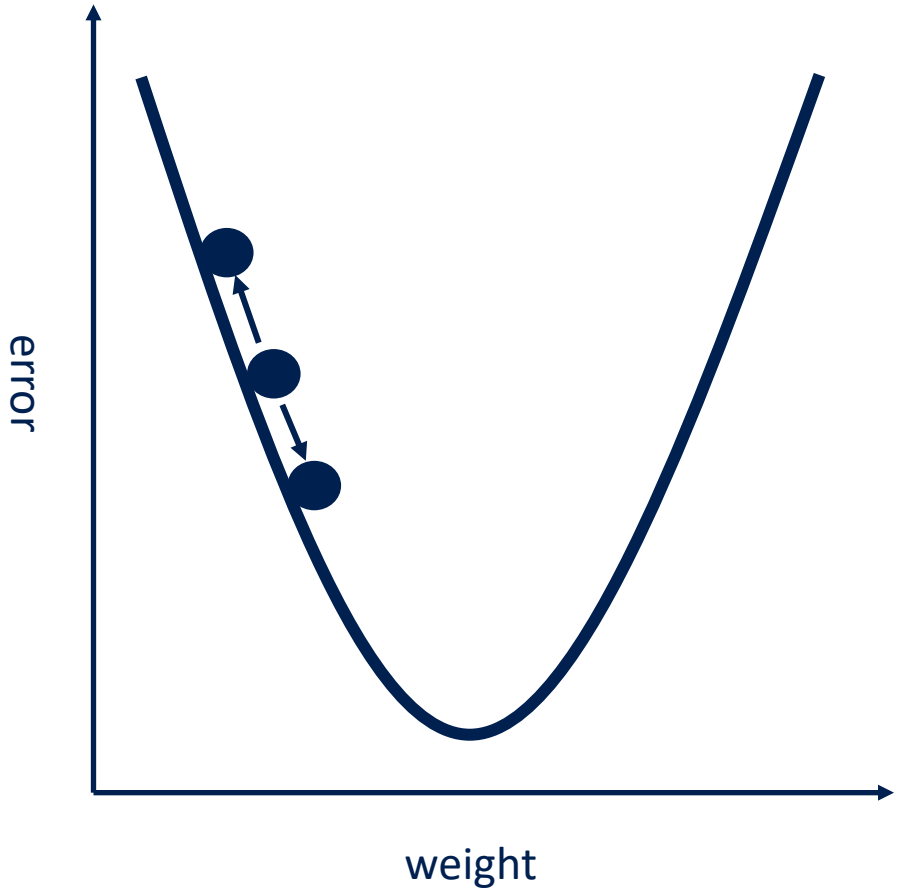
Gradient descent

For each feature pixel and voting weight, adjust it up and down a bit and see how the error changes.



Gradient descent

For each feature pixel and voting weight, adjust it up and down a bit and see how the error changes.



Limitations

- L1: CNNs require large amount and good quality data to get high accuracies: to avoid **underfitting** and **overfitting**
- L2: Deep NNs are not explainable

