

Redes Neuronales Convolucionales

Siham Tabik

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



**UNIVERSIDAD
DE GRANADA**



Outline

1. **What are CNNs?**

2. How CNNs work?

- Convolution layers, pooling layers, FC layers, Gradient, Backpropagation

3. Data augmentation, transfer learning, regulation techniques

4. Explainability

By the way, what is image classification?

Classification



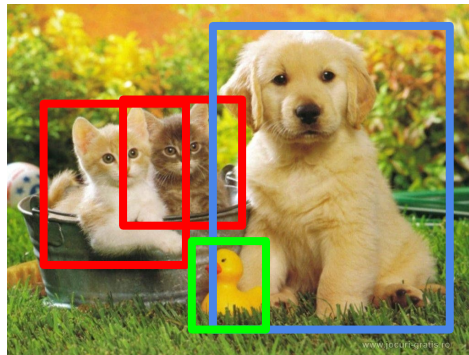
CAT

**Classification
+ Localization**



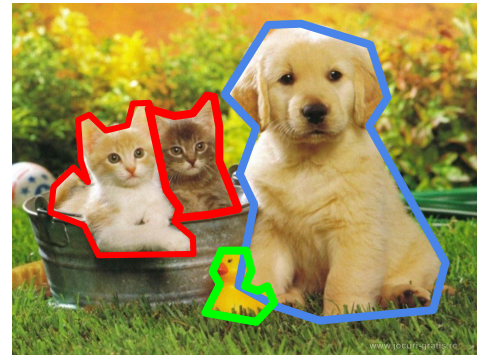
CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**

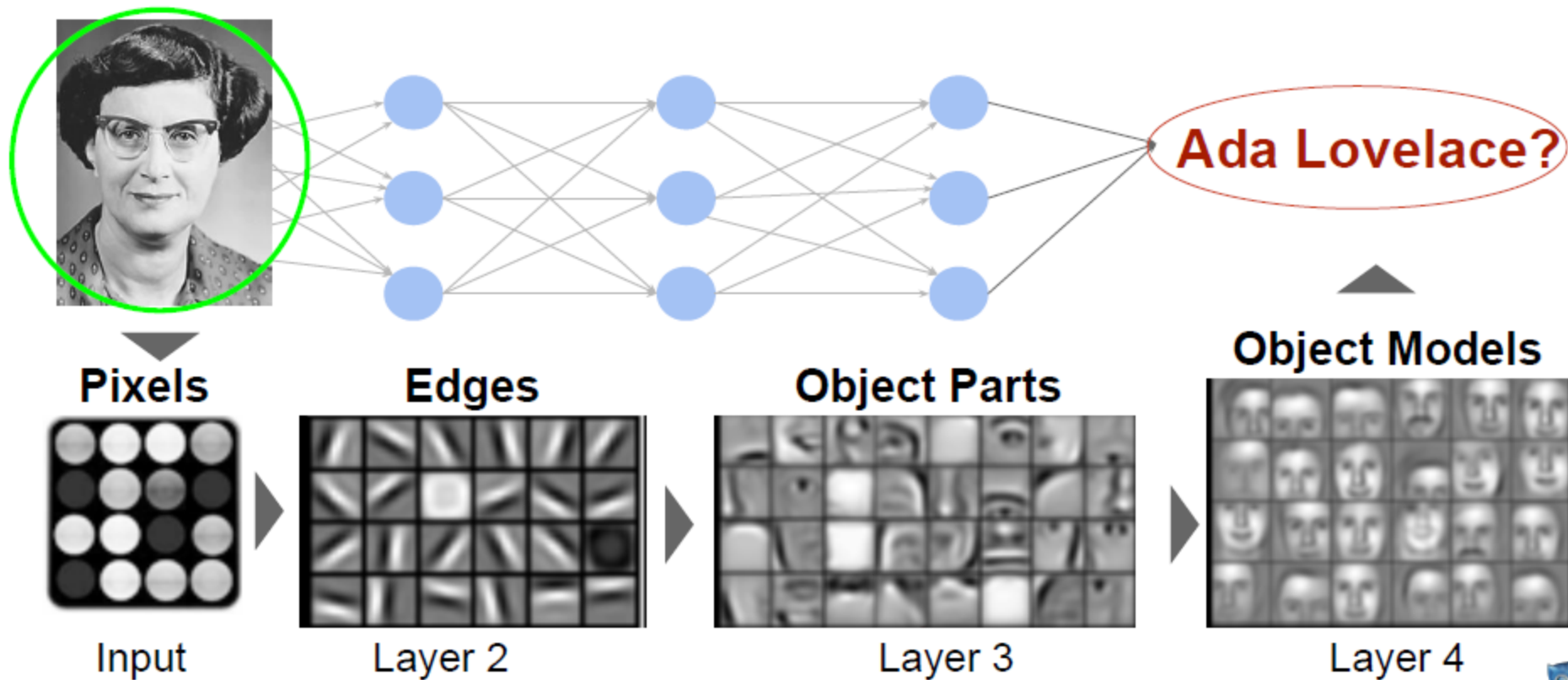


CAT, DOG, DUCK

Single object

Multiple objects

Convolutional Neural Networks



PASCAL Visual Object Classes Challenge (2005-2012)



Aeroplanes



Bicycles



Birds



Boats



Bottles



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



Dining tables



Dogs



Horses



Motorbikes



People



Potted plants



Sheep



Sofas



Trains



TV Monitors

PASCAL Visual Object Classes Challenge



Aeroplanes



Bottles



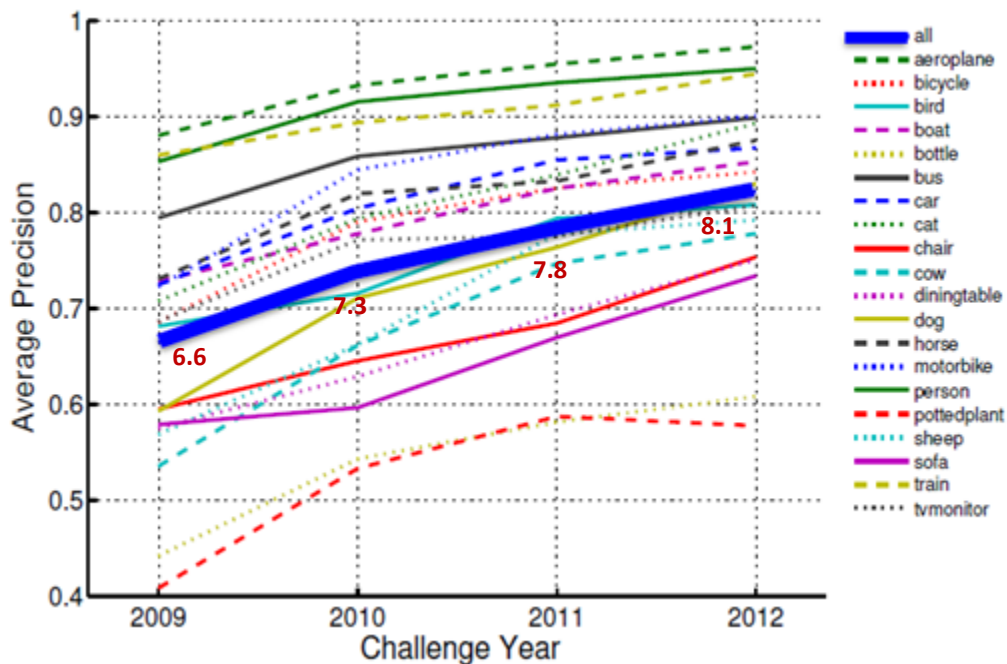
Bus



Dining tables



Potted plants



Cows



People



TV Monitors



Sheep

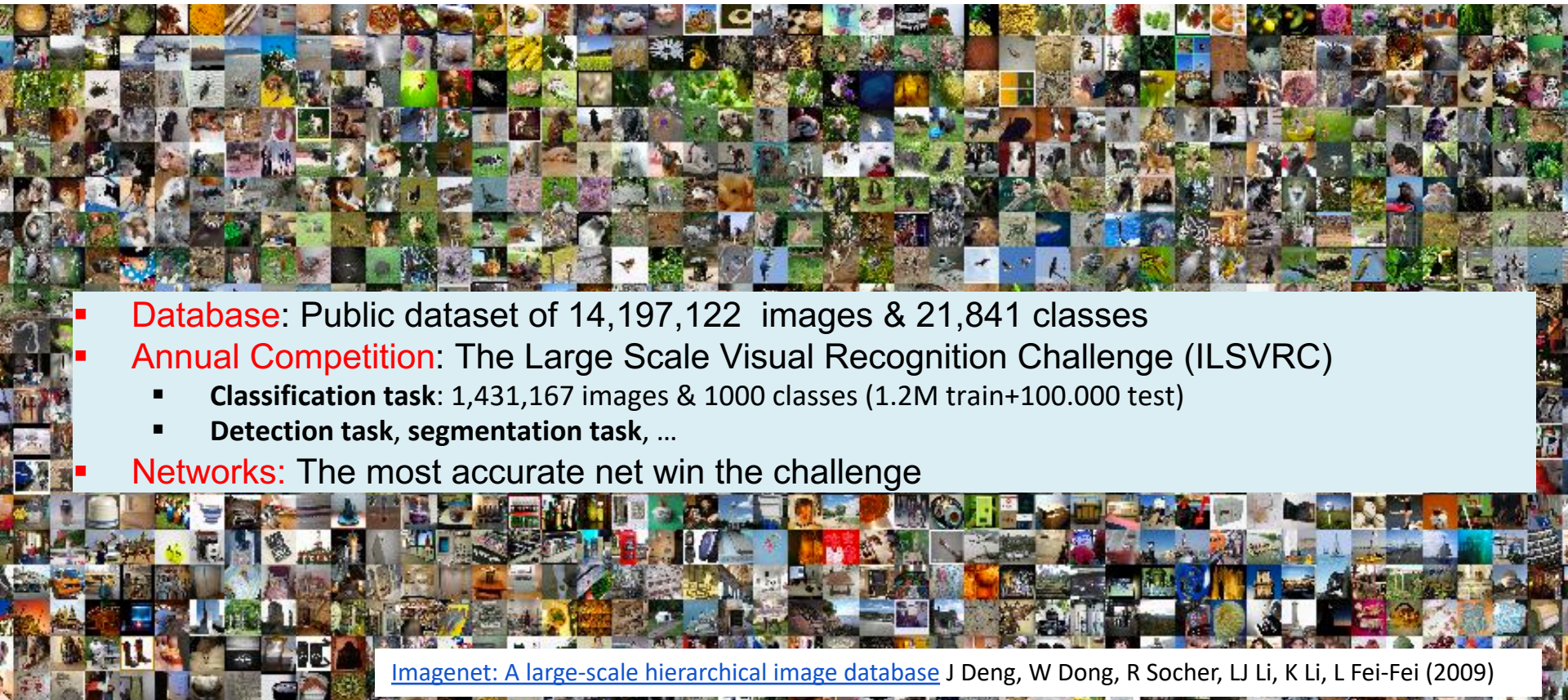


Sofas



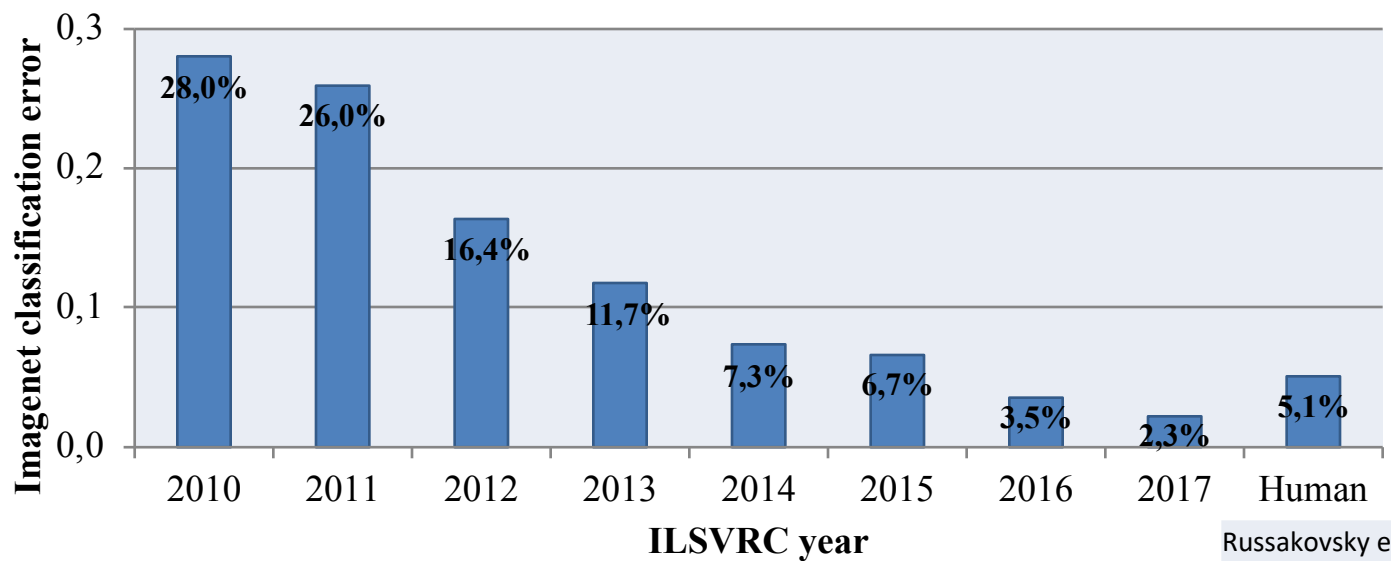
Trains

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


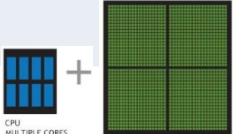
[Imagenet: A large-scale hierarchical image database](#) J Deng, W Dong, R Socher, LJ Li, K Li, L Fei-Fei (2009)

IMAGENET image classification challenge



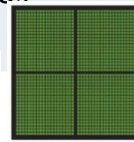
Russakovsky et al. arXiv, 2014

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	1 st massive dataset ImageNet 	
Hardware	Limited capacity	Limited capacity	Limited capacity	Powerful GPUs and CPUs 	



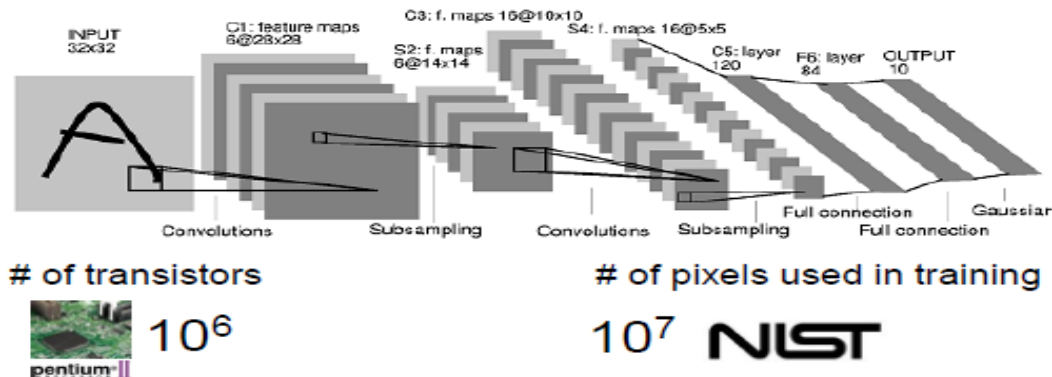
+



CNNs architecture

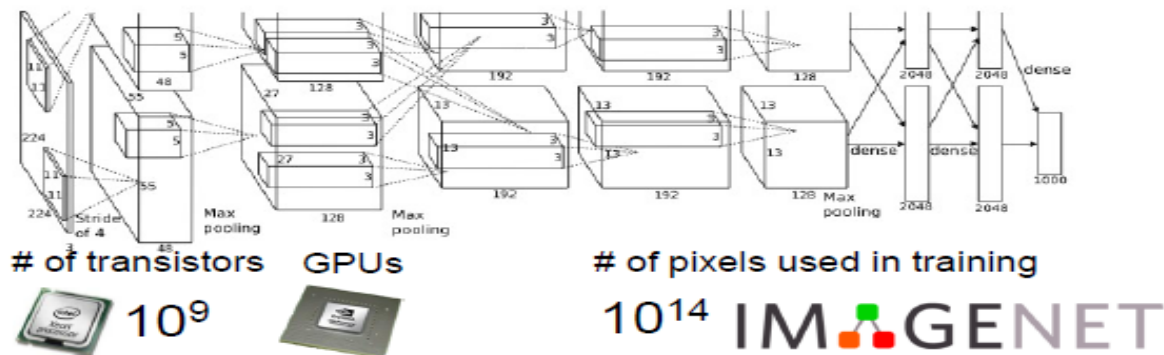
1998

LeCun et al.



