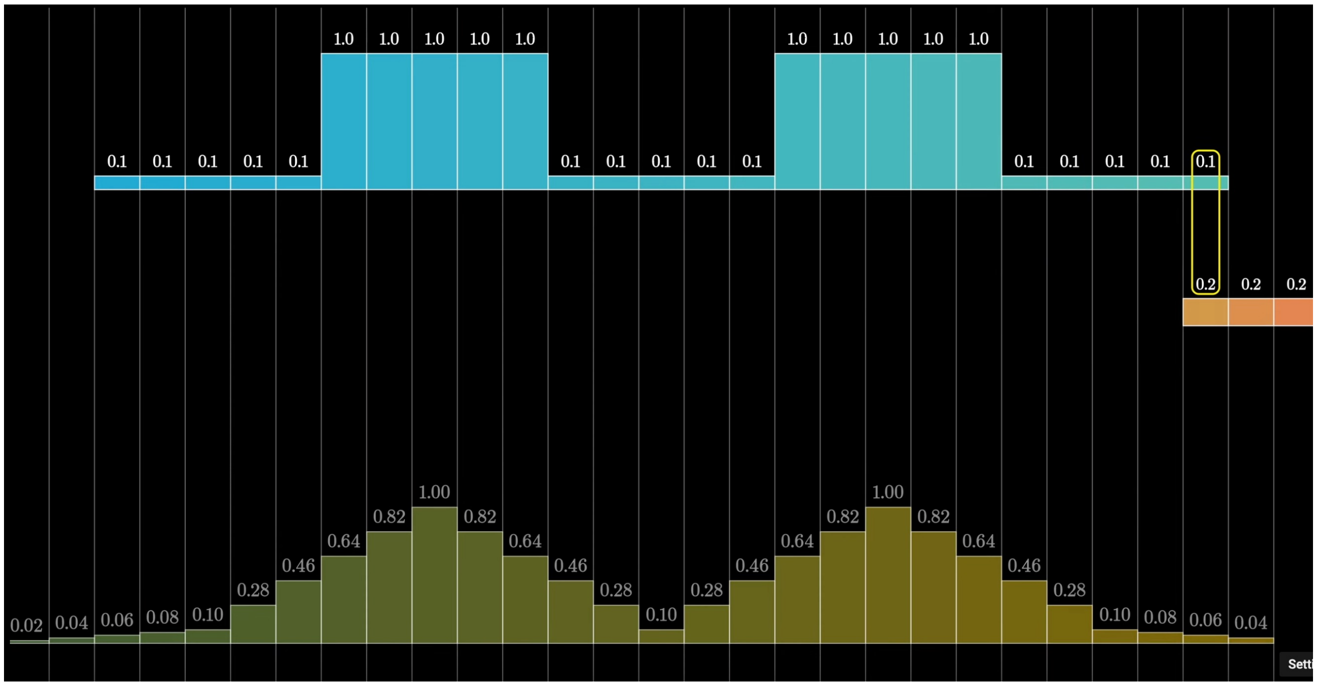
Deep Learning applied to EEG

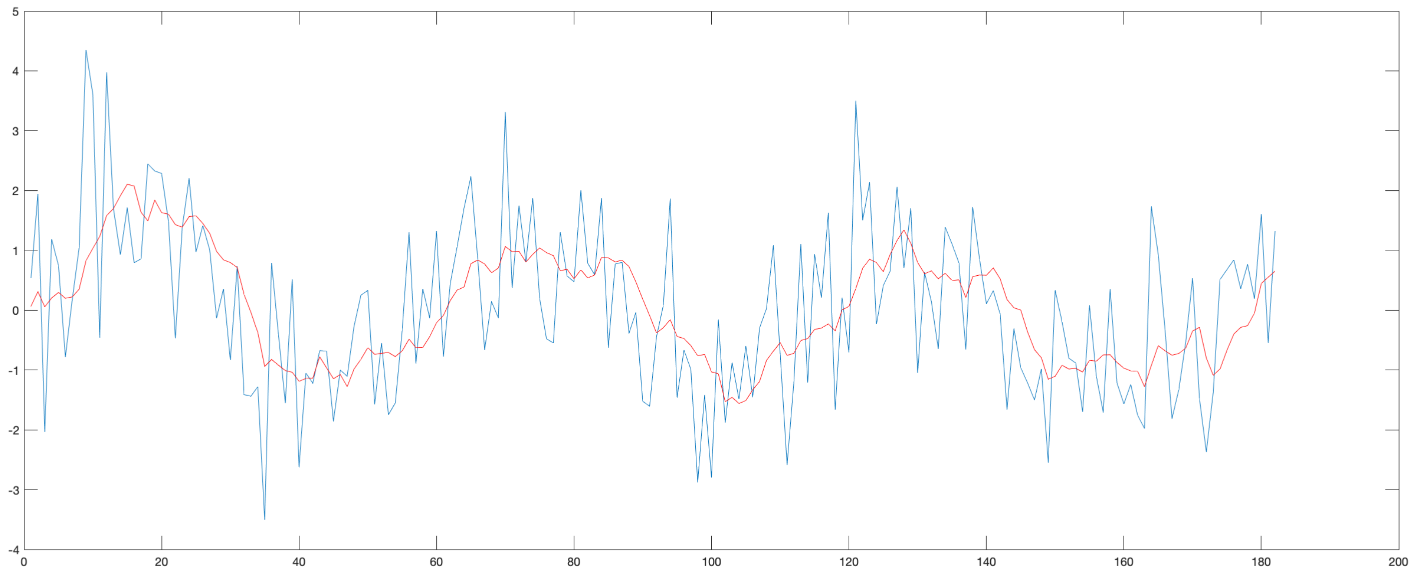
Arnaud Delorme

What kind of operation?



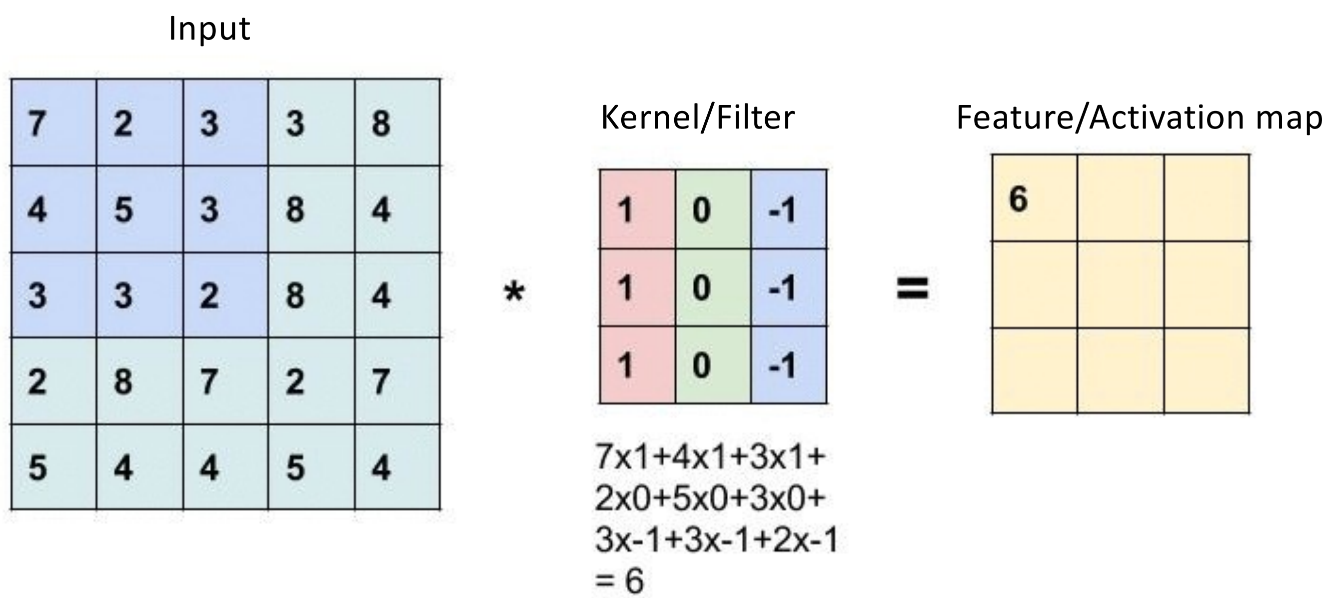
https://youtu.be/KuXjwB4LzSA

Moving average – 1D Convolution



filter = 1/8 * [1 1 1 1 1 1 1 1];

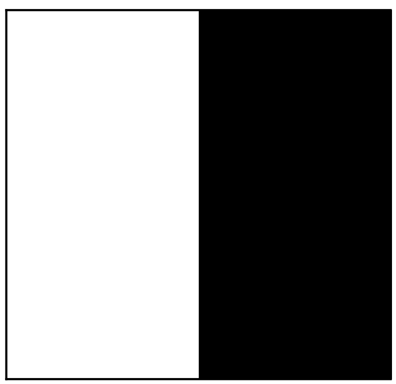
2D Convolution

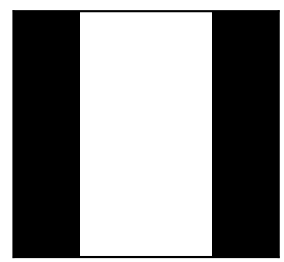


https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

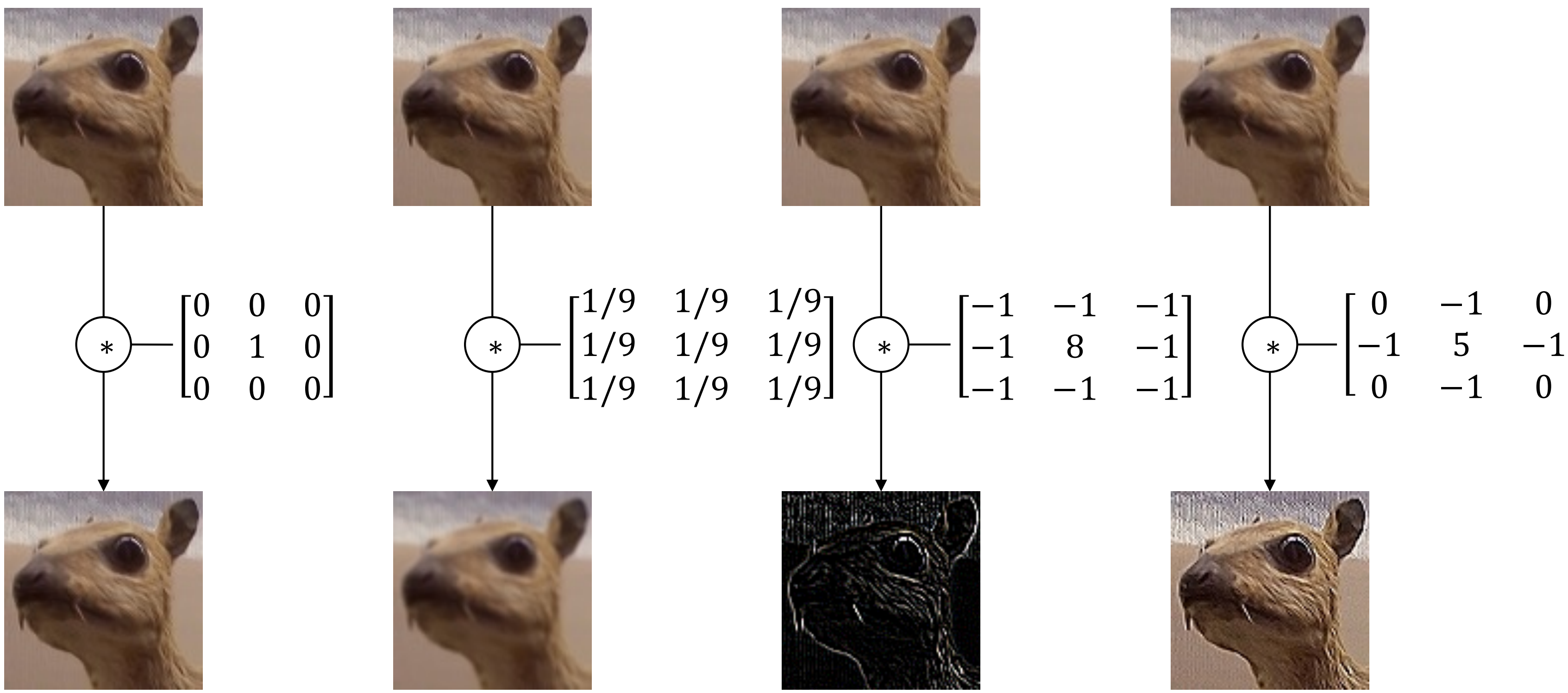
Edge detection filter

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}
*
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1
\end{bmatrix}
=
\begin{bmatrix}
0 & 3 & 3 & 0 \\
0 & 3 & 3 & 0 \\
0 & 3 & 3 & 0 \\
0 & 3 & 3 & 0
\end{bmatrix}
$$





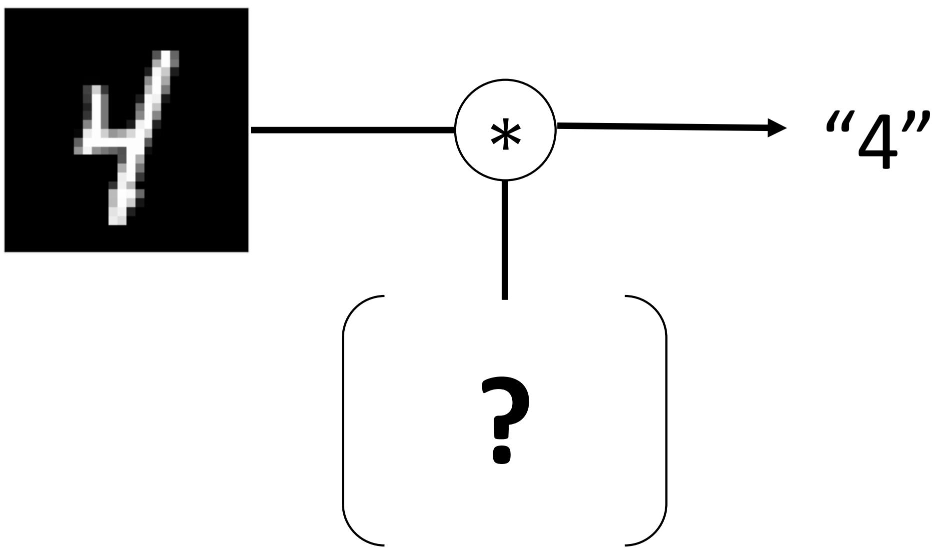
There are many types of filters



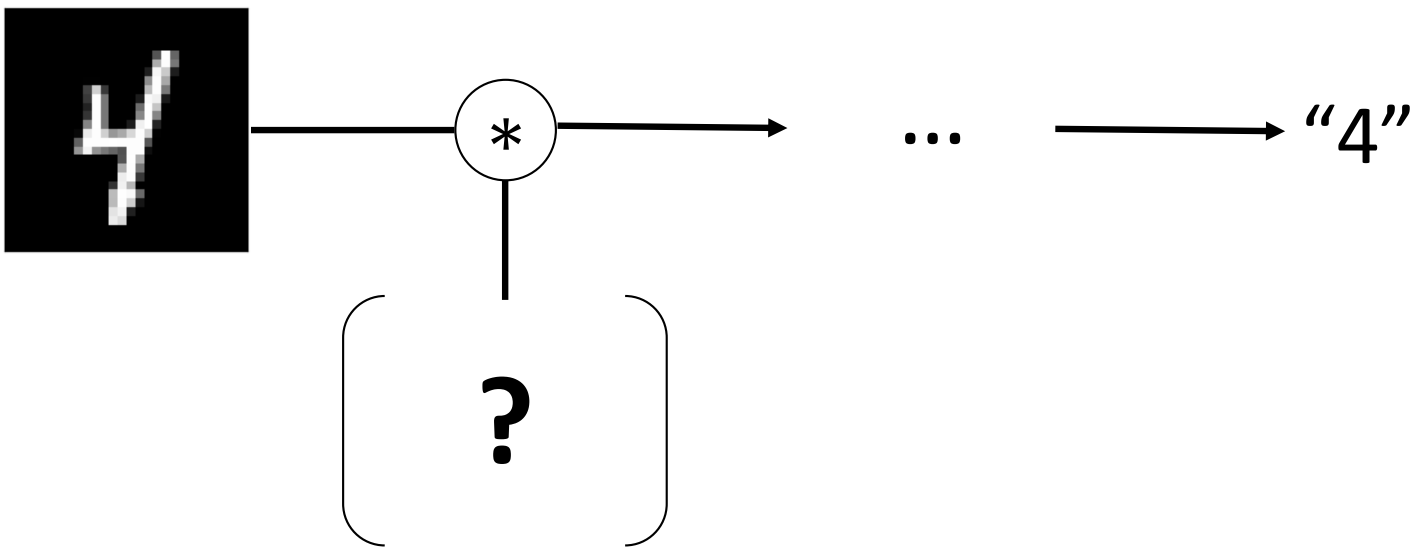
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

https://en.wikipedia.org/wiki/Kernel_(image_processing)

Filter for digit detection?



