6. CONVOLUTIONAL NEURAL NETWORKS

Stephan Robert-Nicoud
HEIG-VD/HES-SO

Credit: Andres Perez-Uribe

Objectives

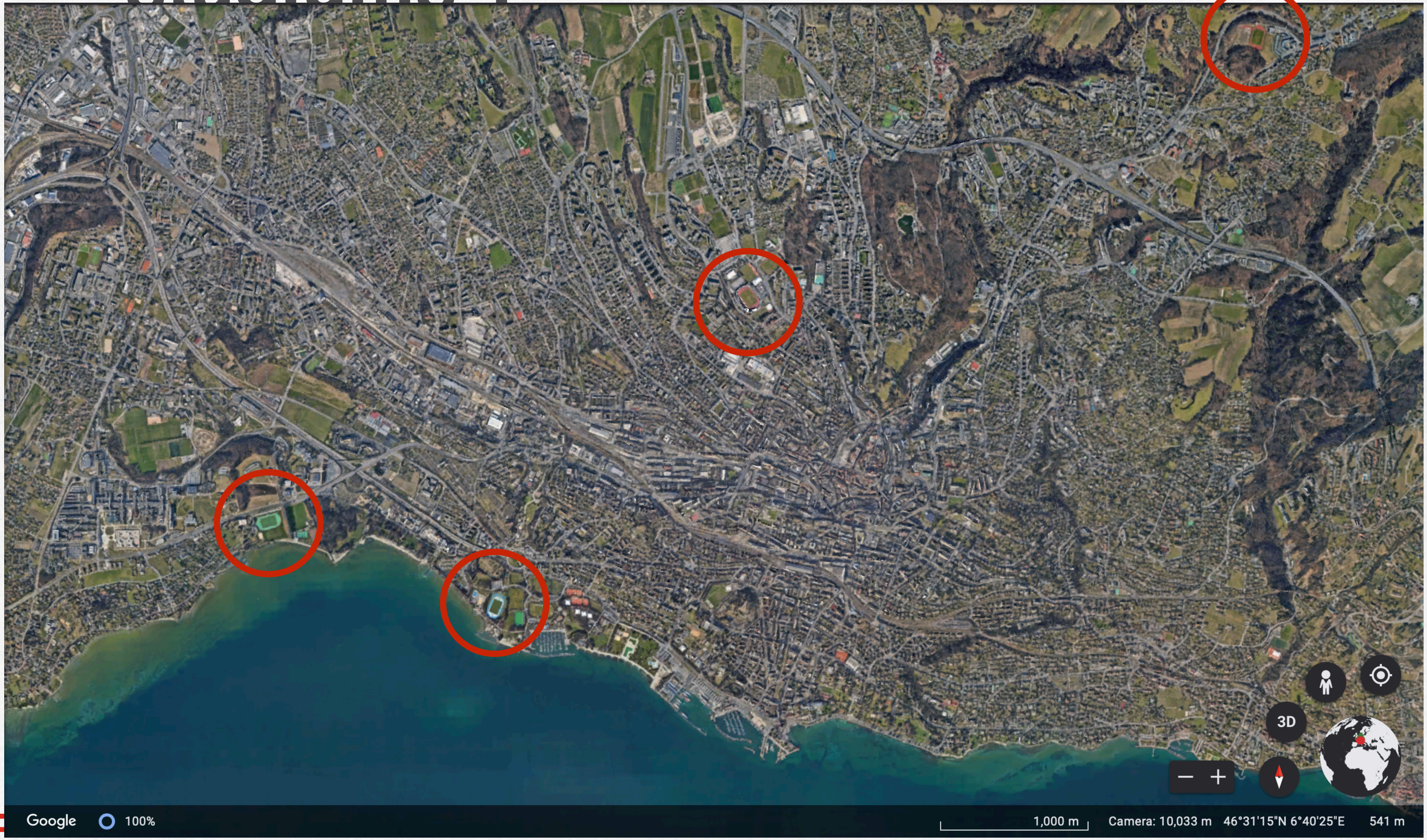
- Understand the architecture of a Convolutional Neural Network (CNN)
- Understand the functioning of a CNN and how it can efficiently process images, as well as multi-dimensional data
- Understand how to implement and train a CNN using high-level libraries like Keras

Contents

- Object recognition by “naive” pattern matching
- How the brain recognizes objects ?
- Image filtering by convolution
- Convolutional Neural Networks (CNN)
- Deep Learning by gradient descent
    - Generalization (performance evaluation)
    - Hands-on deep learning

HE VD IG

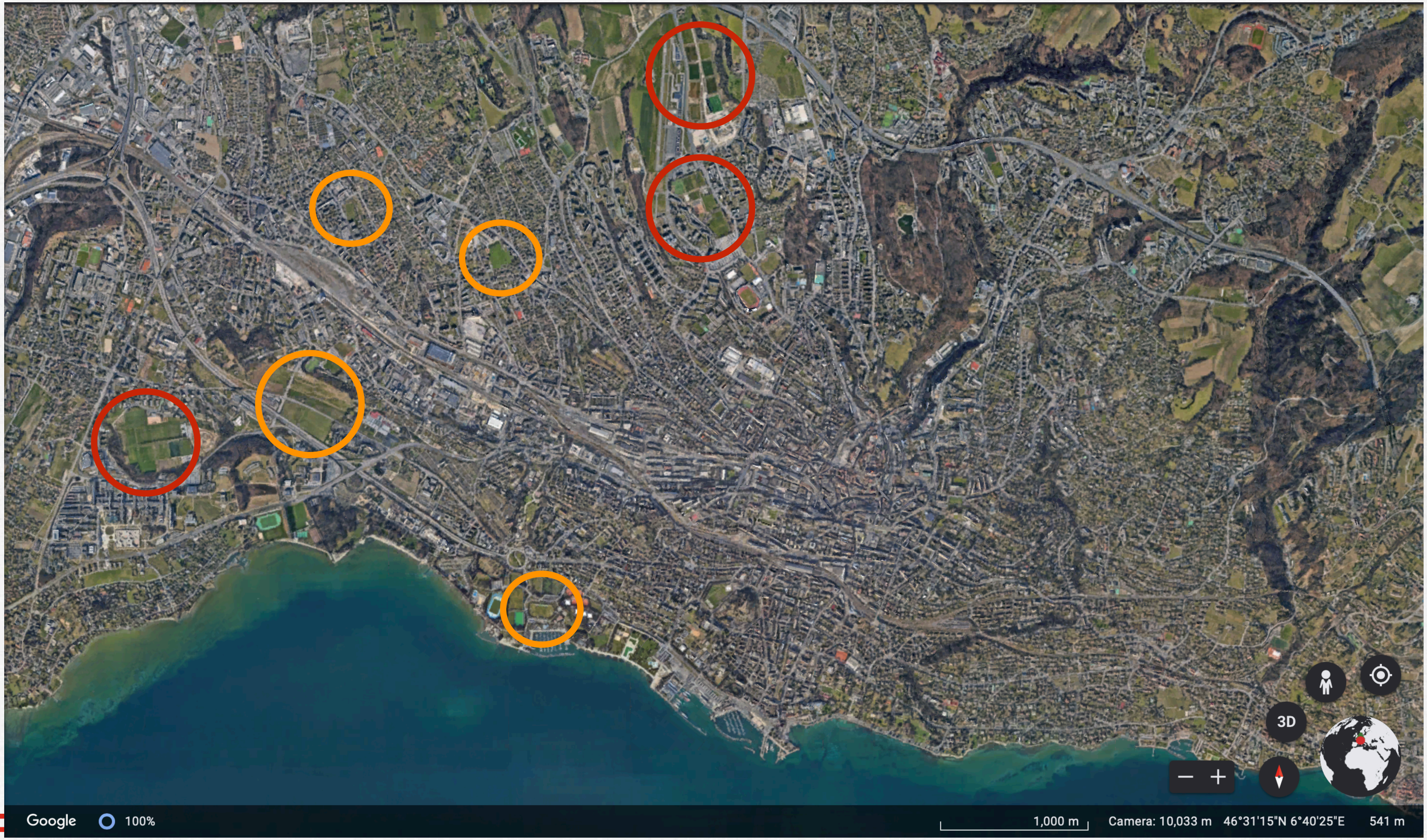
Where are the athletics stadiums ?



https://earth.google.com/web/@46.52707774,6.61885237,506.32773008a,9526.34044459d,35y,0h,0t,0r

HE
IG

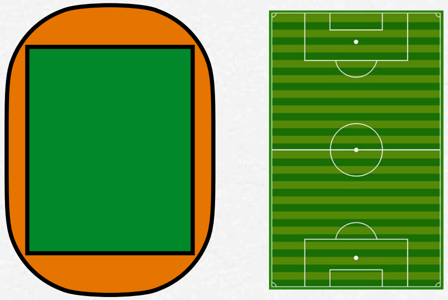
Where are the football fields ?



https://earth.google.com/web/@46.52707774,6.61885237,506.32773008a,9526.34044459d,35y,0h,0t,0r

