DEMYSTIFYING DEEP LEARNING: TUTORIAL SERIES

CHAPTER 5: CNN FOR IMAGE CLASSIFICATION

Vaibhav Verdhan

May 29 2020

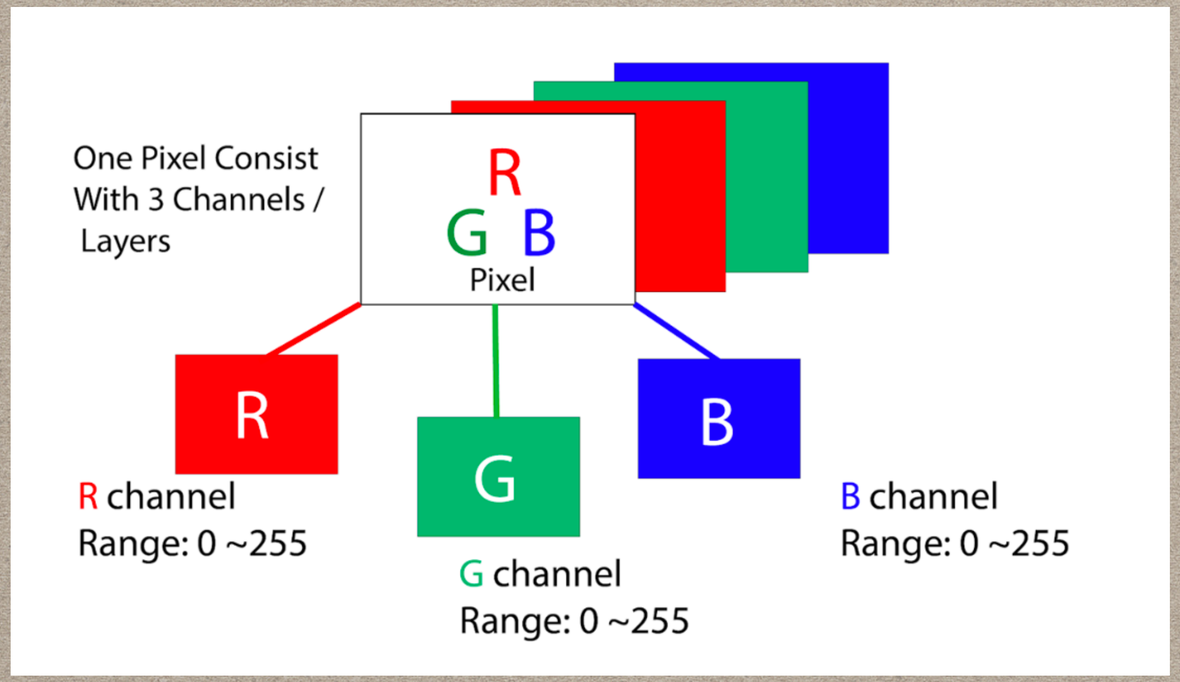
AGENDA FOR THE SERIES

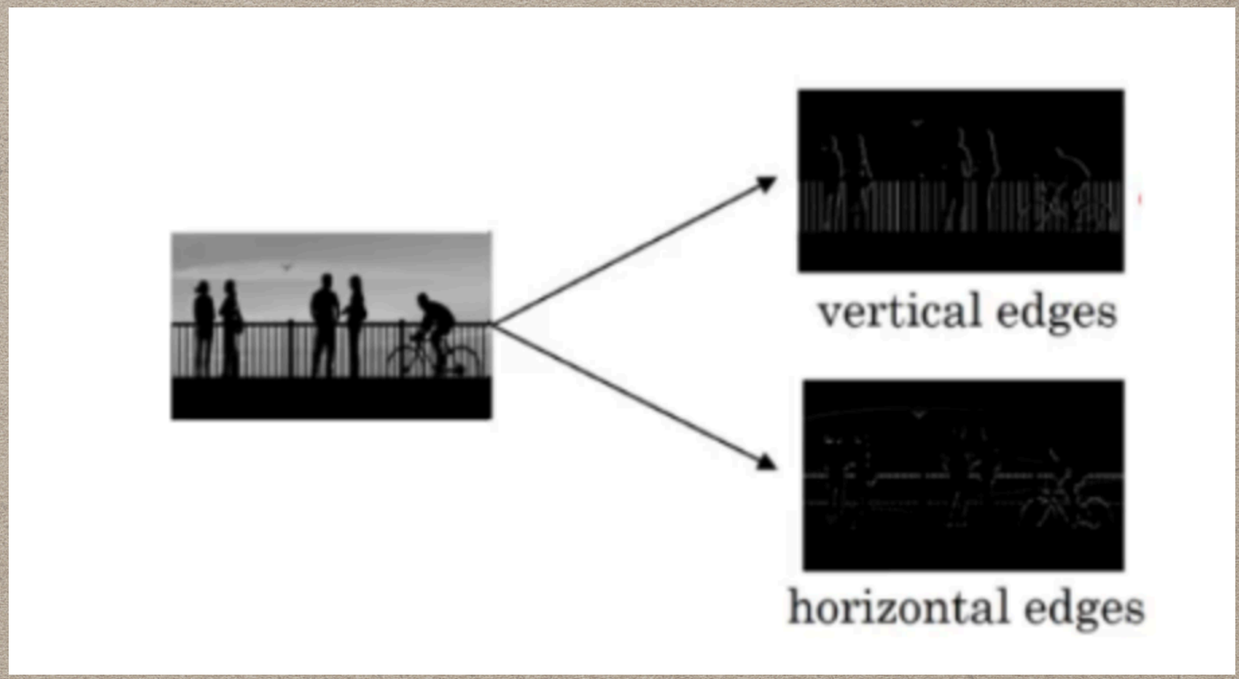
Session	Topic
Session 1	Introduction to Deep Learning
Session 2	Building blocks of Neural Network - 1
Session 3	Building blocks of Neural Network - 2
Session 4	Convolutional Neural Network
Session 5	CNN for Image Classification
Session 6	CNN for Object Detection (June 12)
Session 7	Architectures like AlexNet, Inception etc.
Session 8	Recurrent Neural Network
Session 9	NLP Applications of RNN