Practical solutions

- S1: Transfer learning, full fine-tuning, linear-probing

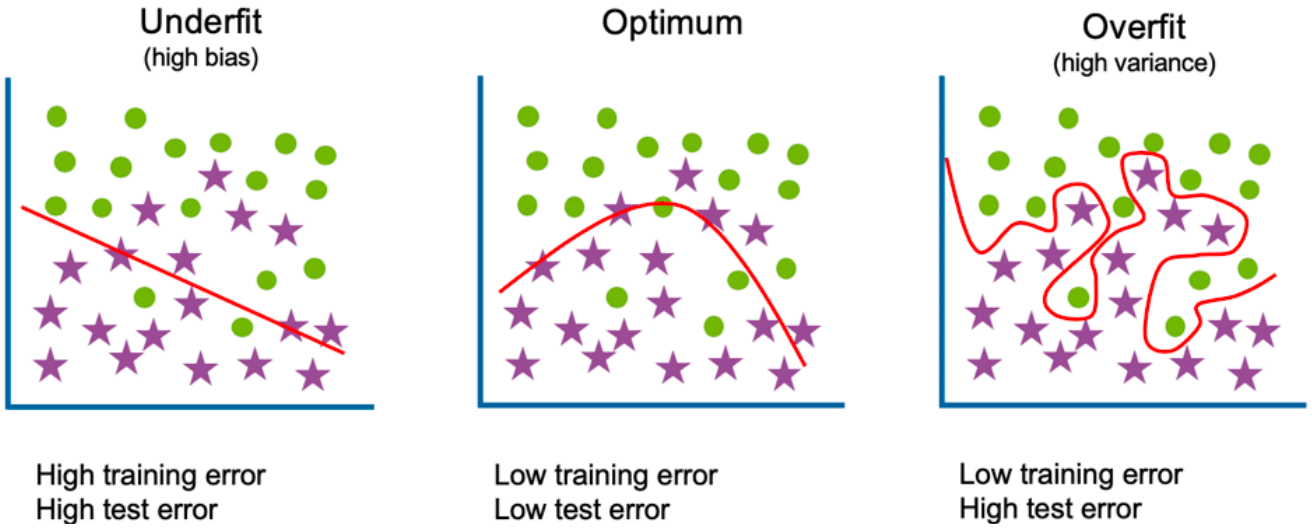
  Data augmentation

  Regulation techniques
- S2: Make Deep NN more explainable

Outline

1. What are CNNs?
2. How CNNs work?
   - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. **Transfer learning**
4. Overfitting vs underfitting
5. Data augmentation
6. Regulation techniques
7. Interpretability

What overfitting and underfitting mean?



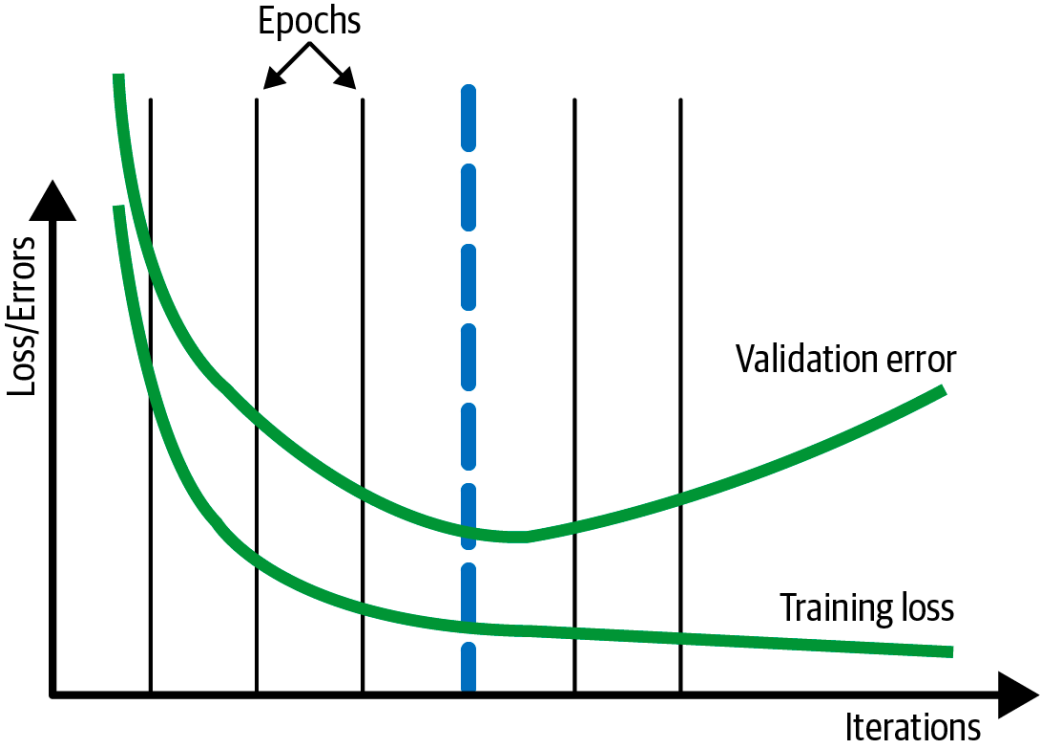
Underfitting	Overfitting
The model is unable to learn the relationship between the data points and target values.	The model tries to learn from the details along noise in the data and tries to fit each data point. The model fails to detect new data points.
Generates a high error in train and test	Generate low error in train and high error in test

When overfitting and underfitting occur?

Underfitting	Overfitting
The training data contains noise	The training data contains noise
The model is too simple.	Model has a high variance
The model is biased.	The model is too complex
Size of the data is not enough	Small training data

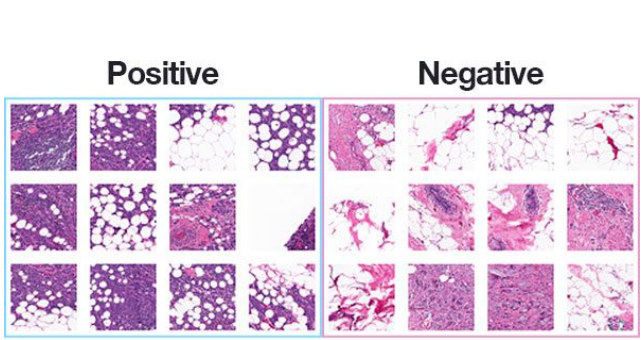
When to stop the training process?

