ECE284 Fall 21 W2S2

Low-power VLSI Implementation for Machine Learning

Prof. Mingu Kang

UCSD Computer Engineering

HW1 Graded & HW2 Posted & Typo in Slides

HW1 graded

- Sign extension
- Unused wire
- Parallel if loop
- Reset use case

HW2 posted

- gedit, and using 4 spaces
- PDF download from jupyter notebook

Typo corrected

- W2S1 slides with purple box

Batch Size Choice

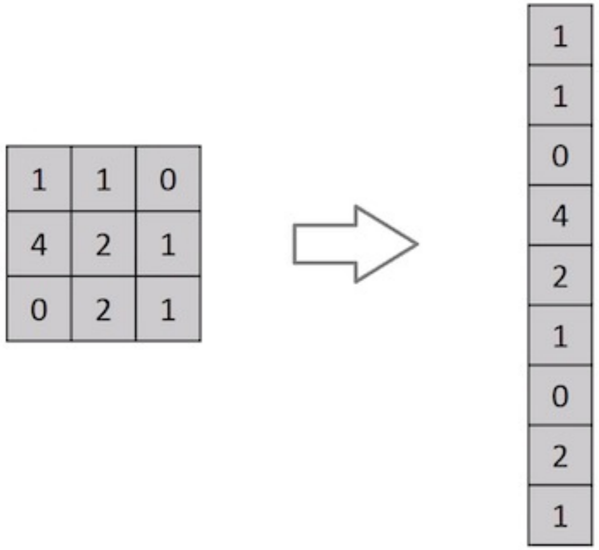
Software

- Pros: fast run-time
- Cons: Large memory consumption -> cause memory fault error (check with “nvidia-smi” command)