Filter for digit detection?



Let machine learn

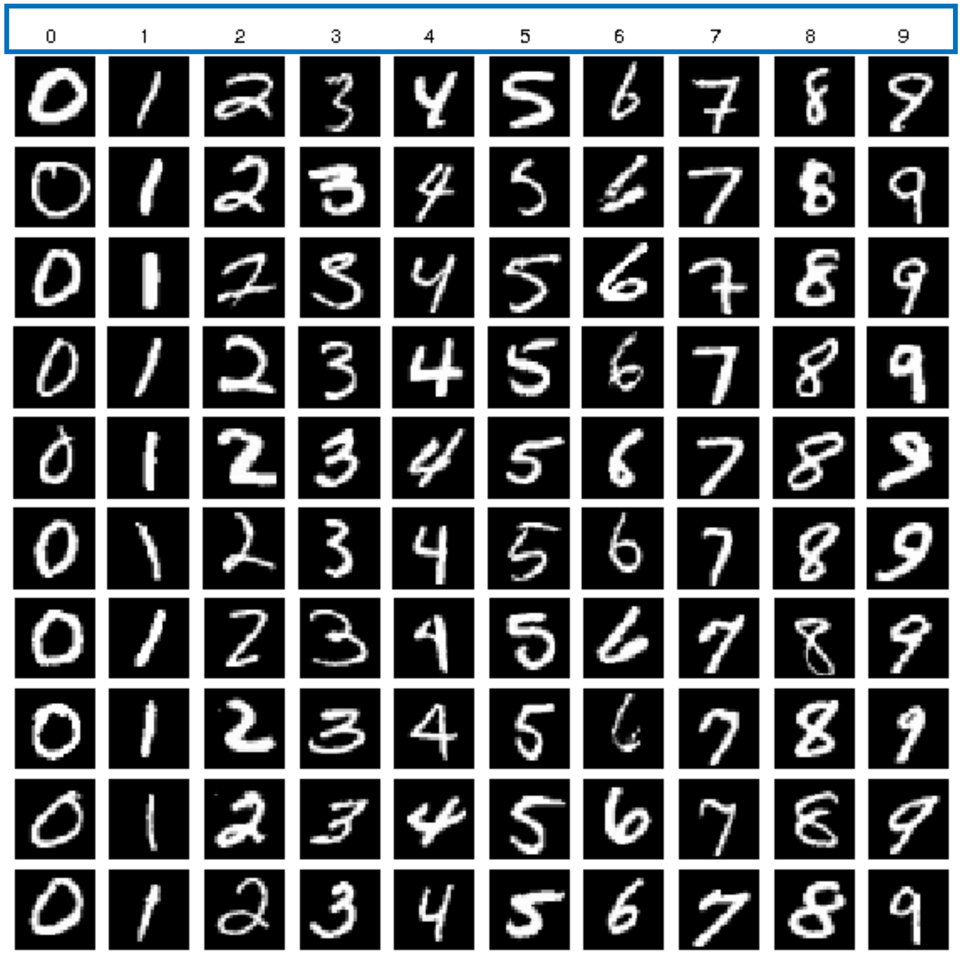
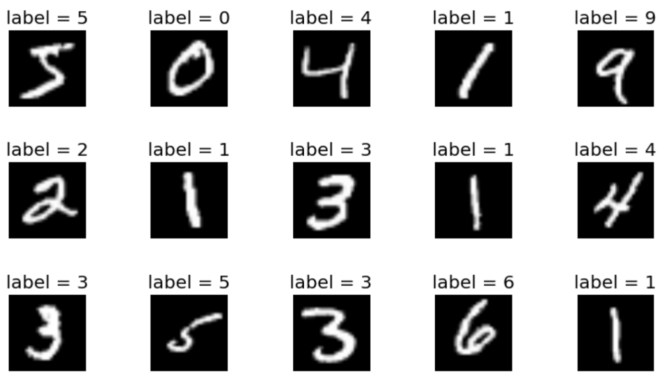
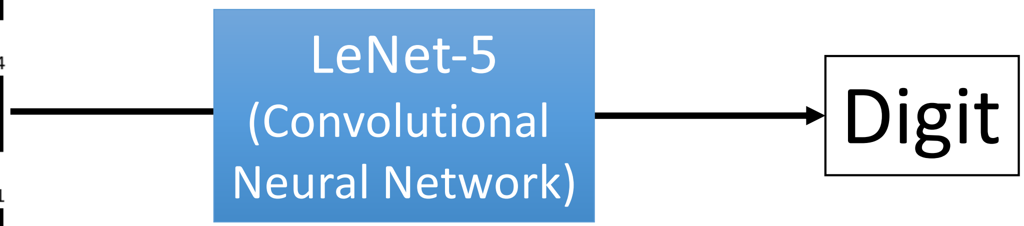


Image processing + Data + Neural network

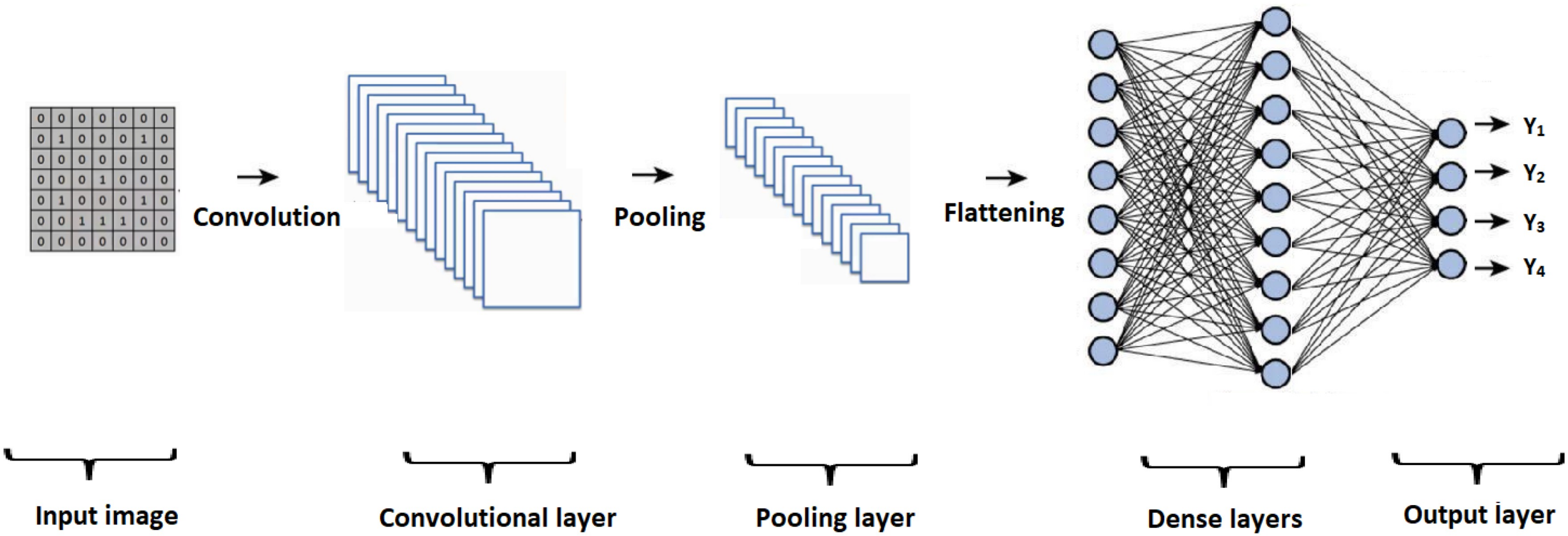


MNIST dataset



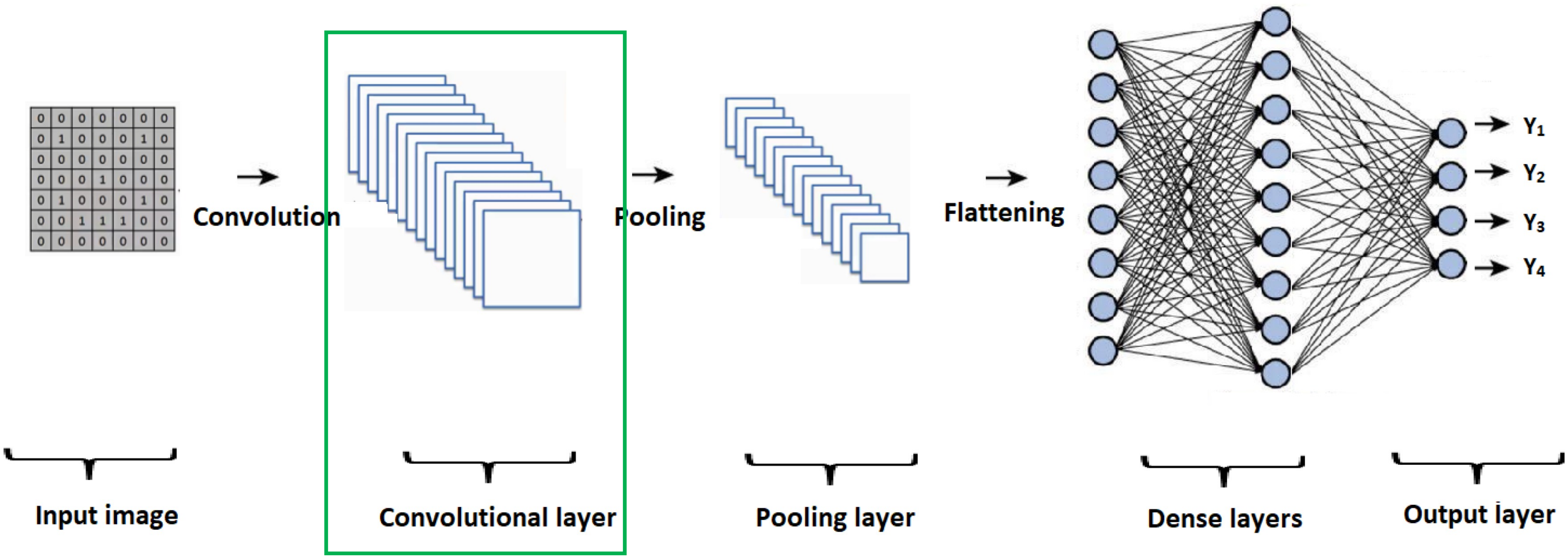
Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

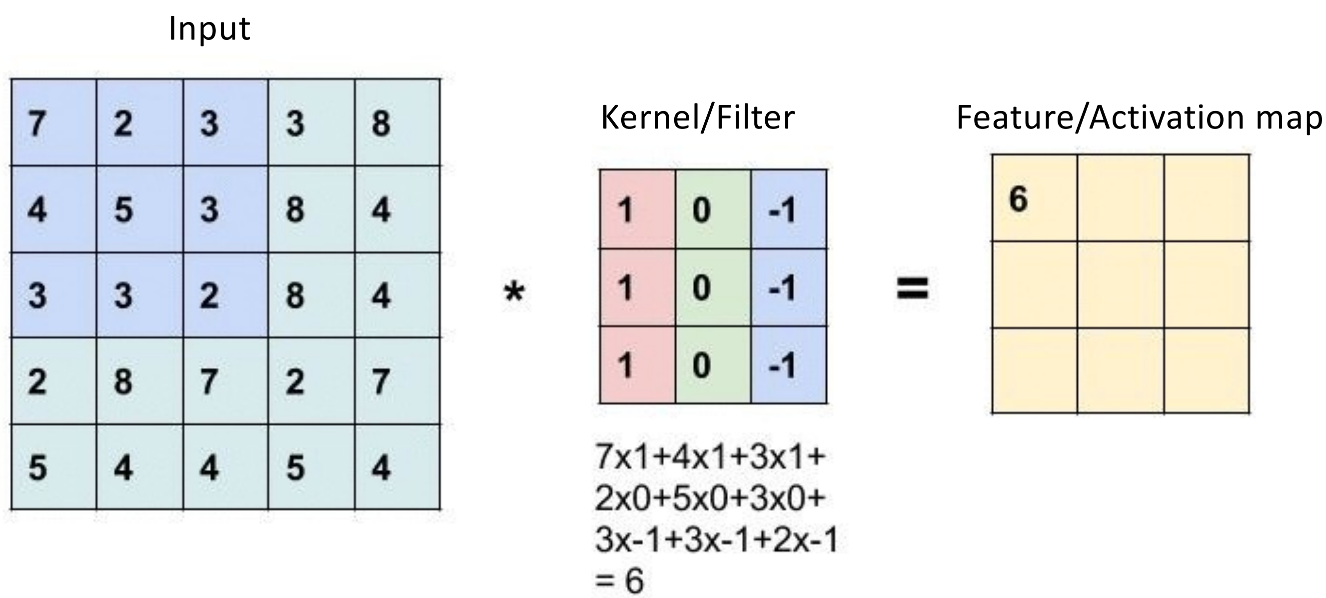
Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