AGENDA FOR SESSION 5

- *Image processing*
- *Python Lab*
- *Convolutional Neural Network Refresh*
- *Image Classification concepts*
- *Python Lab*

IMAGE PROCESSING





EDGE DETECTION

-1	0	1
-2	0	2
-1	0	1

Vertical

1	2	1
0	0	0
-1	-2	-1

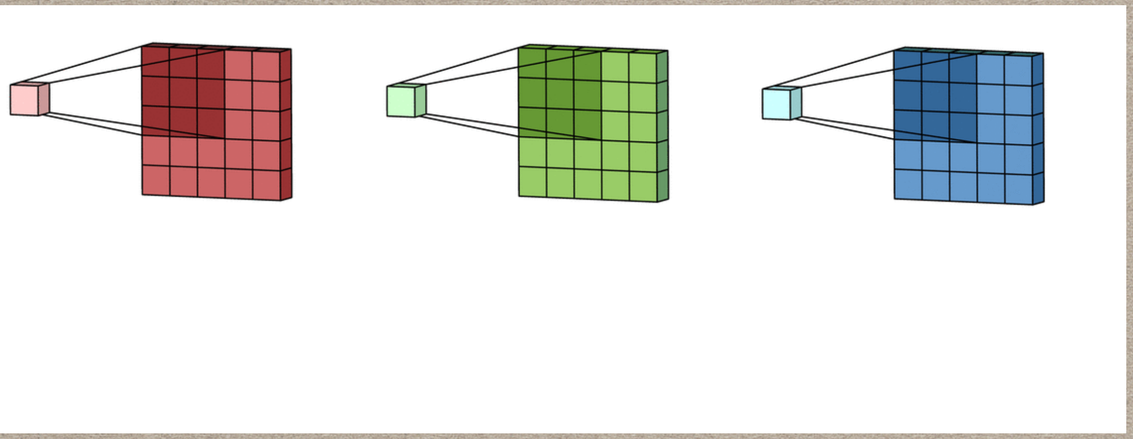
Horizontal

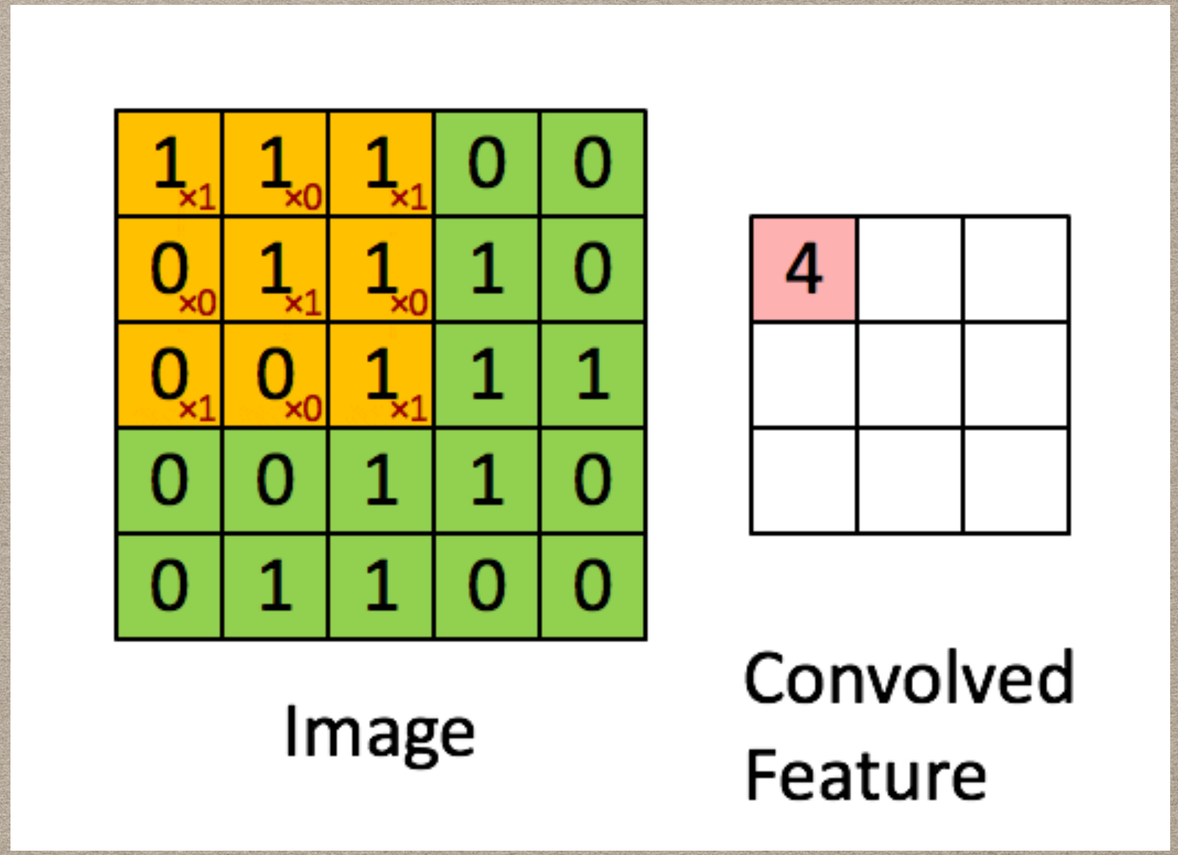
IMAGE AUGMENTATION

- Horizontal flips
- Rotation
- Crop/scale
- Color jitter
- Other creative techniques
- Random mix/combinations of :
  - translation (what about a pure ConvNet?)
  - Rotation
  - Stretching
  - Shearing
  - lens distortions, … (go crazy)