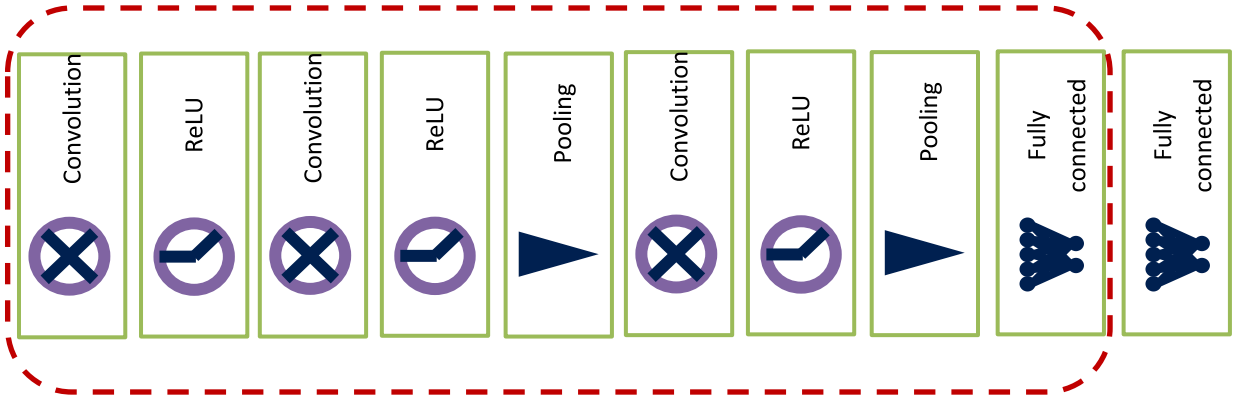
For example, **Early Stopping** is a kind of cross-validation strategy where we keep one part of the training set as the validation set. When we see that the performance on the validation set is getting worse, we immediately stop the training.



Transfer Learning + Fine Tuning

- Given a new problem & new data
- Instead of training from scratch, initializing the the model with the pre-trained weights from problem 1, then retraining all or part of the layers parameters on problem 2