Convolutional layer

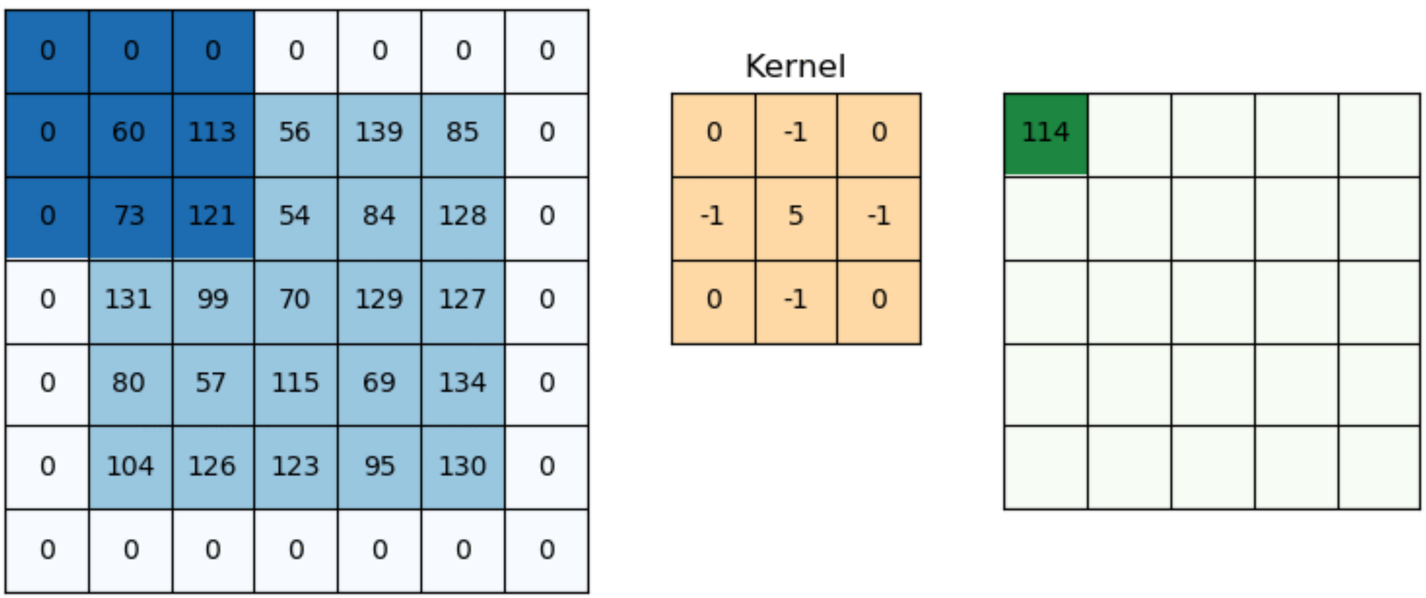
- Convolution operation



https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

Convolutional layer

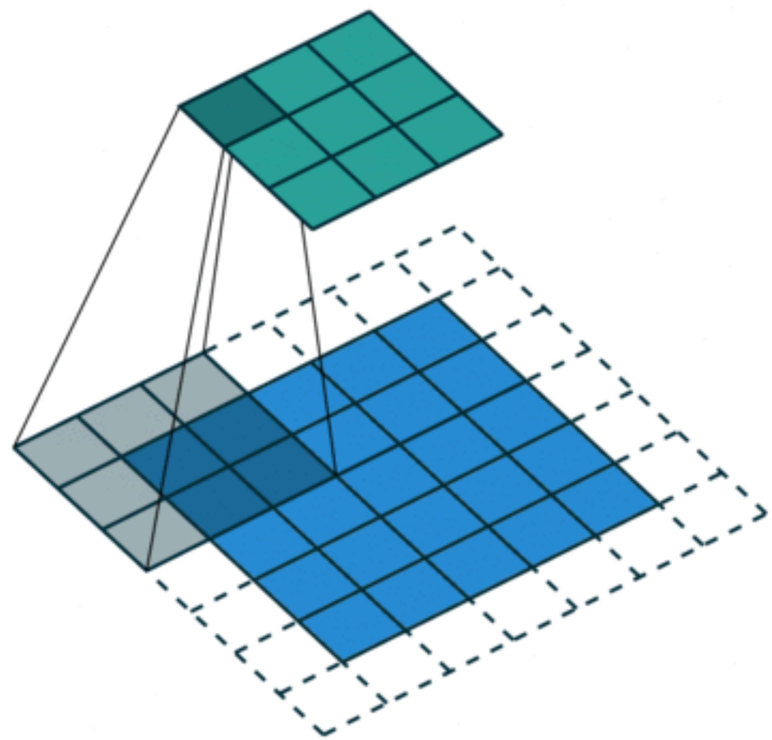
- Padding



https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

Convolutional layer

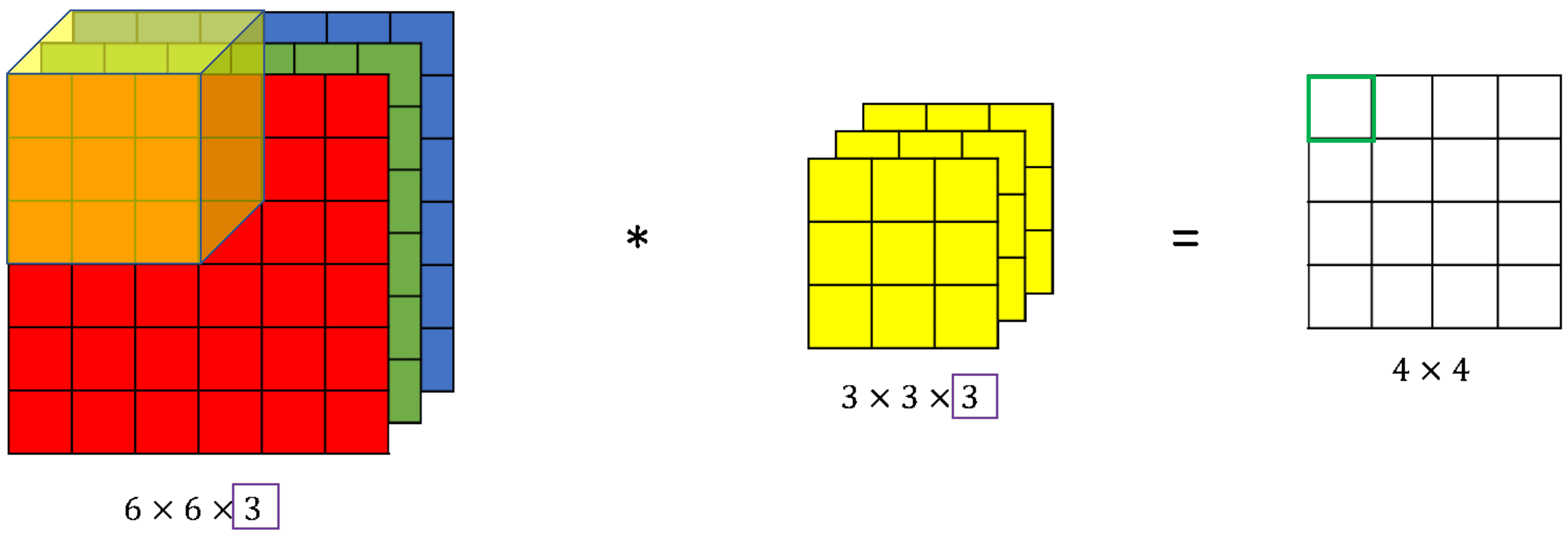
- Stride



Example: Padding = 1, Stride = 2

Convolutional layer

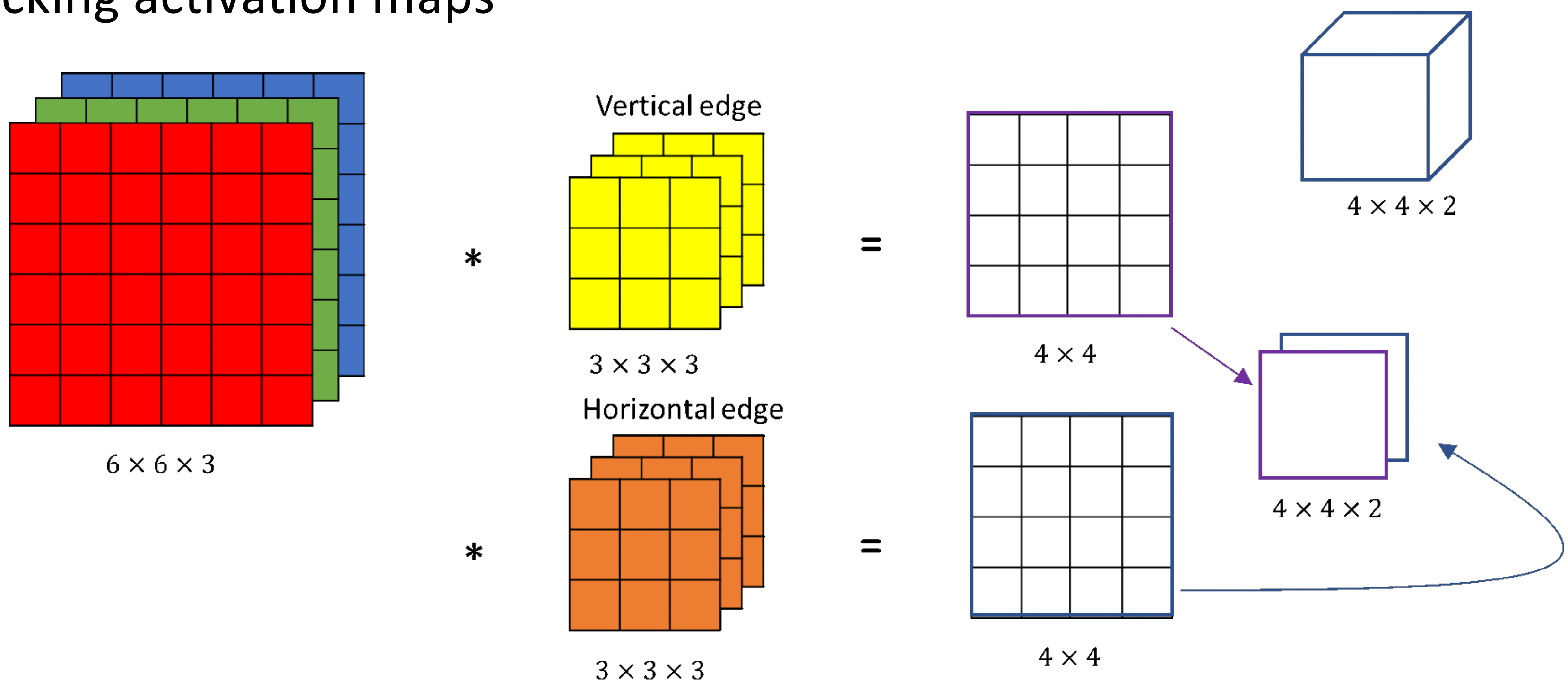
- Convolution over volume



Andrew Ng

Convolutional layer

- Stacking activation maps

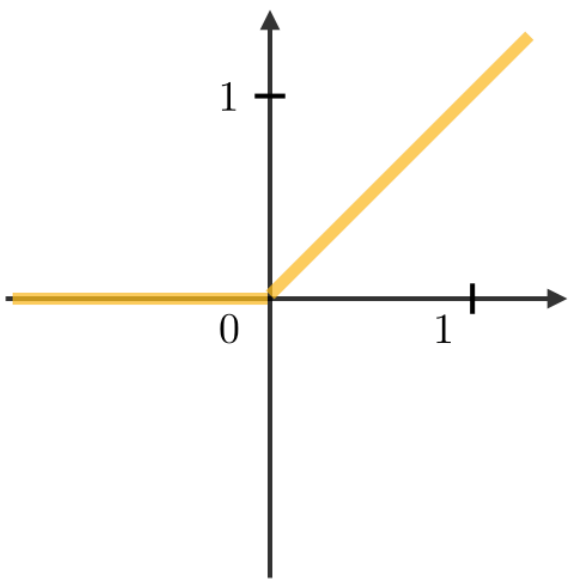


Andrew Ng

Activation functions

- Rectified Linear Unit (ReLU)

$$g(z) = \max(0, z)$$



ReLU Layer

Filter 1 Feature Map

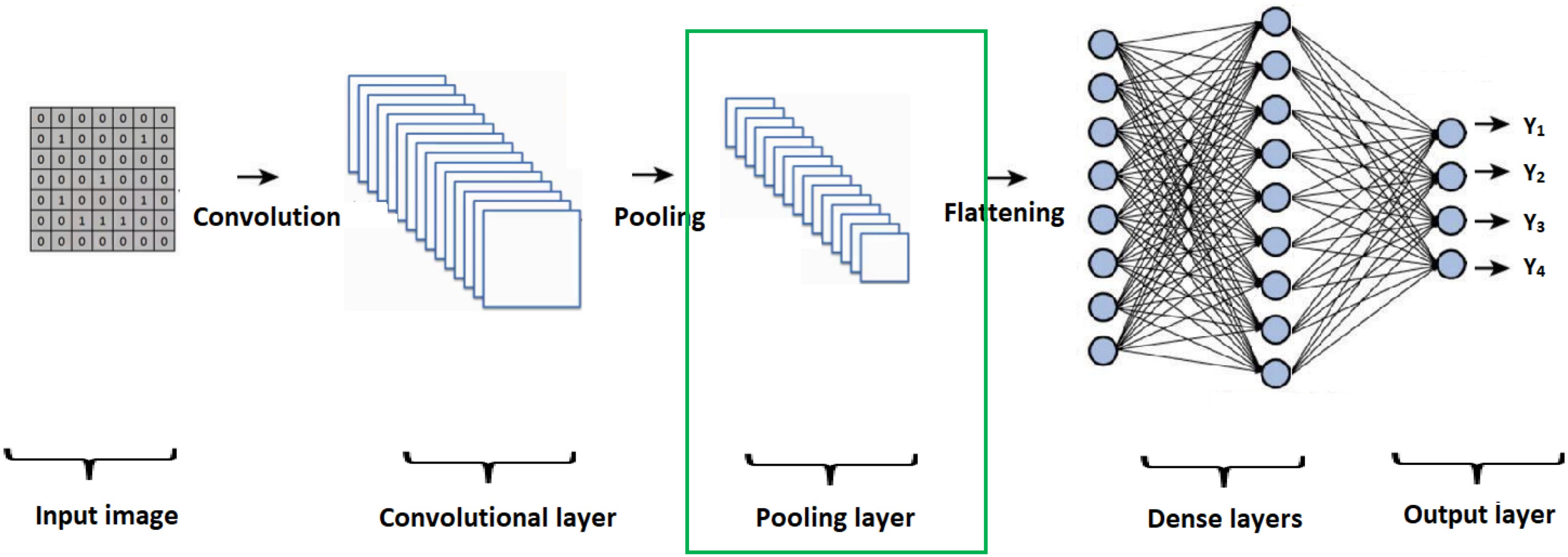
9	3	5	-8
-6	2	-3	1
1	3	4	1
3	-4	5	1

→

9	3	5	0
0	2	0	1
1	3	4	1
3	0	5	1

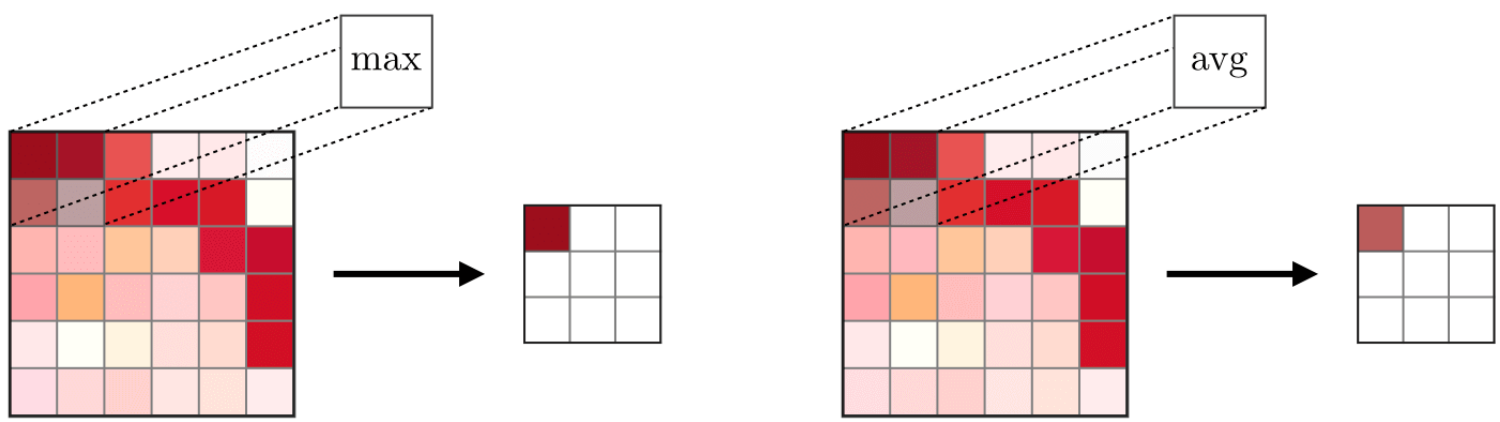
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