2012

Krizhevsky et al.



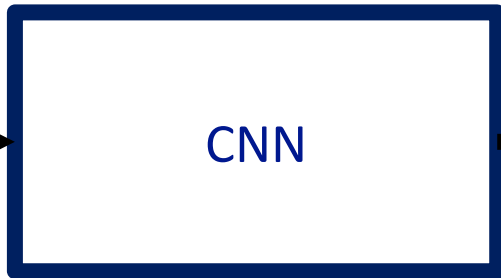
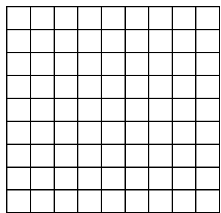
Outline

1. What are CNNs?
- 2. How CNNs work?**
 - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. Data augmentation, transfer learning, regulation techniques
4. Explainability

How CNNs work?

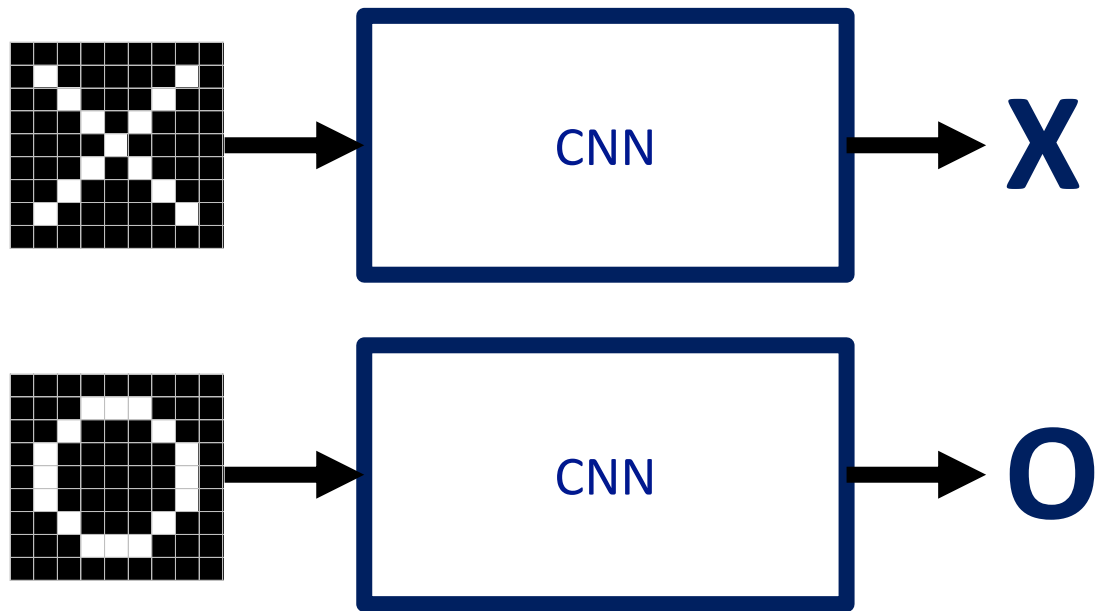
A toy CNN: says whether a picture is of an X or an O

A two-dimensional
array of pixels

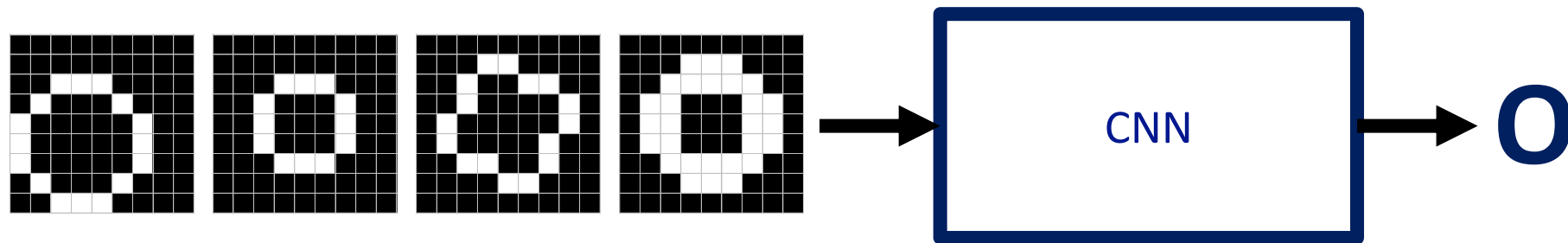
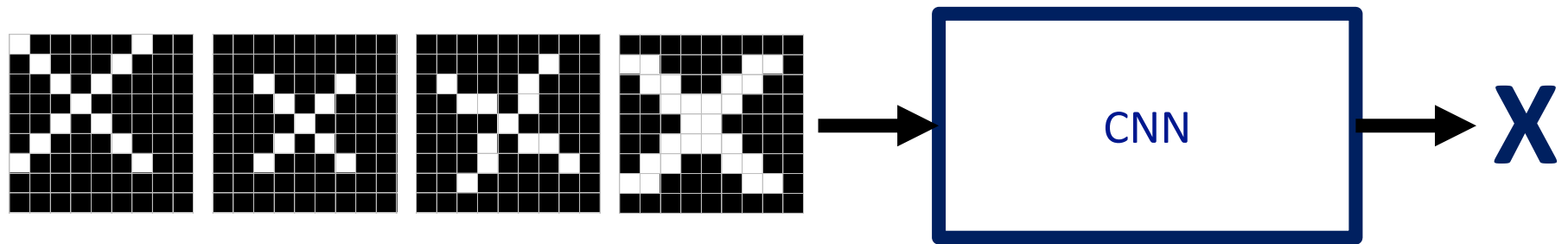


X or **O**

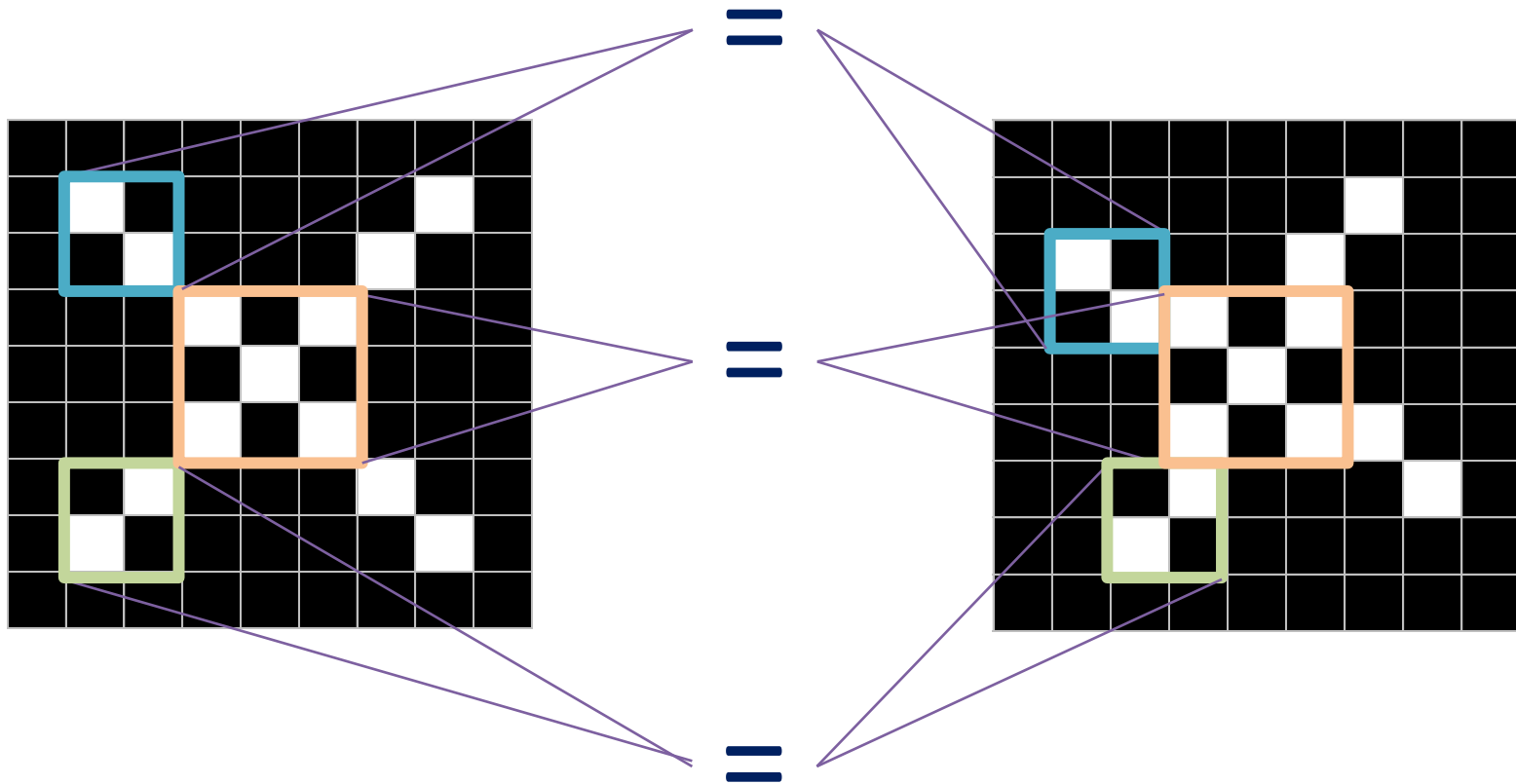
For example



Trickier cases



CNNs match pieces of the image



Features match pieces of the image

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

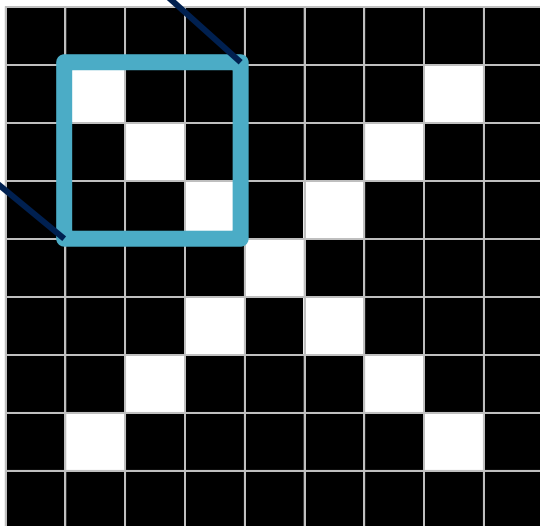
1	-1	1
-1	1	-1
1	-1	1

-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

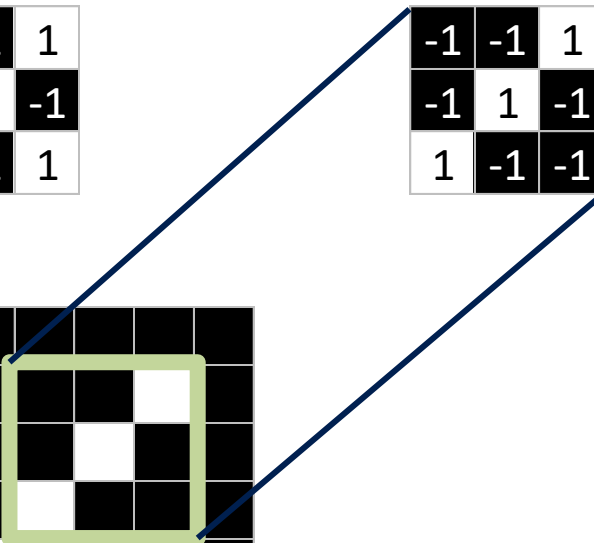
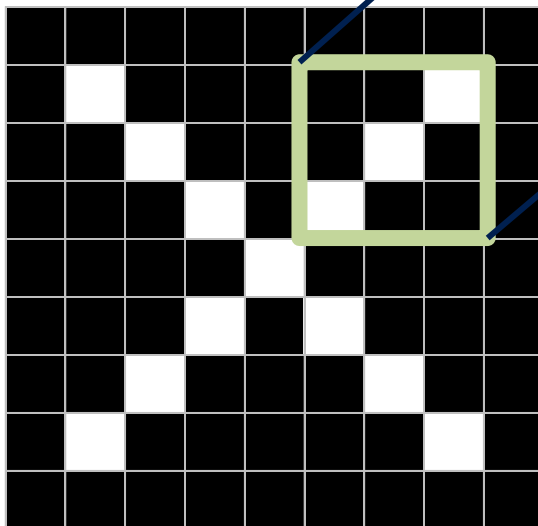
-1	-1	1
-1	1	-1
1	-1	-1