TIME TO HIT THE LAB

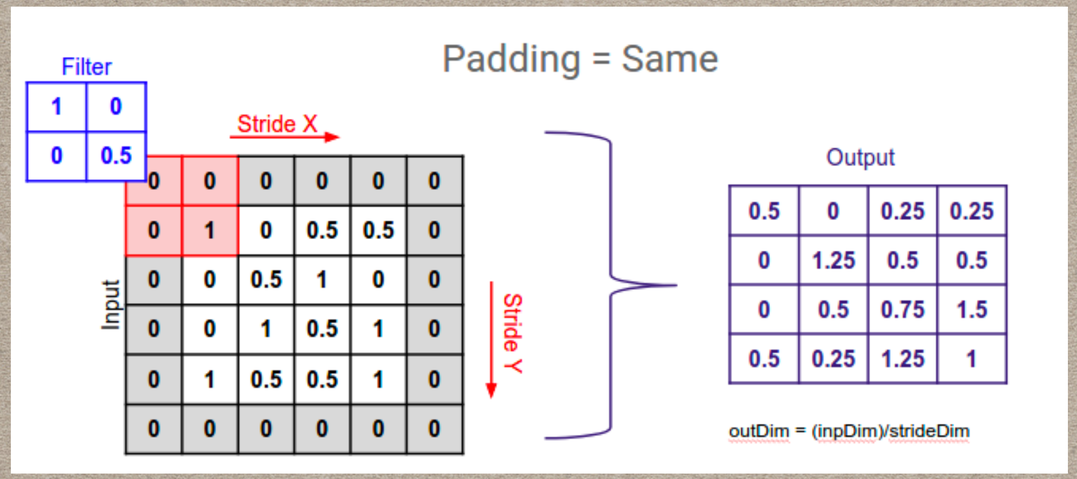
THE CONVOLUTION OPERATION

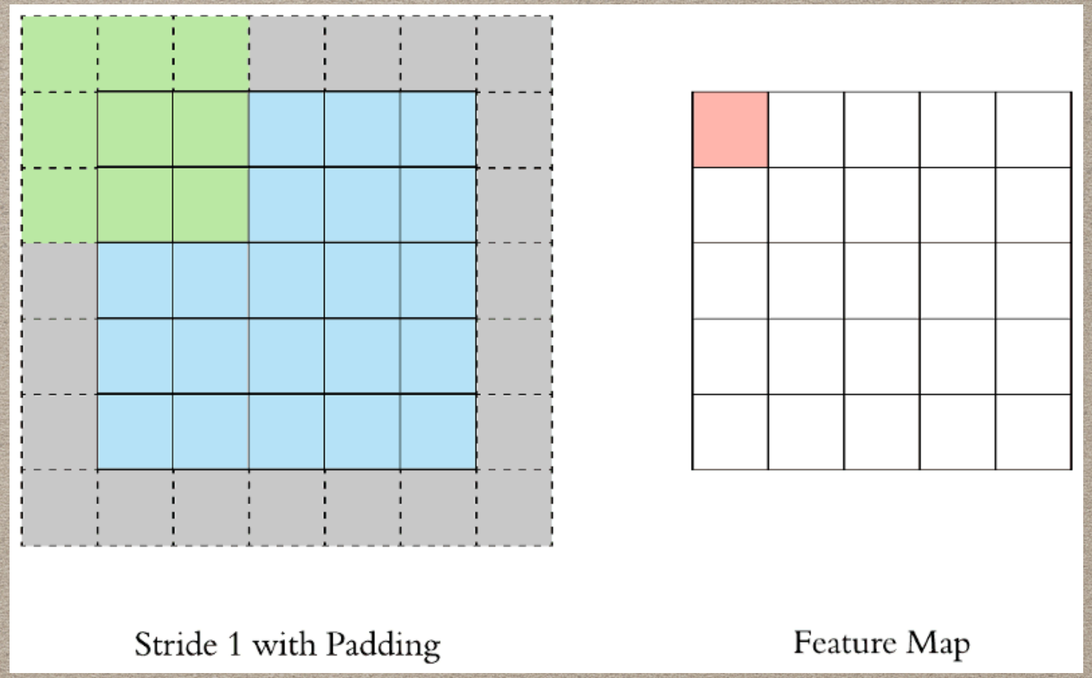




SHOW THE GIF

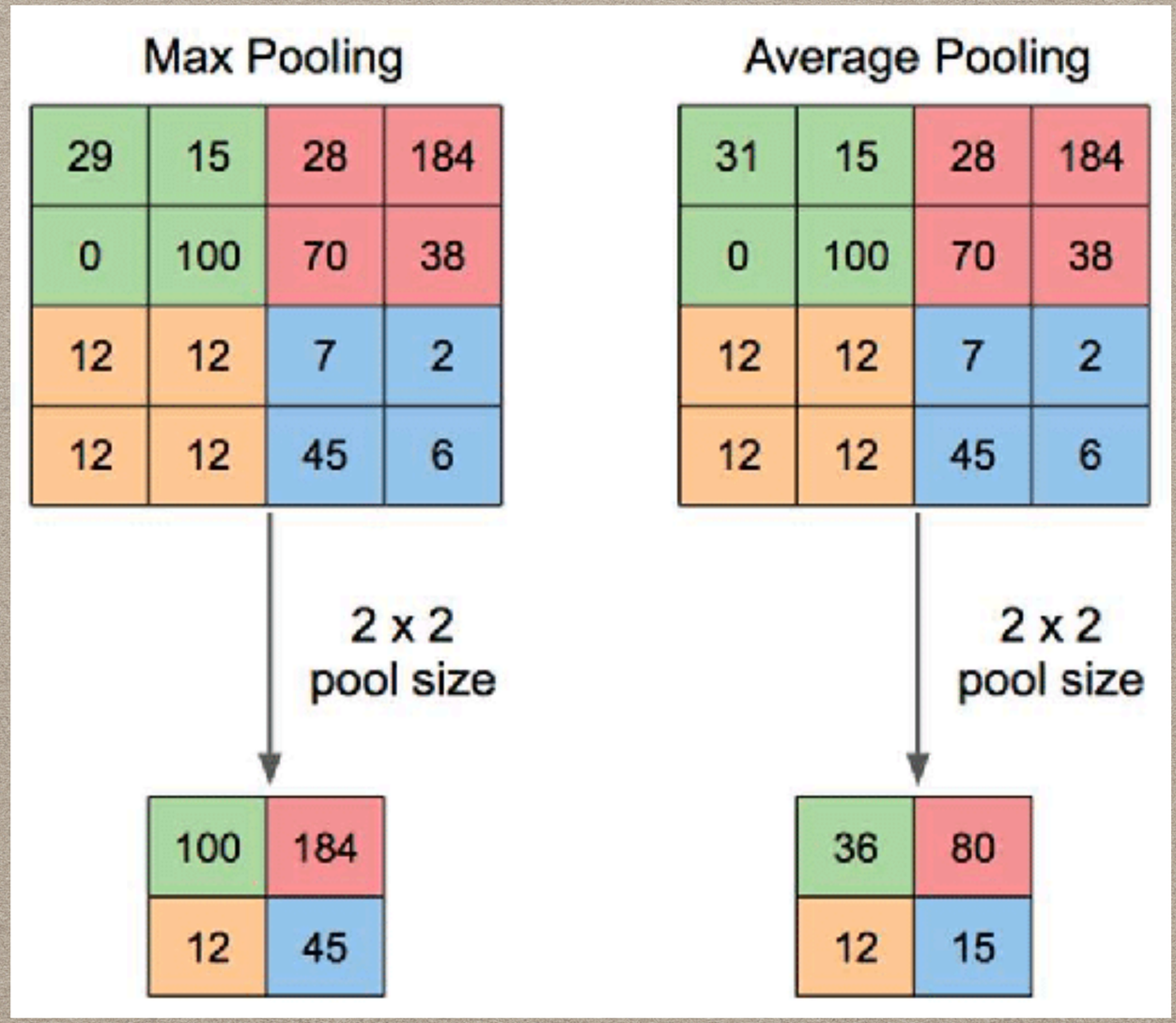
PADDING & STRIDE IN CONVOLUTION





SHOW THE GIF