Convolutional Neural Network Architecture



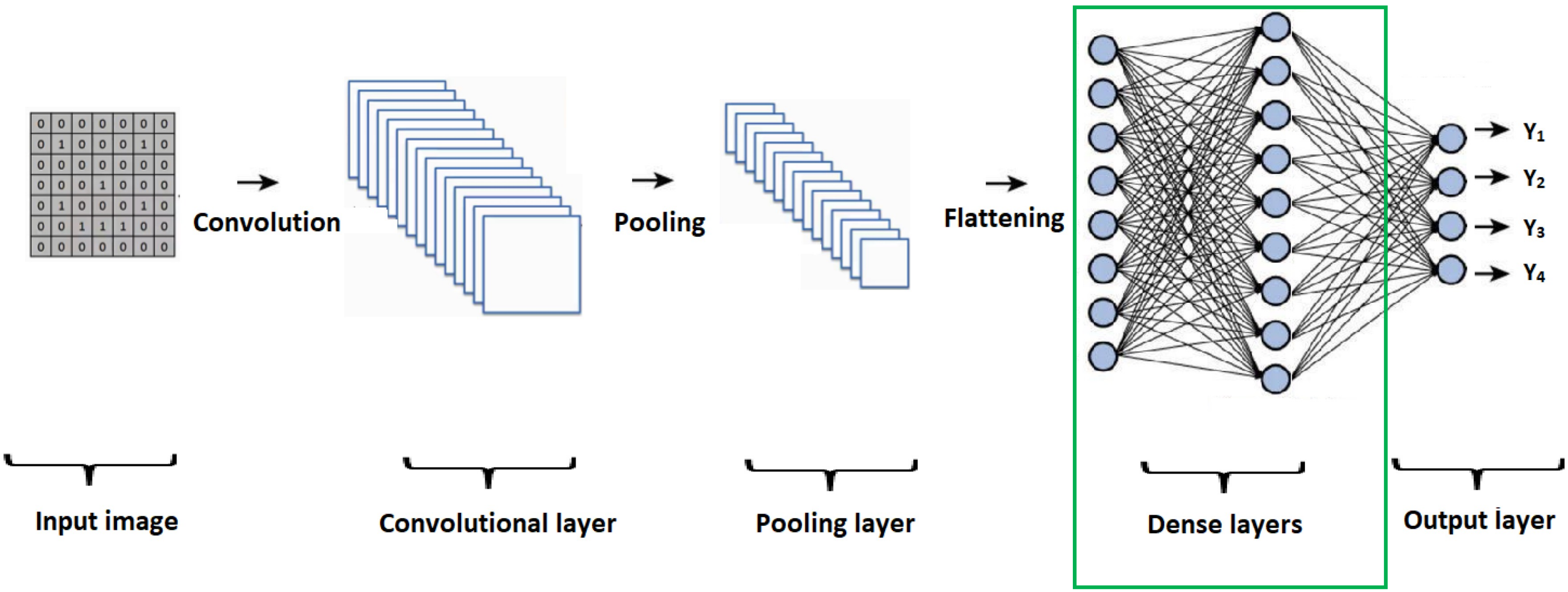
https://www.mdpi.com/2076-3417/10/4/1245/htm

Pooling layer



https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

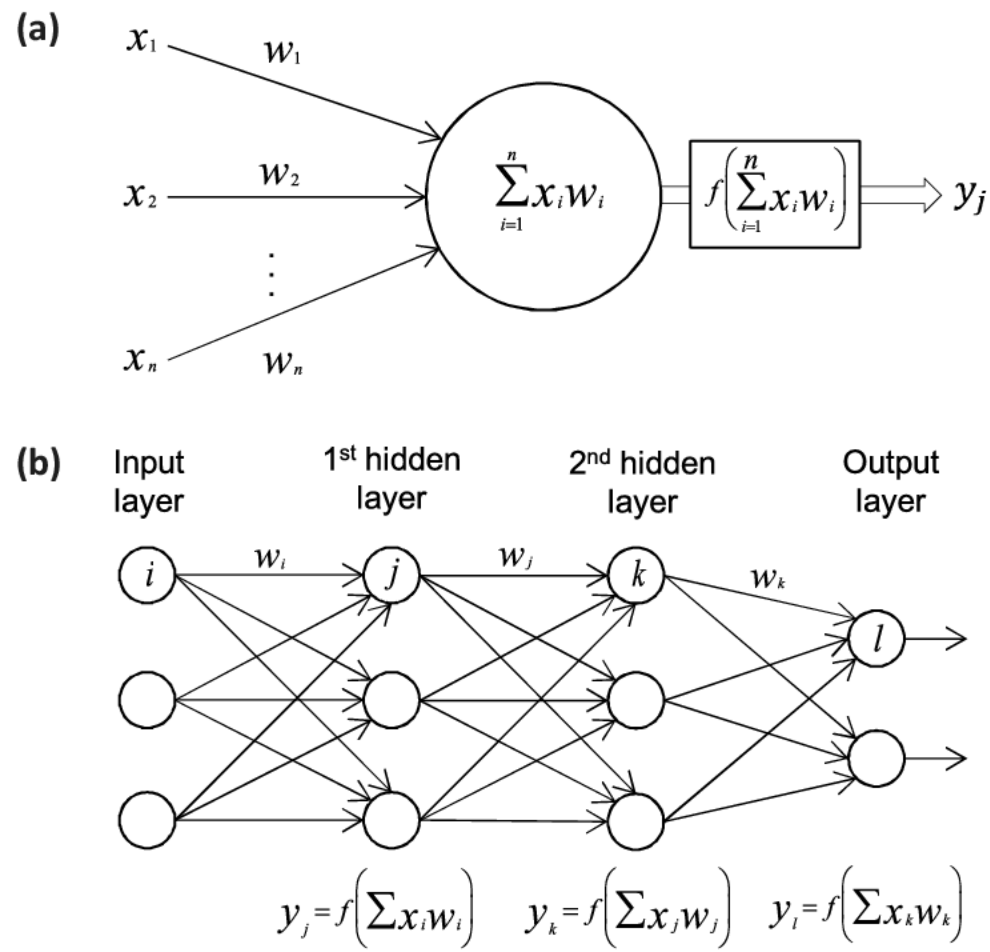
Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm

Fully connected layer

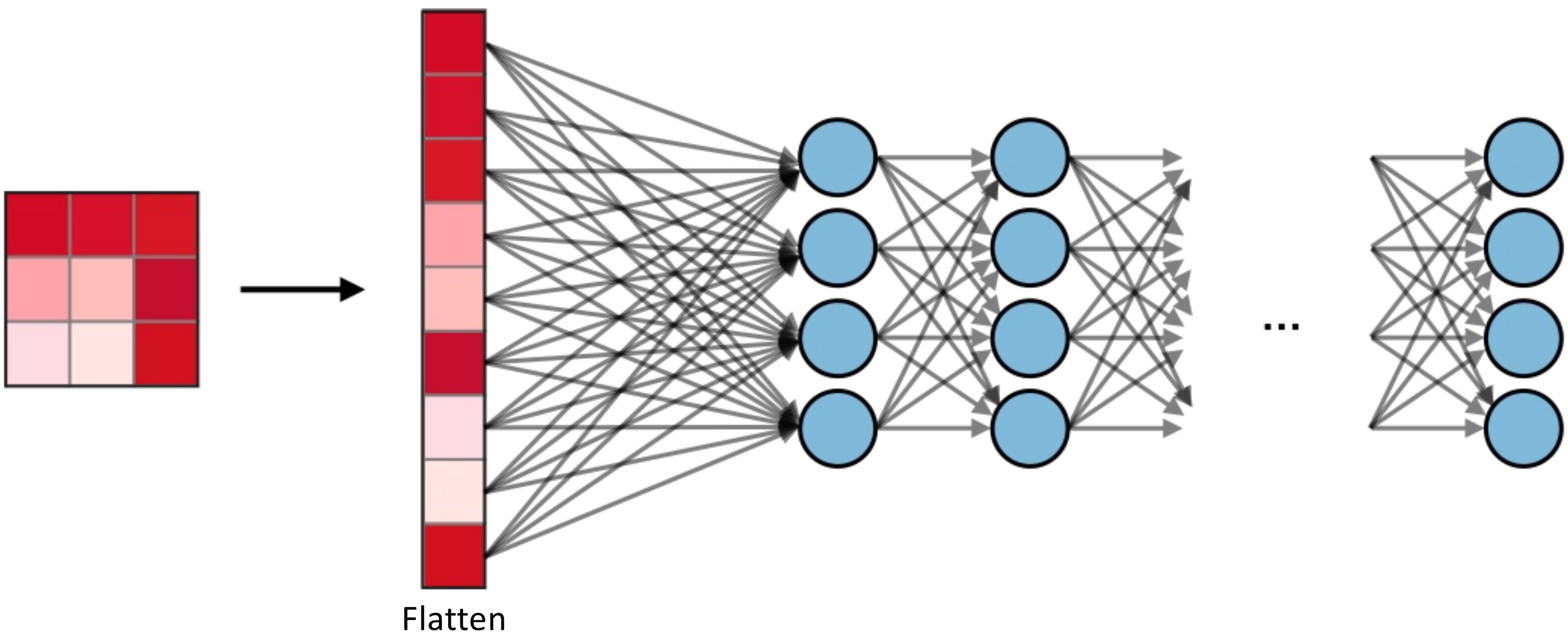
- Also known as dense layer
- Each input is connected to all hidden units



https://www.researchgate.net/figure/a-The-building-block-of-deep-neural-networks-artificial-neuron-or-node-Each-input-x_fig1_312205163

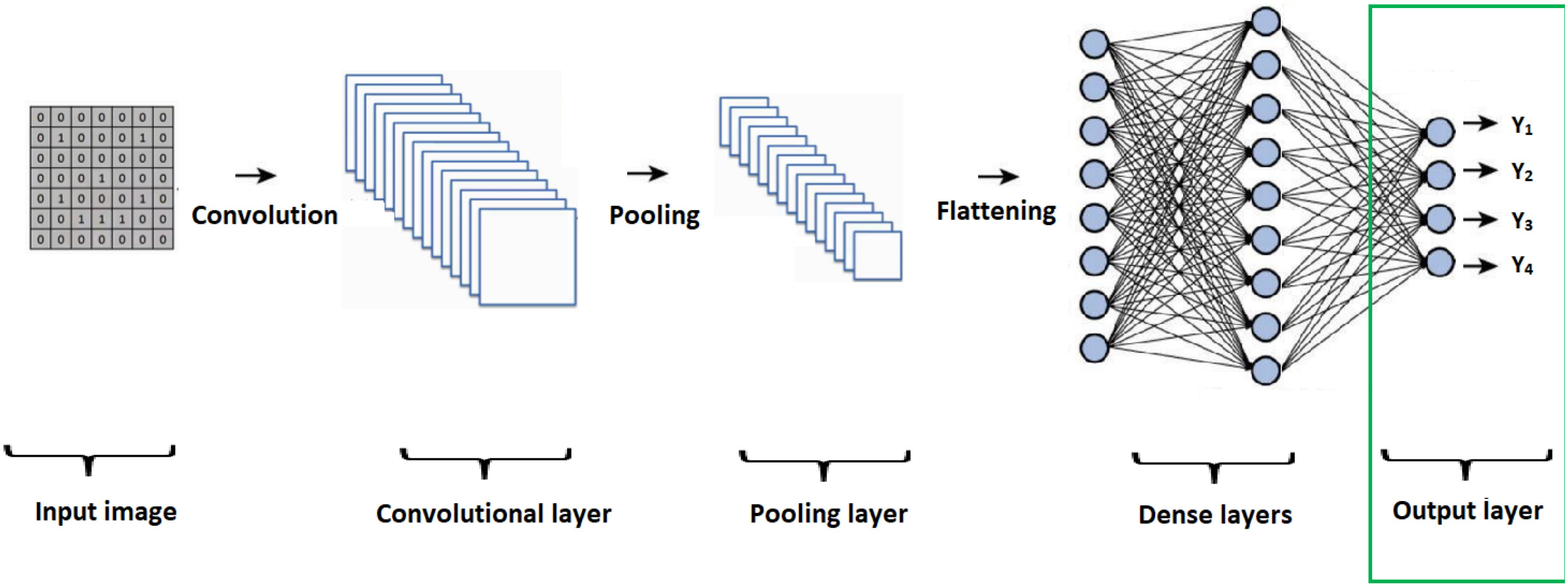
Fully connected layer

- Flattening

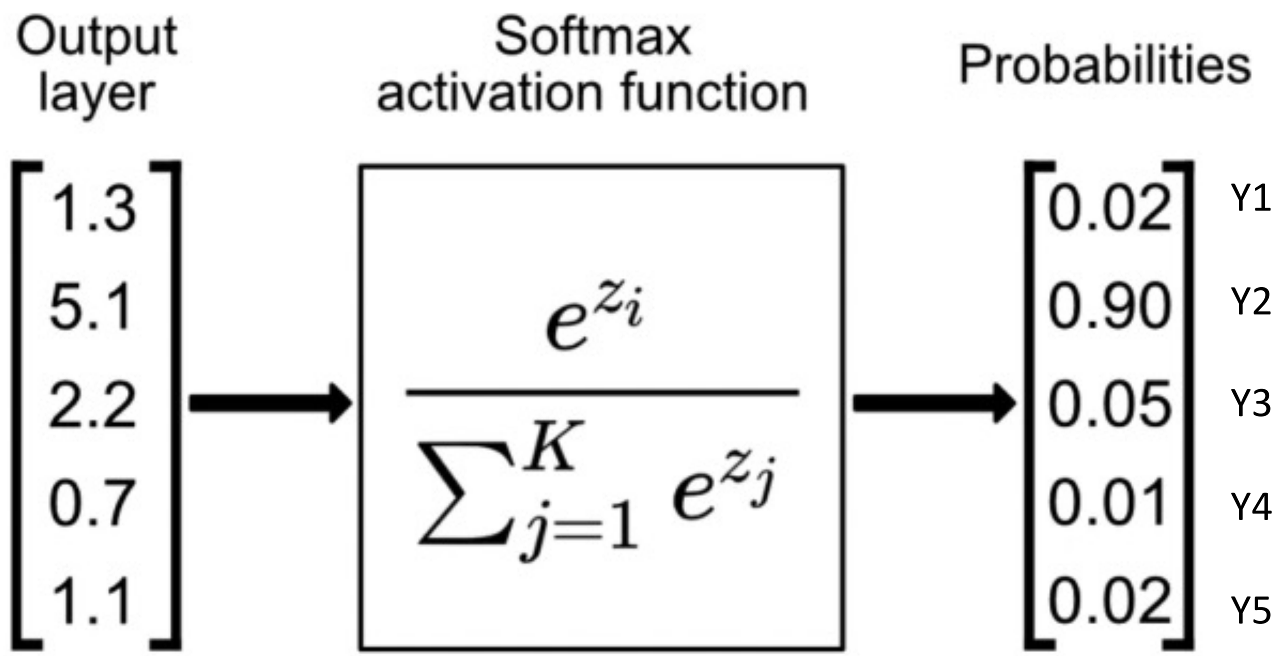


https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#layer

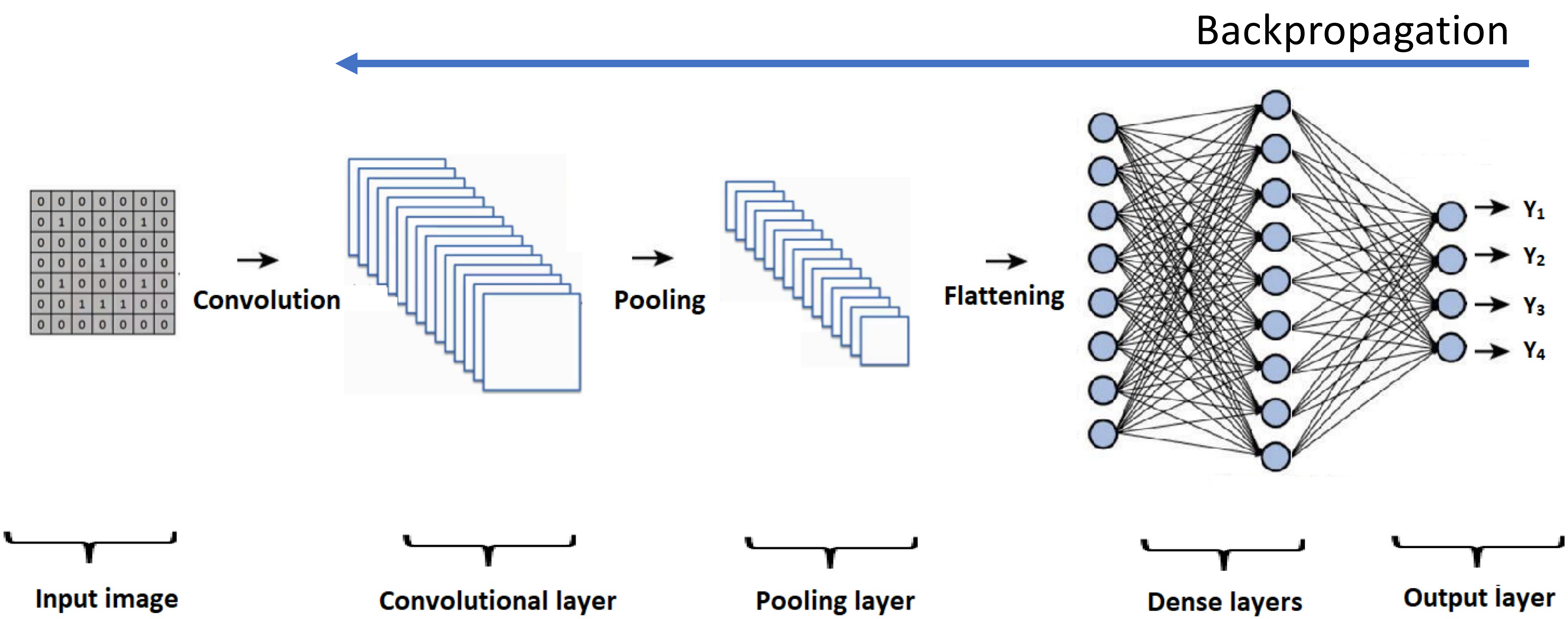
Convolutional Neural Network Architecture



https://www.mdpi.com/2076-3417/10/4/1245/htm


Output layer
Softmax activation function
Probabilities
1.3
5.1
2.2
0.7
1.1
$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
0.02
0.90
0.05
0.01
0.02
Y1
Y2
Y3
Y4
Y5