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

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

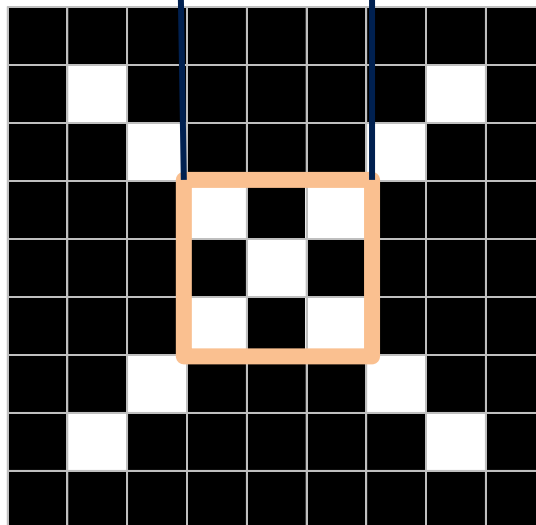
-1	-1	1
-1	1	-1
1	-1	-1



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

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

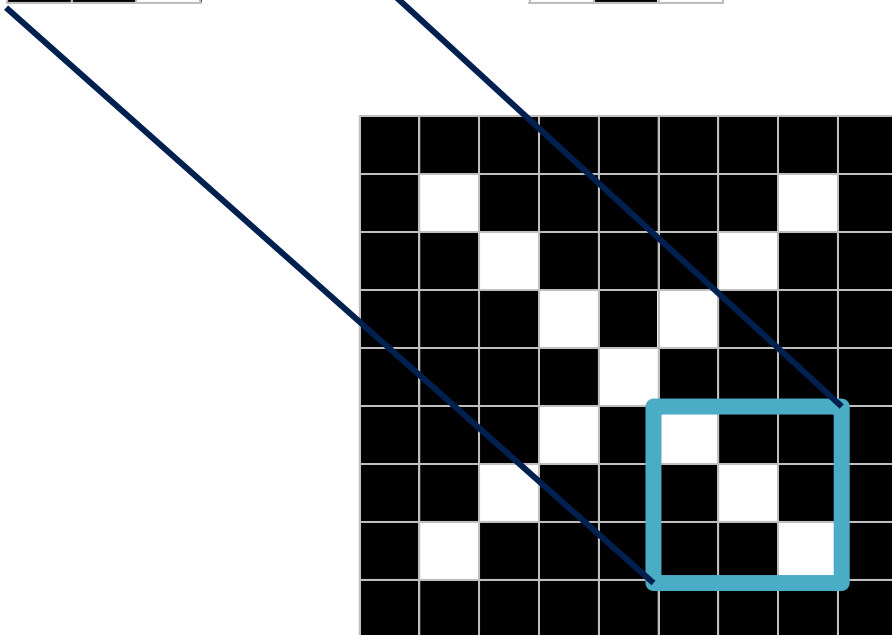
-1	-1	1
-1	1	-1
1	-1	-1



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

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

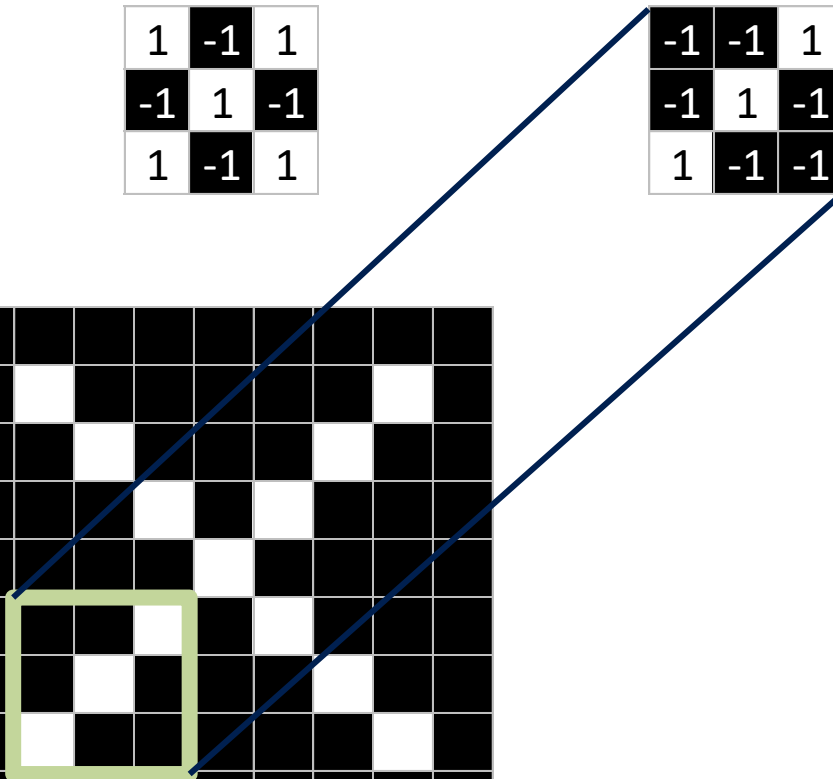
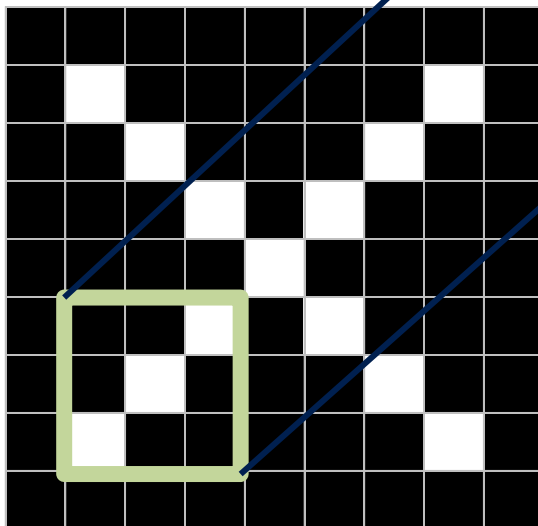
-1	-1	1
-1	1	-1
1	-1	-1



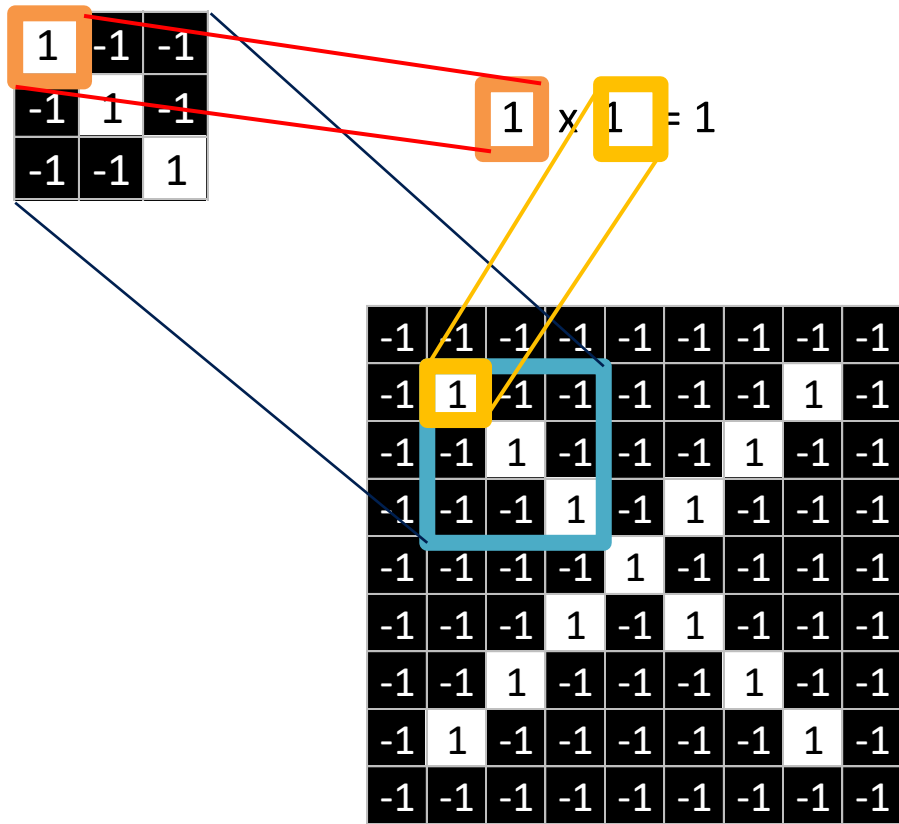
1	-1	-1
-1	1	-1
-1	-1	1

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

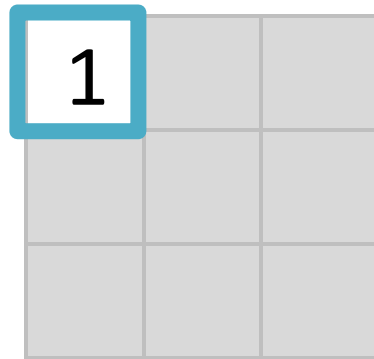
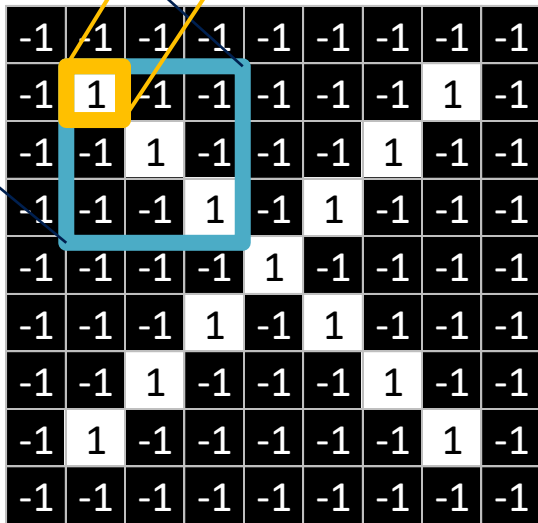
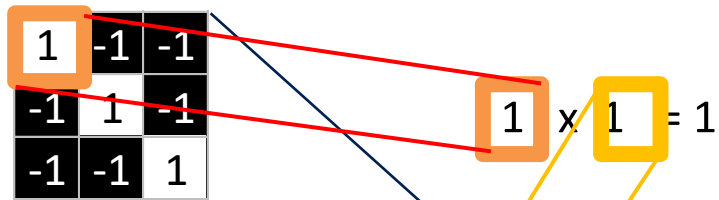
-1	-1	1
-1	1	-1
1	-1	-1



Filtering: The math behind the match



Filtering: The math behind the match



Filtering: The math behind the match

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

$$\begin{array}{|c|} \hline -1 \\ \hline \end{array} \times \begin{array}{|c|} \hline 1 \\ \hline \end{array} = 1$$

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

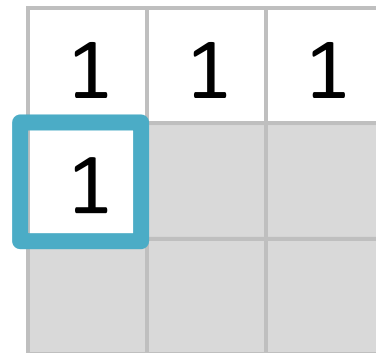
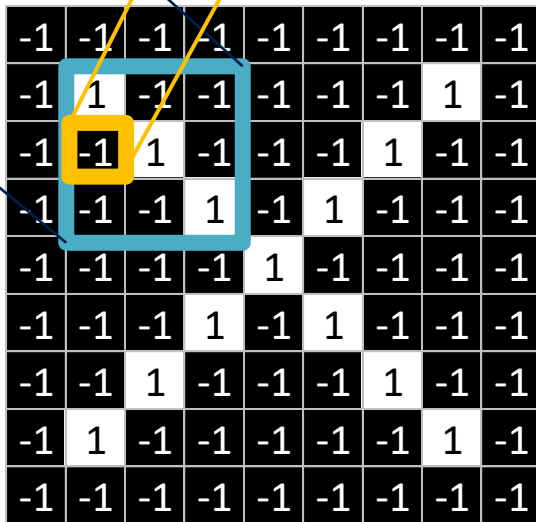
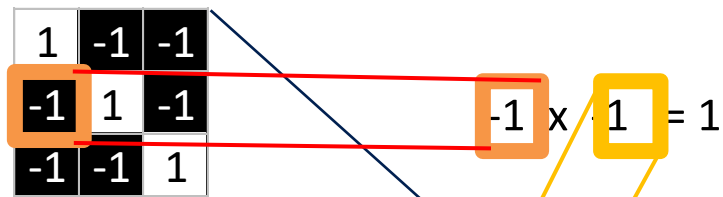
1	-1	-1
-1	1	-1
-1	-1	1

$$\begin{array}{|c|} \hline -1 \\ \hline \end{array} \times \begin{array}{|c|} \hline 1 \\ \hline \end{array} = 1$$

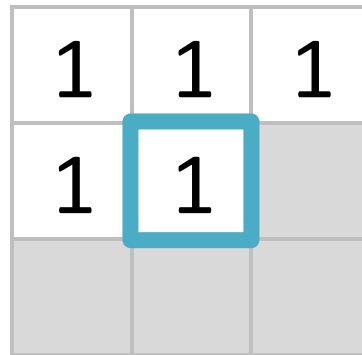
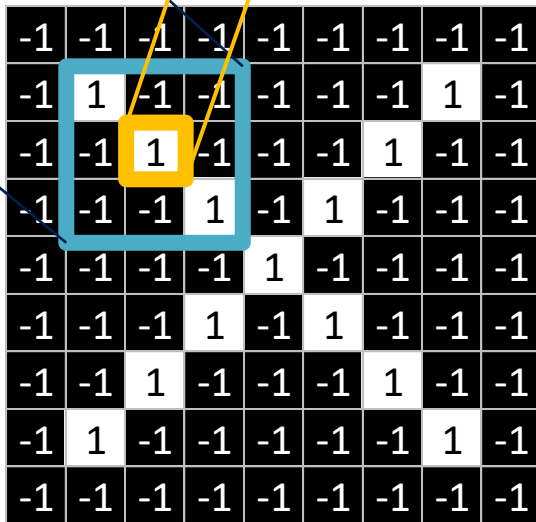
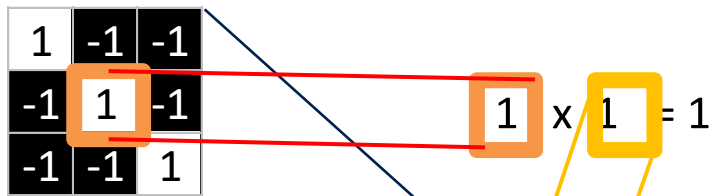
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

1	1	1

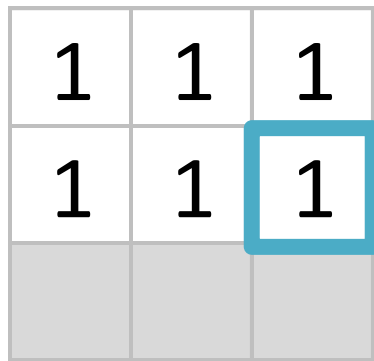
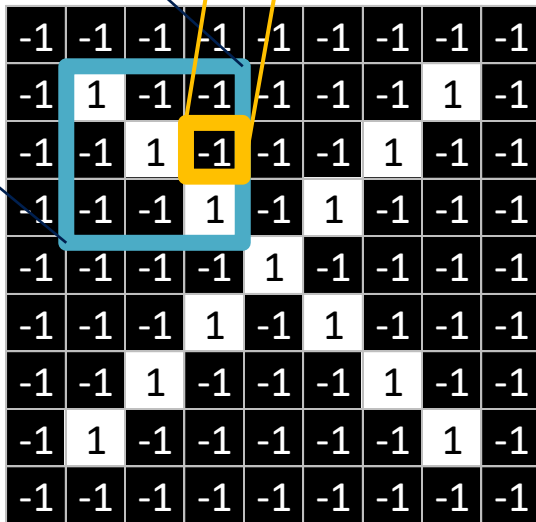
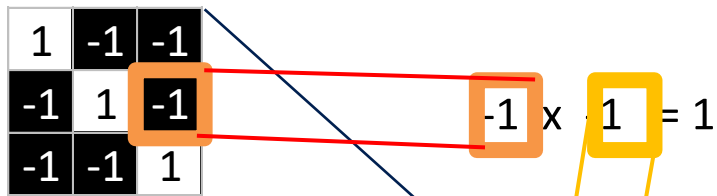
Filtering: The math behind the match



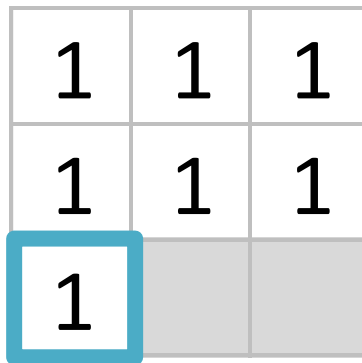
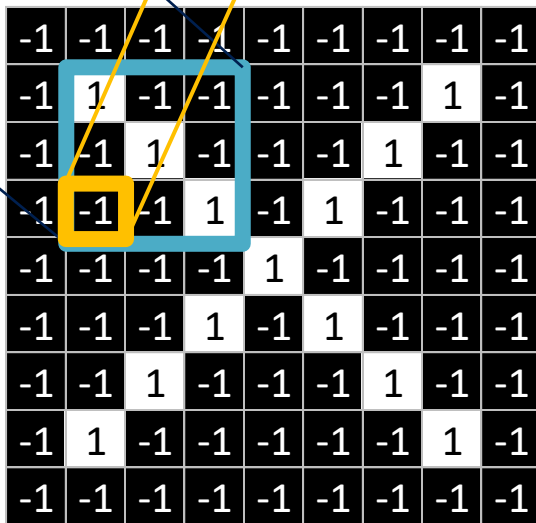
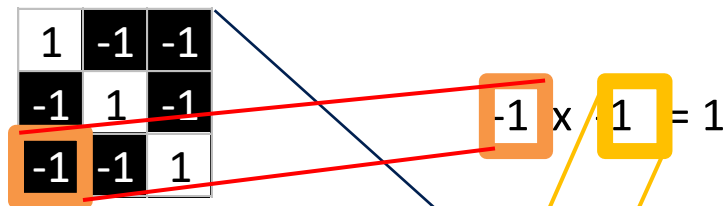
Filtering: The math behind the match