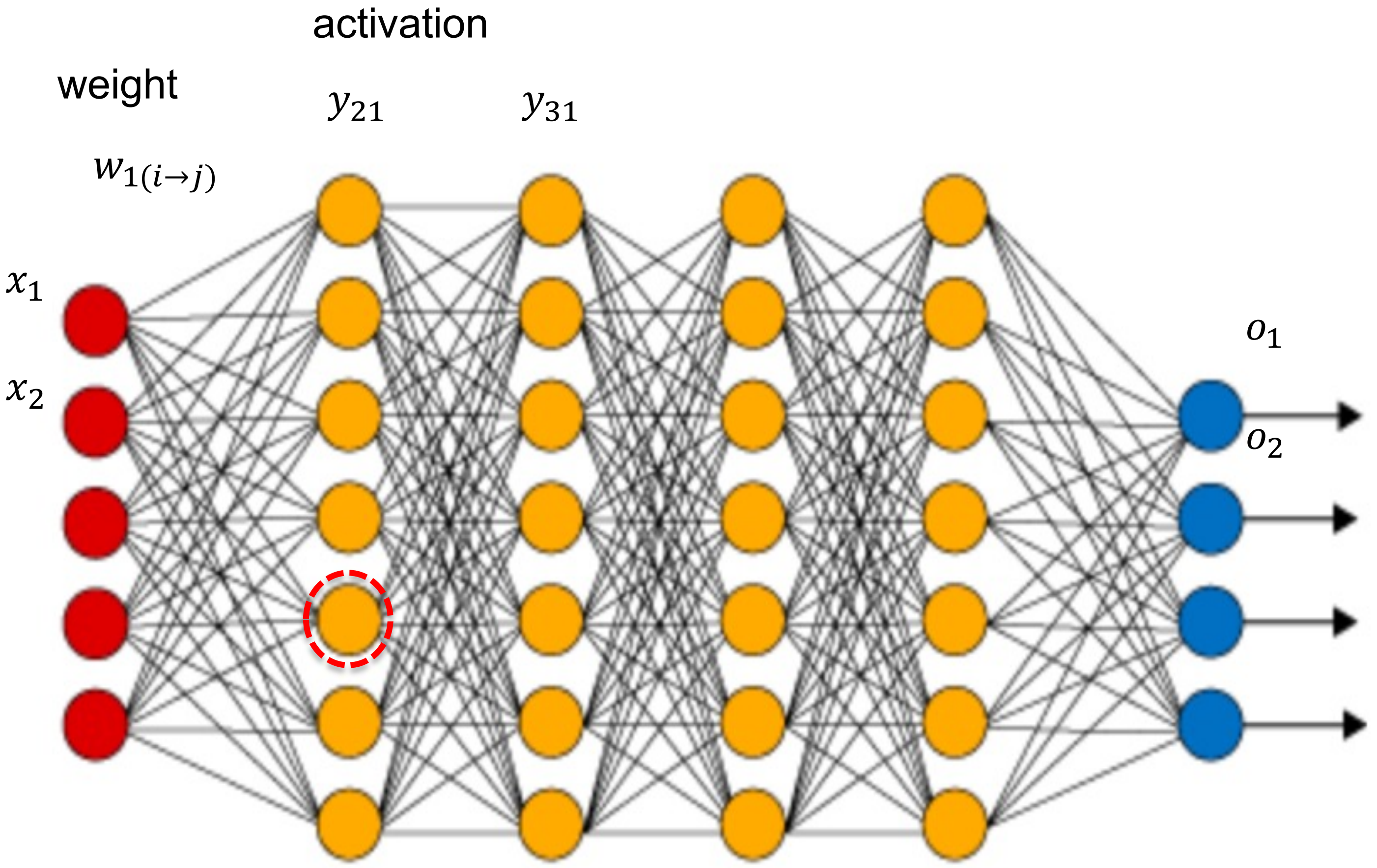
Hardware

- More data re-use opportunity
- Large latency

Multi layer Perceptron

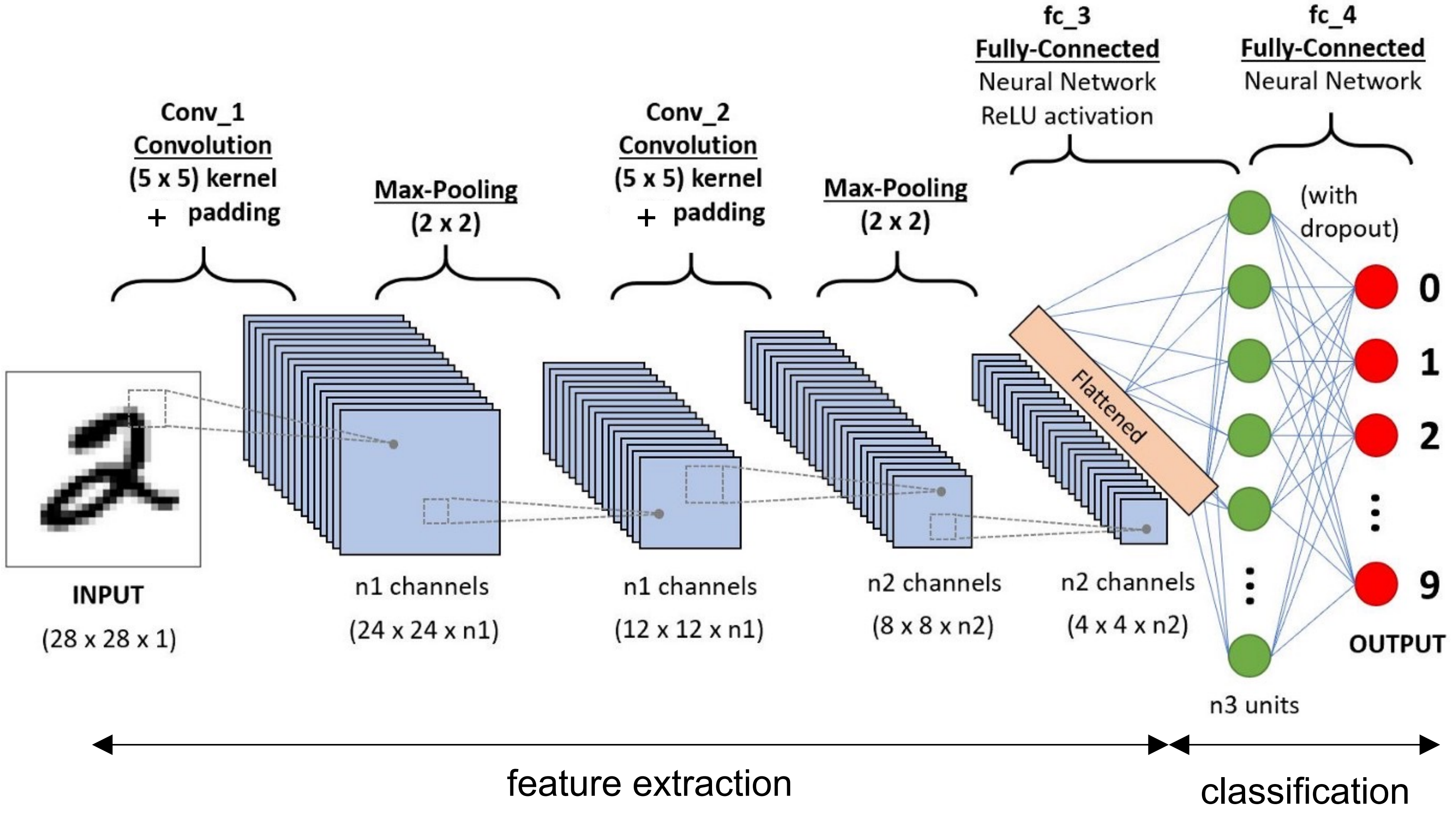


Input layer generation
from 2D image



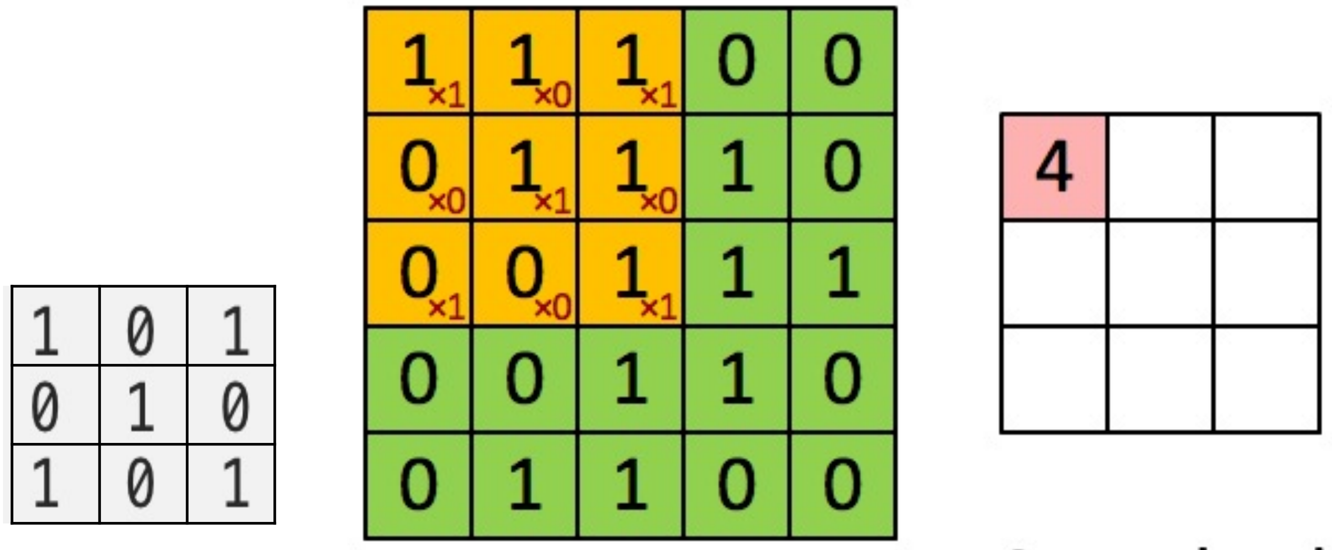
Figures from https://becominghuman.ai/, https://www.learnopencv.com/ https://medium.com/

Convolutional Neural Network



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolution Layer within Single Channel