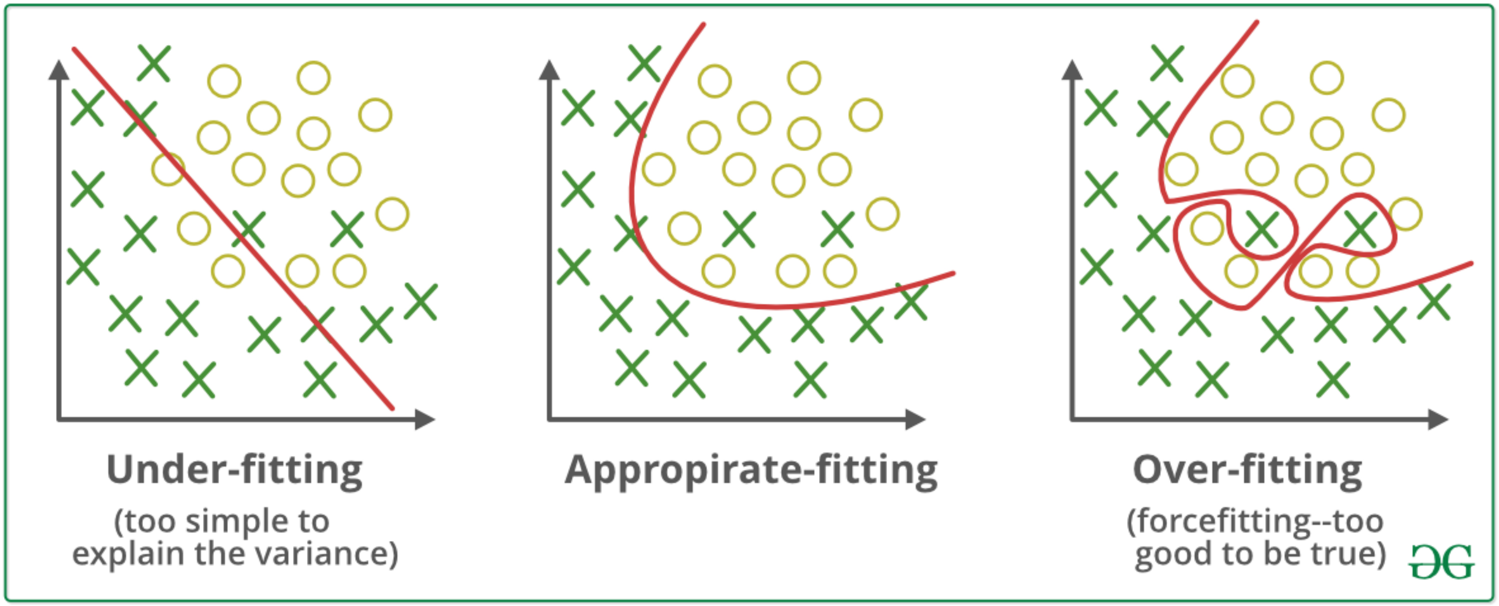
Training the network



https://www.mdpi.com/2076-3417/10/4/1245/htm

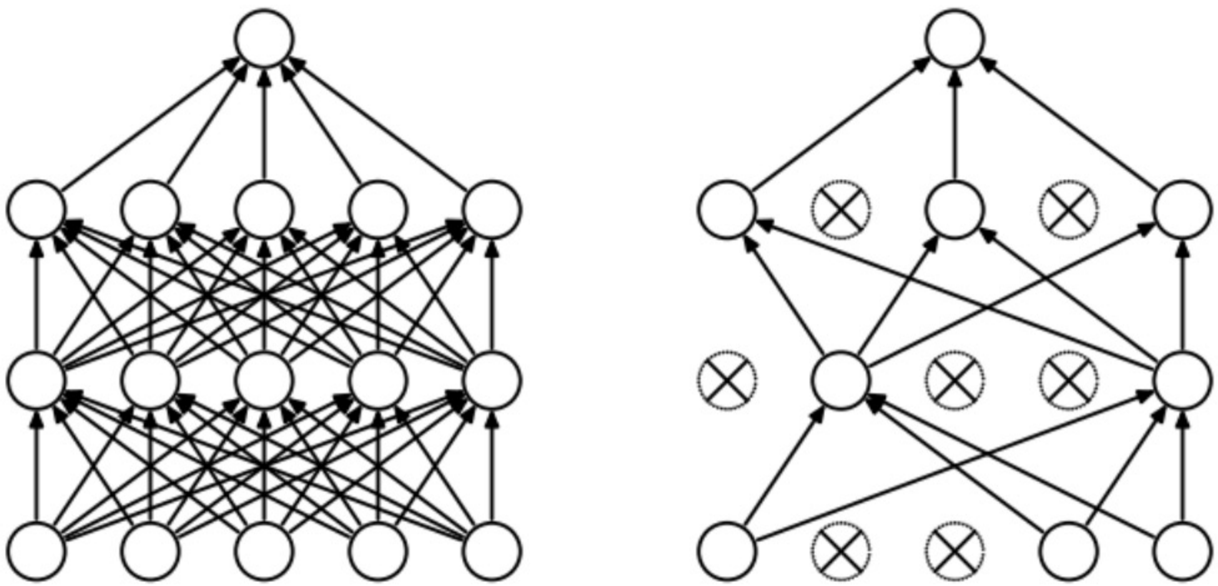
Dropout

- Overfitting



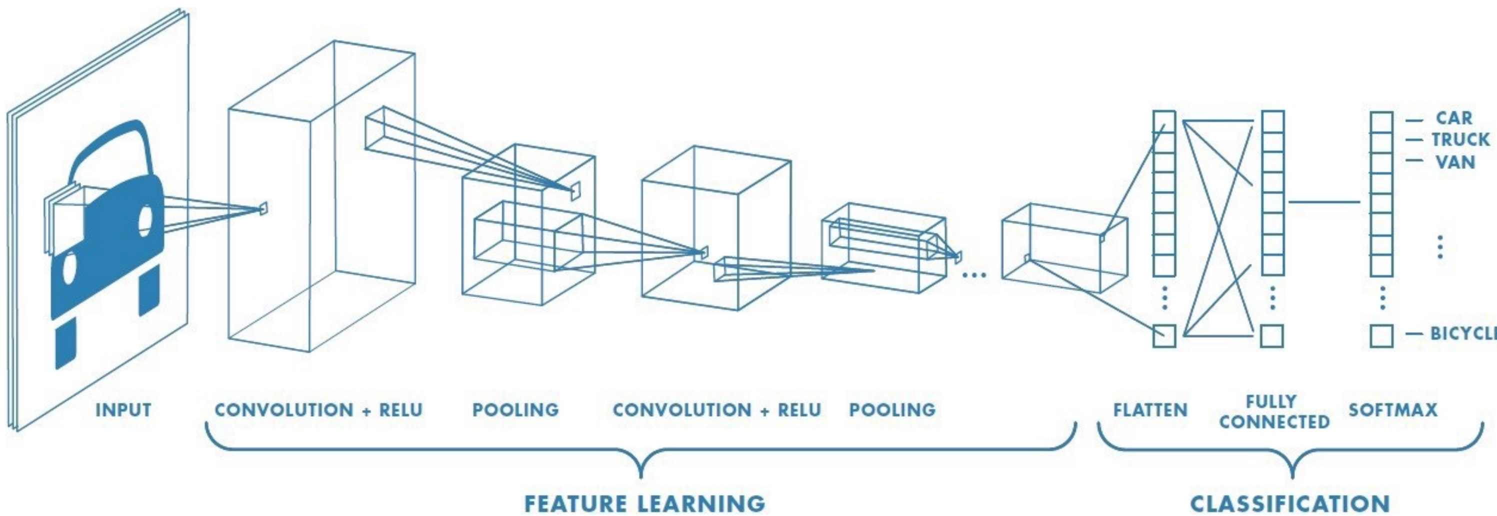
https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

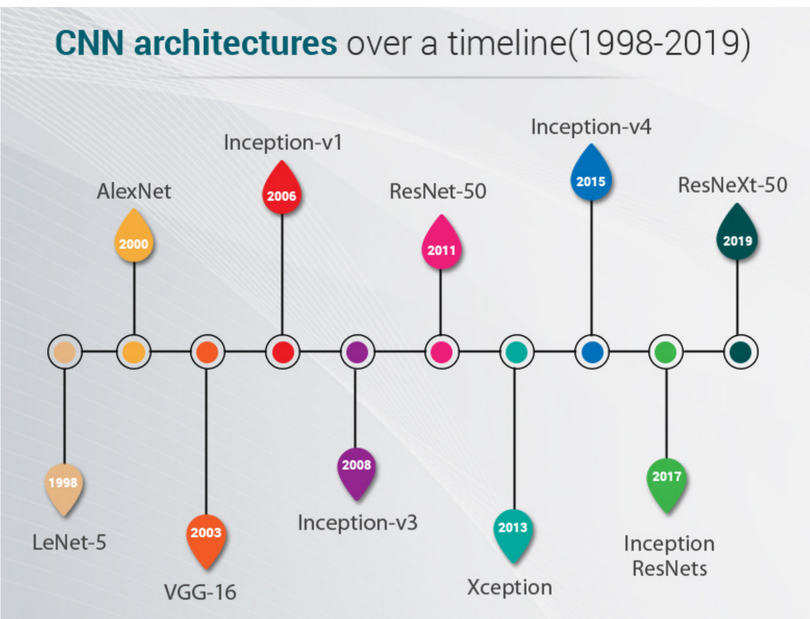
Dropout

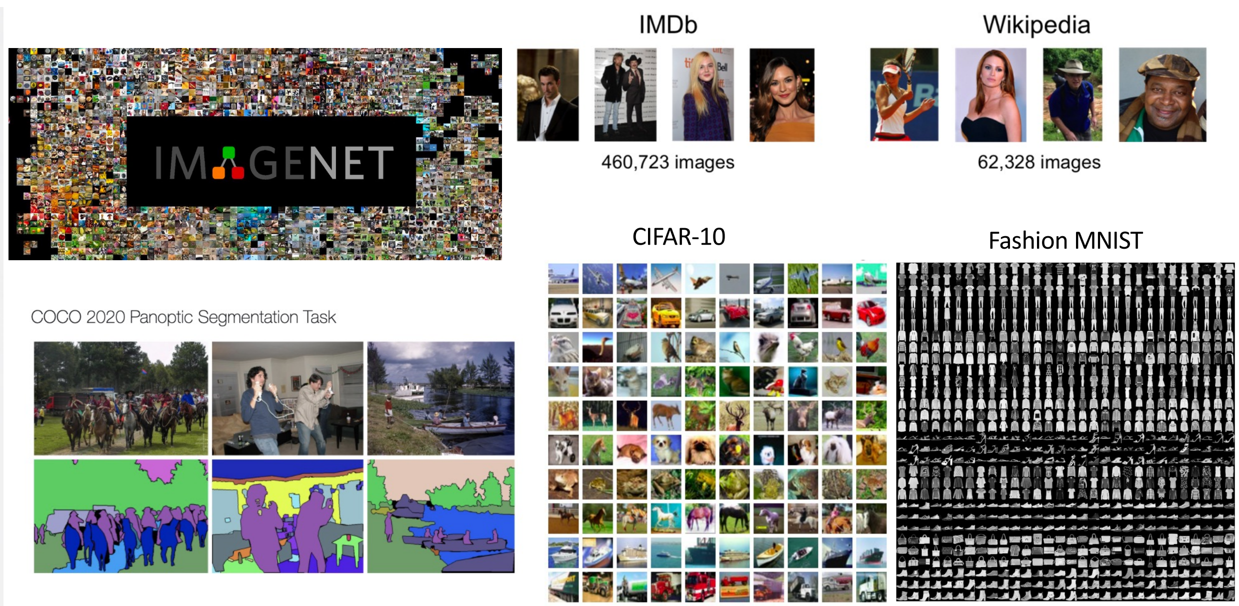


(a) Standard Neural Net (b) After applying dropout.

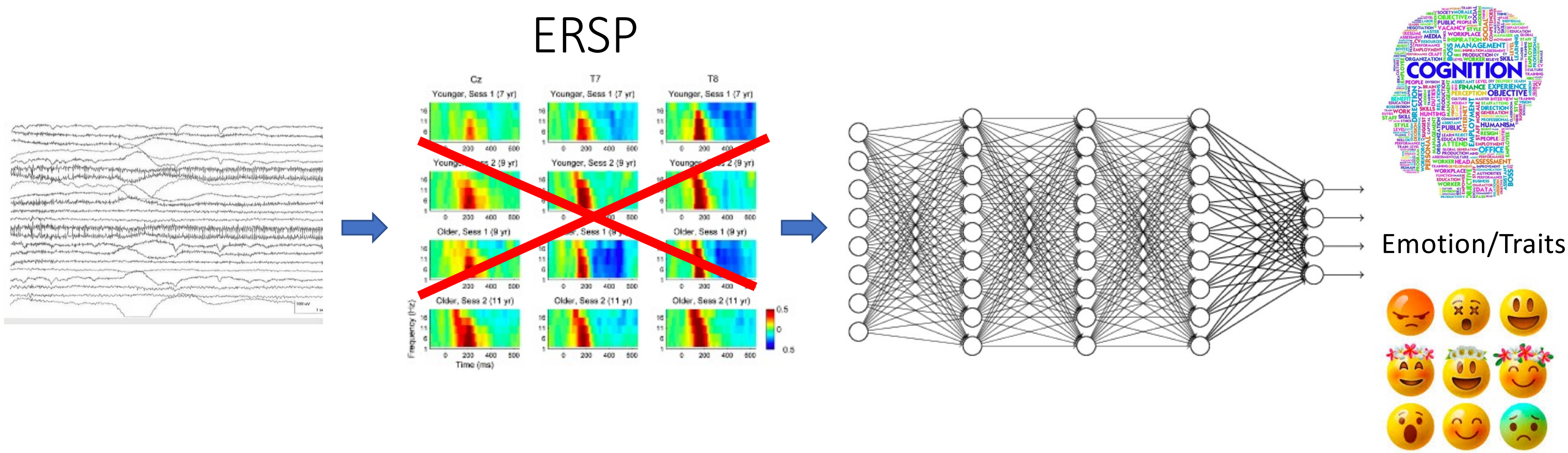
https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5


CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION


CNN architectures over a timeline(1998-2019)
AlexNet
2000
Inception-v1
2006
ResNet-50
2011
Inception-v4
2015
ResNeXt-50
2019
1998
LeNet-5
2003
VGG-16
2008
Inception-v3
2013
Xception
2017
Inception ResNets


IMAGENET
IMDb
460,723 images
Wikipedia
62,328 images
COCO 2020 Panoptic Segmentation Task
CIFAR-10
Fashion MNIST