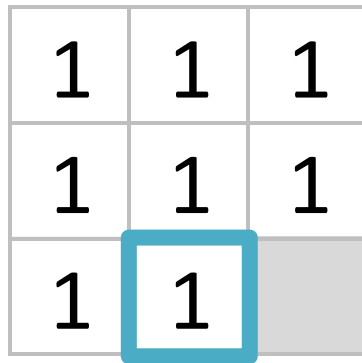
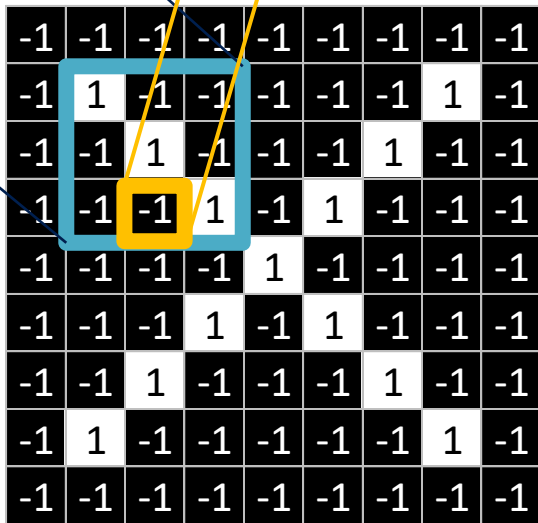
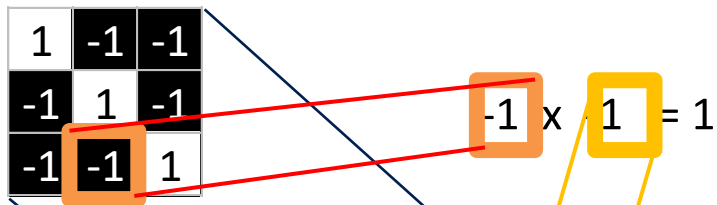
Filtering: The math behind the match



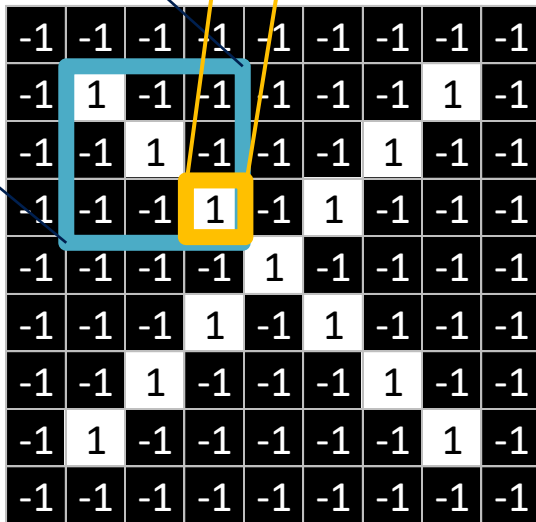
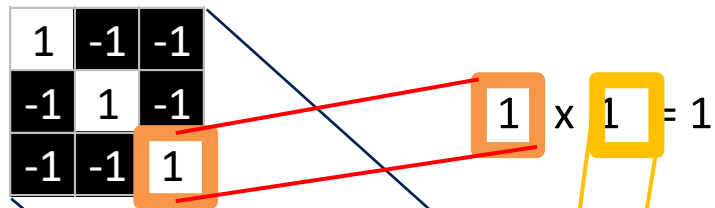
Filtering: The math behind the match



Filtering: The math behind the match



Filtering: The math behind the match



1	1	1
1	1	1
1	1	1

Filtering: The math behind the match

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

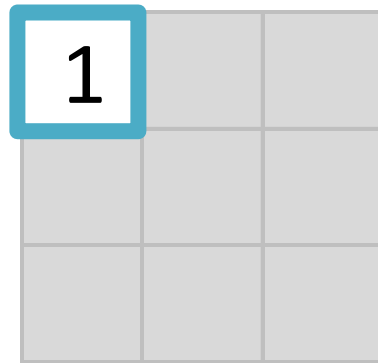
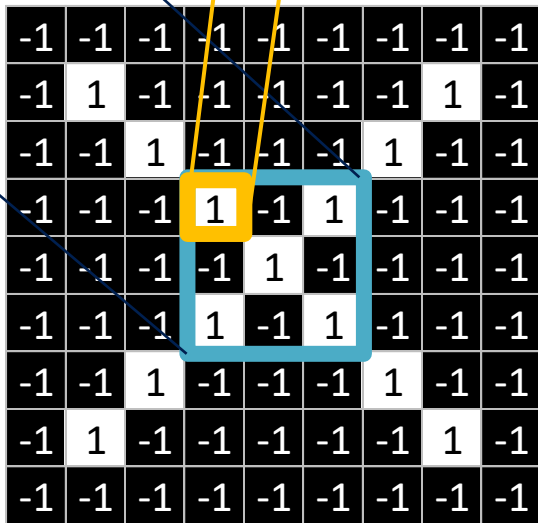
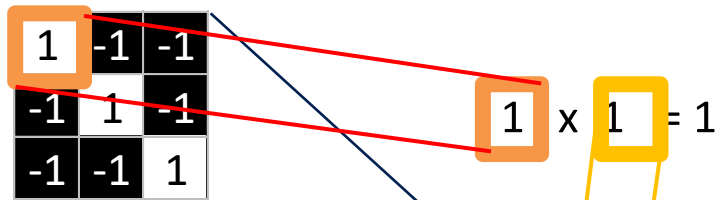
1	1	1
1	1	1
1	1	1

$$\frac{1+1+1+1+1+1+1+1+1}{9} = 1$$

A 9x9 grid representing a sparse matrix A . The diagonal elements are all 1, and the off-diagonal elements are -1. A blue box highlights the 3x3 submatrix starting from row 2, column 2.

A 10x10 grid with a light gray background and thin gray grid lines. A blue square with a thick border is positioned in the top-left corner, covering the first four columns and the first four rows. Inside the blue square, in the second column and second row, is a white square containing the black number '1'. Two blue diagonal lines extend from the top-right corner of the blue square across the grid.

Filtering: The math behind the match



Filtering: The math behind the match

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

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

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

1	1	-1
1	1	1
-1	1	1

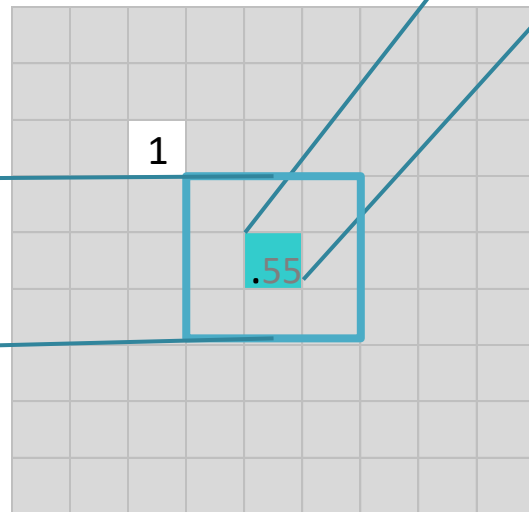
Filtering: The math behind the match

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

1	1	-1
1	1	1
-1	1	1

$$\frac{1+1-1+1+1+1-1+1+1}{9} = .55$$

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



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



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

=

0.33	-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
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1



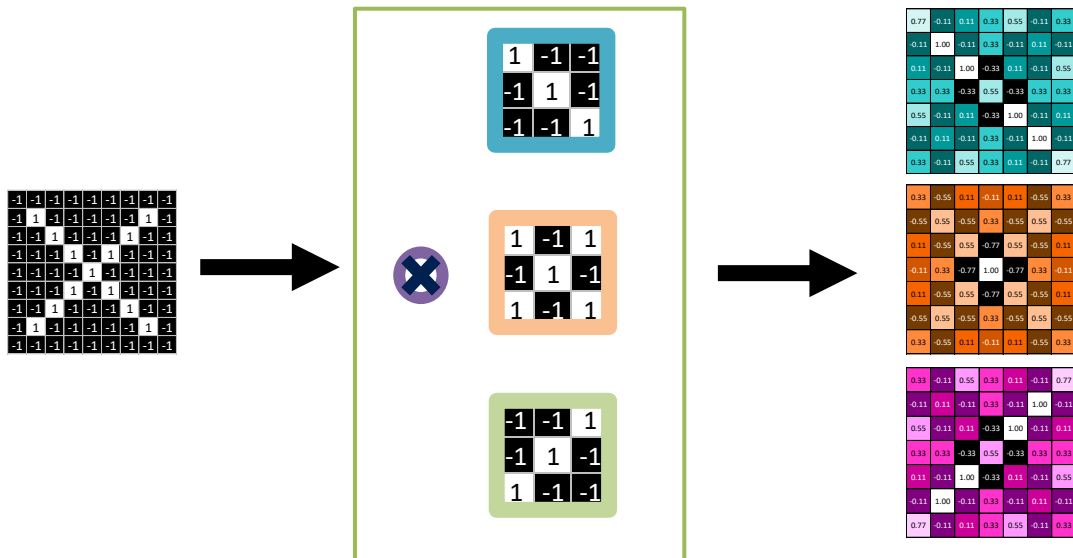
-1	-1	1
-1	1	-1
1	-1	-1

=

0.33	-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

-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1
-1	-1	1	-1	1	1	-1
-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	1	1	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
0.33	-0.55	0.11	0.11	0.11	-0.55	0.33
0.55	0.33	-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	0.33	-0.55	-0.11	-0.55	0.55	0.55
0.33	-0.55	0.11	-0.11	0.11	-0.55	0.33
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