HE
IG

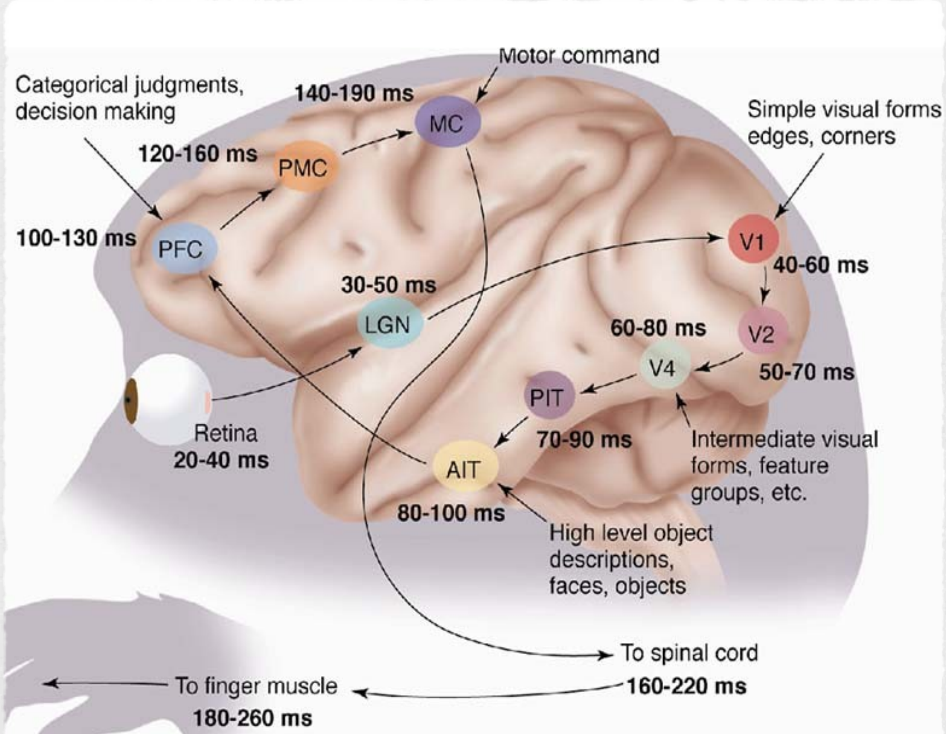
Pattern matching problem



“templates” of athletics stadiums and football fields

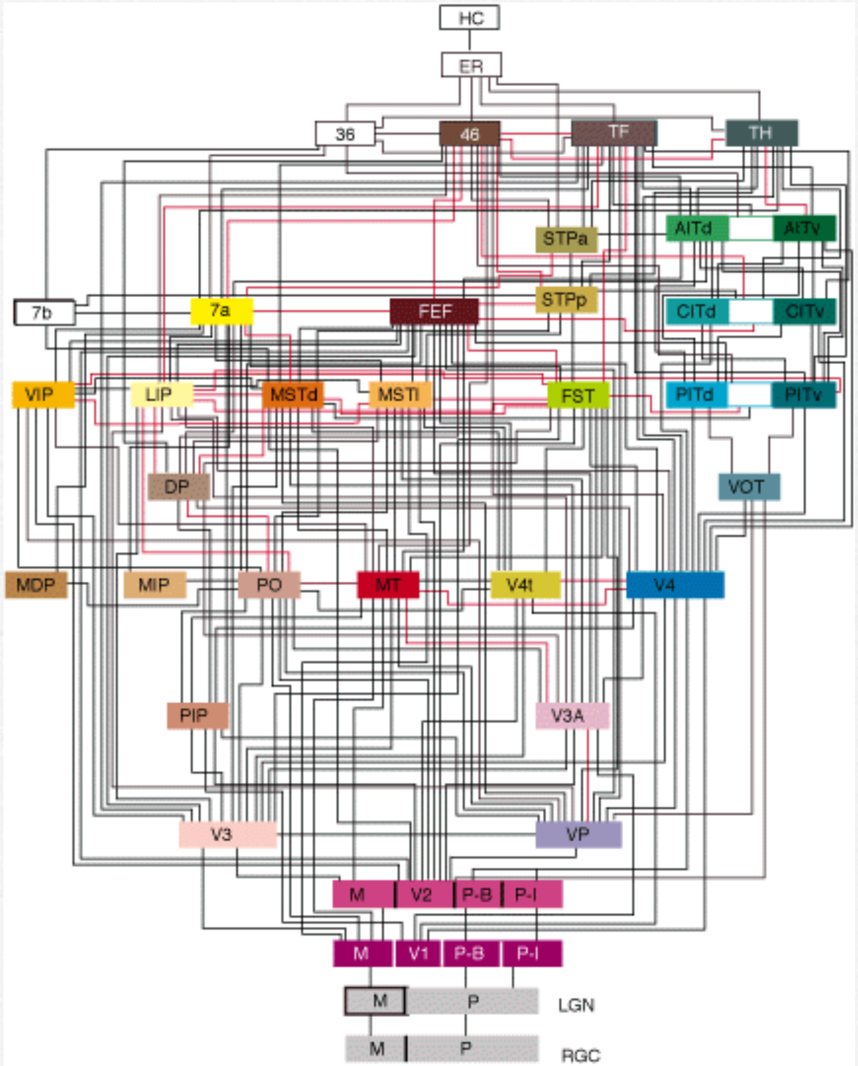
When dealing with real-world data we have to face the problem of large intra-class variability. Thus, we cannot stick to “ideal” templates of objects for recognizing them. Instead, we have to use features: e.g., acceptable colors, terrain texture, size, presence of straight lines, border

How the brain recognizes objects ?



[picture from Simon Thorpe]

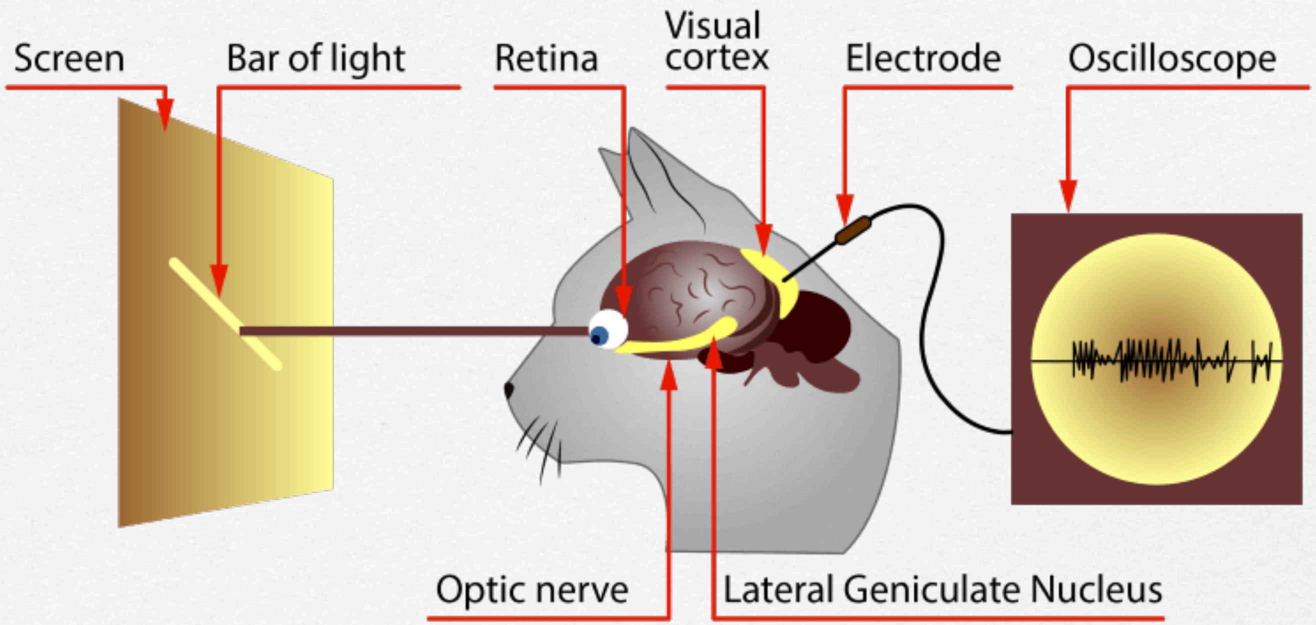
[Gallant & Van Essen]



- Macaque's visual system

from Felleman, D. J. and Van Essen, D. C. (1991) *Cerebral Cortex* **1:**1-47

Receptive fields of single neurons in the cat's cortex



- From Hubel & Wiesel experiments with cats (1959)
  - Single neurons in the visual cortex are preferentially activated by particular patterns (e.g., a bar at a certain angle) appearing in a particular region of the field of vision.

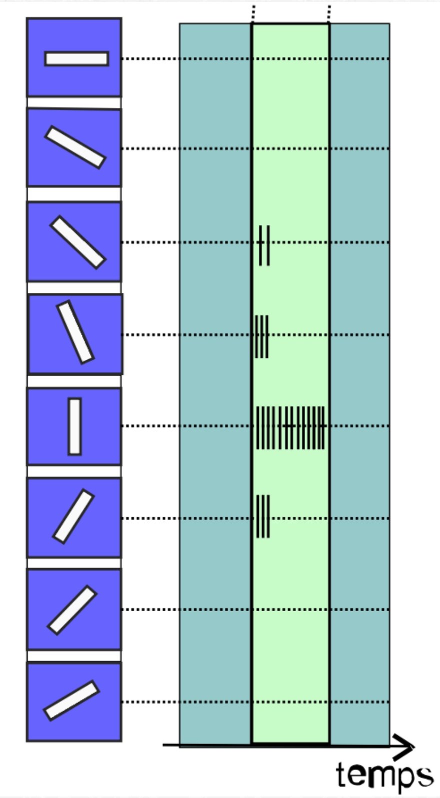
The luck of Hubel and Wiesel

pattern known to activate individual cells in the retina but not in the visual cortex