Applying to EEG



Raw EEG vs. Frequency-domain

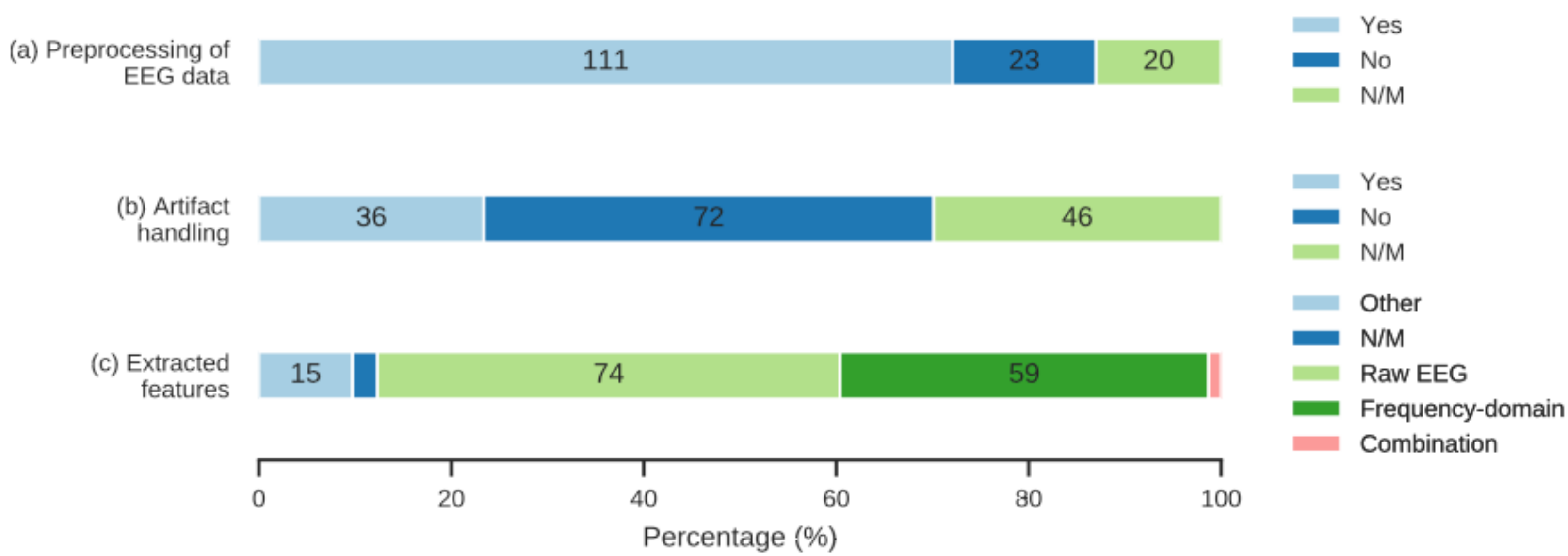


Figure 9. EEG processing choices. (a) Number of studies that used preprocessing steps, such as filtering, (b) number of studies that included, rejected or corrected artifacts in their data and (c) types of features that were used as input to the proposed models.

Roy Y. et al (2019)

Raw EEG vs. Frequency-domain

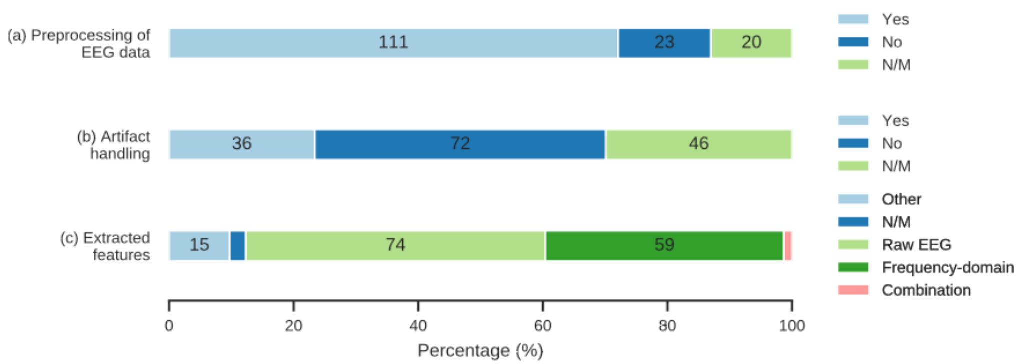
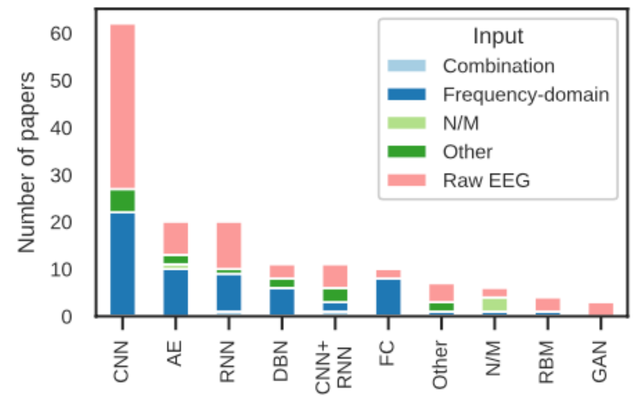


Figure 9. EEG processing choices. (a) Number of studies that used preprocessing steps, such as filtering, (b) number of studies that included, rejected or corrected artifacts in their data and (c) types of features that were used as input to the proposed models.

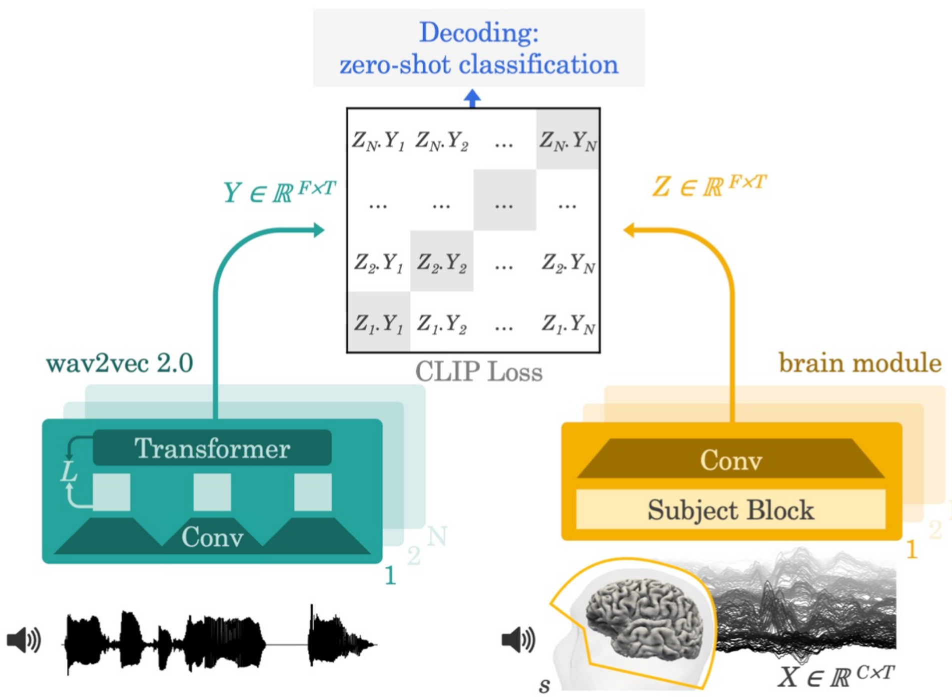


Distribution of input type according to the architecture category.

Roy Y. et al (2019)

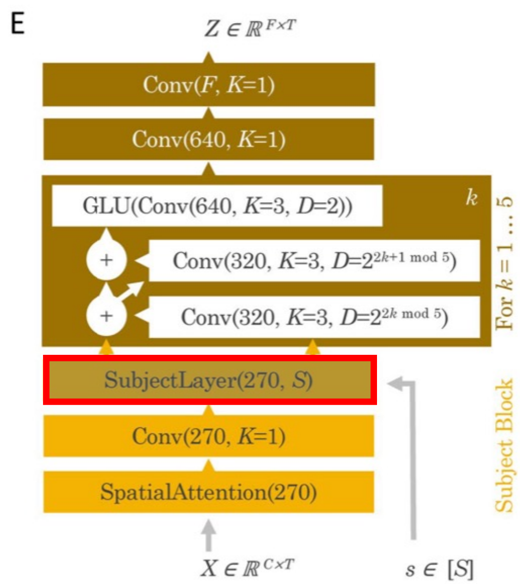
Decoding speech from non-invasive brain recordings

Alexandre Défossez[1,*], Charlotte Caucheteux[1,2], Jérémy Rapin[1], Ori Kabeli[1], and Jean-Rémi King[1,*]

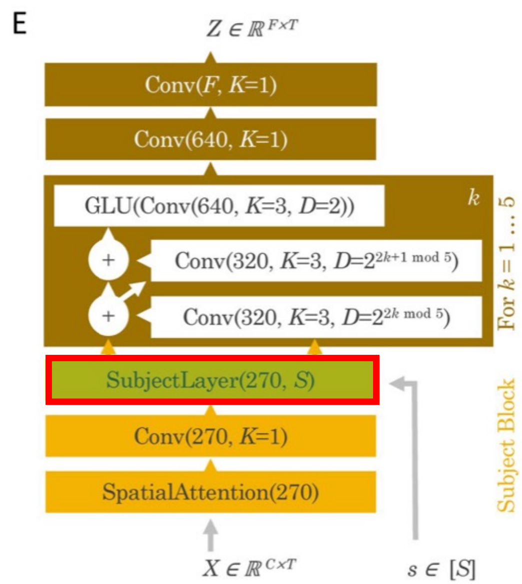


Subject attention

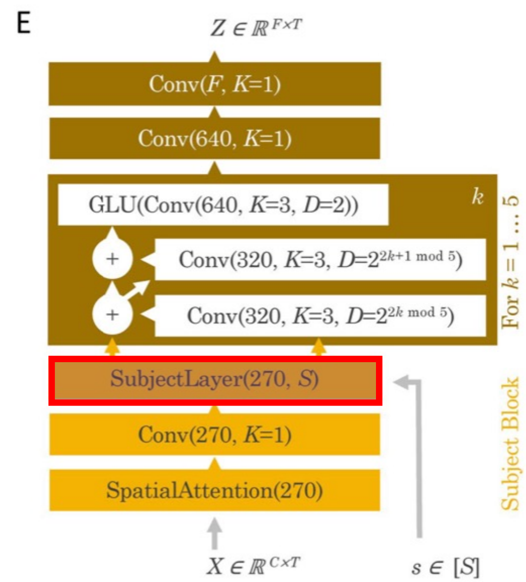
Subject 1



Subject 2

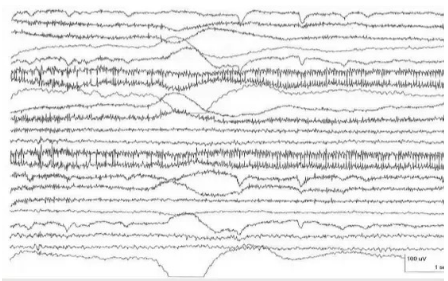


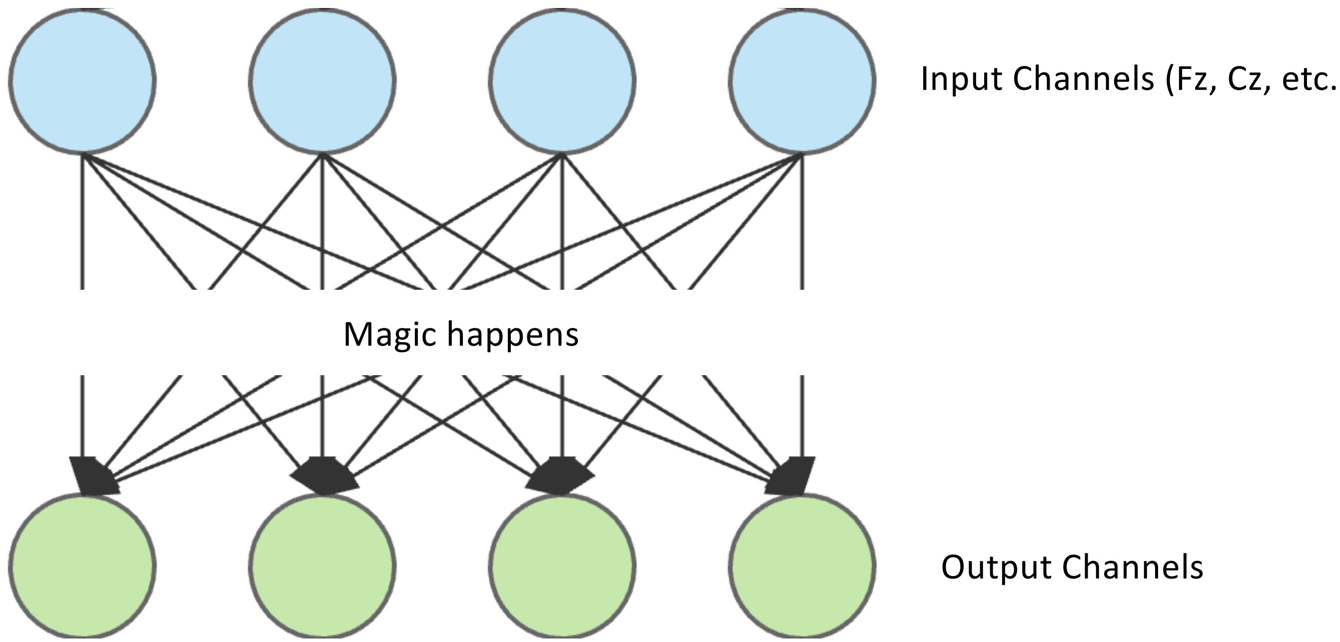
Subject 3



Spatial attention

Which electrodes are neighbors?





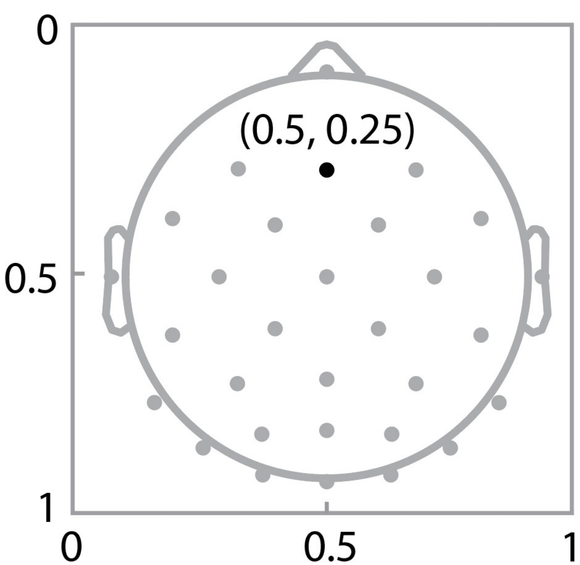
Spatial attention

Learned matrix

Electrode j position

2D Fourier

$$a_j(x,\ y)\ = \boxed{\sum_{k=1}^{K}\ \sum_{l=1}^{K}} Re(z_j^{(k,\ l)})cos(2\pi(kx\ +\ ly))\ +\ \ Im(z_j^{(k,\ l)})sin(2\pi(kx\ +\ ly))$$



Input channels close to each other are weighted in a similar manner by the z_j matrix (especially for low values of k and l at low spatial frequencies)

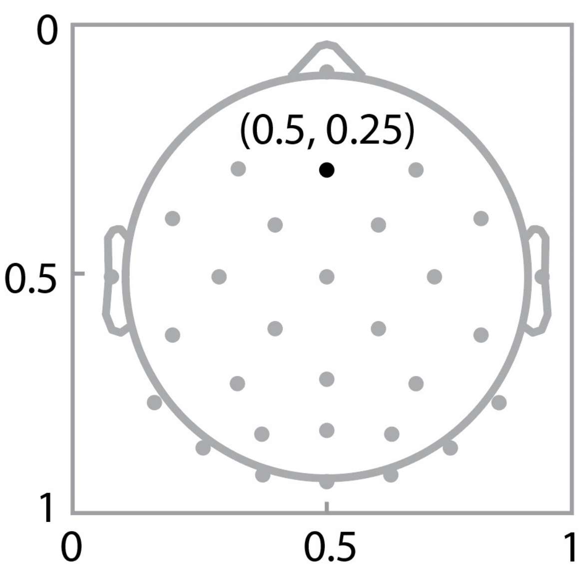
Spatial attention

Learned matrix

Electrode j position

2D Fourier

$$a_j(x, y) = \boxed{\sum_{k=1}^{K} \sum_{l=1}^{K}} Re(z_j^{(k,l)}) cos(2\pi(kx + ly)) + Im(z_j^{(k,l)}) sin(2\pi(kx + ly))$$