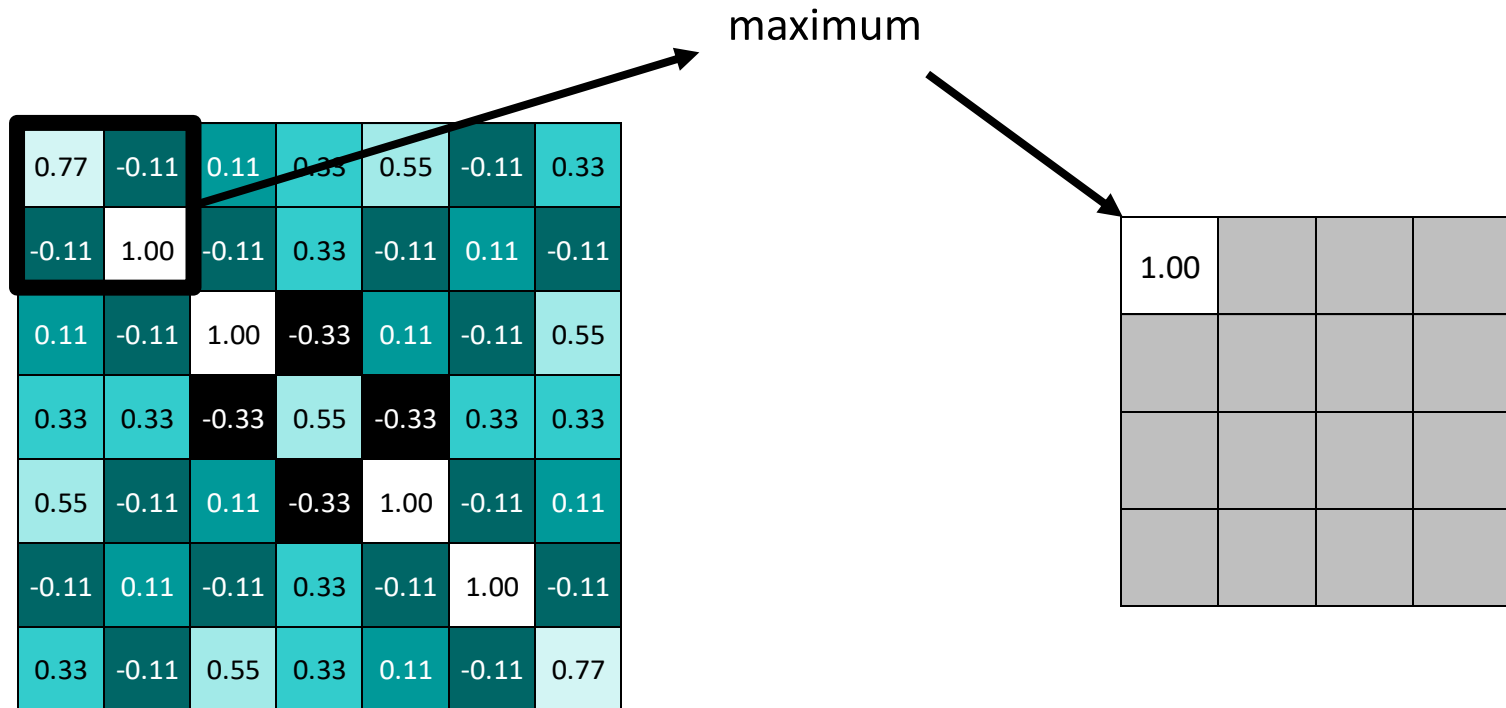
Pooling: Shrinking the image stack

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



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		

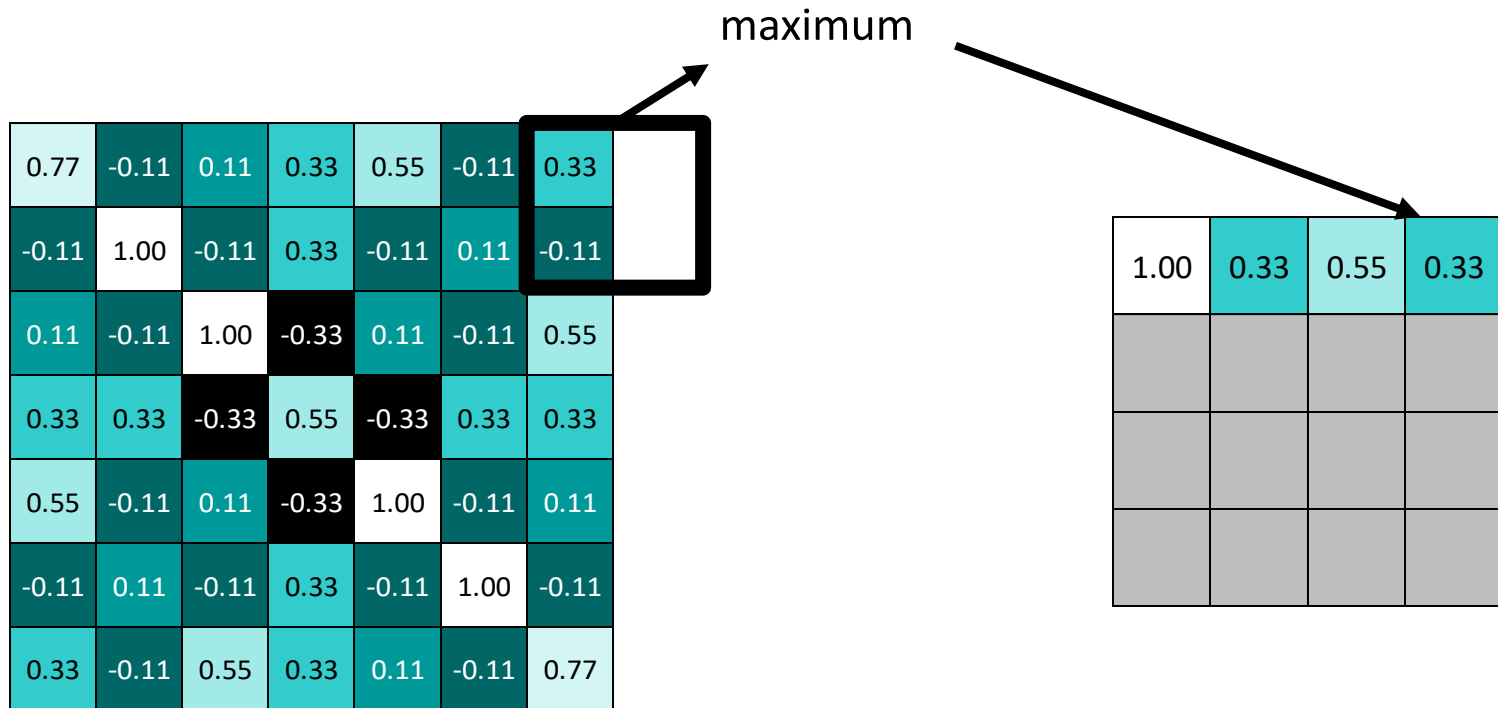
Pooling

maximum

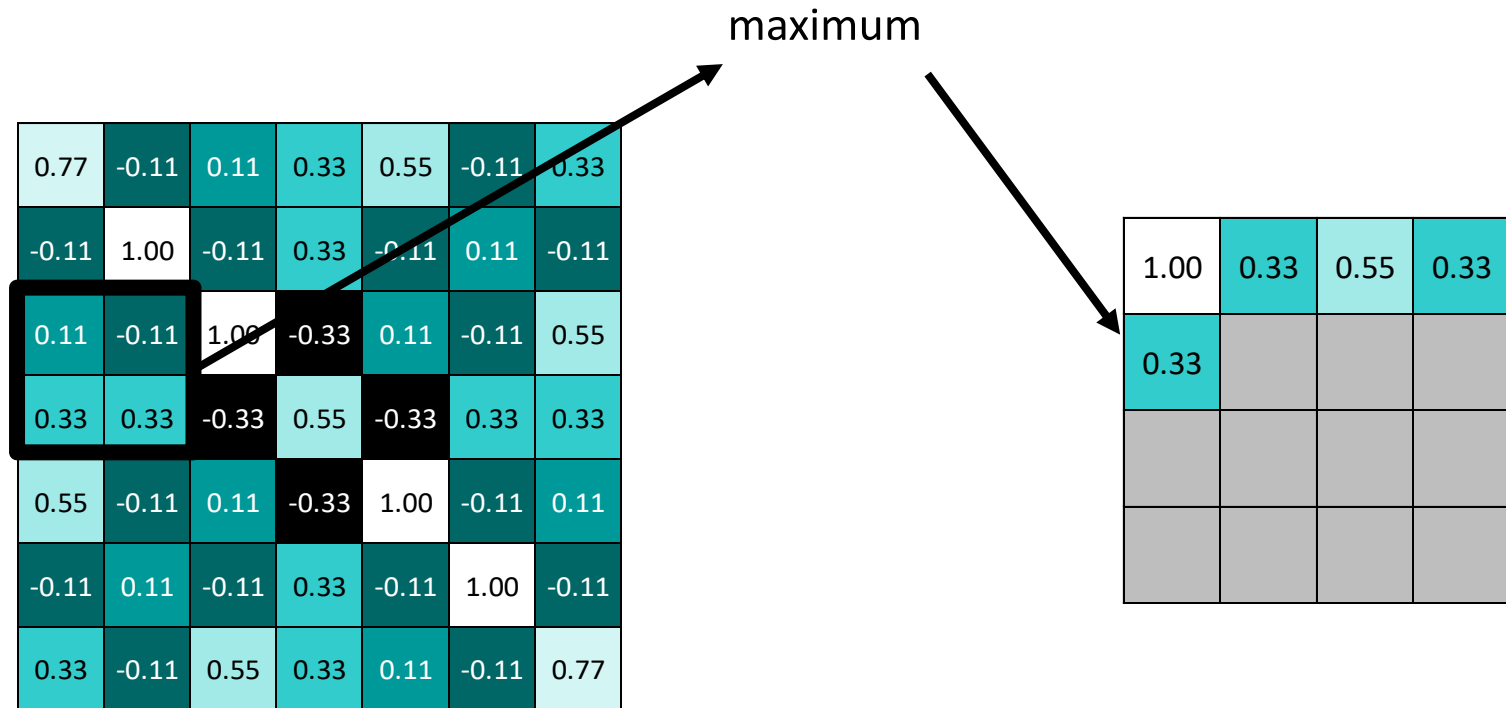
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

1.00	0.33	0.55	

Pooling



Pooling



Pooling

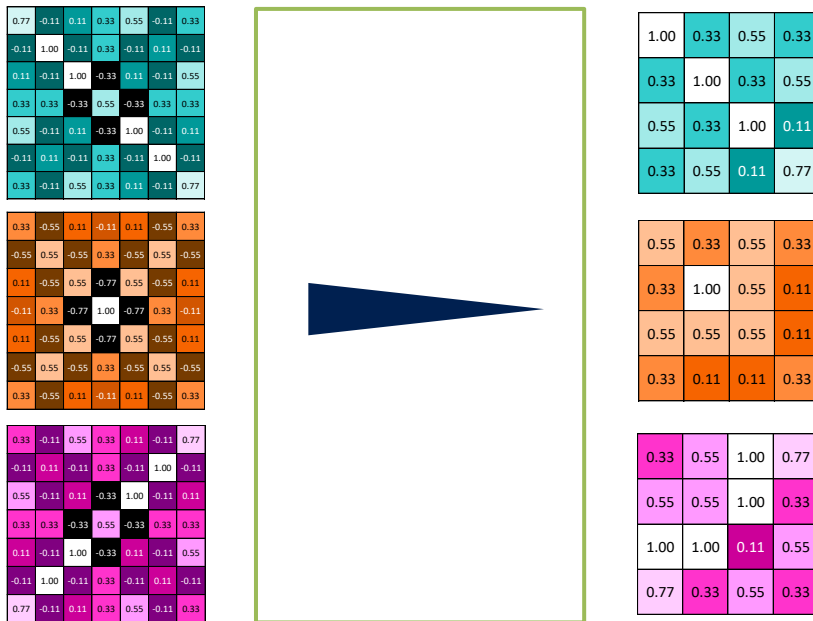
0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

max pooling

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

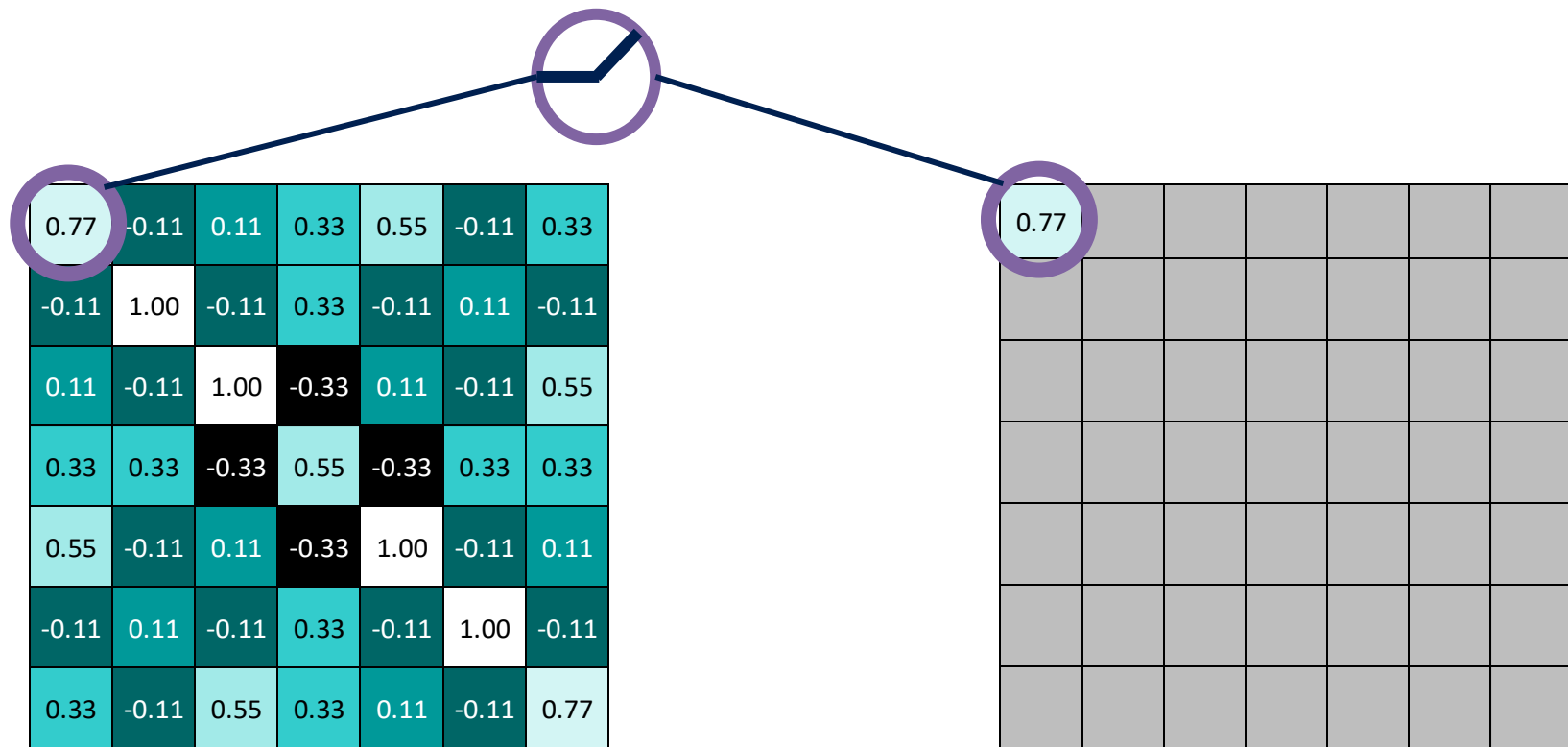
A stack of images becomes a stack of smaller images.



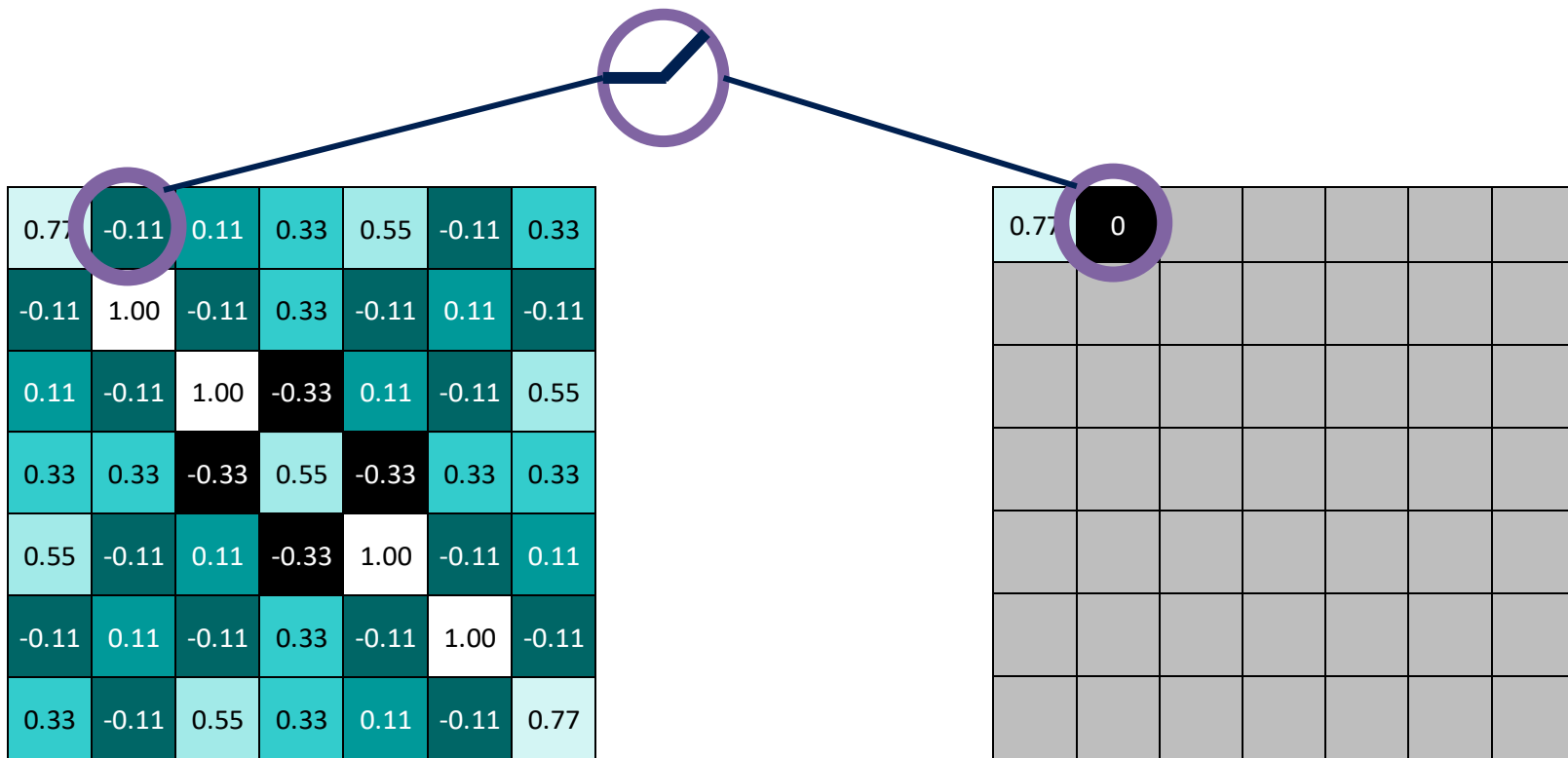
Normalization

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

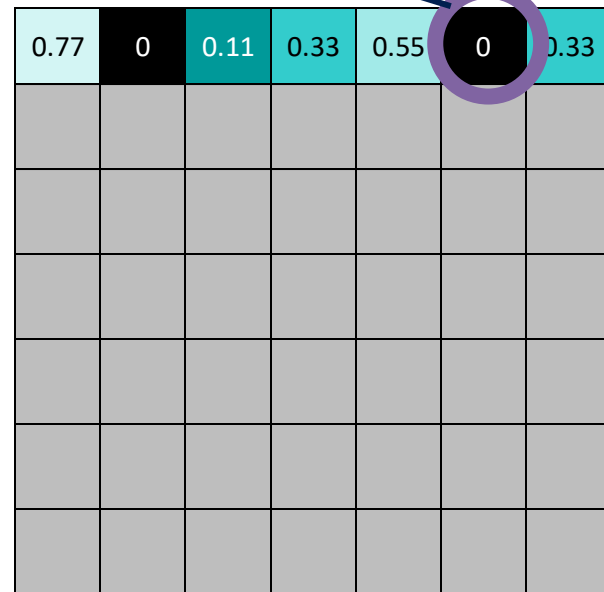
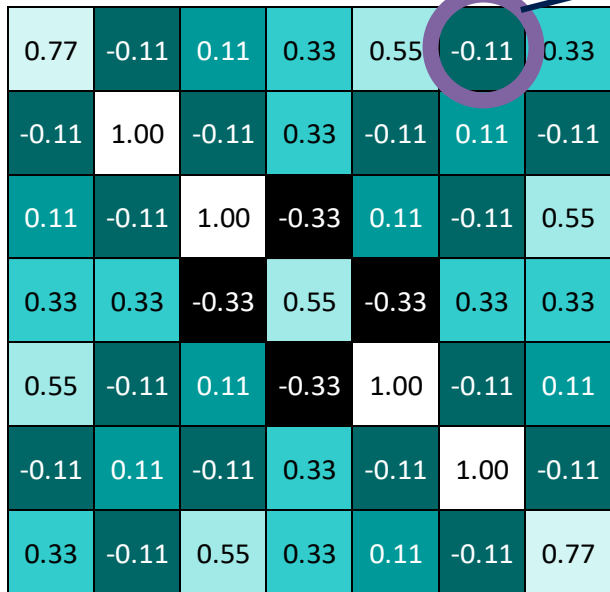
Rectified Linear Units (ReLUs)