Kernel Filter

Image

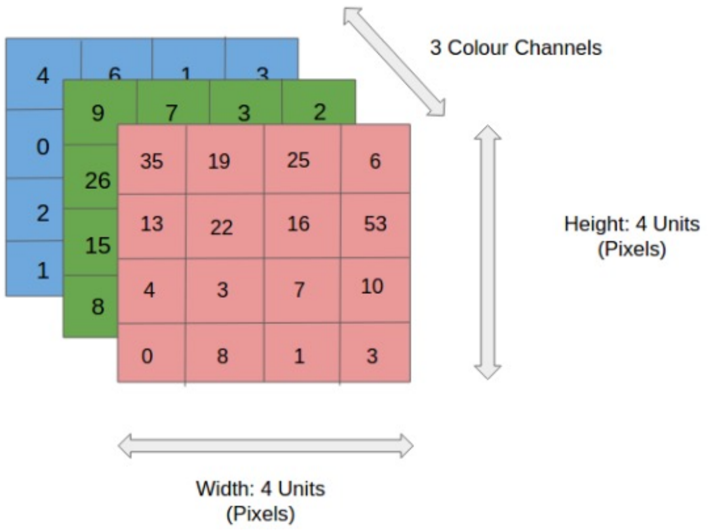
Convolved Feature

Stride = 1

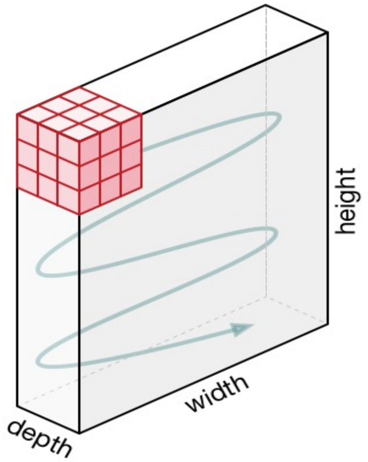
Why convolution ?
- capture the Spatial or temporal dependency well
- reduced the data volume in kernel

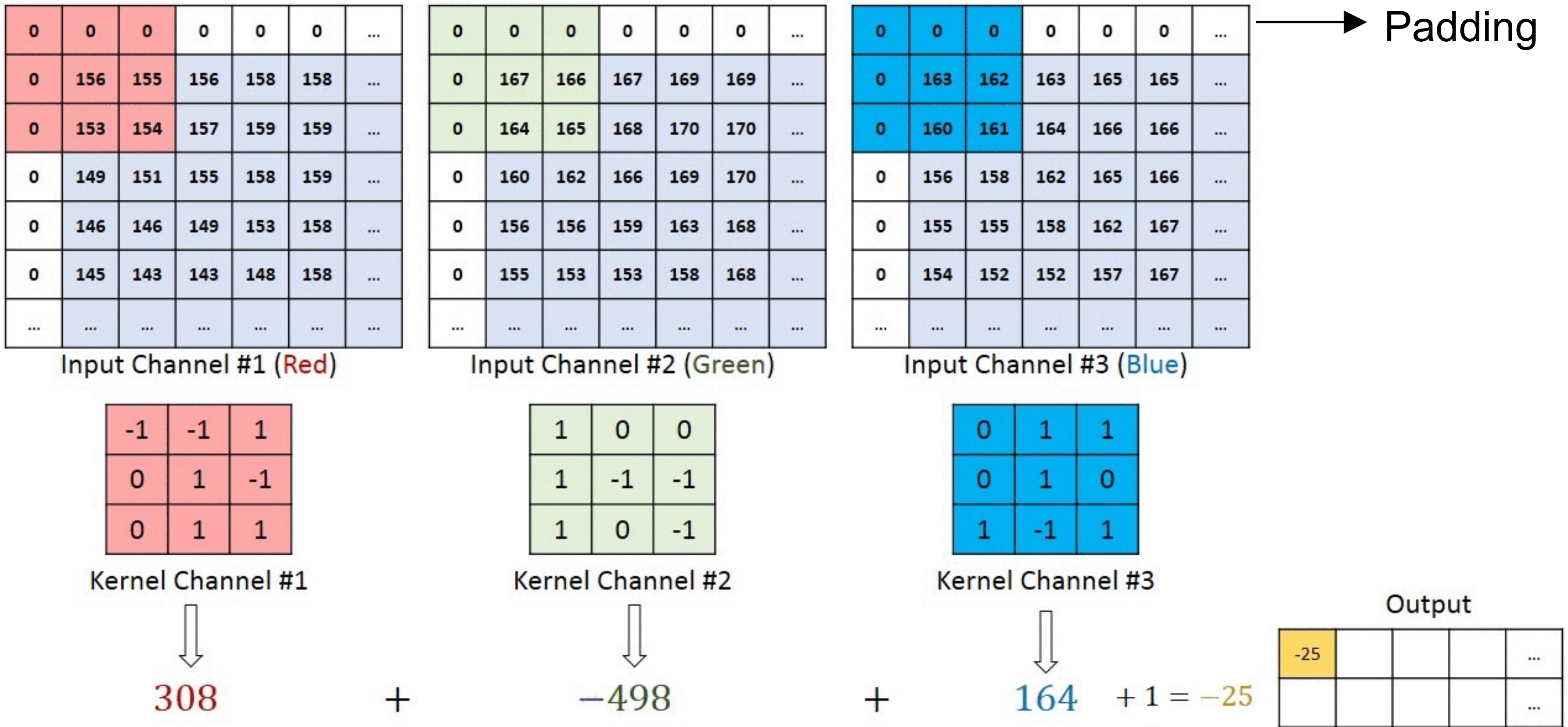
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