Input channels close to each other are weighted in a similar manner by the z_j matrix (especially for low values of k and l at low spatial frequencies)

Output channel j

$$SA(X)^{(j)} = \underbrace{\frac{1}{\sum_{i=1}^{C} e^{a_j(x_i, y_i)}} \left(\sum_{i=1}^{C} e^{a_j(x_i, y_i)} X^{(i)} \right)}_{\text{Softmax}}$$

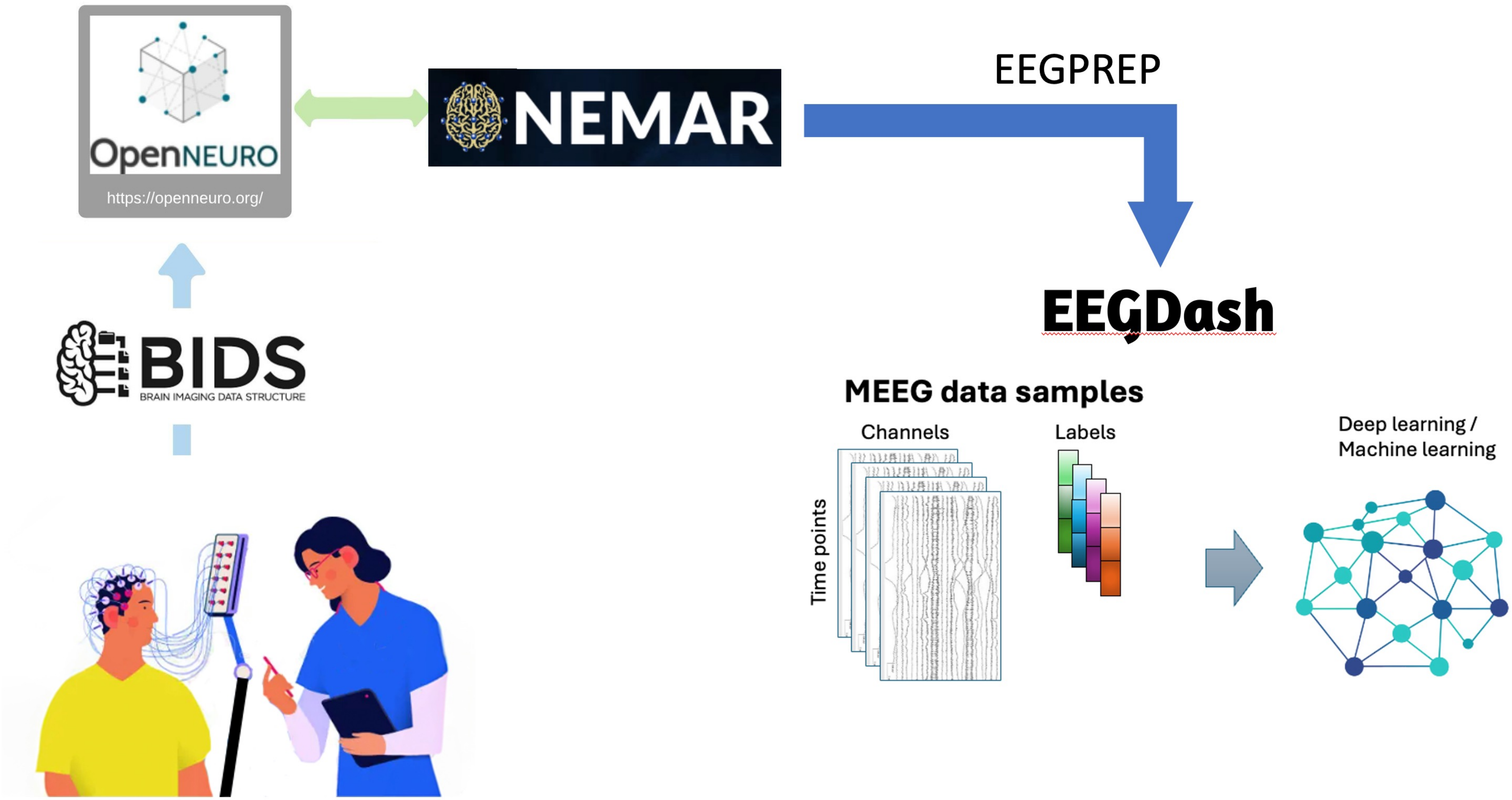
Deep learning applied to EEG data with different montages using spatial attention

Dung Truong*
SCCN, INC, UCSD, La Jolla CA, USA
dutruong@ucsd.edu
https://orcid.org/0000-0003-4540-3551

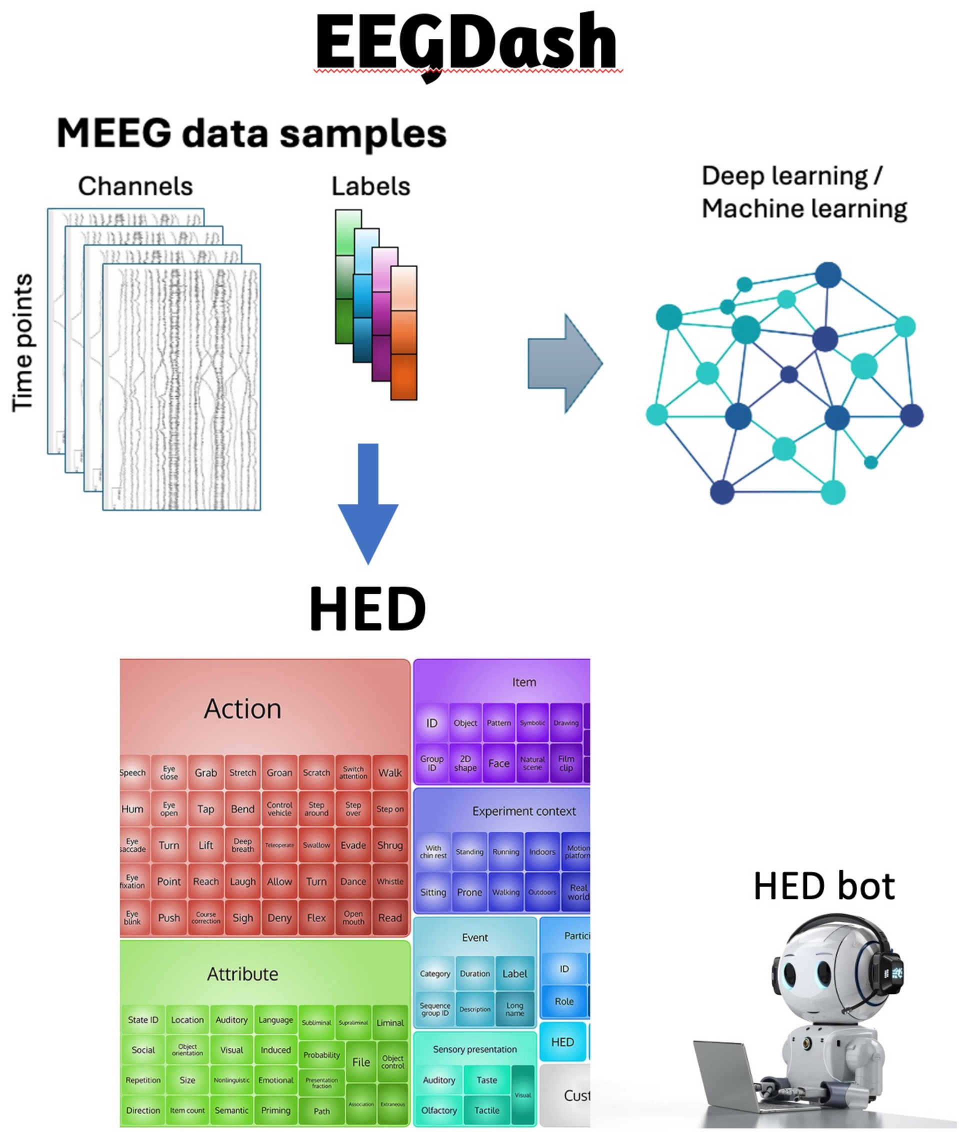
Muhammad Abdullah Khalid*
SCCN, INC, UCSD, La Jolla CA, USA
mkhalid.bee18seecs@seecs.edu.pk

Arnaud Delorme
SCCN, INC, UCSD, La Jolla CA, USA
CerCo CNRS, Paul Sabatier University, Toulouse, France
arnodelorme@gmail.com
https://orcid.org/0000-0002-0799-3557

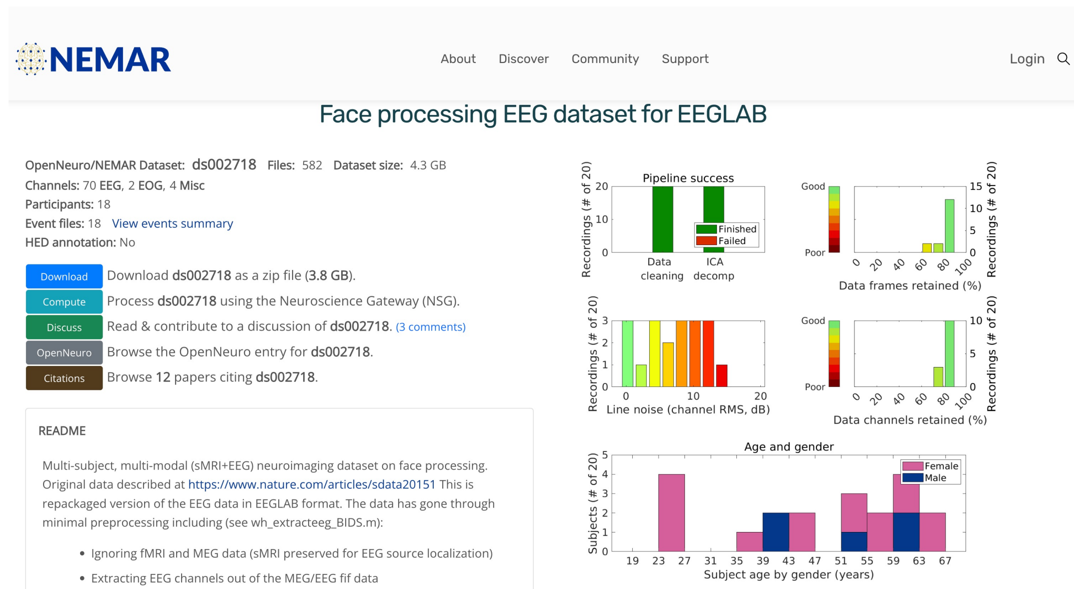
Model→ Attention↓	128-channel model	23-channel model
No spatial attention	80.4 (0.8)	78.0 (1.8)
Spatial attention	83.7 (1.5)†	80.3 (1.4)†

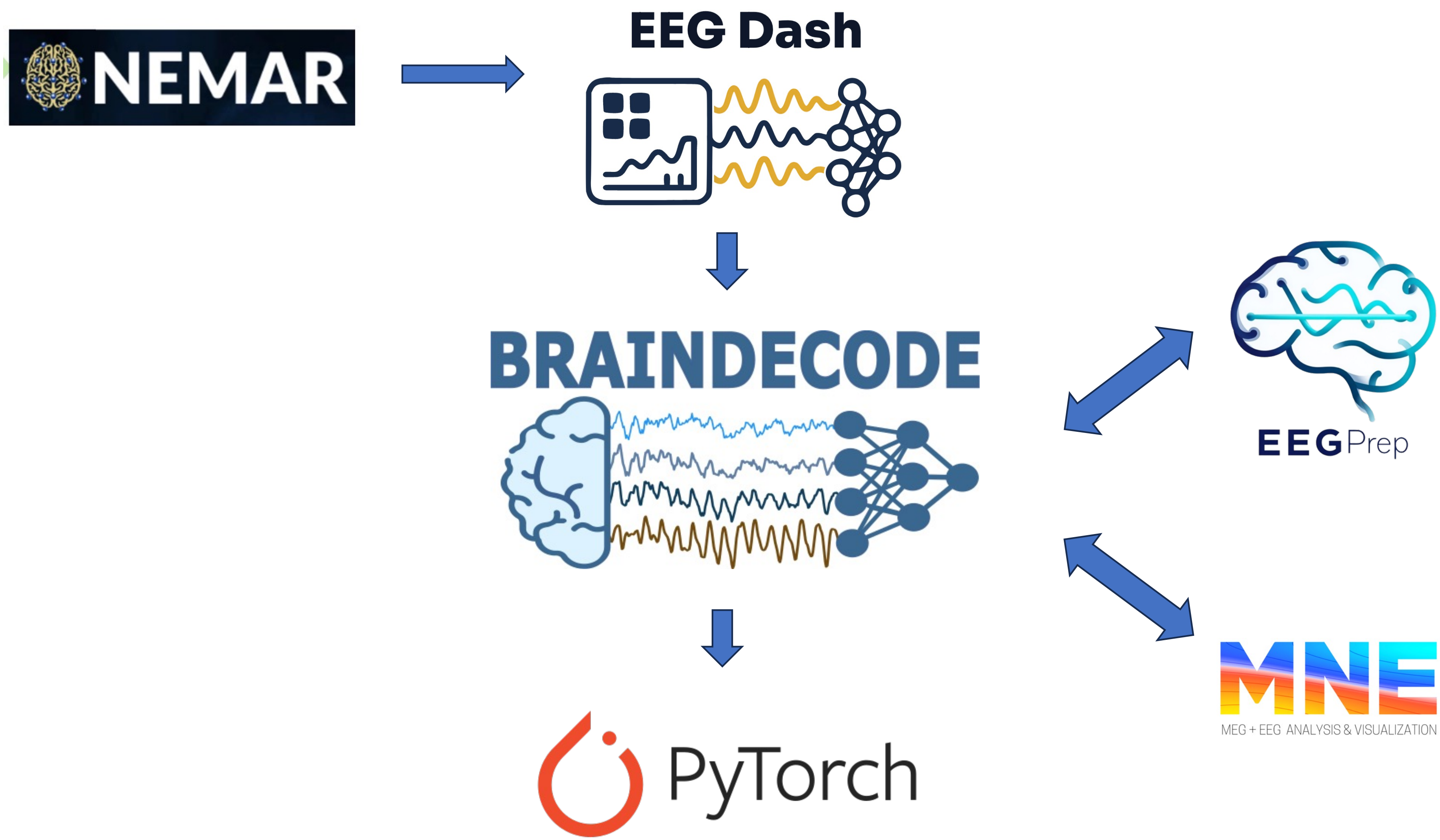


OpenNEURO
https://openneuro.org/
NEMAR
EEGPREP
EEGDash
BIDS
BRAIN IMAGING DATA STRUCTURE
MEEG data samples
Channels
Labels
Time points
Deep learning /
Machine learning


EEGDash
MEEG data samples
Channels
Labels
Time points
Deep learning /
Machine learning
HED
Action
Attribute
Item
Experiment context
Event
Sensory presentation
HED bot

DatasetID	Participants	Files	Sessions	Population	Channels	Is 10-20?	Modality	Size
ds001785	18	242	1	Healthy	63	10-20	Tactile	0.027 TB
ds001787	24	141	3	Healthy	64	10-20	Auditory	5.7 GB
ds001810	47	1678	6	Healthy	64	10-20	Visual	0.049 TB
ds001849	20	363	1	Healthy	30	10-20	Multisensory	0.048 TB
ds001971	20	1917	1	Healthy	108	10-20	Auditory	0.034 TB
ds002034	14	882	3	Healthy	64	10-20	Visual	0.011 TB
ds002094	20	282	1	0	30	10-20	Resting State	0.042 TB
ds002158	20	949	1	Healthy	63	10-20	Visual	0.082 TB
ds002181	20	949	1	Healthy	63	10-20	Visual	0.163 GB
ds002218	18	133	1	Healthy	32	10-20	Multisensory	0.002 TB
ds002336	10	325	1	Healthy	63	other	Visual	0.018 TB
ds002338	17	484	1	Healthy	63	other	Visual	0.026 TB
ds002578	2	22	1	Healthy	256	10-20	Visual	0.001 TB
ds002680	14	4977	2	Healthy	0	10-20	Visual	0.01 TB
ds002691	20	146	1	Healthy	32	other	Visual	0.001 TB
ds002718	18	582	1	Healthy	70	other	Visual	0.005 TB
ds002720	18	828	1	Healthy	19	10-20	Auditory	0.003 TB
ds002721	31	929	1	Healthy	19	10-20	Auditory	0.004 TB
ds002722	19	582	1	Healthy	32	10-20	Auditory	0.007 TB
ds002723	8	204	1	Healthy	32	10-20	Auditory	0.003 TB
ds002724	10	700	3	Healthy	32	10-20	Auditory	0.009 TB
ds002725	21	1098	1	Healthy	30	10-20	Auditory	0.016 TB
ds002778	31	328	3	Parkinson's	40	10-20	Resting State	0.001 TB
ds002814	21	1139	2	Healthy	68	10-20	Visual	0.03 TB
ds002833	20	623	4	?	257	10-20	Auditory	0.043 TB
ds002893	49	370	1	Healthy	33	10-20	Multisensory	7.7 GB
ds003004	34	277	1	Healthy	224	10-20	Auditory	0.039 TB
ds003039	19	157	1	Healthy	64	10-20	Motor	0.008 TB
ds003061	13	282	1	?	64	10-20	Auditory	0.002 TB




NEMAR
EEG Dash
BRAINDECODE
EEGPrep
MNE
MEG + EEG ANALYSIS & VISUALIZATION
PyTorch

Data Retrieval Using EEGDash

First we find one resting state dataset. This dataset contains both eyes open and eyes closed data.