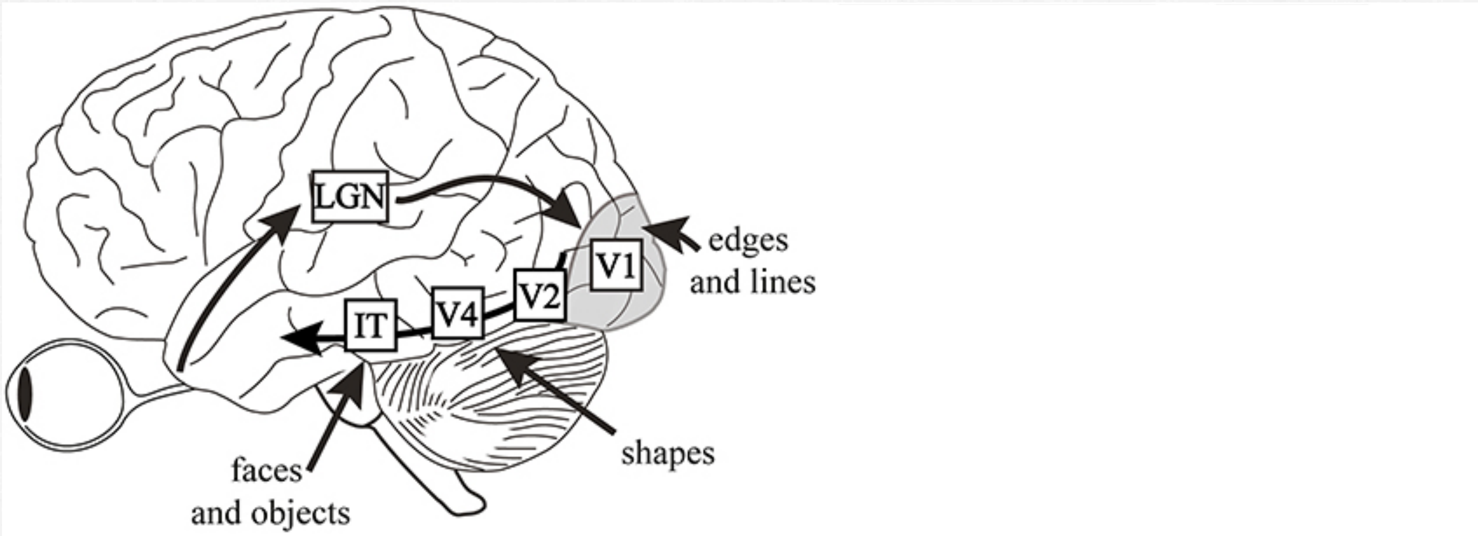
pattern found to activate individual cells in the cortex (V1)

Hubel & Wiesel cat experiments in 1959
they will receive the Nobel Prize in 1981

Hierarchical representation of objects in the visual cortex



patterns found to activate individual cells in the cortex (V1)



Higher layers appear to recognize more high-level features

The Jennifer Aniston neuron

- There is a neuron in the hippocampus of certain people that preferentially fires when a picture of the actress (or other celebrities, or certain objects) is perceived.
- More recent experiments showed associate learning in those neurons after a single perception of a new image showing an actor and an object (e.g., the Eiffel tower).

from Nature (July, 2005)

Face pareidolia in the cortex





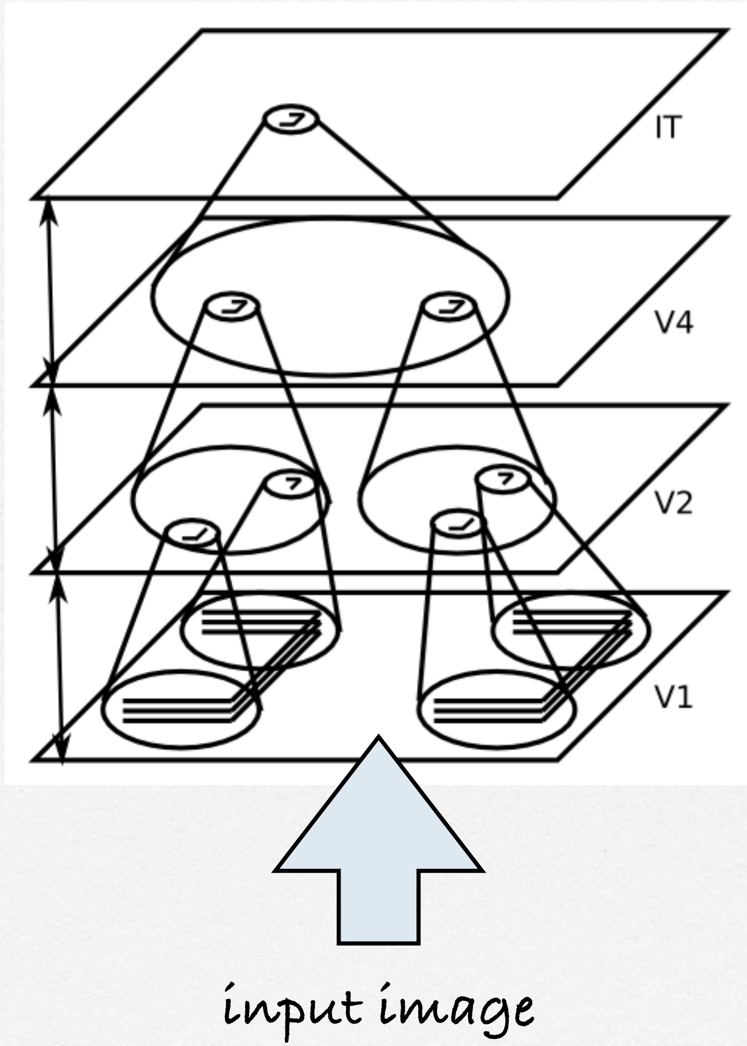



- A brain region called FFA (fusiform face area) is known to detect faces and is also activated by illusory faces.
- Face pareidolia is the perception of illusory faces inanimate objects.

HE IG VD

Lessons from neuroscience

- Start by extracting localized low-level features of the image (e.g., edges)
- Use multiple levels of processing to incrementally allow the system to appropriately bind together features and their relationships (e.g., extract higher-level features)
- Gradually build-up overall spatial invariance (i.e., be able to find a pattern anywhere in the input image)



Convolutions on image processing

- Blurring, sharpening, embossing, and edge detection are typical functions of image processing. They are accomplished by means of convolution between a kernel and an image.

- For each 3x3 block of pixels in the image on the left, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image on the right.