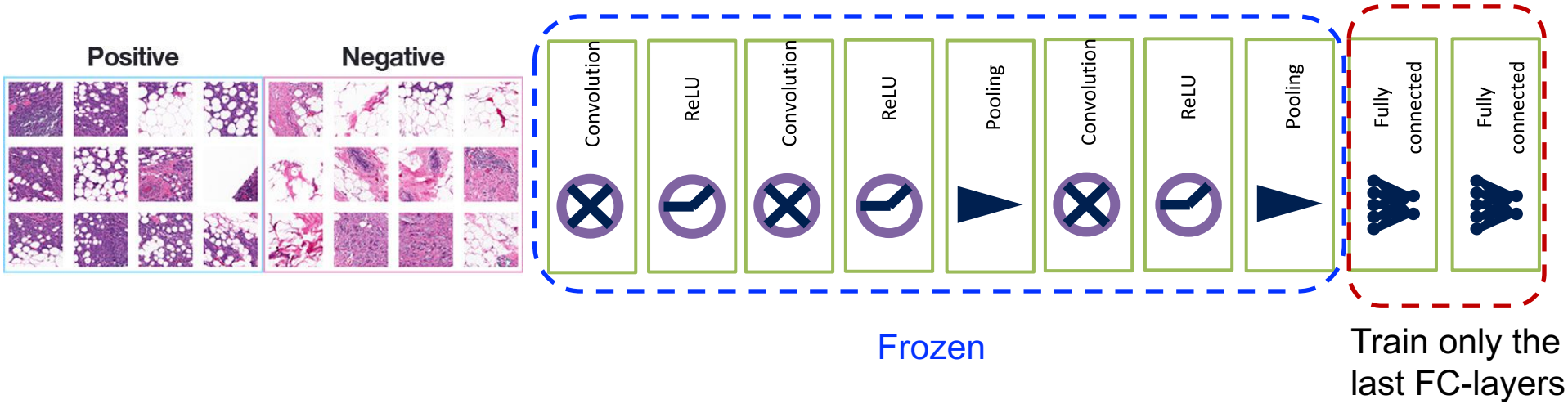




Full fine-tuning = retrain all the layers parameters

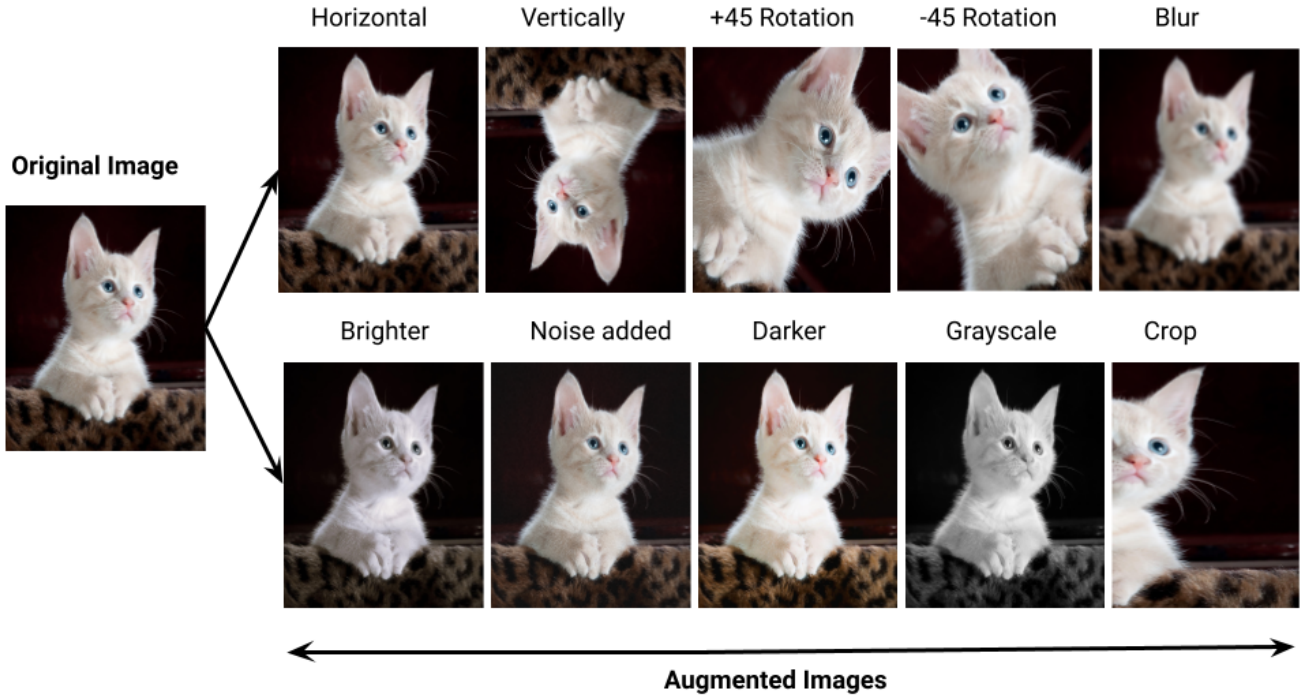
Transfer Learning + Linear Probing

- Given a new problem & new data
- Instead of training from scratch, initializing the the model with the pre-trained weights from problem 1, then retraining all or part of the layers parameters on problem 2

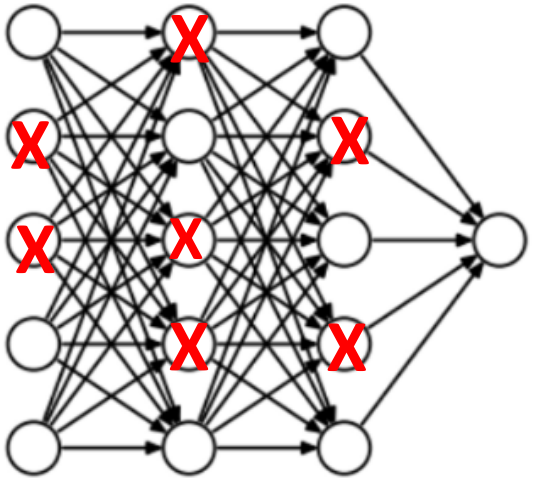


Data augmentation

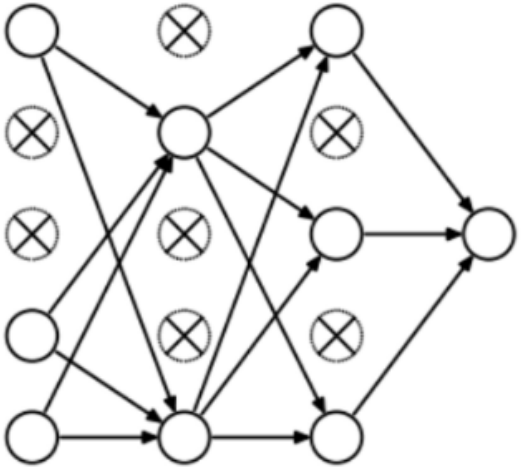
- Increase the training dataset volume artificially using transformations: zooming in/out, rotating along the axis, vertical/horizontal flips, adjusting the brightness and sheer intensity
- Objective: Improve the generalization capacity of the model