AVERAGE AND MAX POOLING



FULLY CONNECTED LAYERS AND USE

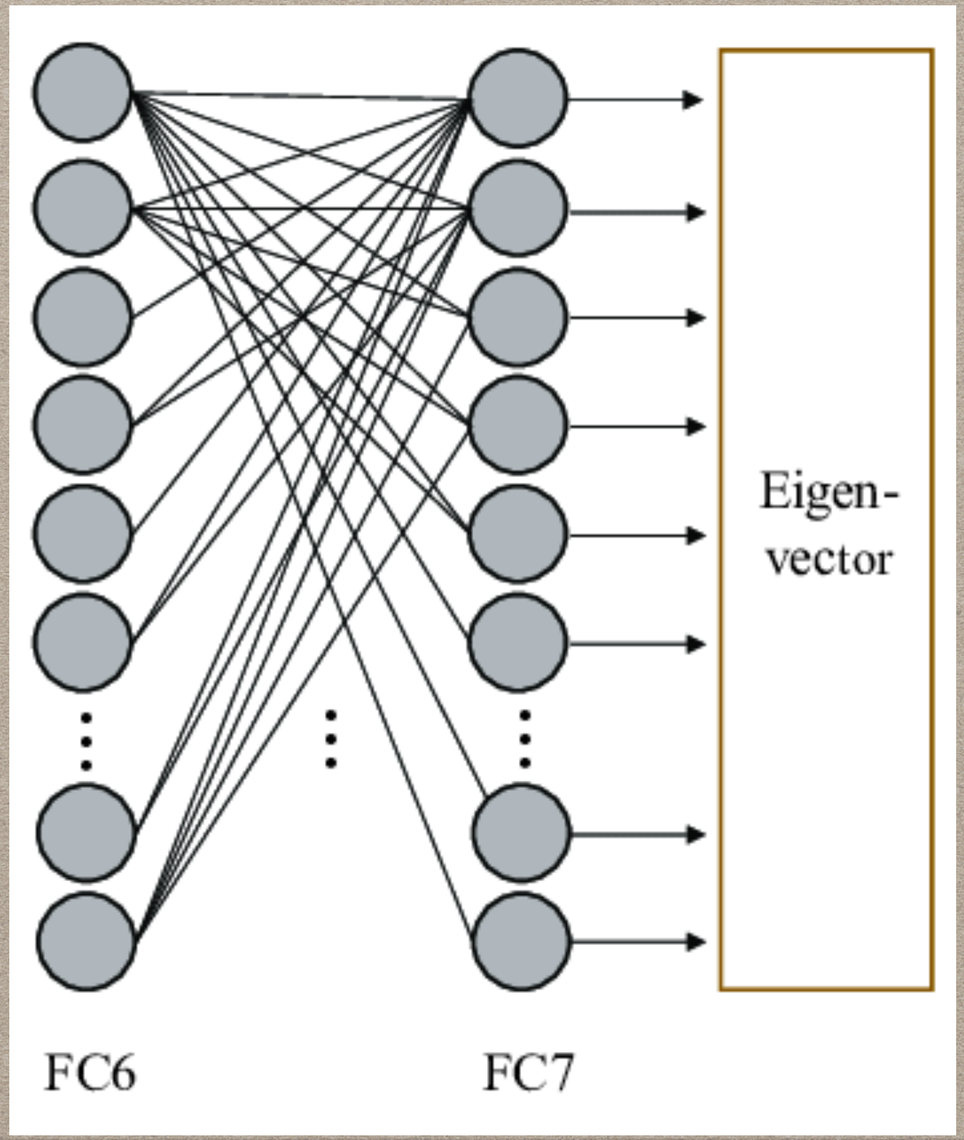


IMAGE CLASSIFICATION

Image classification is the task of taking an input image and outputting a class or a probability of classes that best describes the image