X

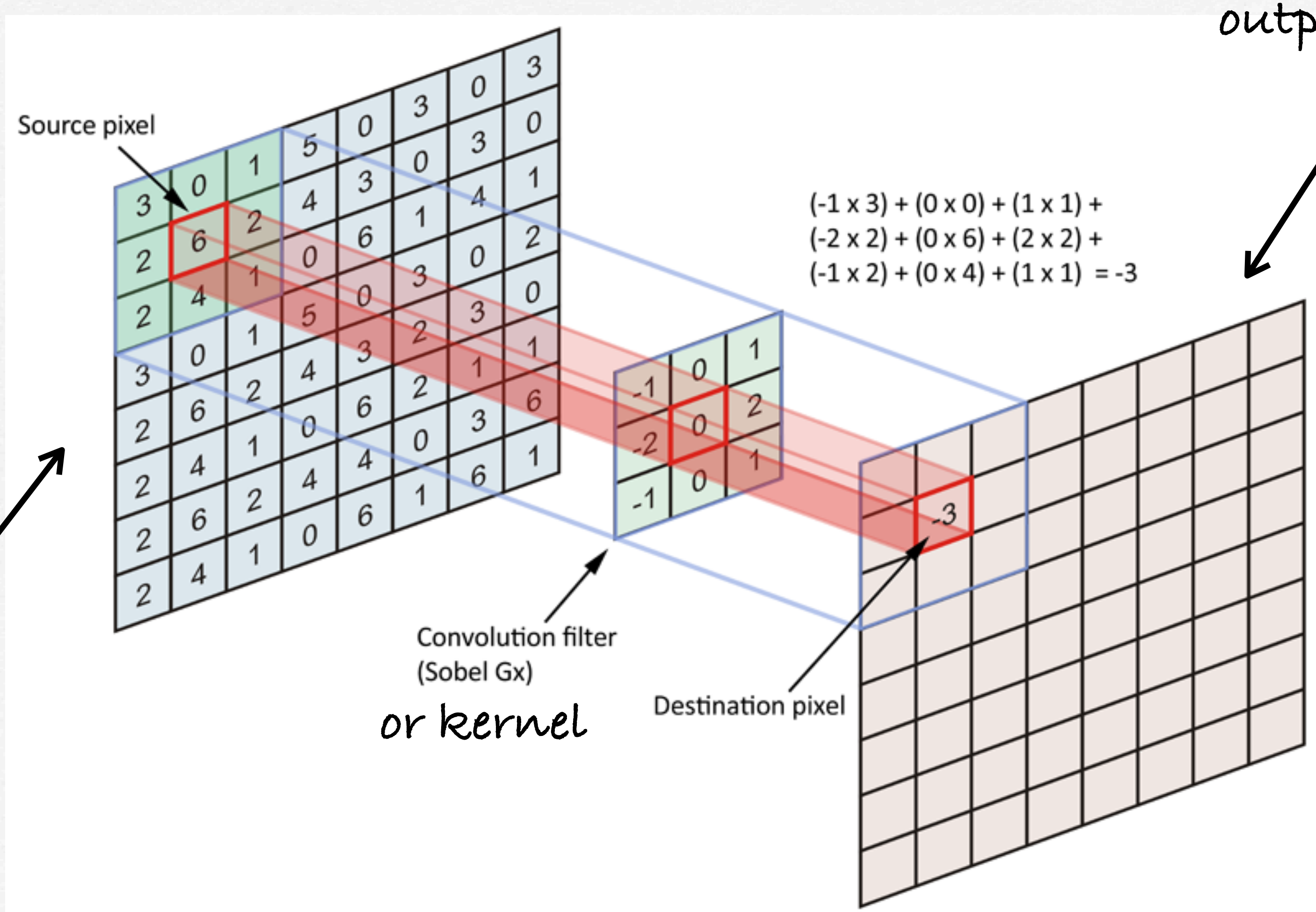
1	1	1
1	-8	1
1	1	1

kernel

=

Image filtering by convolution

output image

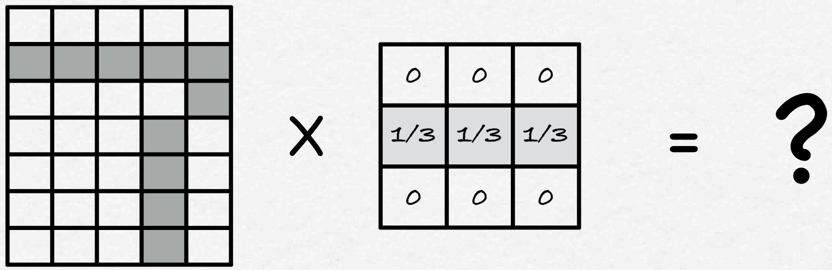


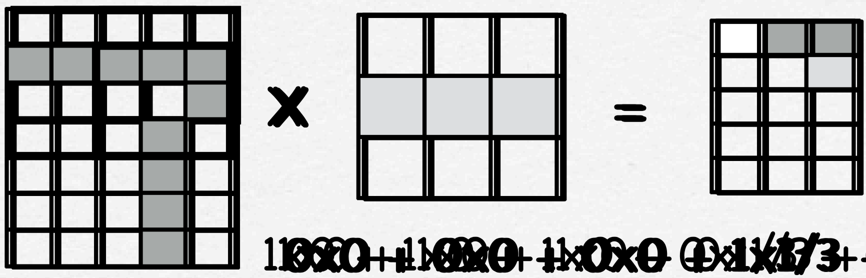
input image

or kernel

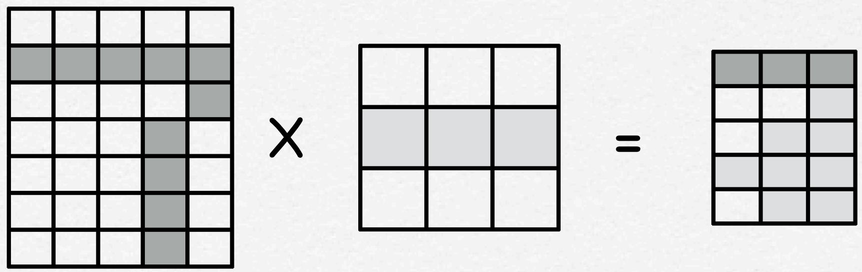
from Analytics Vidhya

Convolution kernels as pattern detectors



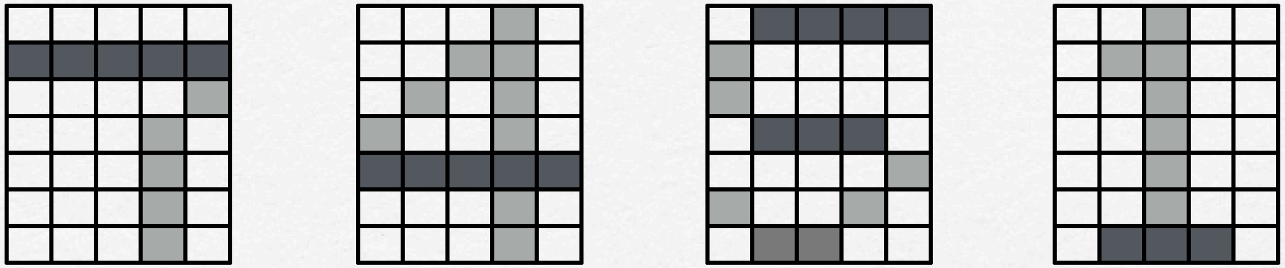


0x0 + 0x0 + 0x0 + 1x1/3 + 1x1/3 + 1x1/3 + 0x0 + 0x0 + 0x0 = 1

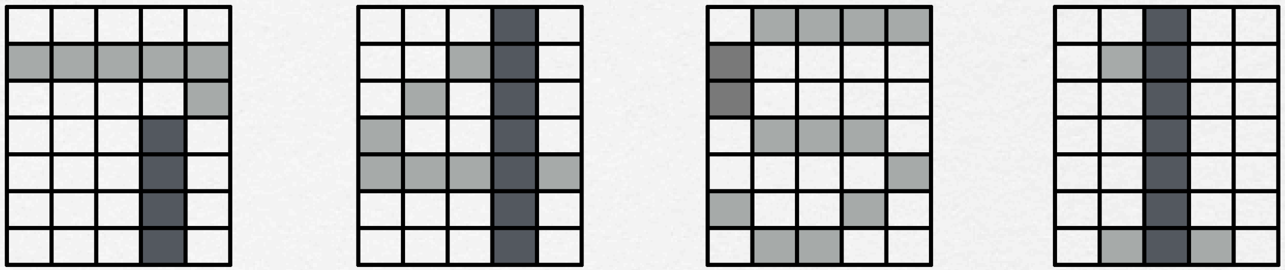


Convolution kernels as pattern detectors

0	0	0
1/3	1/3	1/3
0	0	0



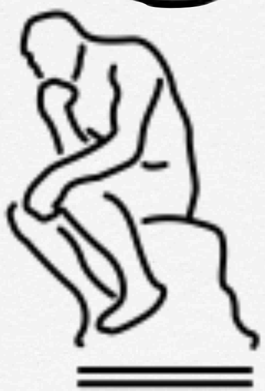
0	1/3	0
0	1/3	0
0	1/3	0



0	0	1/3
0	1/3	0
1/3	0	0

etc...

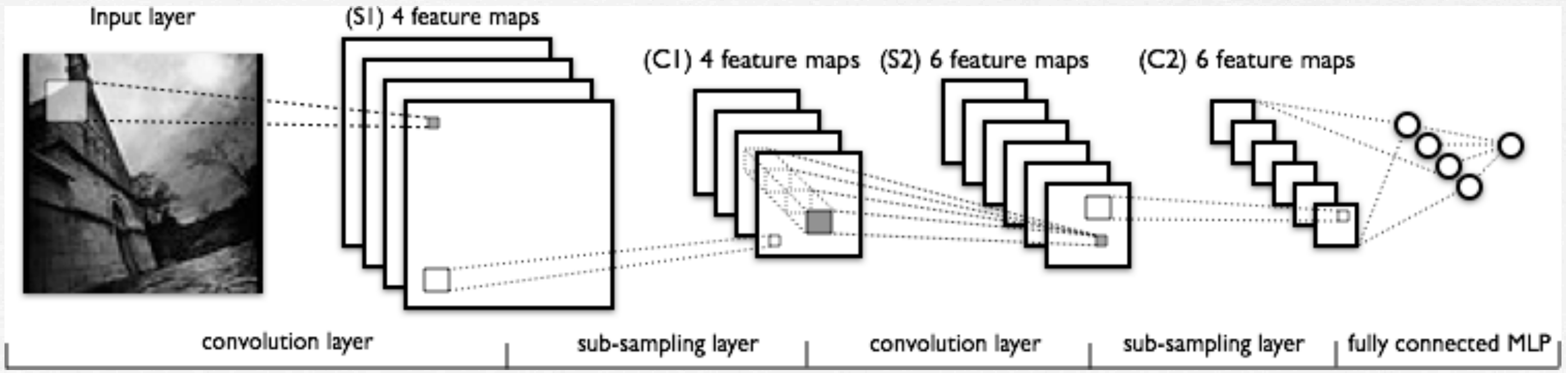
We would like a system that automatically finds the kernels that allow the detection of relevant low-level features (e.g., edges) and their combinations (e.g., higher-level features) to recognize objects



Convolutional Neural Networks (CNN)

- A neural network architecture (with local connections and shared weights) introduced by Yann LeCun in 1989.
- After joining AT&T Bell Labs in 1988, he applied convolutional networks to the task of recognizing handwritten characters (the initial goal was to build automatic mail-sorting machines). This work was one of the first (and one of the most cited) demonstrations that Neural Networks could be applied to "real-world" applications