Regulation techniques: dropout

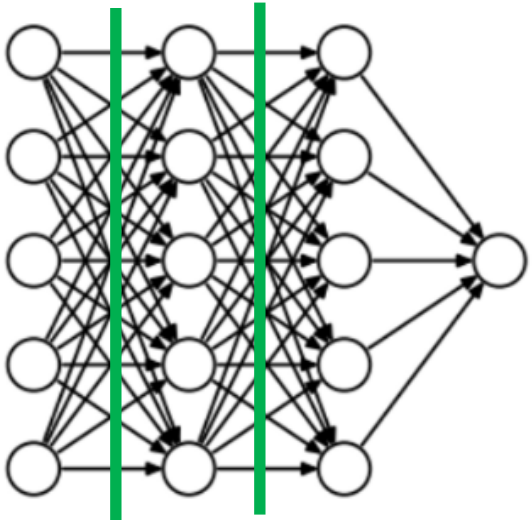


After applying dropout (0.5)

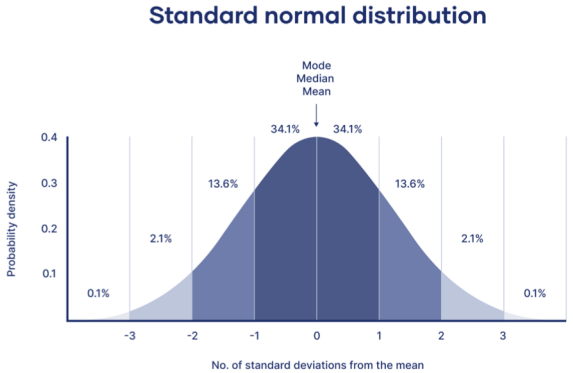


Regulation techniques: BatchNormalization

- Data of very different ranges make convergance very hard
- BN transforms values to make their mean=0 and variance=1



Convert the output to:



➡ layer ➡ x ➡ $\hat{x} = \frac{x - \mu}{\sigma}$ ➡ $y = \gamma\hat{x} + \beta$

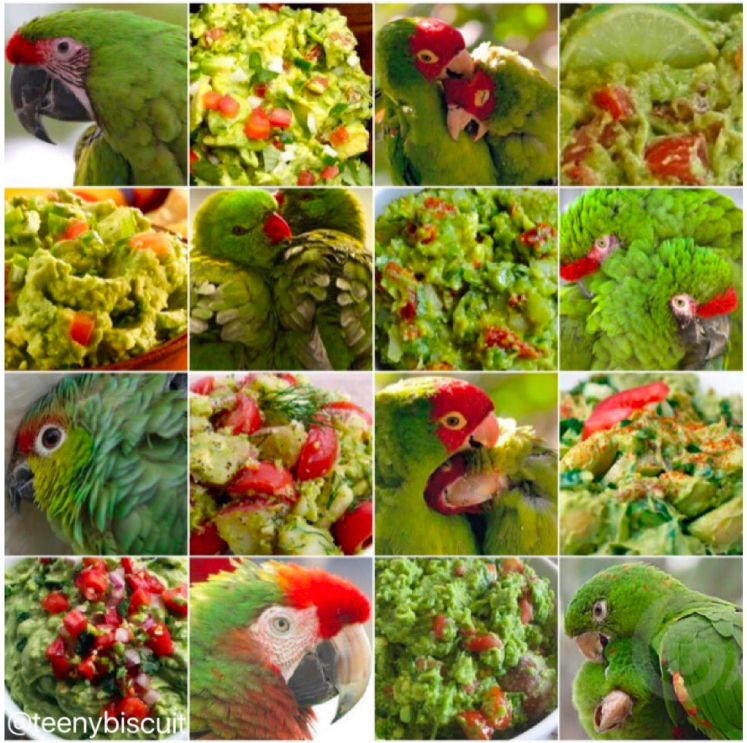
- μ: mean of x in mini-batch
- σ: std of x in mini-batch
- γ: scale
- β: shift

- μ, σ: functions of x, analogous to responses
- γ, β: parameters to be learned, analogous to weights