For humans, this task is one of the first skills we learn and it comes naturally and effortlessly as adults

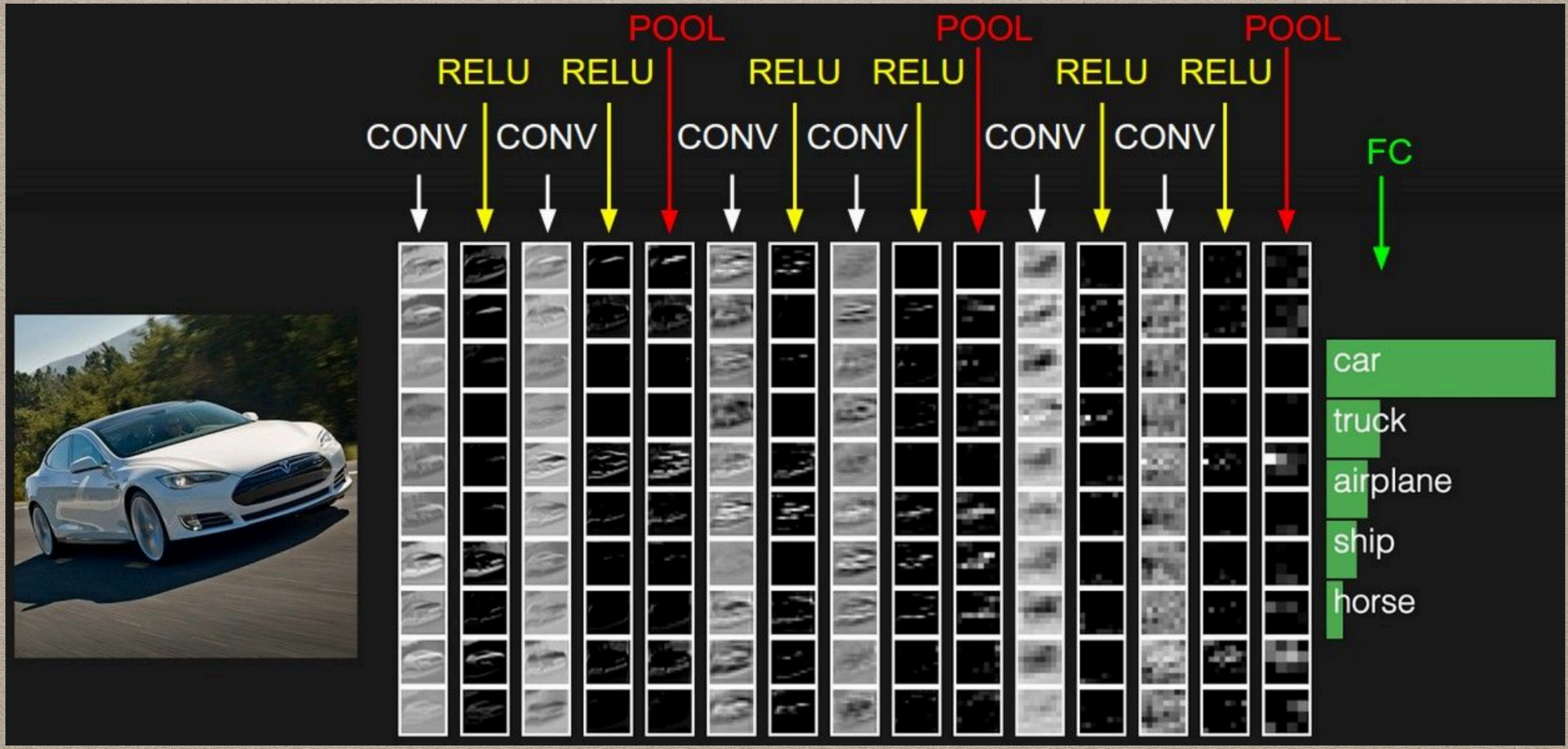
Being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are difficult tasks for machines