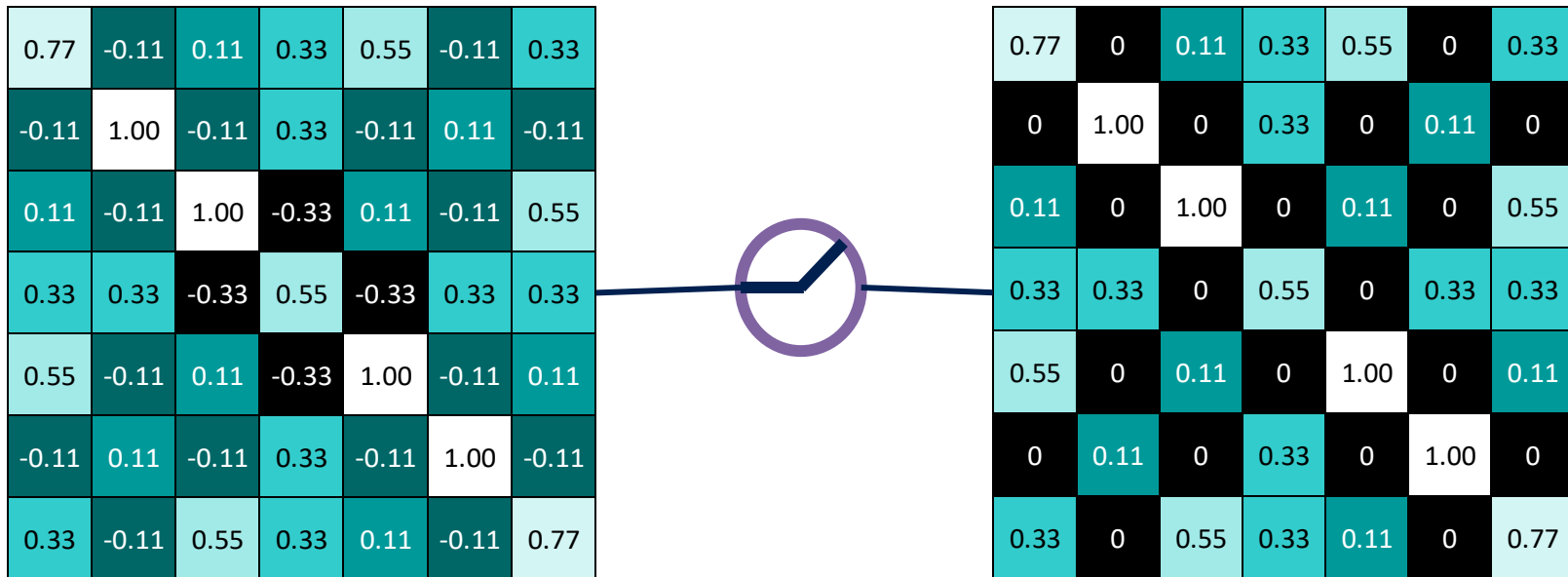
Rectified Linear Units (ReLUs)



Rectified Linear Units (ReLUs)



Rectified Linear Units (ReLUs)



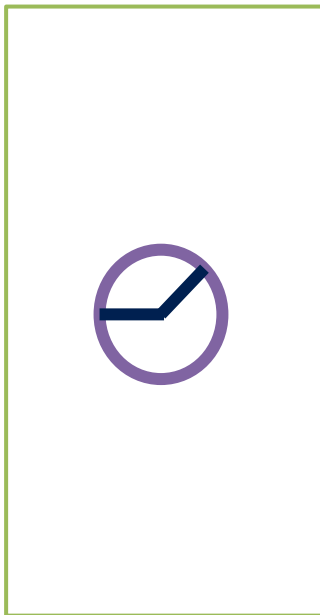
ReLU layer

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

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

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

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



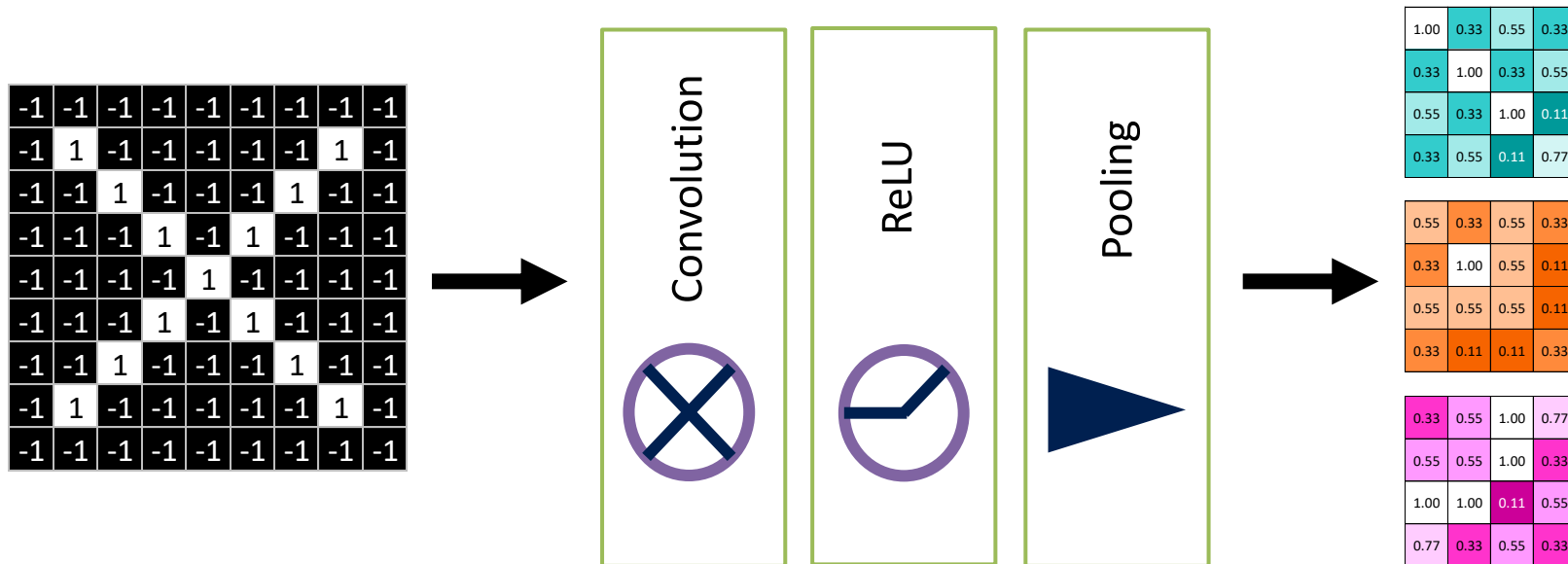
0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

0.33	0	0.11	0	0.11	0	0.33
0	0.55	0	0.33	0	0.55	0
0.11	0	0.55	0	0.55	0	0.11
0	0.33	0	1.00	0	0.33	0
0.11	0	0.55	0	0.55	0	0.11
0	0.55	0	0.33	0	0.55	0
0.33	0	0.11	0	0.11	0	0.33

0.33	0	0.55	0.33	0.11	0	0.77
0	0.11	0	0.33	0	1.00	0
0.55	0	0.11	0	1.00	0	0.11
0.33	0.33	0	0.55	0	0.33	0.33
0.11	0	1.00	0	0.11	0	0.55
0	1.00	0	0.33	0	0.11	0
0.77	0	0.11	0.33	0.55	0	0.33

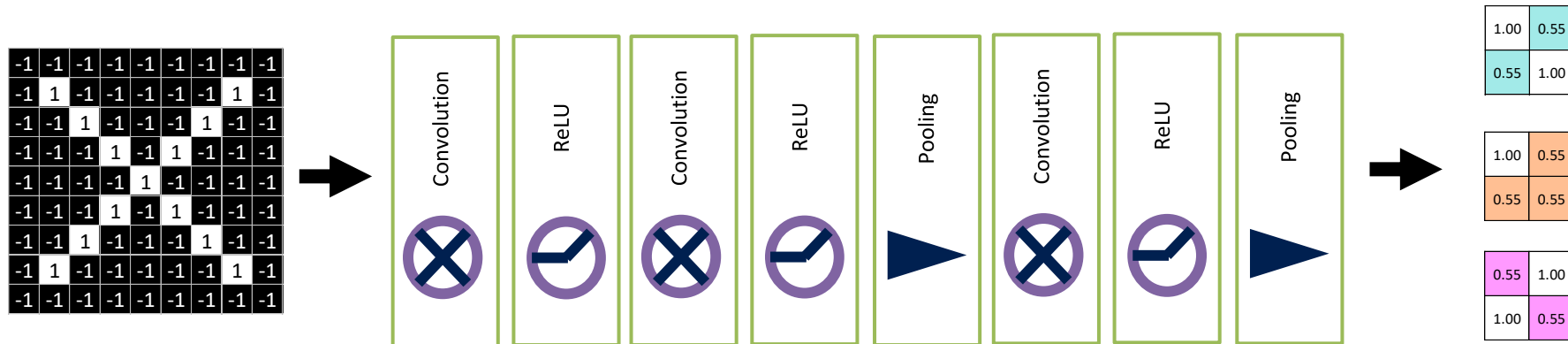
Layers get stacked

The output of one becomes the input of the next.



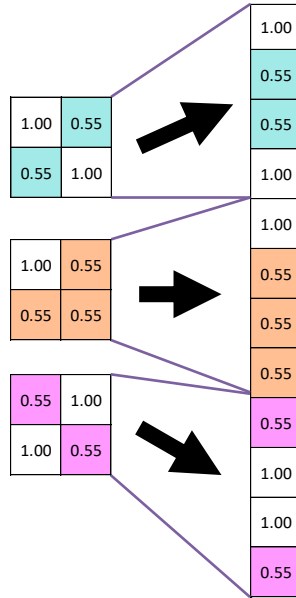
Deep stacking

Layers can be repeated several (or many) times.



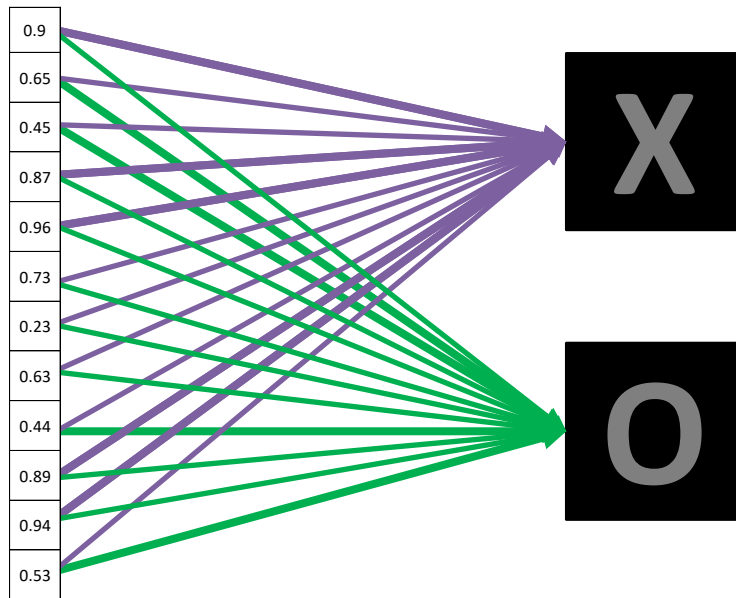
Fully connected layer

Every value gets a vote



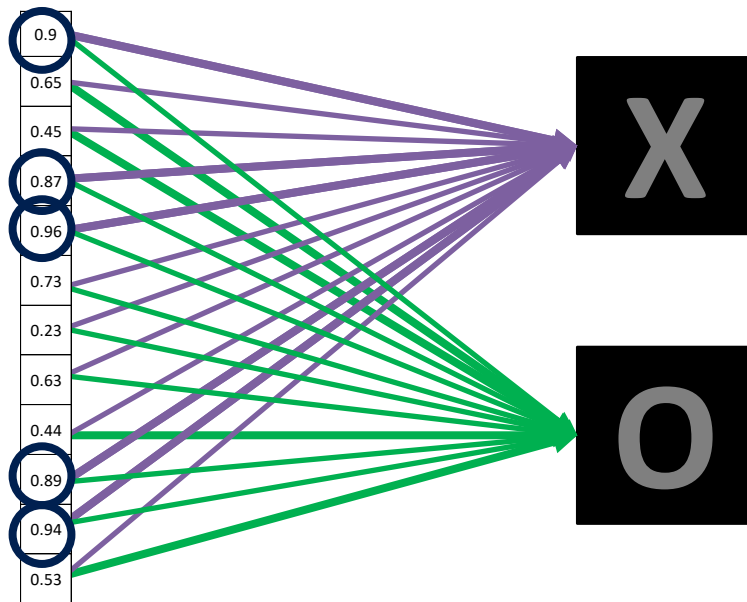
Fully connected layer

Future values vote on X or O



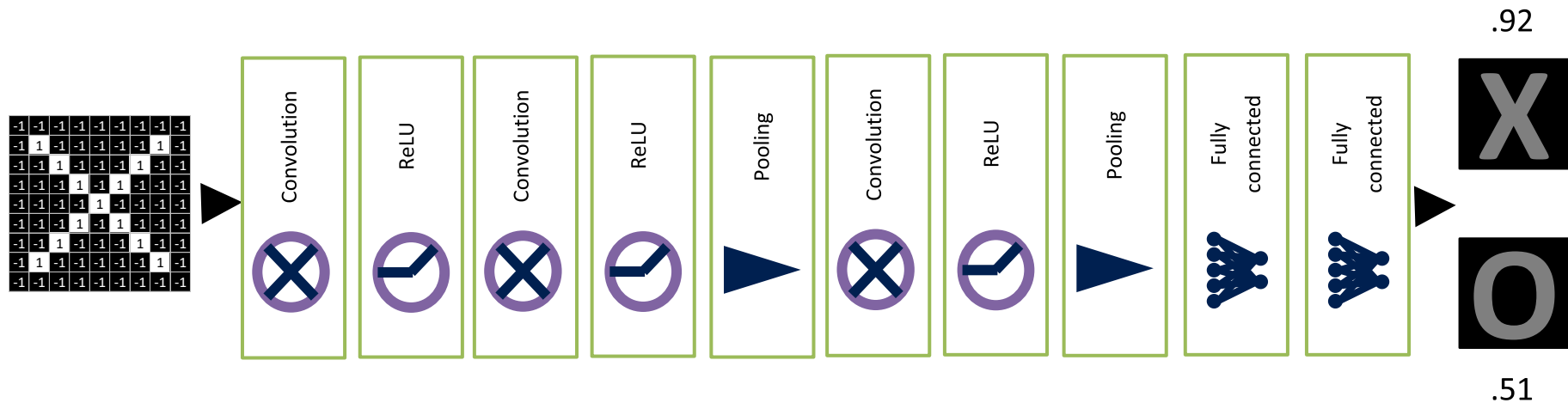
Fully connected layer

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?