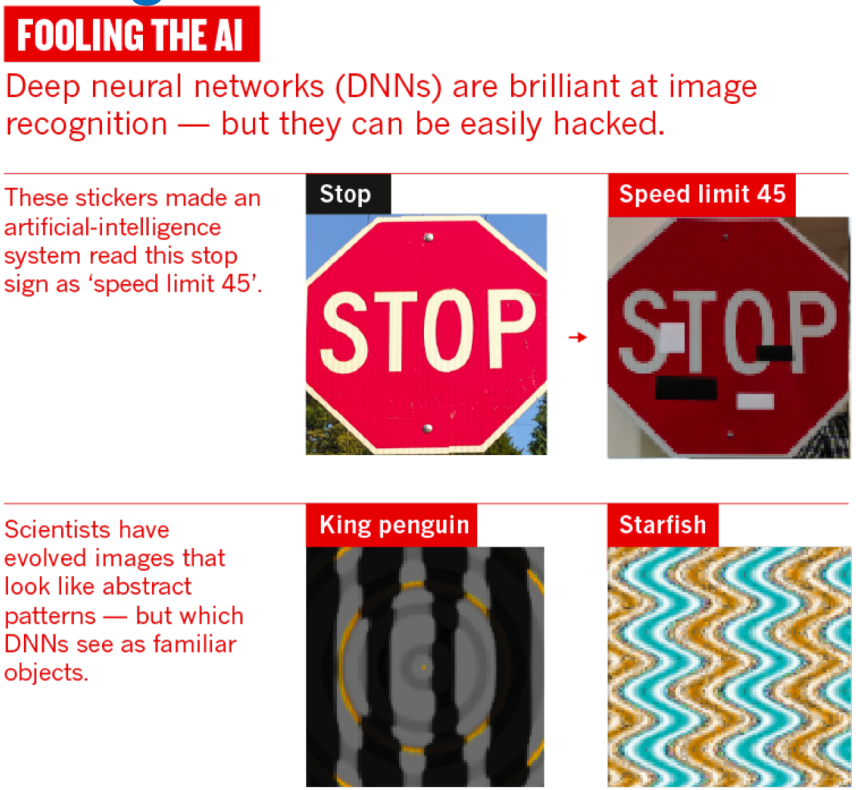
Outline

1. What are CNNs?
2. How CNNs work?
   - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. Overfitting/underfitting
4. Transfer learning
5. Data augmentation
6. Regulation techniques
7. **Interpretability**

CNN can distinguish complex objects

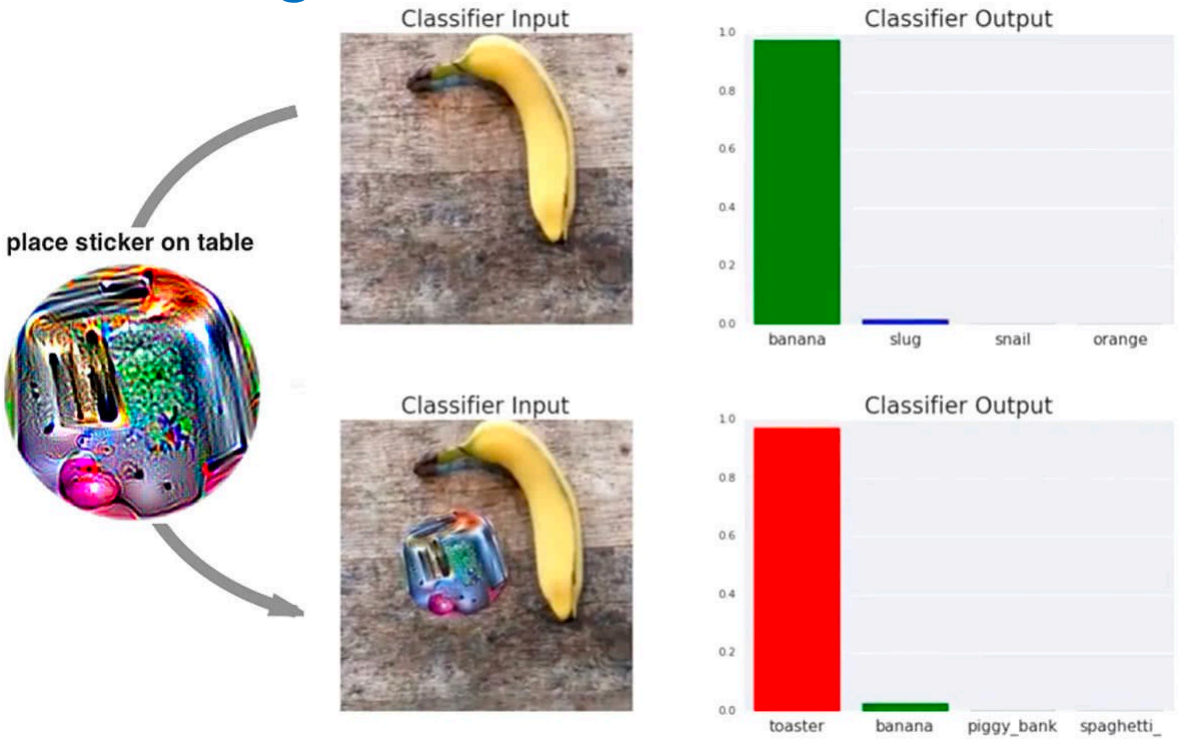


Deep learning can be fooled



Heaven, D. (2019). Why deep-learning AIs are so easy to fool. *Nature*, *574*(7777), 163-166.