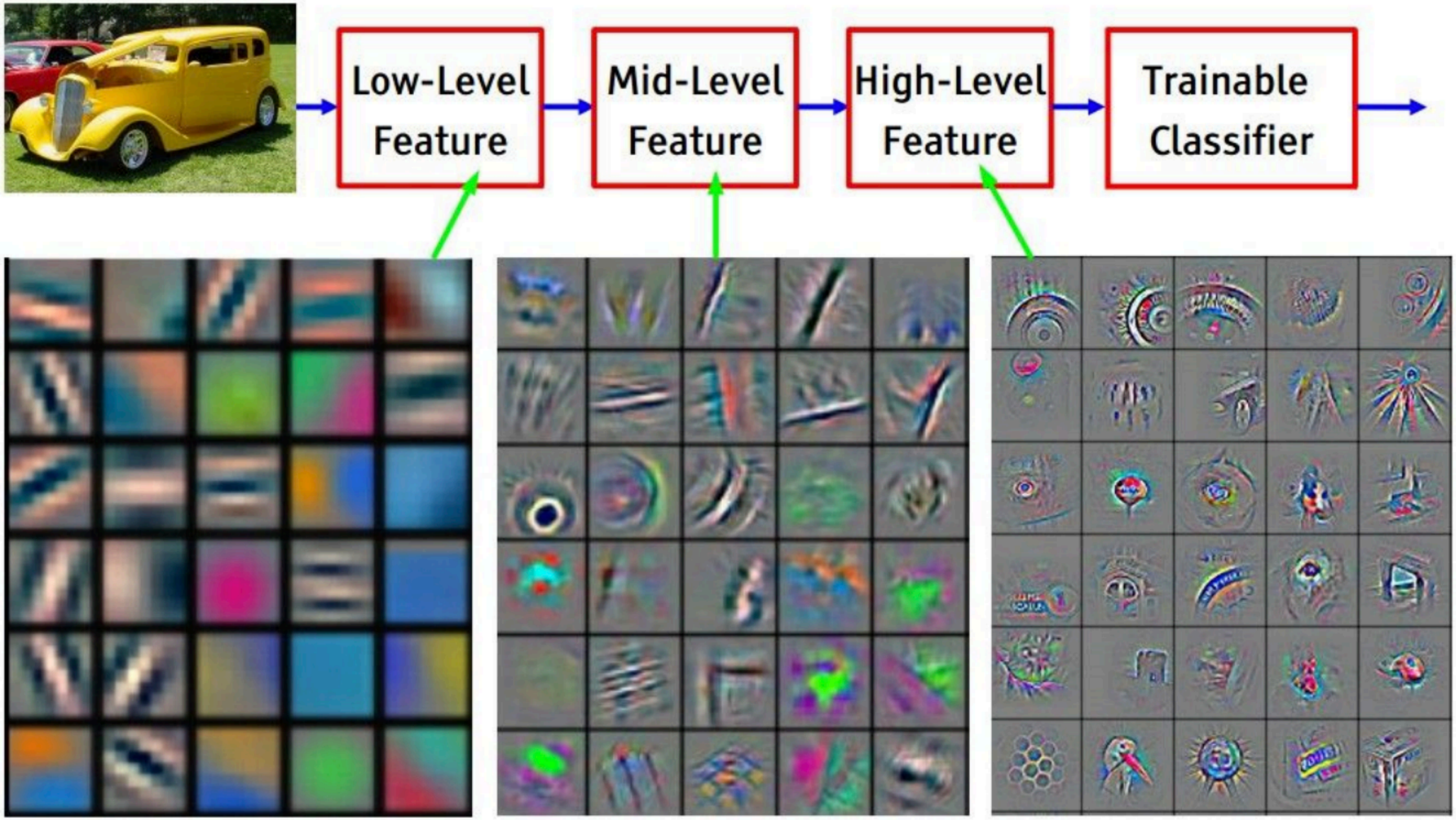
IMAGE CLASSIFICATION

- Hand-Craft Features

- Texture Features: Histogram based, Entropy, Haralick features (Cooccurrence matrix), Gray-level run length metrics, Local Binary Patterr Fractal, etc.