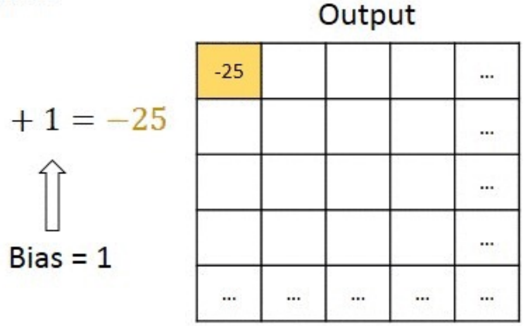
Convolution Layer across Channels (Color Image)



Input (3D data)





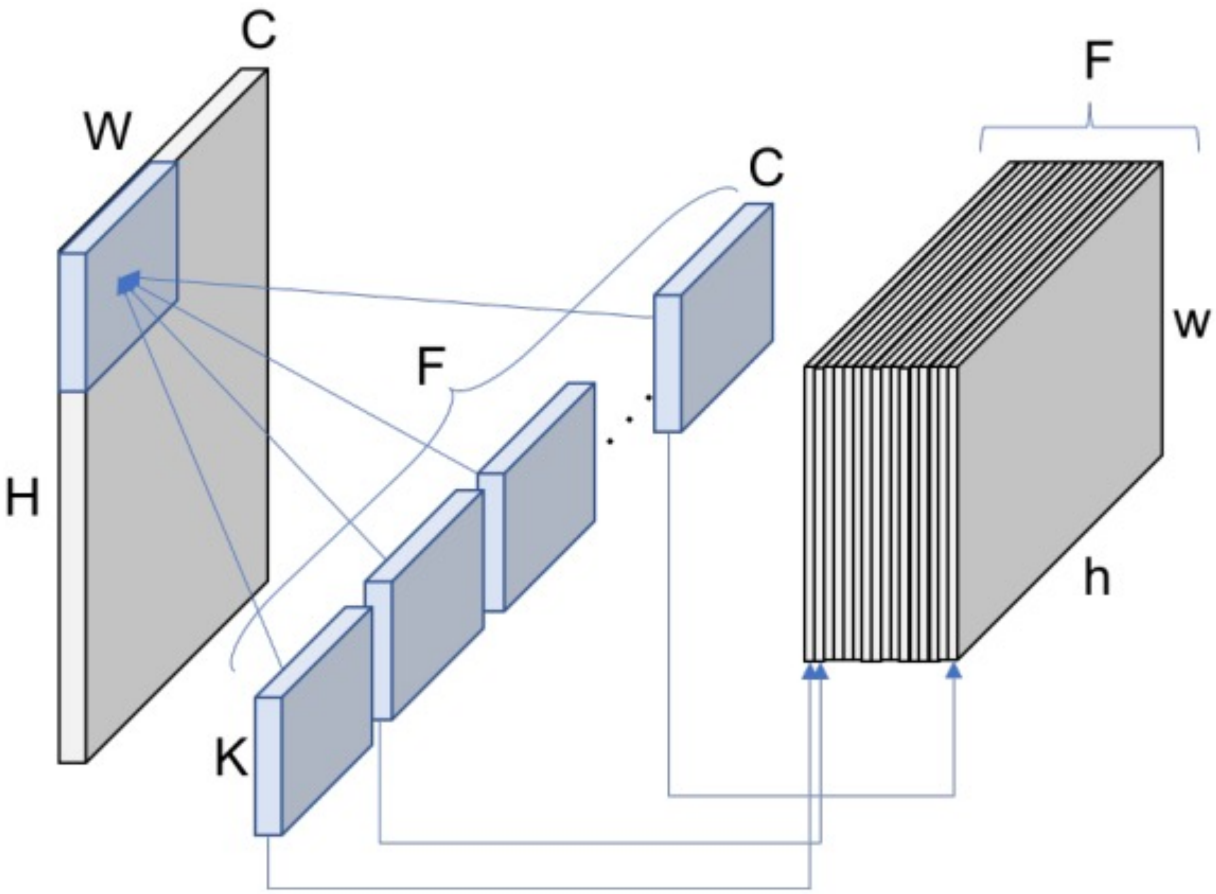


- Above filters are for only 1 output channel
- We need “num_out_ch” such sets of filters

Convolution calculation across channel

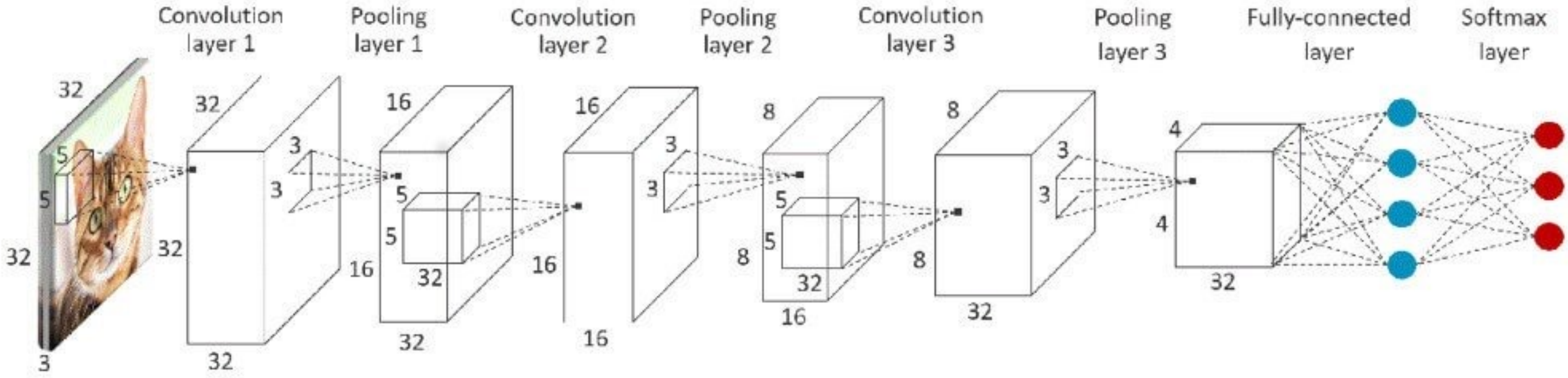
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3D Representation of Convolution



- C: number of input channels, F: number of output channels
- For each output channel, different kernel filters are required

Convolutional Neural Network for Color Image



Neural Network model definition

https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/deploying-convolutional-neural-network-on-cortex-m-with-cmsis-nn