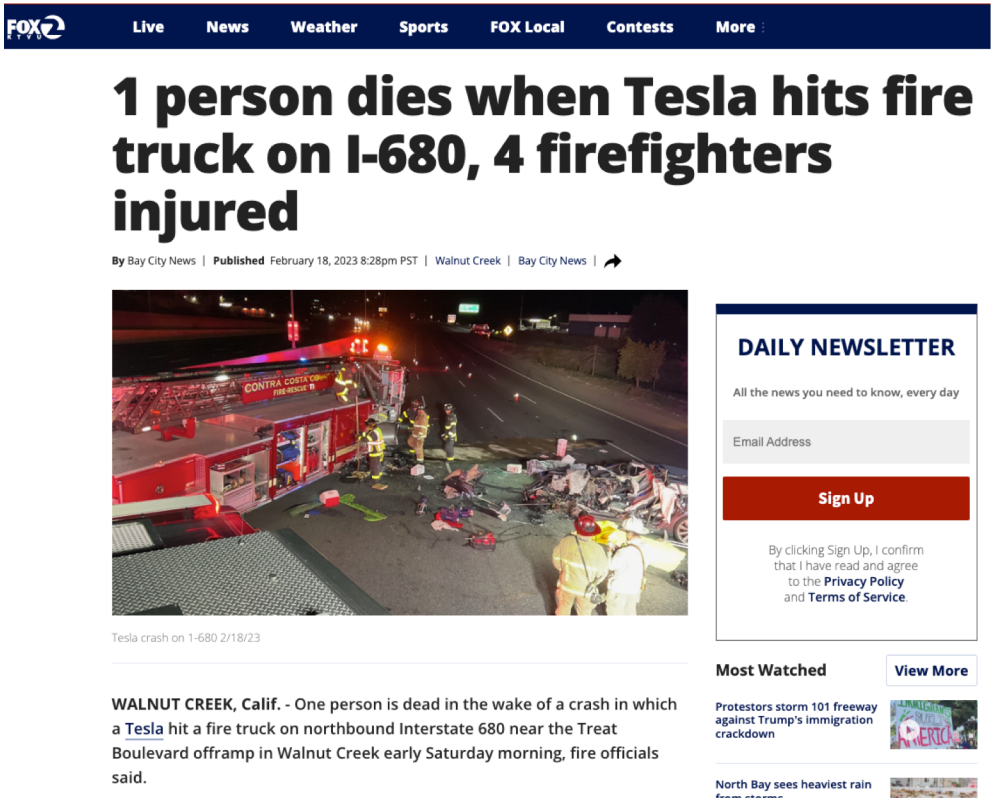
Deep learning can be fooled



Evtimov, I., Eykholt, K., Fernandes, E., Kohno, T., Li, B., Prakash, A., ... & Song, D. (2017). Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*.

Deep learning can fail

- From 2018 to present: 39 accidents around the world
- Of which 27 accident are fatal accidents
- https://en.wikipedia.org/wiki/List_of_Tesla_Autopilot_crashes



FOX 2 KTVU

Live | News | Weather | Sports | FOX Local | Contests | More

1 person dies when Tesla hits fire truck on I-680, 4 firefighters injured

By Bay City News | Published February 18, 2023 8:28pm PST | Walnut Creek | Bay City News |

Tesla crash on 1-680 2/18/23

WALNUT CREEK, Calif. - One person is dead in the wake of a crash in which a Tesla hit a fire truck on northbound Interstate 680 near the Treat Boulevard offramp in Walnut Creek early Saturday morning, fire officials said.

DAILY NEWSLETTER

All the news you need to know, every day

Email Address

Sign Up

By clicking Sign Up, I confirm that I have read and agree to the Privacy Policy and Terms of Service.