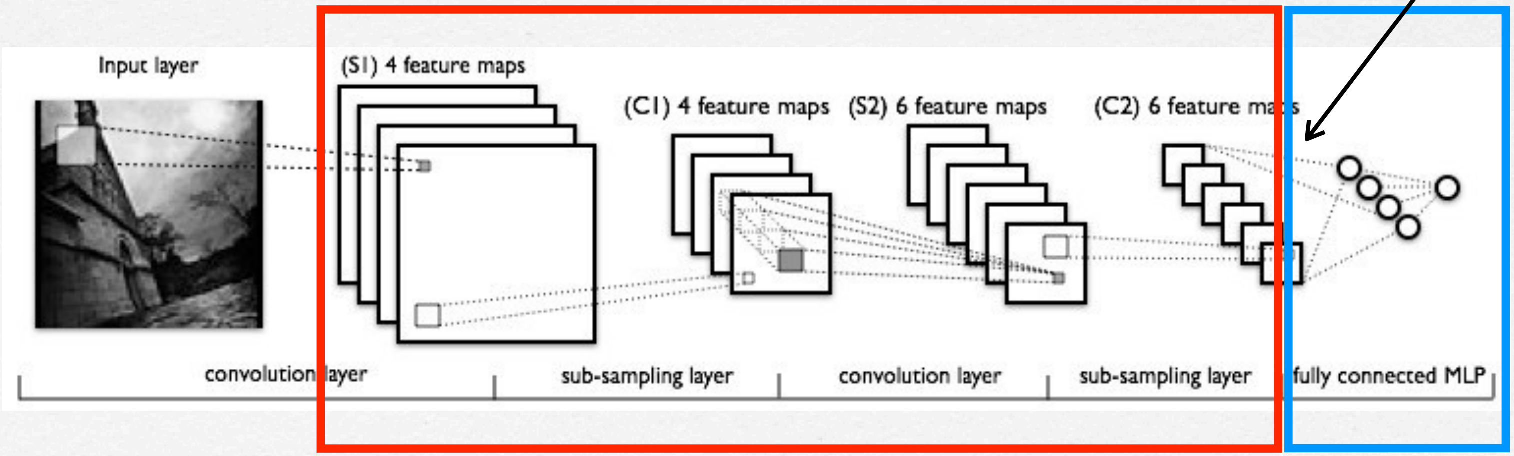
Le Net5



CNN: typical architecture

- convolution layers
- sub-sampling layers
- fully-connected layers

flattened images
2D -> 1D



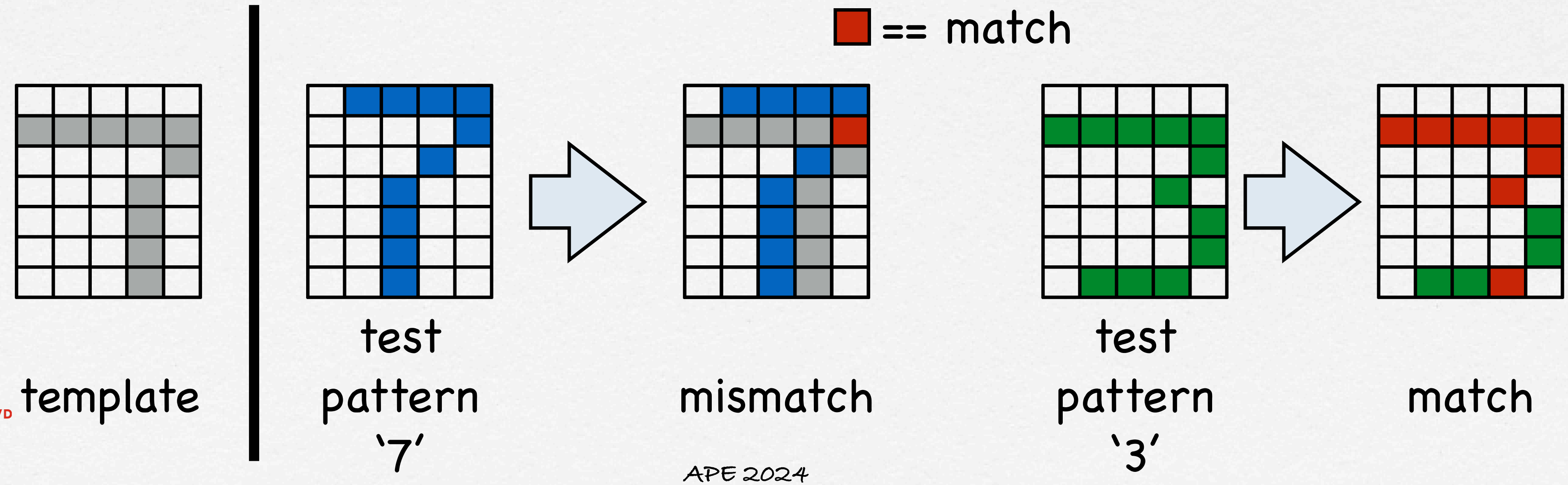
feature extraction

classification

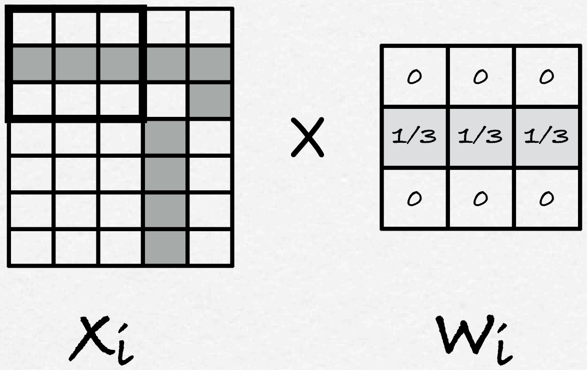
Towards invariant object recognition

A CNN solves a problem that arises with classical object recognition techniques based on naive pattern matching. That problem is:

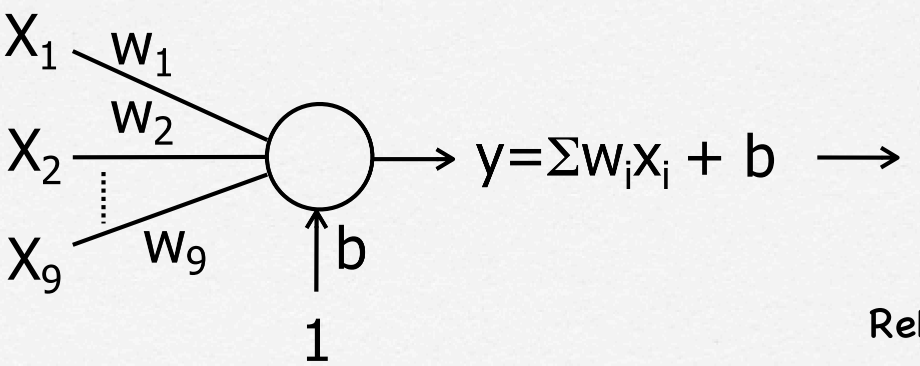
If an object does not appear in the inputs with the same size and at the same location, the overlap between the « template » and the object being recognized can be low.

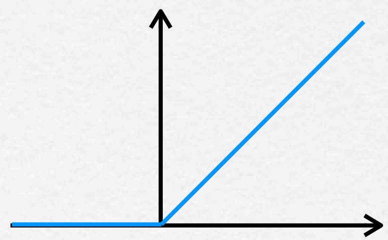


Convolution layers (1)



$= x1\ w1 + x2\ w2 + x3\ w3 + \ldots\ x9\ w9 \ = \sum w_i\ x_i$





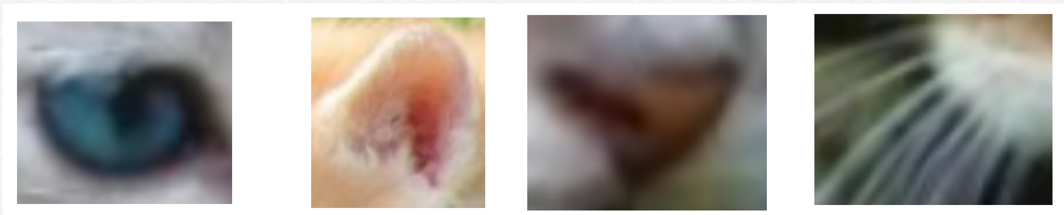
ReLU == Rectified Linear Unit

y = x if x > 0, else y = 0

Perceptron or a simple model of an artificial neuron

Convolution layers (2)

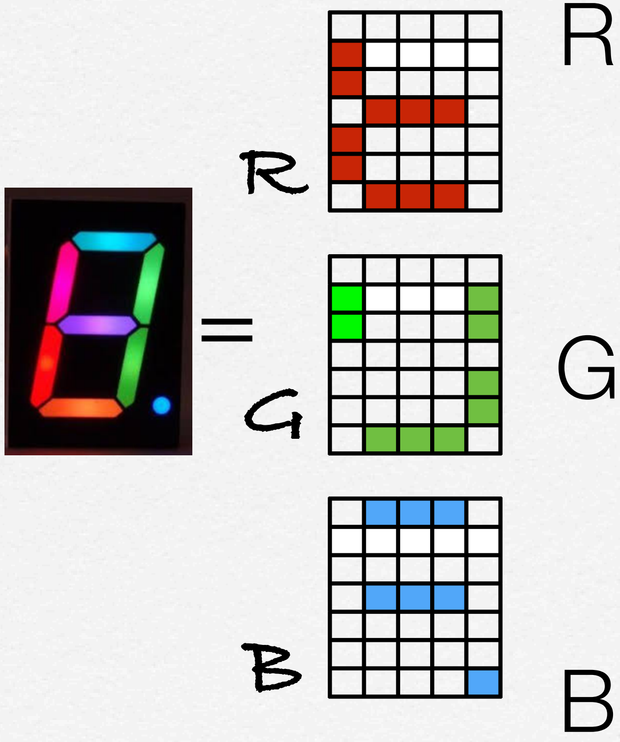
- In an example of learning to recognize cats, the objective of training a CNN is to learn the filters that **detect the features** that allow the network to recognize a cat independently of its position in the image, its orientation, its size, etc…
- **Examples of filters to be learned** are: eye, ear, nose and whiskers' detection filters:

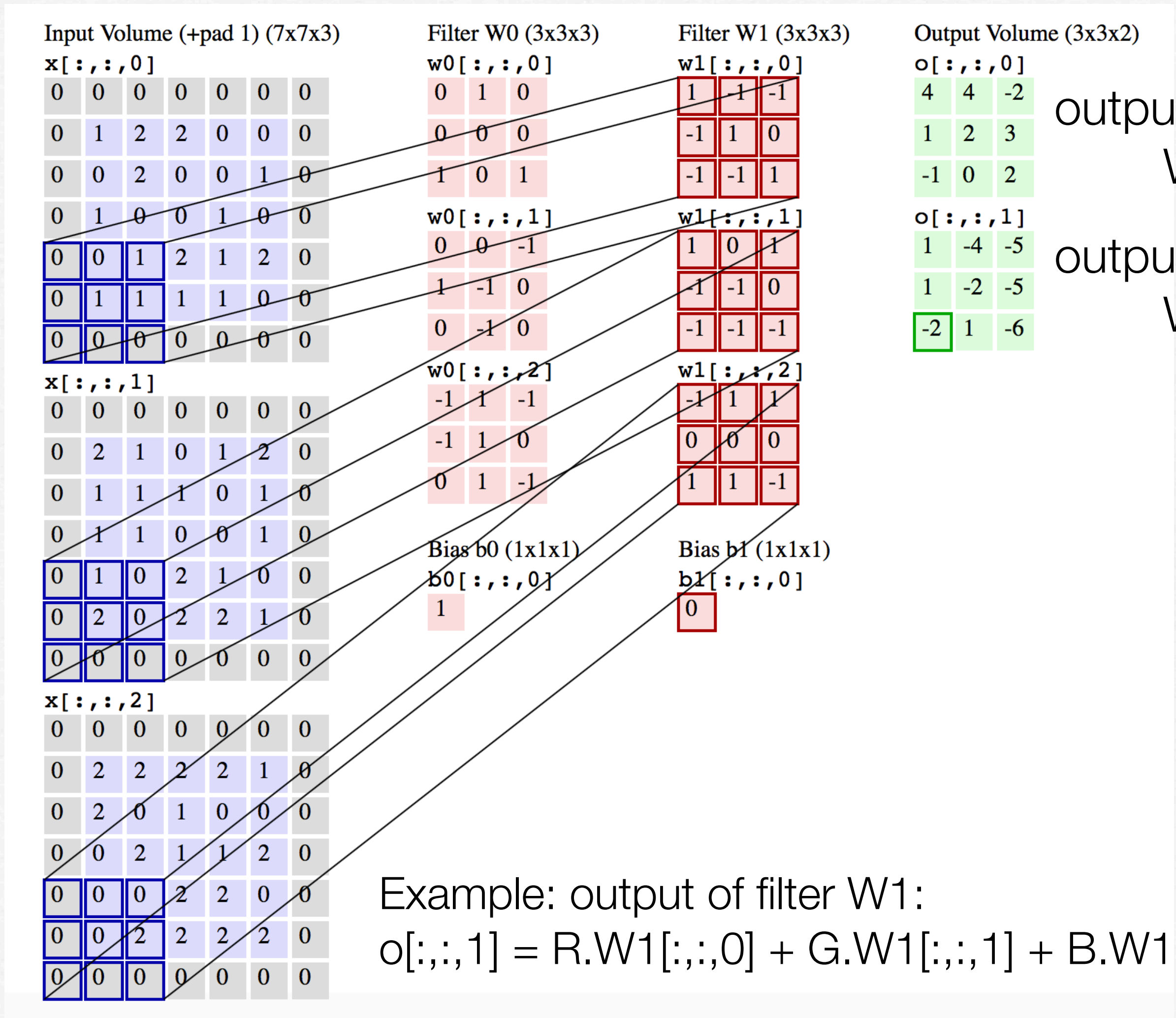


Convolution layers (3)

- Different kernel sizes (3x3, 5x5, 7x7, etc) allows the identification of features at different scales
- Researchers have found that multiple layers of 3x3 kernels can implement other kernel sizes.
- Some architectures (e.g., inception) use different kernel sizes in parallel at each layer.

Dealing with RGB-inputs





output of filter W0

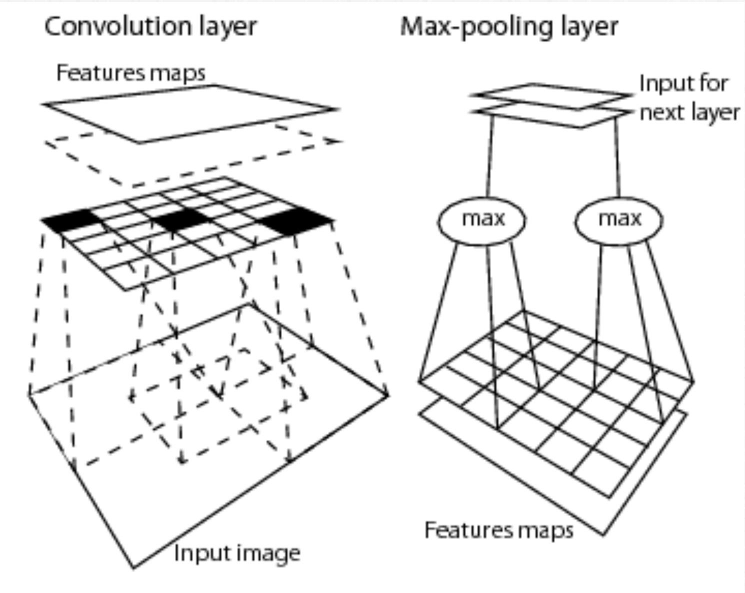
output of filter W1

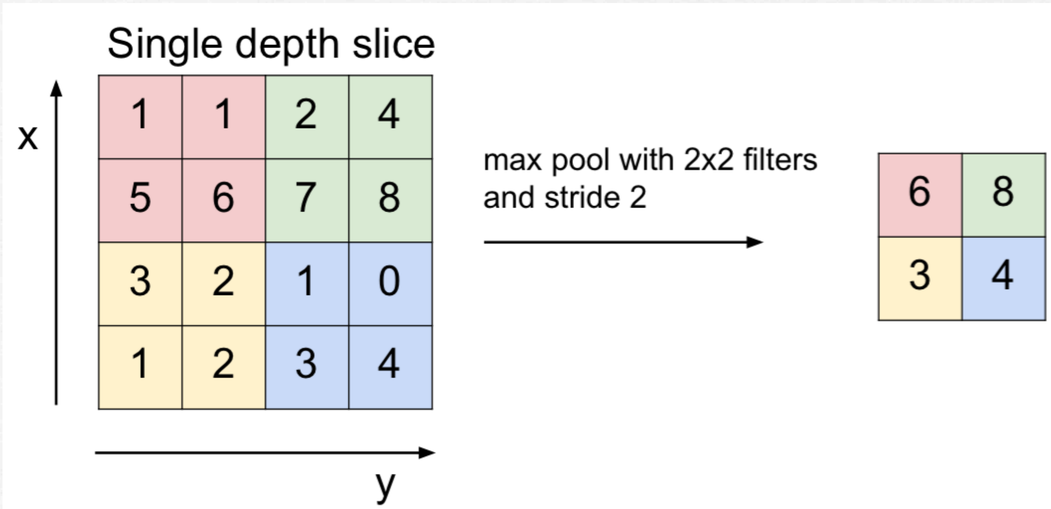
Example: output of filter W1:
o[:,:,1] = R.W1[:,:,0] + G.W1[:,:,1] + B.W1[:,:,2] + b1

from http://cs231n.github.io/convolutional-networks/

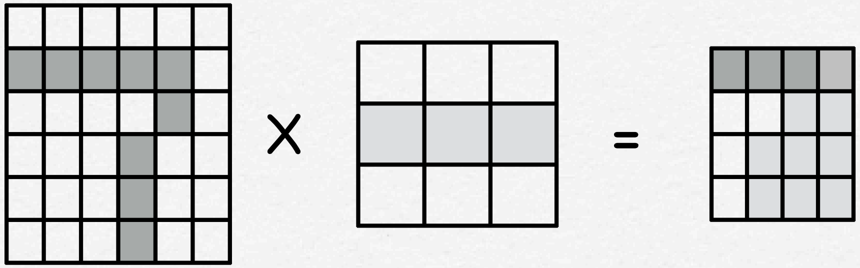
Sub-sampling layers

- Maxpool after a convolution layer eliminates non maximal values: it is a form of non-linear down-sampling that reduces computation for upper layers and provides a « summary » of statistics of features in lower layers. The resulting images are smaller than those having been processed in previous layers. /* average pooling is an option too */