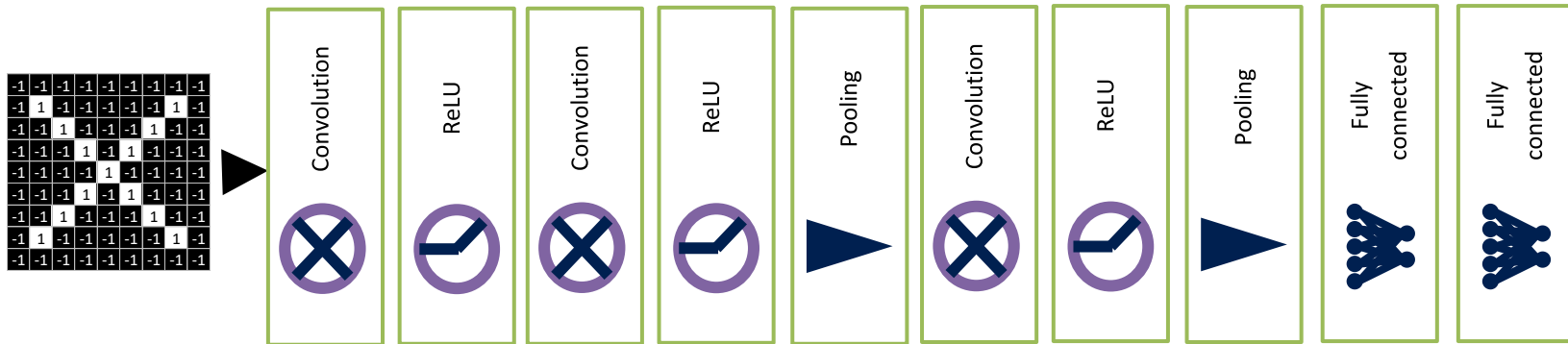
Features in convolutional layers

Voting weights in fully connected layers

A: Backpropagation

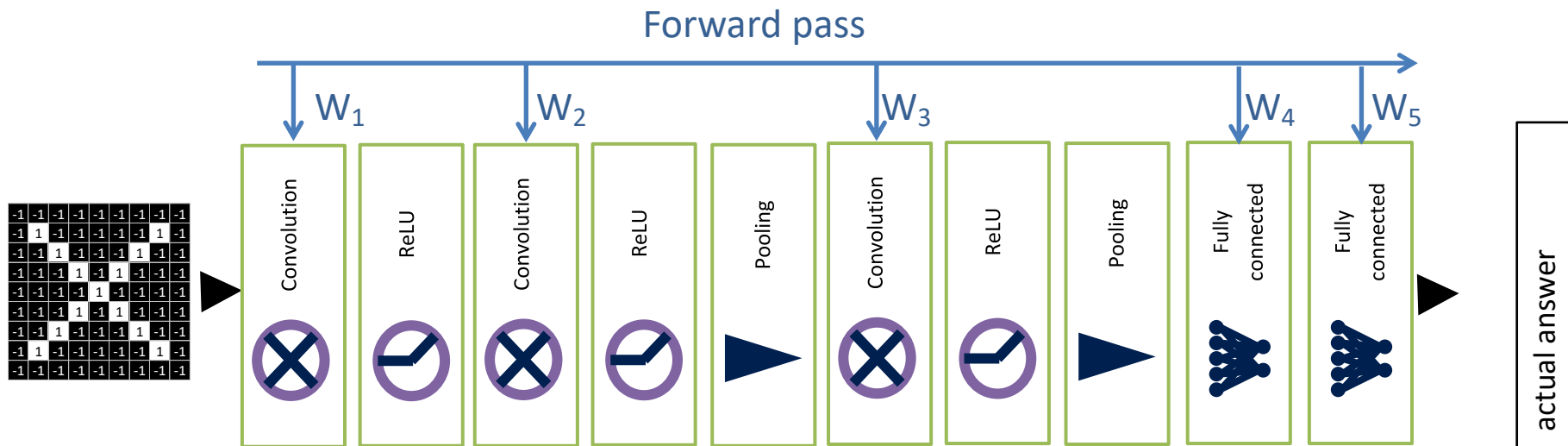
Backpropagation

In the first iteration W_i are set randomly



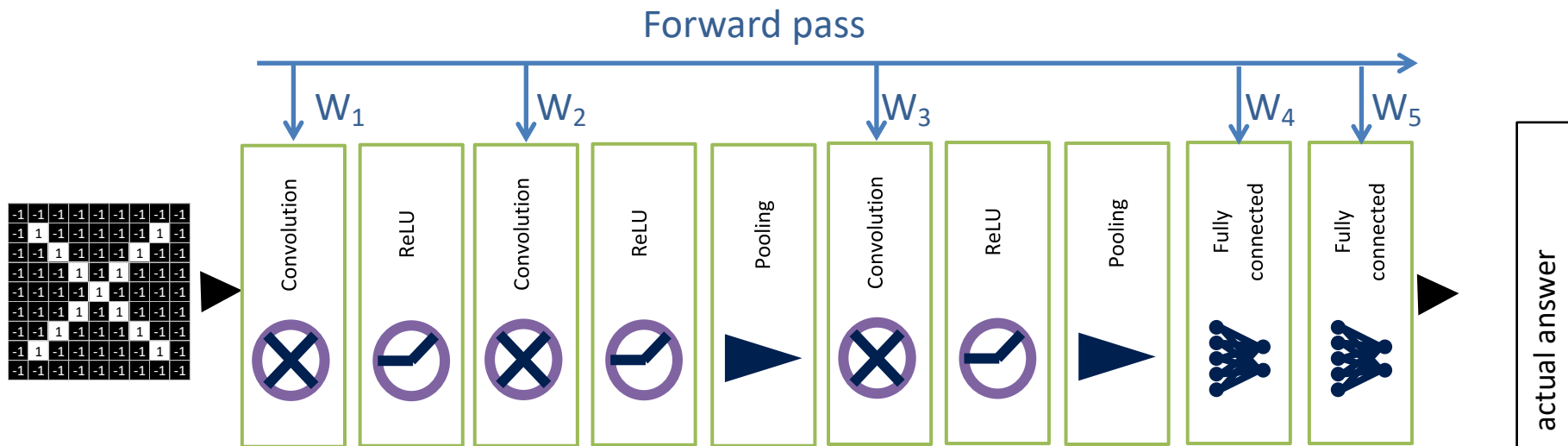
Backpropagation

In the first iteration W_i are set randomly



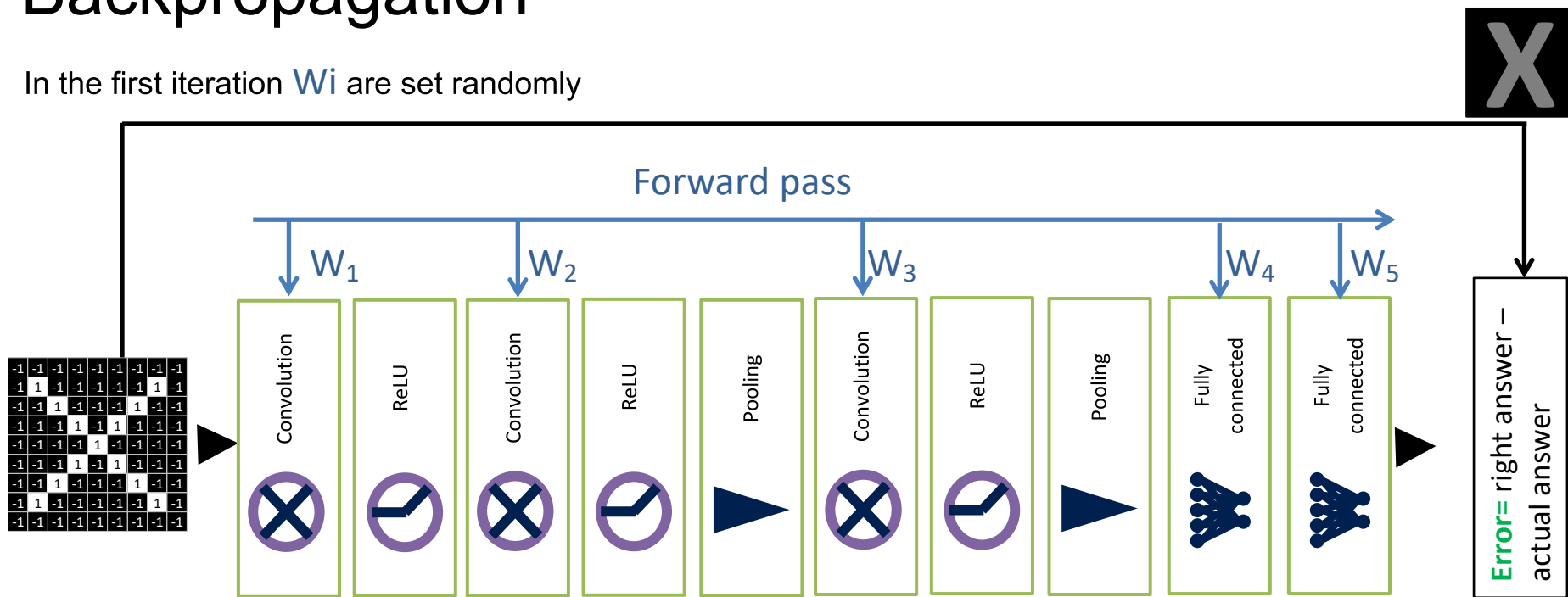
Backpropagation

In the first iteration W_i are set randomly



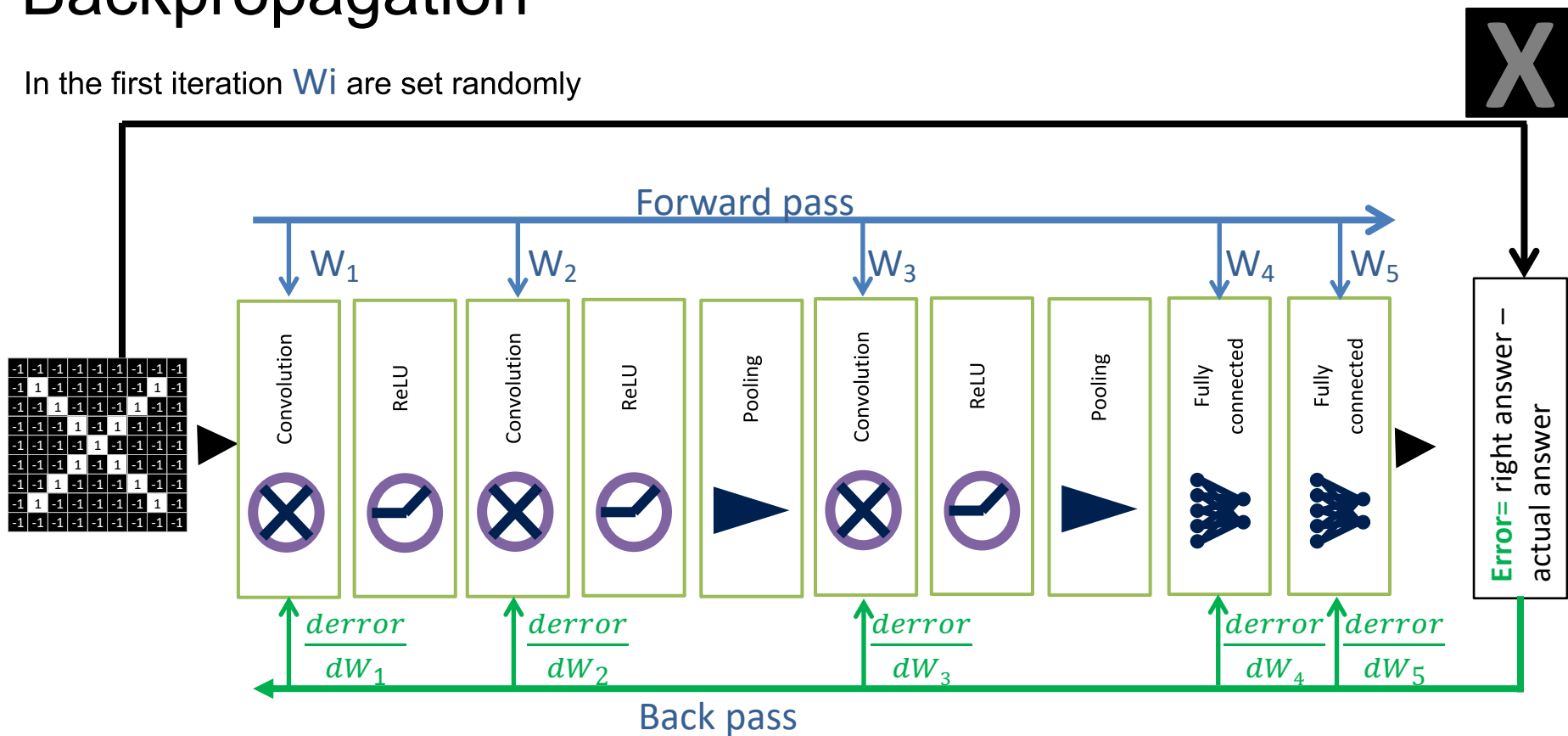
Backpropagation

In the first iteration W_i are set randomly



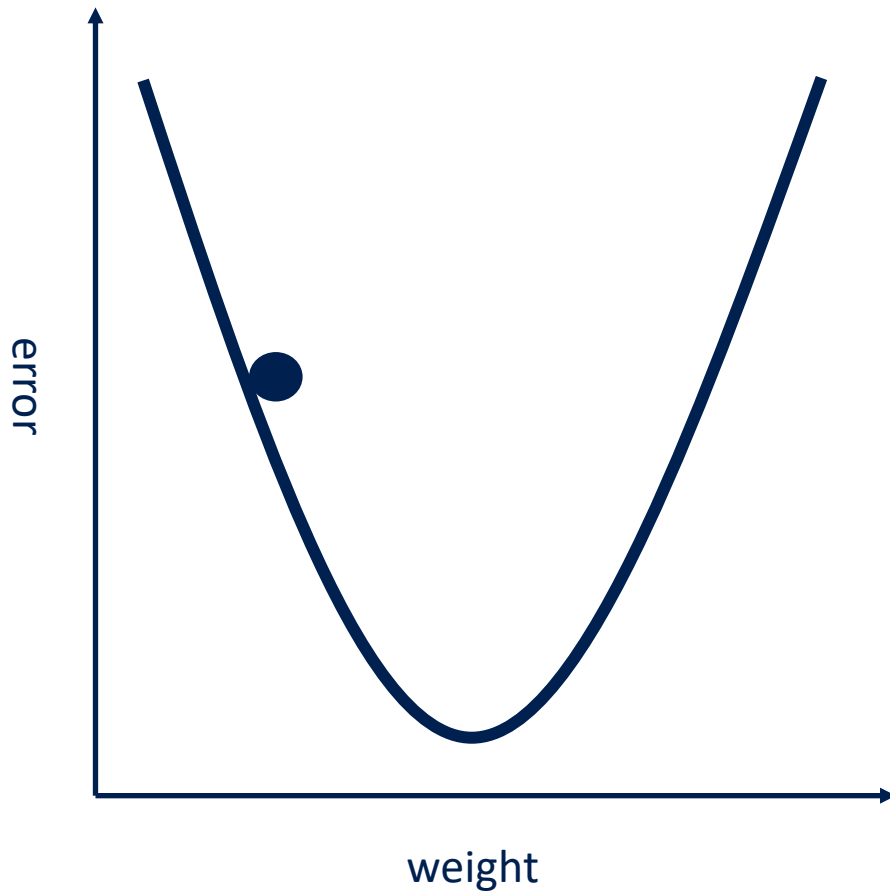
Backpropagation

In the first iteration W_i are set randomly



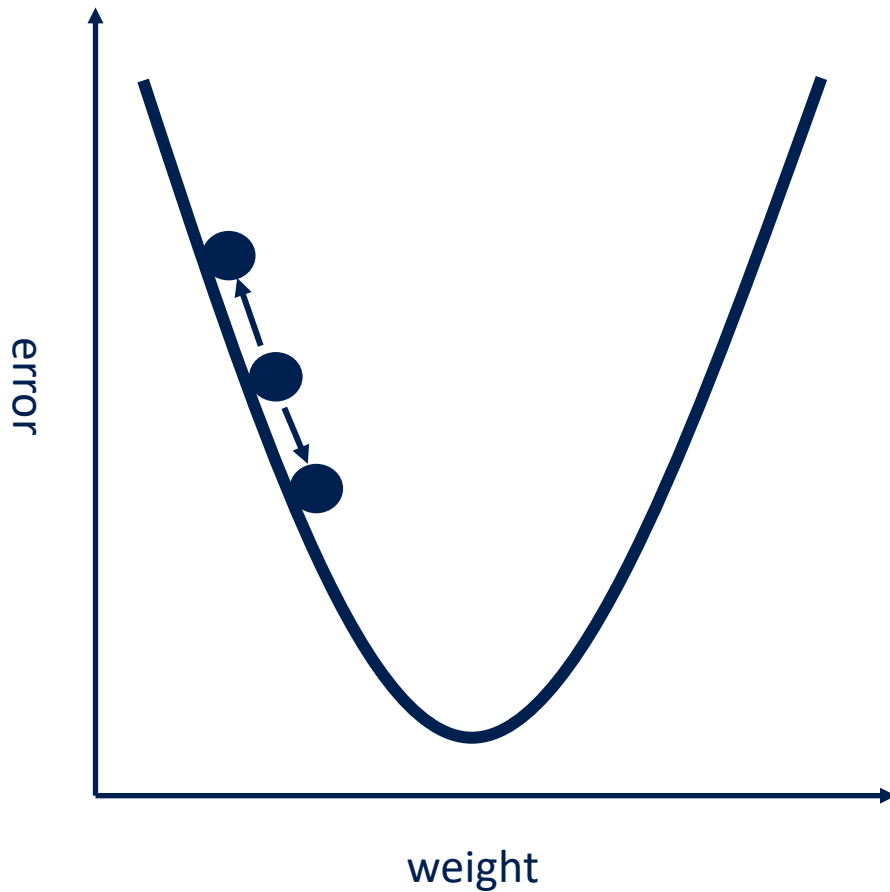
Gradient descent

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

- CNNs require large amount of data to get high accuracies
- Deep NNs are not explainable