- Morphological Features: Hu's moments, Shape features, Granulometry, Bending Energy, Roundness ratio, etc


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

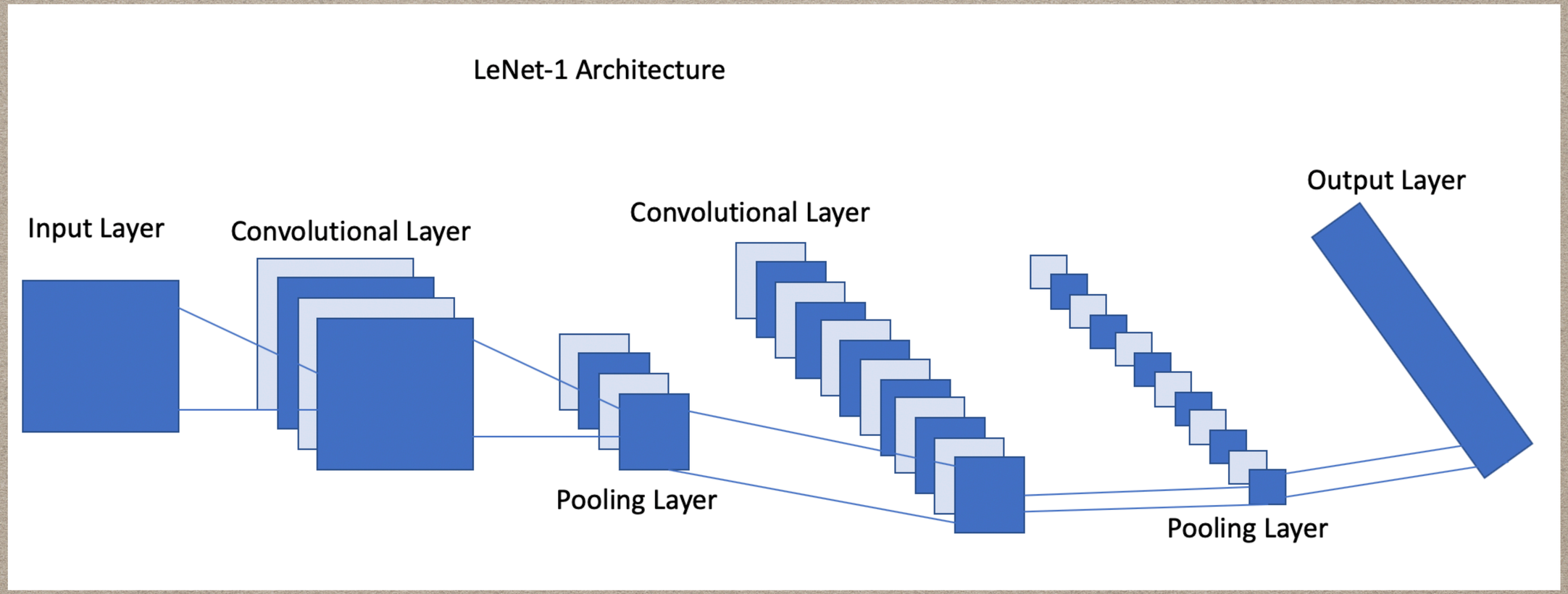
TYPICAL ARCHITECTURE IN CNN

`INPUT -> [[CONV -> RELU]*N -> POOL?]*M -> [FC -> RELU]*K -> FC`

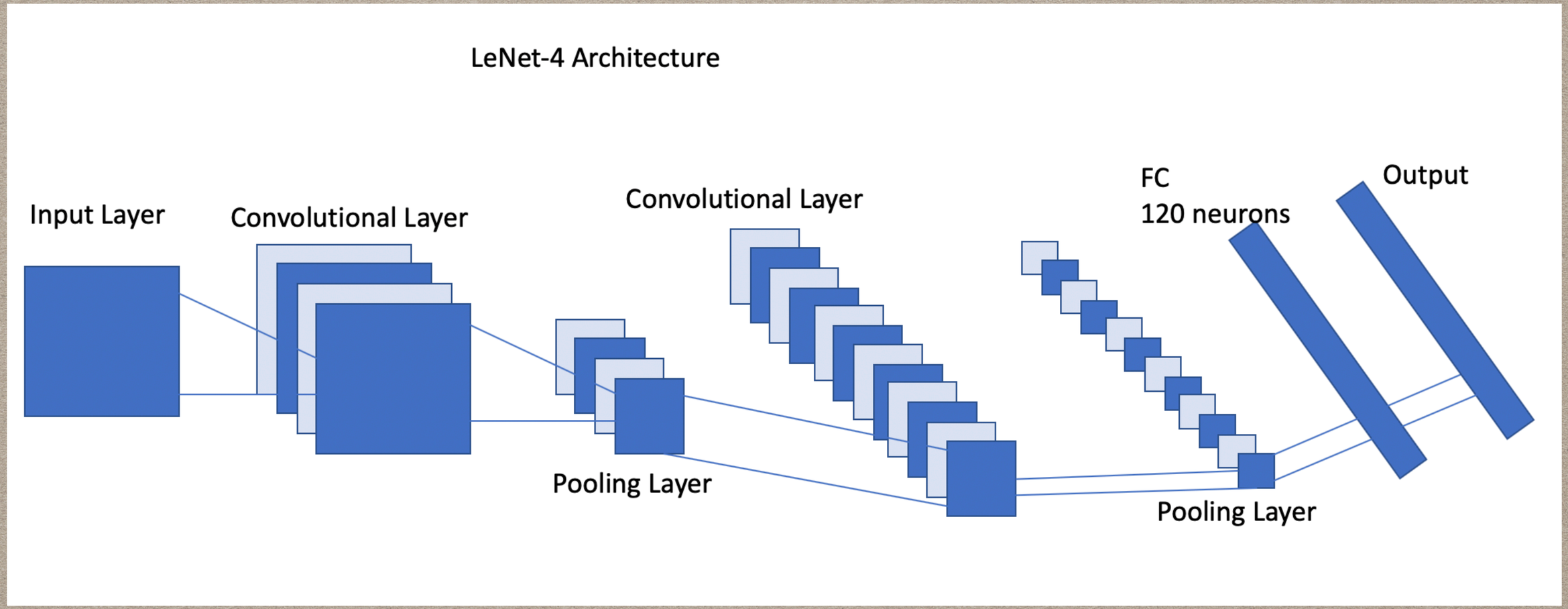
where the `*` indicates repetition, and the `POOL?` indicates an optional pooling layer. Moreover, `N >= 0` (and usually `N <= 3`), `M >= 0`, `K >= 0` (and usually `K < 3`). For example, here are some common ConvNet architectures you may see that follow this pattern:

- `INPUT -> FC`, implements a linear classifier. Here `N = M = K = 0`.
- `INPUT -> CONV -> RELU -> FC`