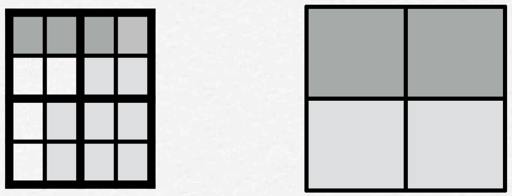




Maxpooling example



convolution -> pattern detection over all the input image



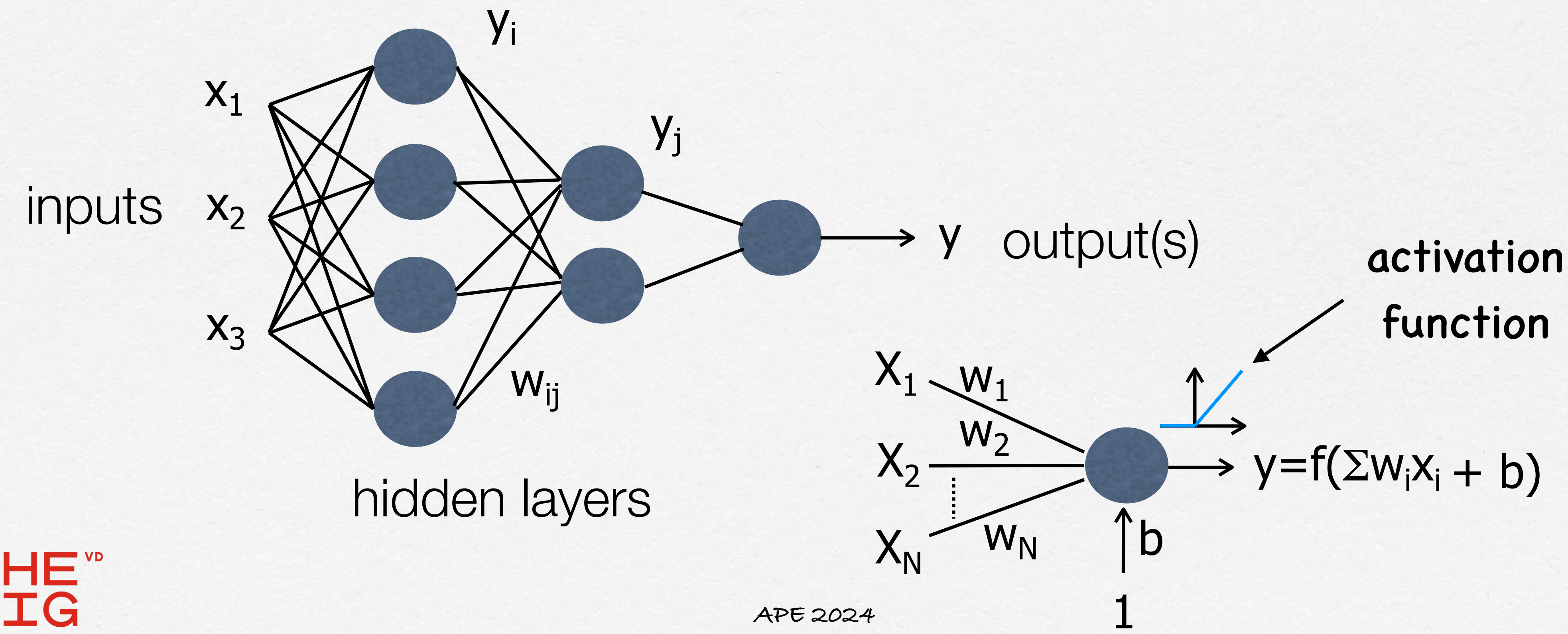
maxpooling -> summary statistics of pattern detection

"a horizontal line is present in the upper part of the image"

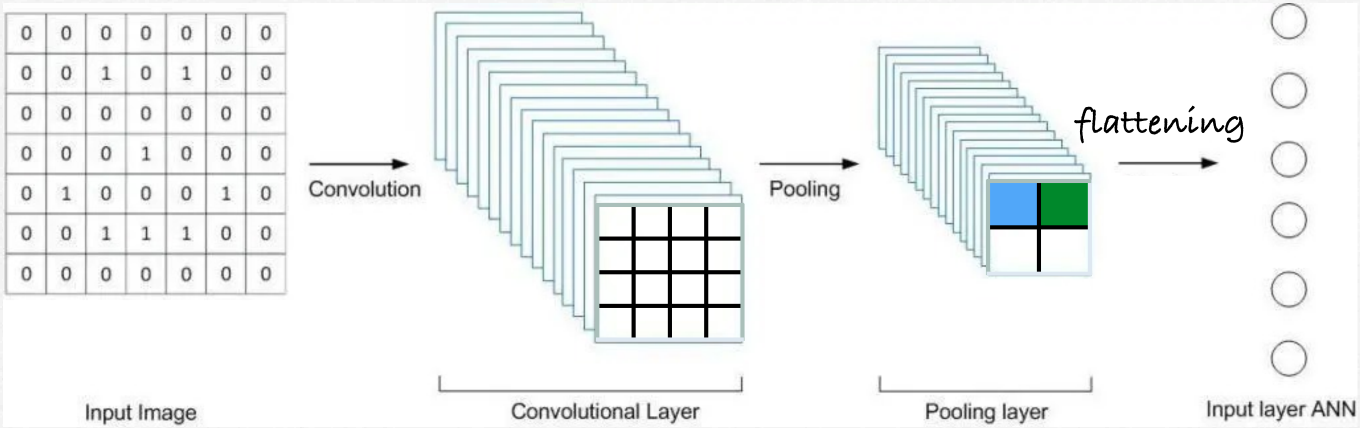
HEIG VD

Fully-connected layers

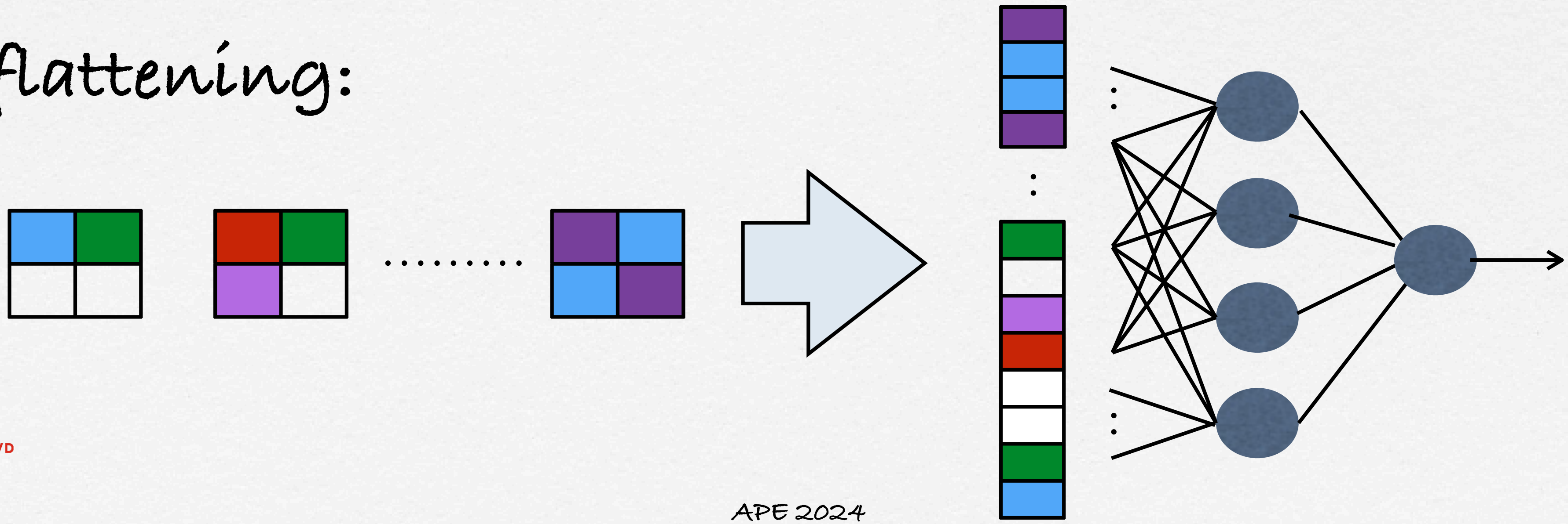
- A fully-connected layer is what we used to call a Multi-Layer Perceptron or an artificial neural network. Today we refer to them to as "shallow networks", because they usually consist of few layers of Perceptron units.