In [23]:
```
from eegdash import EEGDash
from eegdash import EEGDashDataset

ds_eoec = EEGDashDataset({'dataset': 'ds005514', 'task': 'RestingState', 'subject': 'NDARDB033FW5'})
```

...

```
Epoch 0, Train accuracy: 0.52, Test accuracy: 0.50
Epoch 1, Train accuracy: 0.82, Test accuracy: 0.50
Epoch 2, Train accuracy: 0.91, Test accuracy: 0.64
Epoch 3, Train accuracy: 0.93, Test accuracy: 0.64
Epoch 4, Train accuracy: 0.88, Test accuracy: 0.64
Epoch 5, Train accuracy: 0.96, Test accuracy: 0.71
```

https://eegdash.org

EEGPrep

MATLAB

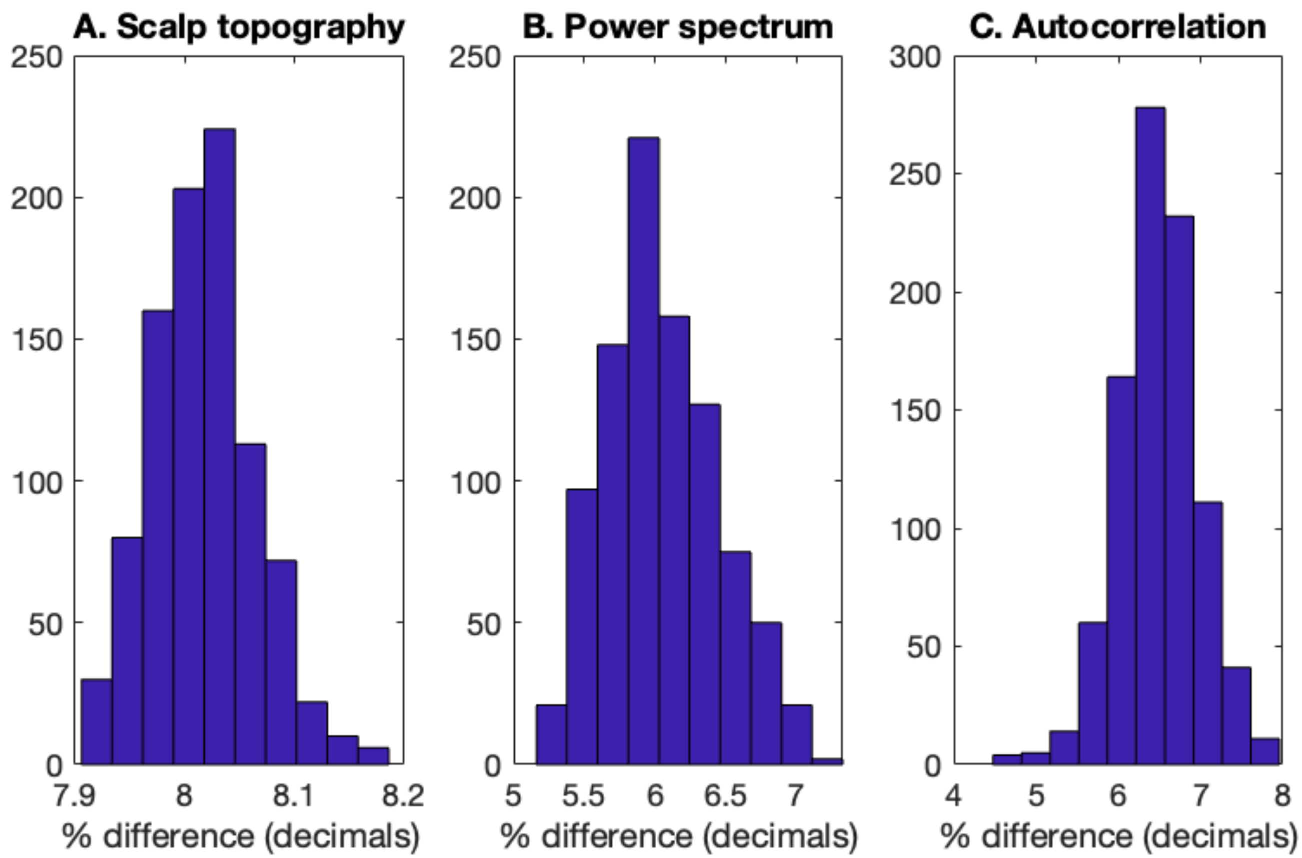
```
eeglab

fname = "data/eeglab_data_with_ica_tmp.set";
EEG = pop_loadset(fname);
EEG = clean_artifacts(EEG, 'FlatlineCriterion',5,'ChannelCriterion',0.87, 'LineNoiseCriterion',4, ...
                 'Highpass',[0.25, 0.75], 'BurstCriterion',20, 'WindowCriterion',0.25, ...
                 'BurstRejection',true, 'WindowCriterionTolerances',[-inf 7]);
EEG = eeg_picard(EEG);
EEG = iclabel(EEG);
pop_saveset(EEG, [ fname(1:end-4) '_out.set']);
```

Python

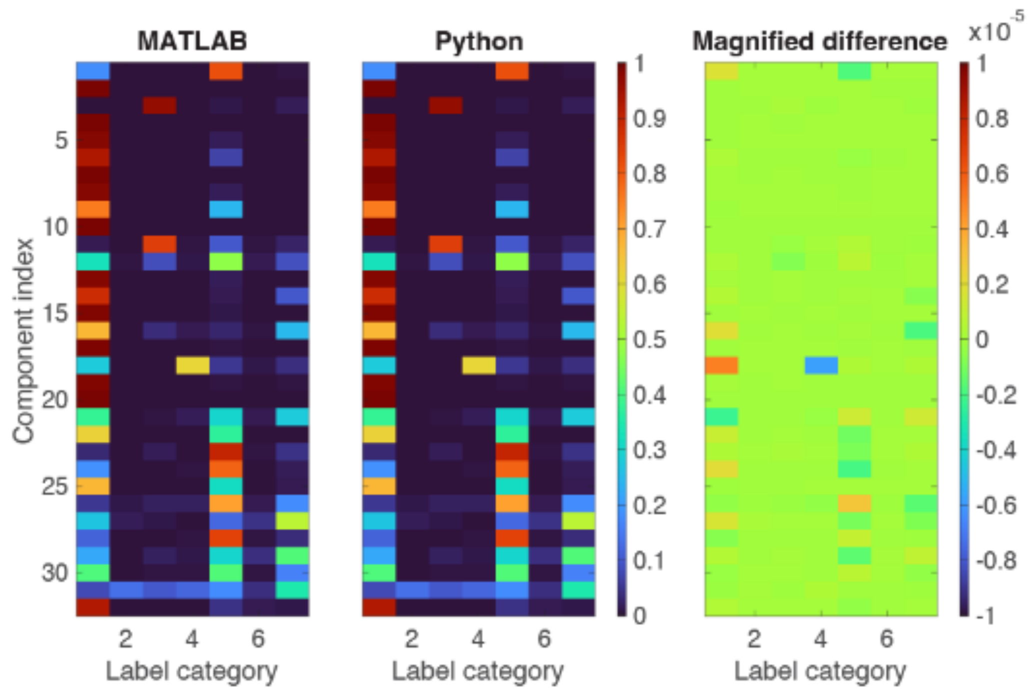
```
from eegprep import iclabel, pop_loadset, pop_saveset, pop_eegfiltnew, clean_artifacts, eeg_picard

fname = "data/eeglab_data_with_ica_tmp.set"
EEG = pop_loadset(fname)
EEG = clean_artifacts(EEG, FlatlineCriterion=5,ChannelCriterion=0.87, LineNoiseCriterion=4, \
                      Highpass=[0.25, 0.75], BurstCriterion= 20, WindowCriterion=0.25, \
                      BurstRejection=True, WindowCriterionTolerances=[float('-inf'), 7])
EEG = eeg_picard(EEG)
EEG = iclabel(EEG)
pop_saveset(EEG, fname.replace('.set', '_out.set'))
```

Stage	StageName	MaxErr(all)
1	BIDS Import	0
2	ChannelSelection	0
3	Resampling	0
4	FlatlineRemoval	0
5	HighpassFilter	0.001842876518
6	BadChannelRemoval	9.54E-07
7	BurstRemoval (ASR)	3.81E-06
8	BadWindowRemoval	3.81E-06
9	PICARD	3.81E-06
10	ICLabel	3.81E-06
11	Reinterpolate	0.0001857959412
12	Epoching	7.13E-05
13	BaselineRemoval	6.78E-05
14	CommonAverageRef	6.50E-05

Percentage difference MATLAB vs Python

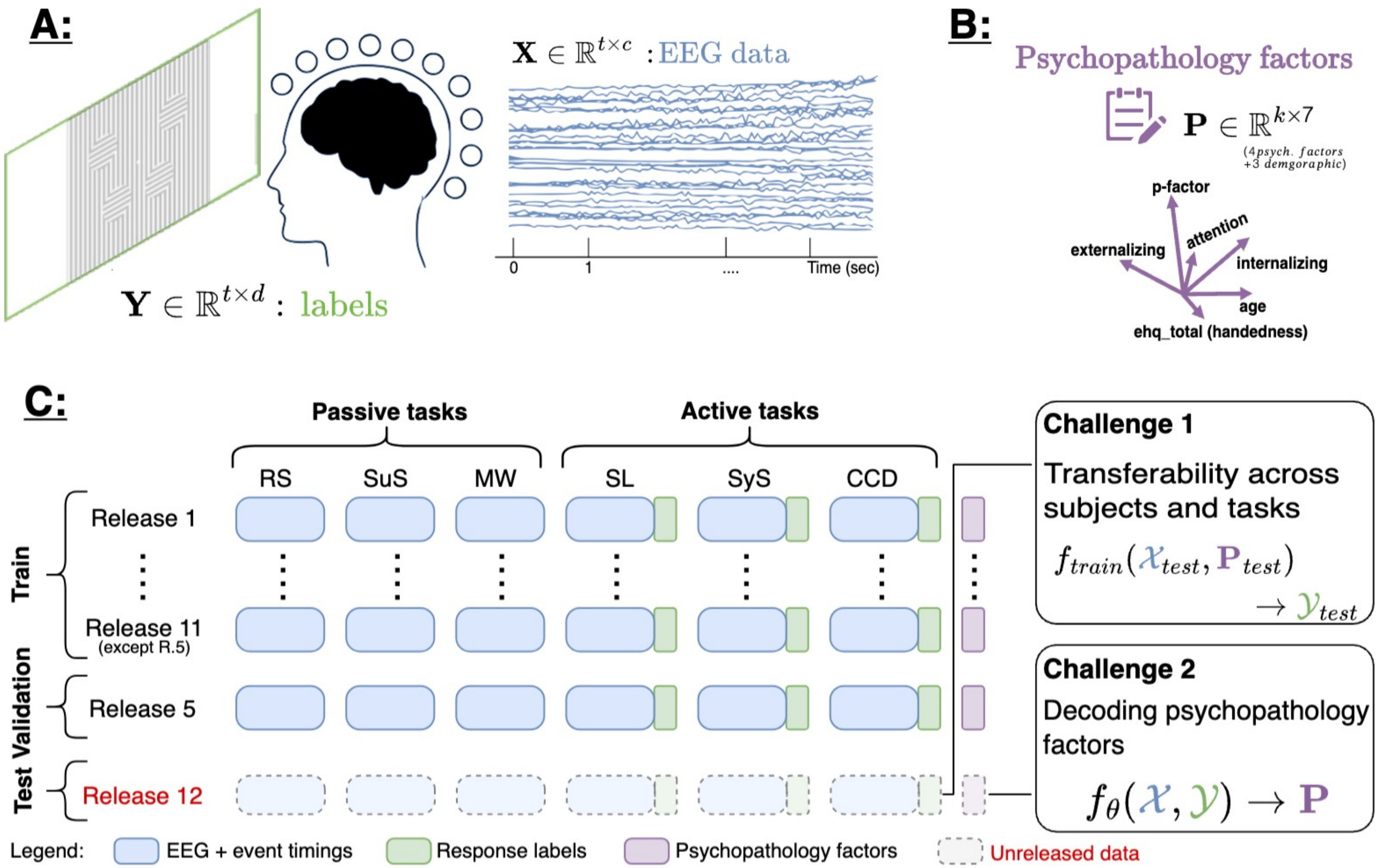


Difference in classification: ICLabel



EEG Foundation Challenge: From Cross-Task to Cross-Subject EEG Decoding

Bruno Aristimunha[*,†,♠] **Dung Truong**[‡] **Pierre Guetschel**[§] **Seyed Yahya Shirazi**[‡]
Isabelle Guyon[¶] **Alexandre R. Franco**[||] **Michael P. Milham**[||] **Aviv Dotan**[♣]
Scott Makeig[‡] **Alexandre Gramfort**[✠] **Jean-Remi King**[✠] **Marie-Constance Corsi**[†]
Pedro A. Valdés-Sosa[♡] **Amit Majumdar**[‡] **Alan Evans**[◇] **Terrence J Sejnowski**[‡]
Oren Shriki[♣] **Sylvain Chevallier**[*]
Arnaud Delorme[‡,◇]
neurips2025-eeg-competition@googlegroups.com



A:
$\mathbf{X} \in \mathbb{R}^{t \times c}$: EEG data
0
1
....
Time (sec)
$\mathbf{Y} \in \mathbb{R}^{t \times d}$: labels
B:
Psychopathology factors
$\mathbf{P} \in \mathbb{R}^{k \times 7}$
(4psych. factors +3 demgoraphic)
p-factor
attention
externalizing
internalizing
age
ehq_total (handedness)
C:
Passive tasks
Active tasks
RS
SuS
MW
SL
SyS
CCD
Train
Release 1
Release 11
(except R.5)
Validation
Release 5
Test
Release 12
Challenge 1
Transferability across subjects and tasks
$f_{train}(\mathcal{X}_{test}, \mathbf{P}_{test}) \rightarrow \mathcal{Y}_{test}$
Challenge 2
Decoding psychopathology factors
$f_{\theta}(\mathcal{X}, \mathcal{Y}) \rightarrow \mathbf{P}$
Legend:
EEG + event timings
Response labels
Psychopathology factors
Unreleased data

Thank you