Practical solutions

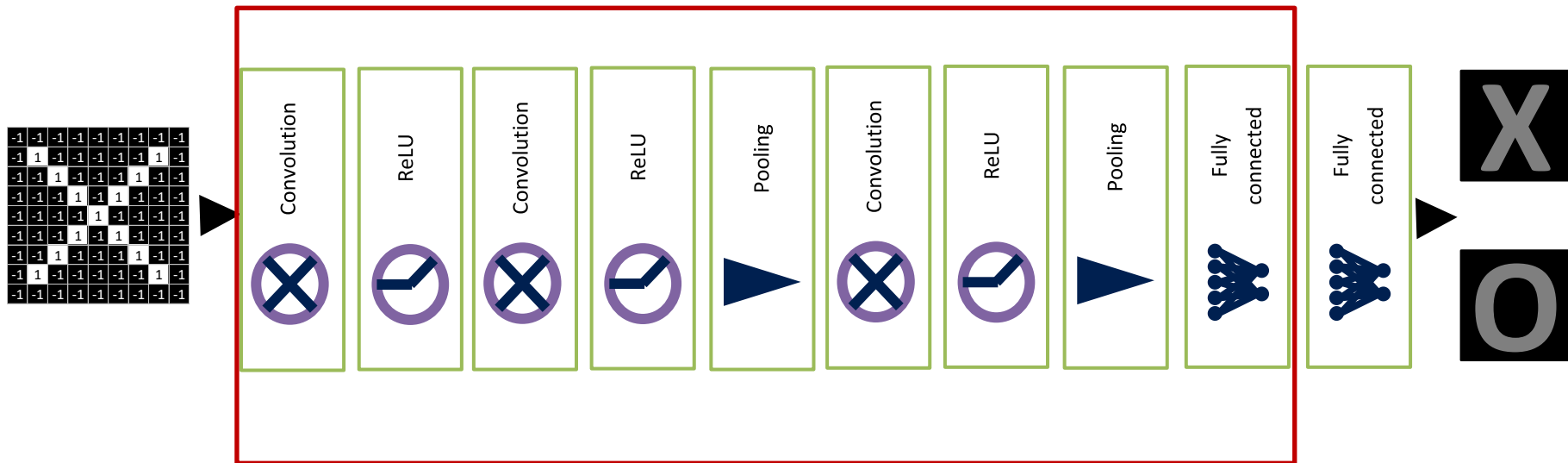
- Transfer learning, Data augmentation, Regulation techniques
- Make Deep NN more explainable

Outline

1. What are CNNs?
2. How CNNs work?
 - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
- 3. Data augmentation, transfer learning, regulation techniques**
4. Explainability

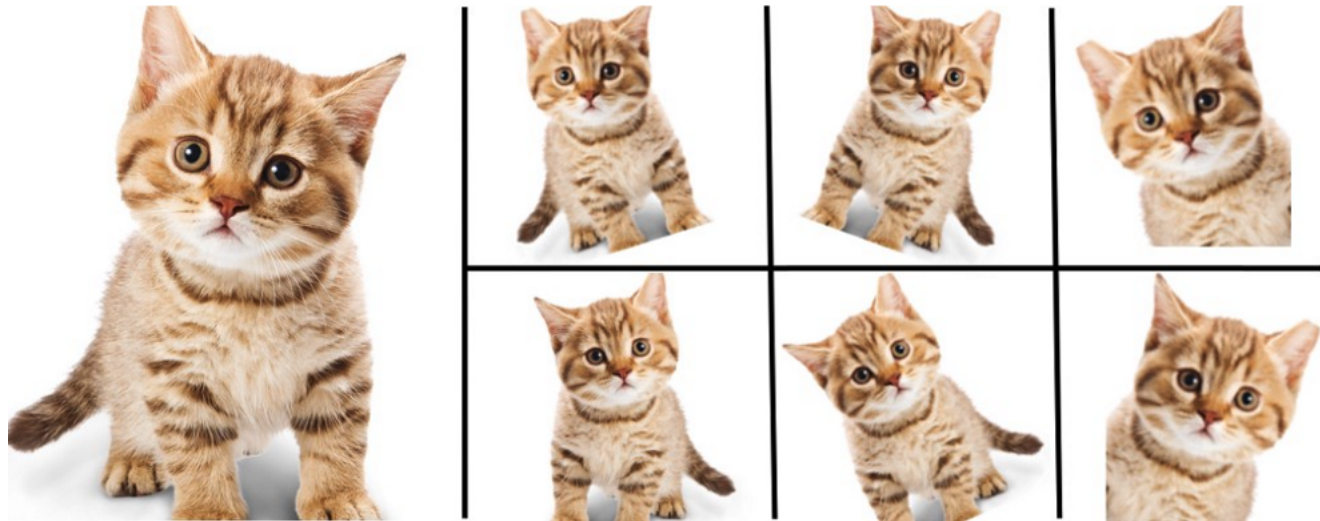
Transfer Learning

- ❑ Given a new problem & new data
- ❑ Instead of training from scratch, train only the last layers



Data augmentation

- ❑ Increase the training dataset volume artificially using transformations
- ❑ Objective: Improve model robustness



Original :

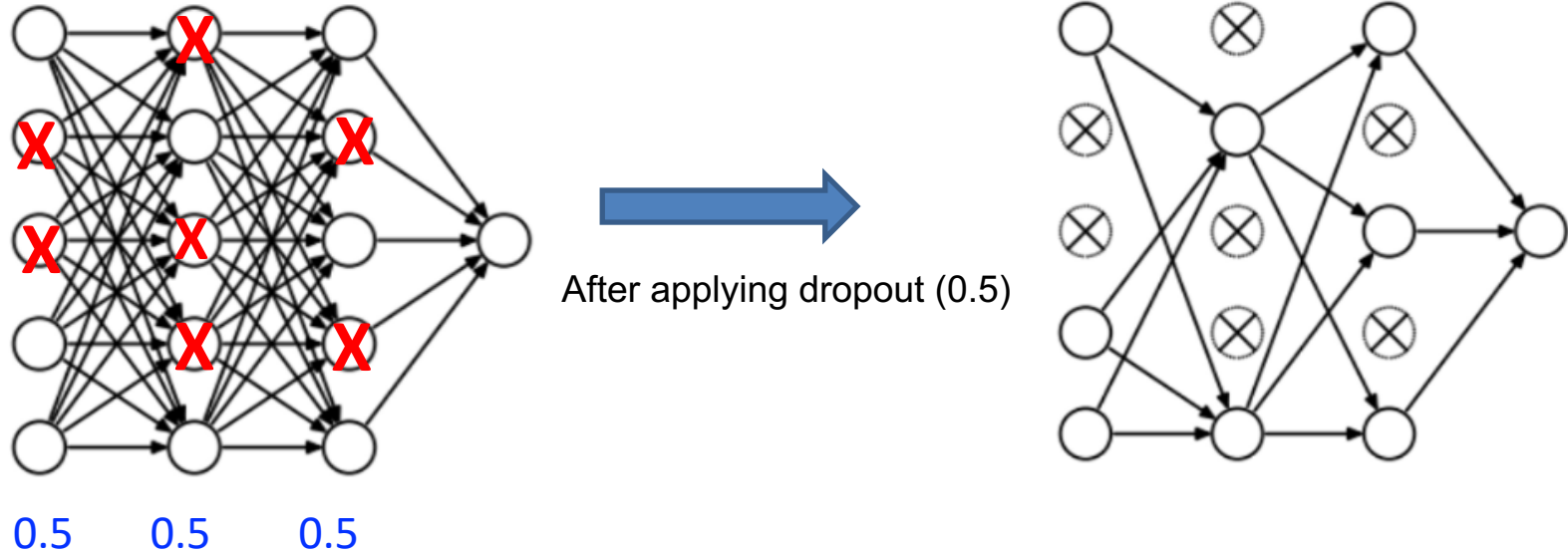
Scaling

Centring

translation

rotation Etc

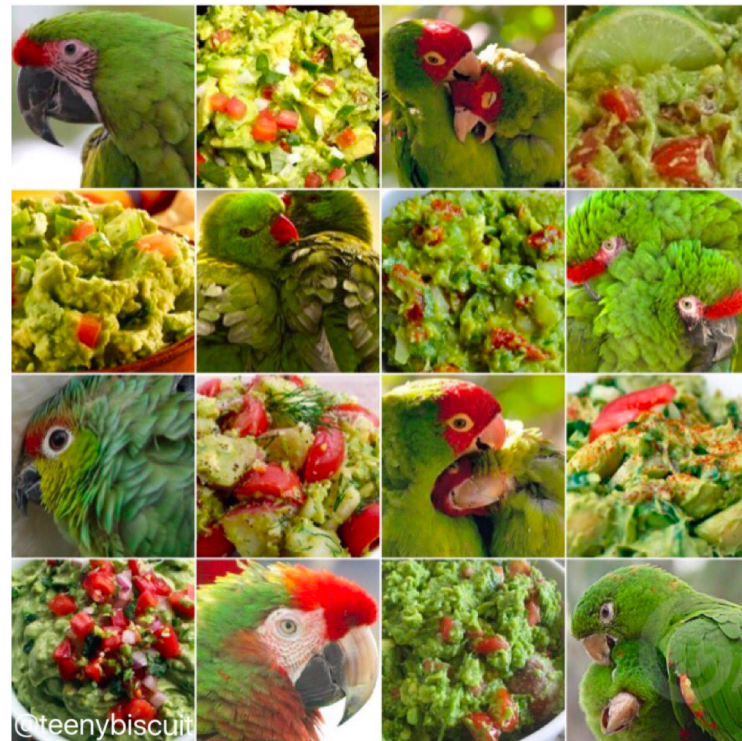
Regulation techniques: dropout



Outline

1. What are CNNs?
2. How CNNs work?
 - Convolution layers, pooling layers, FC layers, Gradient, Backpropagation
3. Data augmentation, transfer learning, regulation techniques
- 4. Explainability**

CNN can distinguish complex objects



Deep learning can fail



Dos muertos en
California en 2018

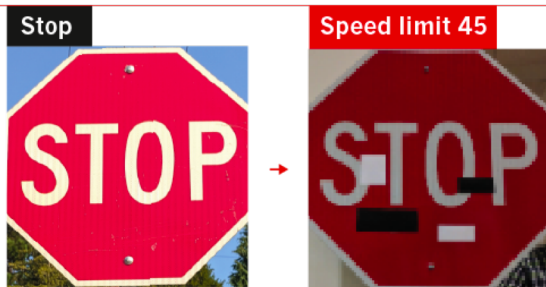
https://www.washingtonpost.com/local/trafficandcommuting/deadly-tesla-crash-tied-to-technology-and-human-failures-ntsb-says/2020/02/25/86b710bc-574d-11ea-9b35-def5a027d470_story.html

Deep learning can be fooled

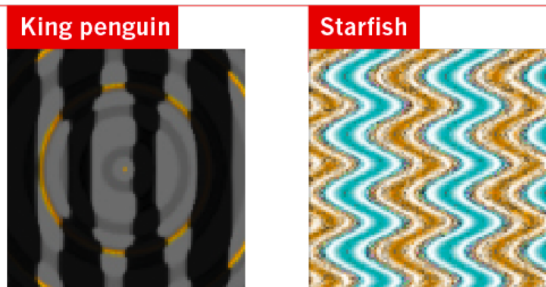
FOOLING THE AI

Deep neural networks (DNNs) are brilliant at image recognition — but they can be easily hacked.

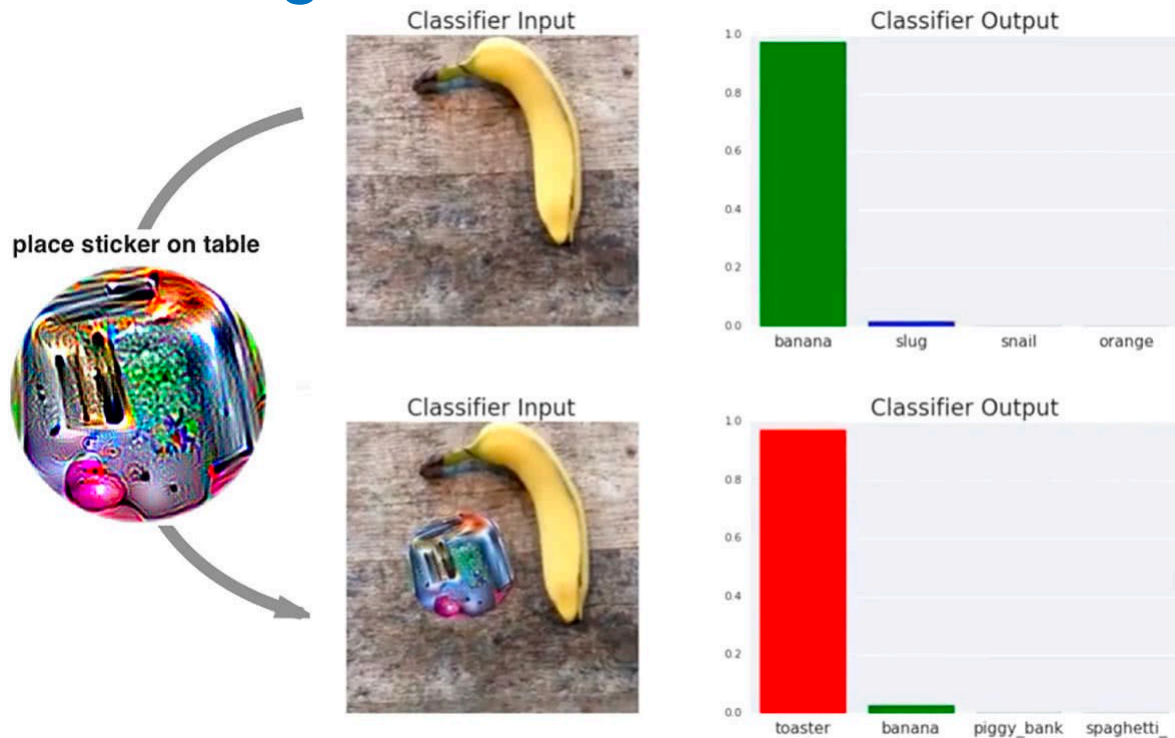
These stickers made an artificial-intelligence system read this stop sign as 'speed limit 45'.



Scientists have evolved images that look like abstract patterns — but which DNNs see as familiar objects.



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*.

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

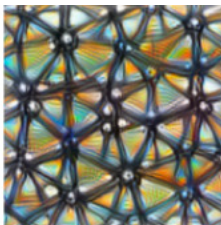
Arrieta, A. B et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82-115.

Visual explanation tools for CNNs

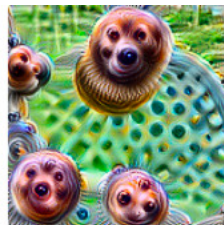
Watch the inside:



(a) Neuron



(b) Channel

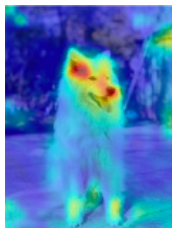


(c) Layer

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

Watch the outside:

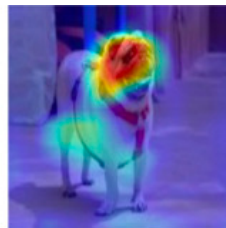
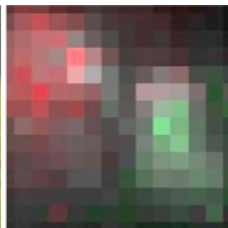
- Saliency maps
- Grad-CAM



(a) Heatmap [168]



(b) Attribution [293]



(c) Grad-CAM [292]

- LIME - Local Interpretable Model-Agnostic Explanations

LIME




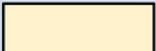



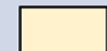
Original Image



Interpretable
Components

$P(\text{tree frog}) = 0.54$



Perturbed Instances	$P(\text{tree frog})$
	 0.85
	 0.00001
	 0.52