- `INPUT -> [CONV -> RELU -> POOL]*2 -> FC -> RELU -> FC`. Here we see that there is a single CONV layer between every POOL layer.
- `INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL]*3 -> [FC -> RELU]*2 -> FC` Here we see two CONV layers stacked before every POOL layer. This is generally a good idea for larger and deeper networks, because multiple stacked CONV layers can develop more complex features of the input volume before the destructive pooling operation.

LE-NET 1 ARCHITECTURE



LE-NET 4 ARCHITECTURE



LE-NET 5 ARCHITECTURE

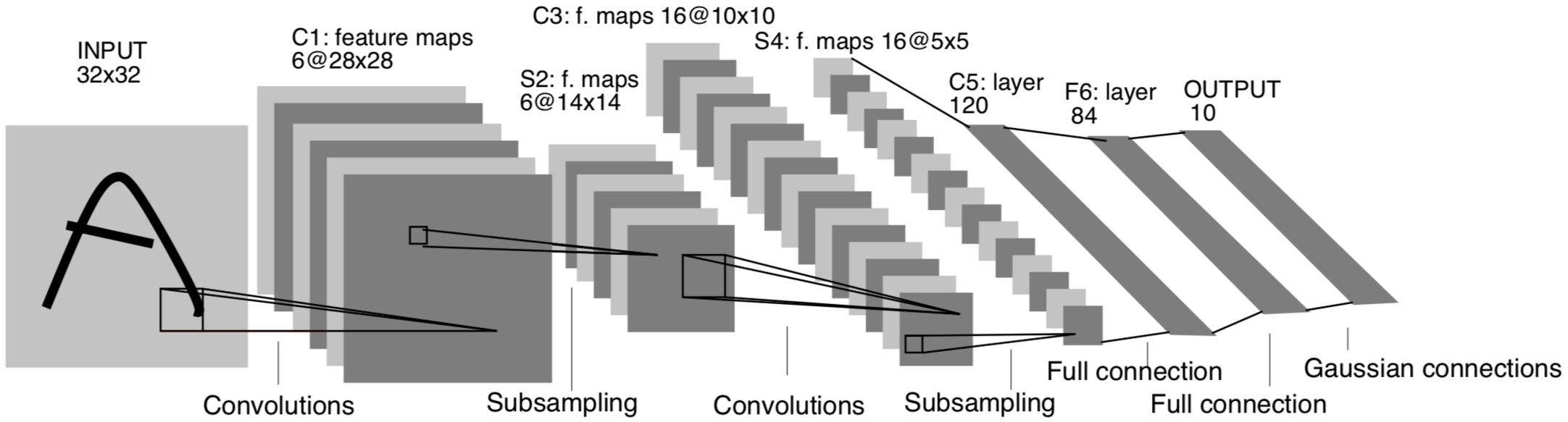


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

TIME TO HIT THE LAB

QUESTIONS PLEASE!

github.com/vverdhan

Thanks