Most Watched View More

Protestors storm 101 freeway against Trump's immigration crackdown

North Bay sees heaviest rain

https://www.ktvu.com/news/1-person-dies-when-tesla-hits-fire-truck-on-i-680-4-firefighters-injured

Innovate Healthcare
HealthImaging
FOR LEADERS OPTIMIZING CARE WITH IMAGES, INTELLIGENCE & IT

Algorithm's 'unexpected' weakness raises larger concerns about AI's potential in broader populations

Matt O'Connor | April 05, 2021 | Health Imaging | Artificial Intelligence

A new investigation revealed "unexpected" shortcomings when using a federally cleared artificial intelligence tool to detect intracranial hemorrhages. The findings pushed researchers to call for more standardization when evaluating AI-based clinical decision support platforms.

Why should I trust NN?

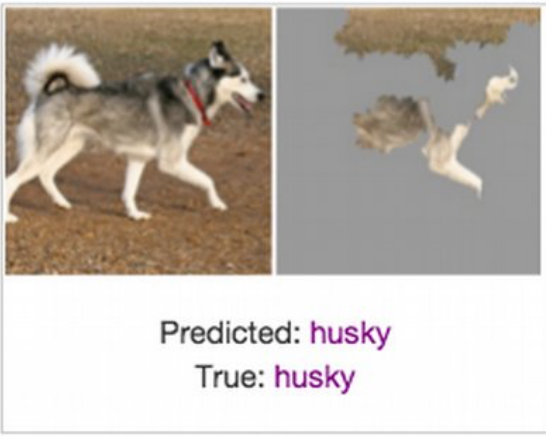
Predicted: wolf
True: husky

Predicted: husky
True: husky

Predicted: wolf
True: wolf

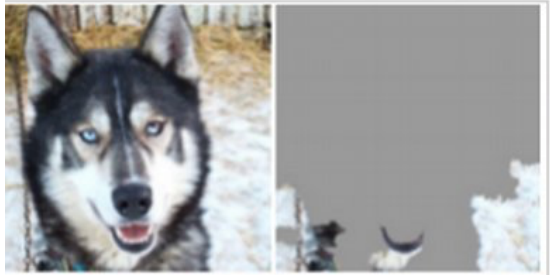
Why should I trust NN?

Predicted: wolf
True: wolf



Predicted: husky
True: husky

Predicted: wolf
True: wolf



Predicted: wolf
True: husky

Predicted: husky
True: husky

Predicted: wolf
True: wolf

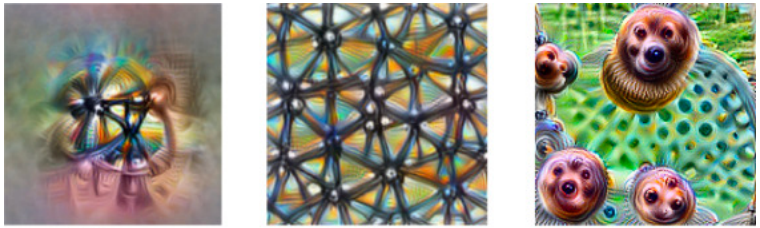
Explainability tools for CNNs

- Who needs to understand the NN? Expert or user?
- Model-specific techniques for post-hoc explainability.
- Explanation by simplification
- Feature relevance explanation
- Visual explanation

Arrieta, A. B et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82-115.

Visual explanation tools for CNNs

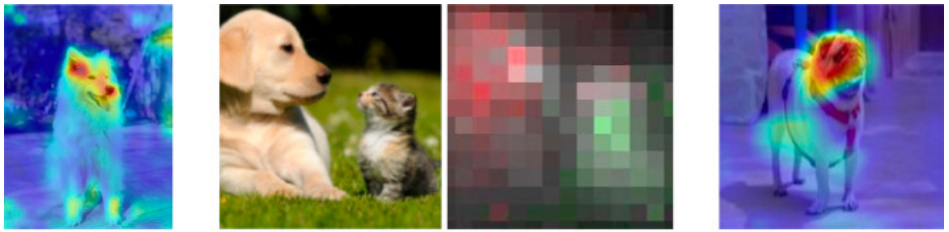
Watch the inside:



(a) Neuron (b) Channel (c) Layer

https://ai.googleblog.com/2018/03/the-building-blocks-of-interpretability.html

Watch the outside:



(a) Heatmap [168] (b) Attribution [293] (c) Grad-CAM [292]

- Saliency maps
- Grad-CAM
- LIME - Local Interpretable Model-Agnostic Explanations

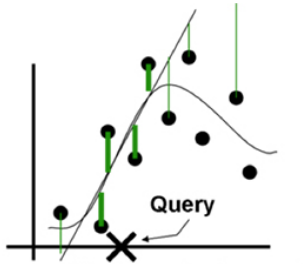
LIME

Original Image

Interpretable Components

P(tree frog) = 0.54

Perturbed Instances	P(tree frog)
	0.85
	0.00001
	0.52



Locally weighted regression

Explanation