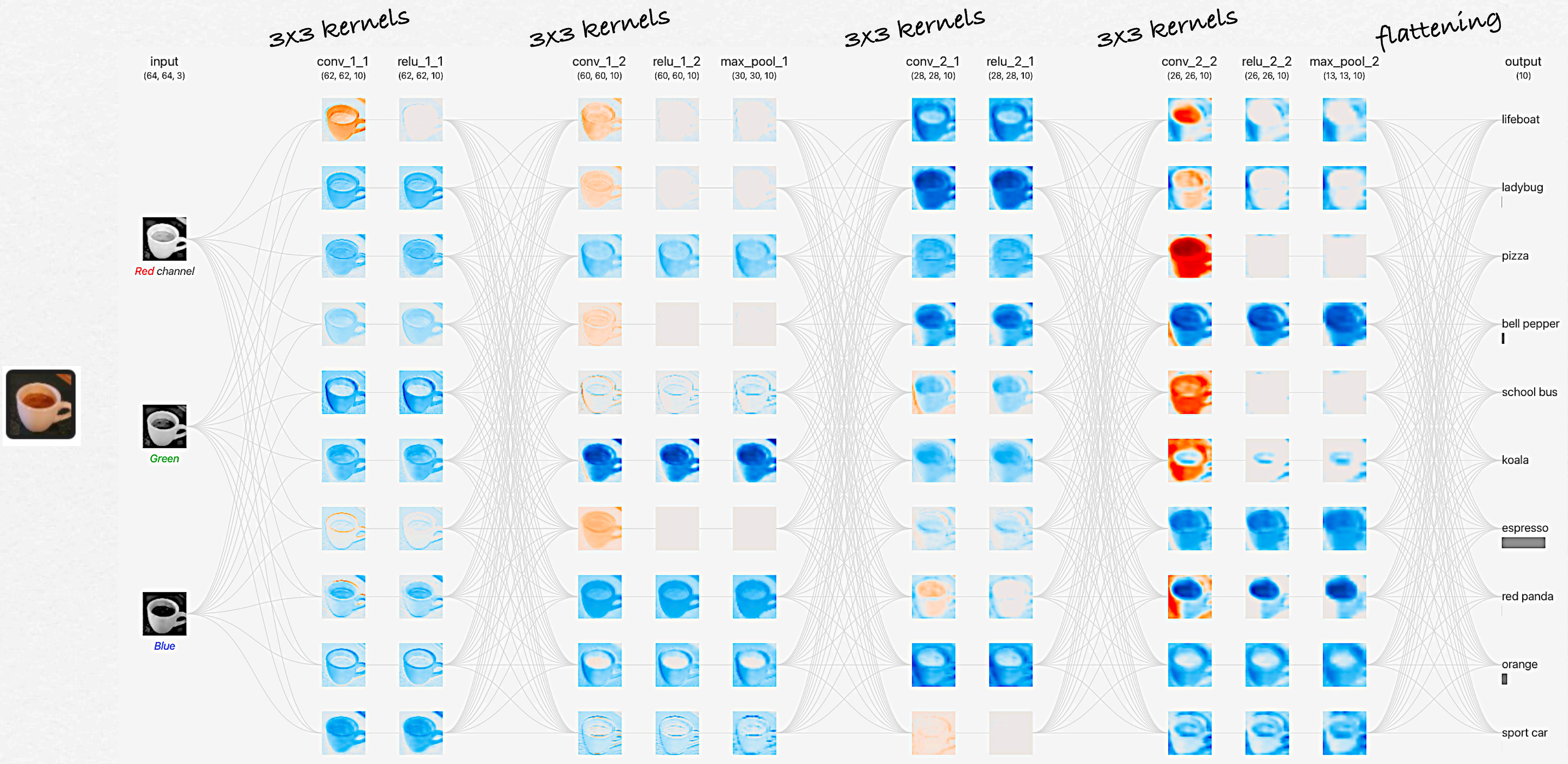


Flattening phase



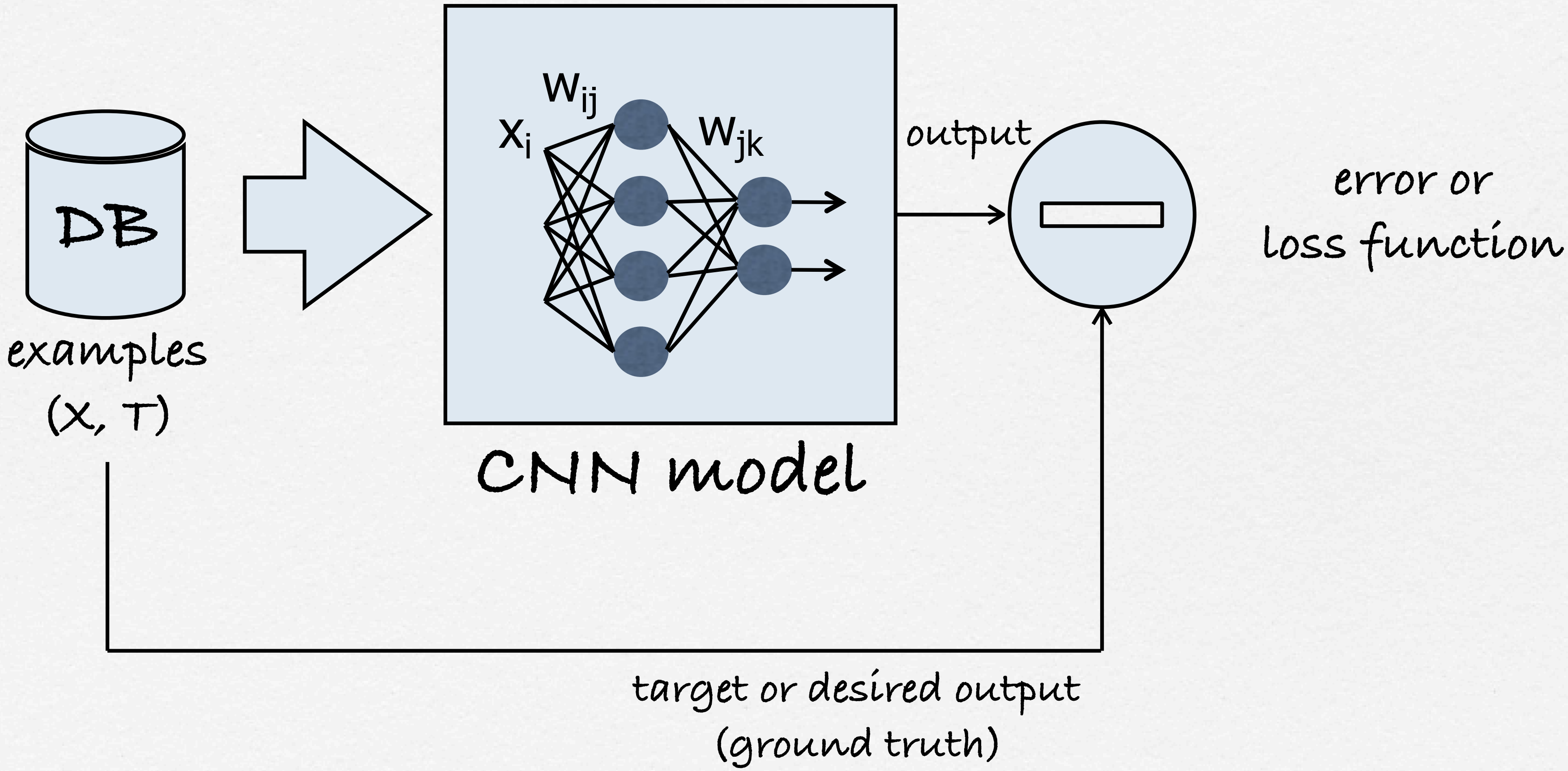
flattening:





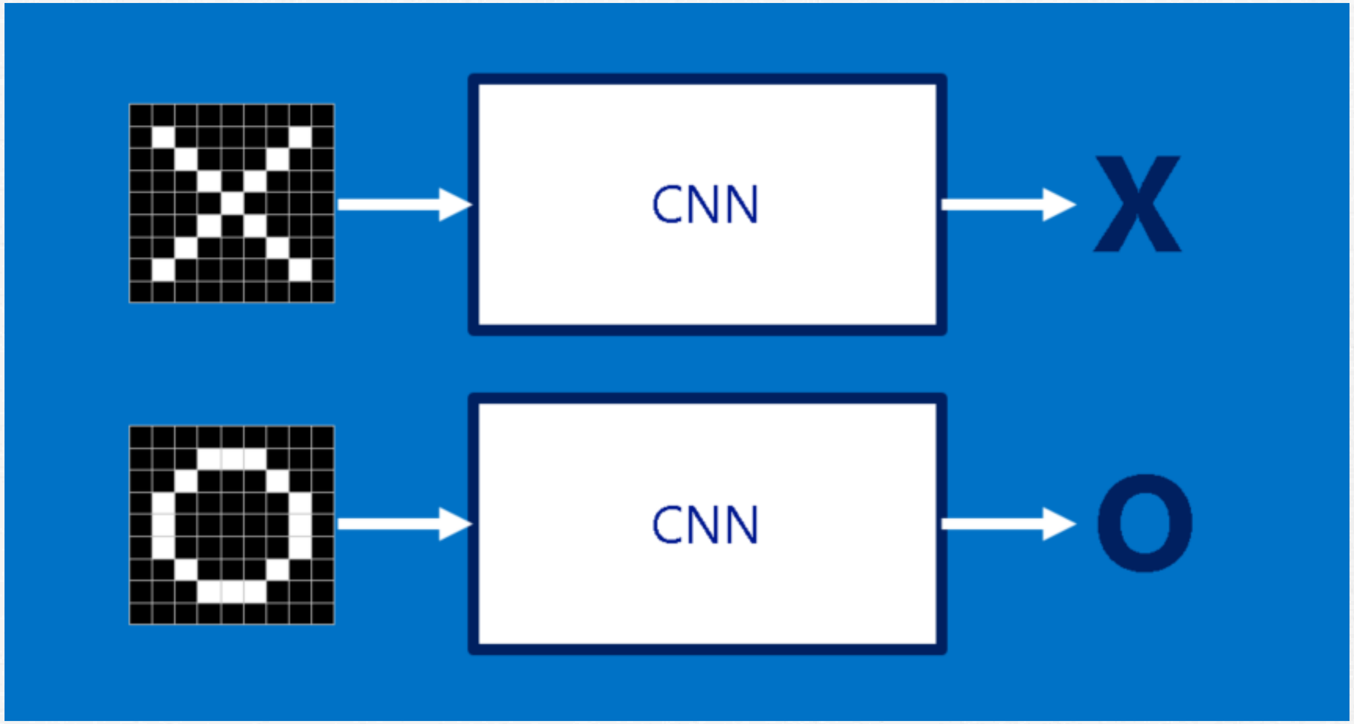
https://poloclub.github.io/cnn-explainer/

The gradient descent trick



How do CNN's work ?

(by Brandon Rohrer)



https://brohrer.github.io/how_convolutional_neural_networks_work.html

https://www.youtube.com/watch?v=JB8T_zN7ZC0

Hands-on Deep Learning (1)

Keras

TensorFlow numerical computation using data flow graphs

Google Colaboratory

- Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or ~~Theano~~. Its primary author and maintainer is François Chollet.
- Keras is a leading deep learning framework for Python, with over 50,000 users and over 200 open-source contributors. Keras is in use at a considerable number of startups, research labs (including CERN, Microsoft Research and OpenAI), and large companies such as Netflix, Yelp, Square, Google, etc.

Hands-on Deep Learning (2)

```
# 0. Import libraries and modules

# 1. Prepare training and test data

# 2. Define model architecture
model = Sequential()

model.add(Convolution2D(32, 3, 3, activation='relu', input_shape=(1,28,28)))
model.add(Convolution2D(32, 3, 3, activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))

# 3. Compile model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

# 4. Fit model on training data
model.fit(X_train, Y_train,
          batch_size=32, epochs=300, validation_data=(val_input, val_target))

# 5. Evaluate model on test data
score = model.evaluate(X_test, Y_test, verbose=0)
```

number of filters

size of filters (3x3)

convolutional layer

sampling layer

flatten after conv. layers

fully connected layer of 128 neurons

number of outputs

loss function

learning algorithm

CNN-based digit recognition