Three Data Reuse Opportunities

1. Filter (kernel) reuse across input feature map coordinate in convolution

2. Input feature map reuse across output channels

3. Filter (kernel) reuse across data points in the batch

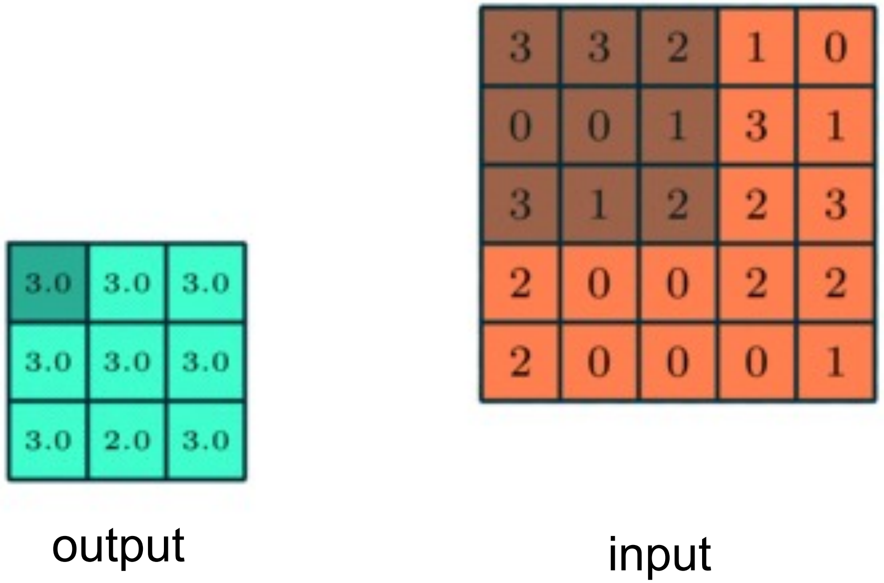
Y.Chen, “Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks”, JSSC16

- Still has difficulty as:

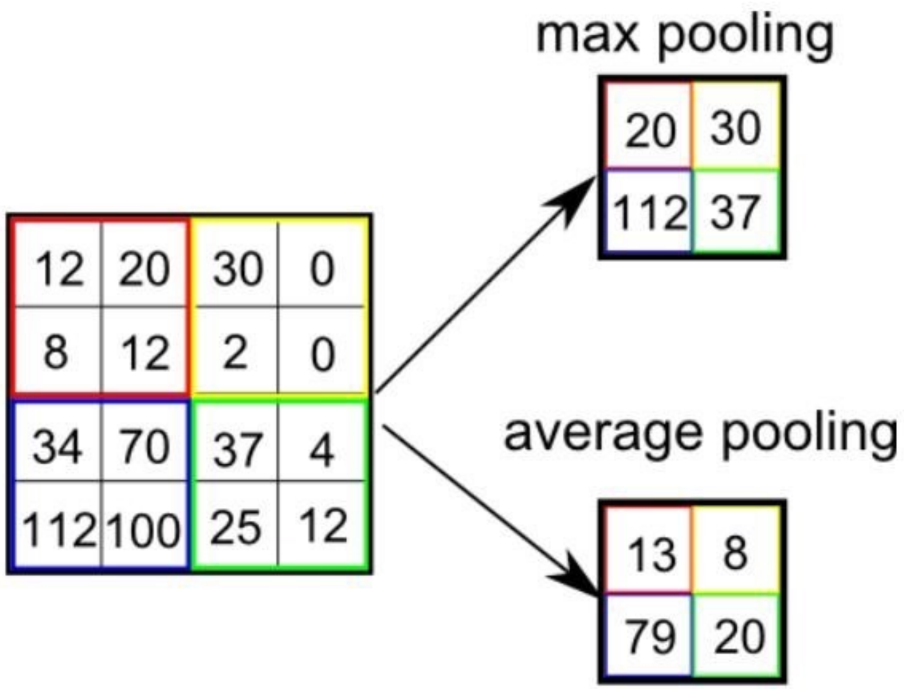
1 .Convolution kernel is compute-intensive whereas

2. Fully-connected layer is memory-bounded

Pooling Layer



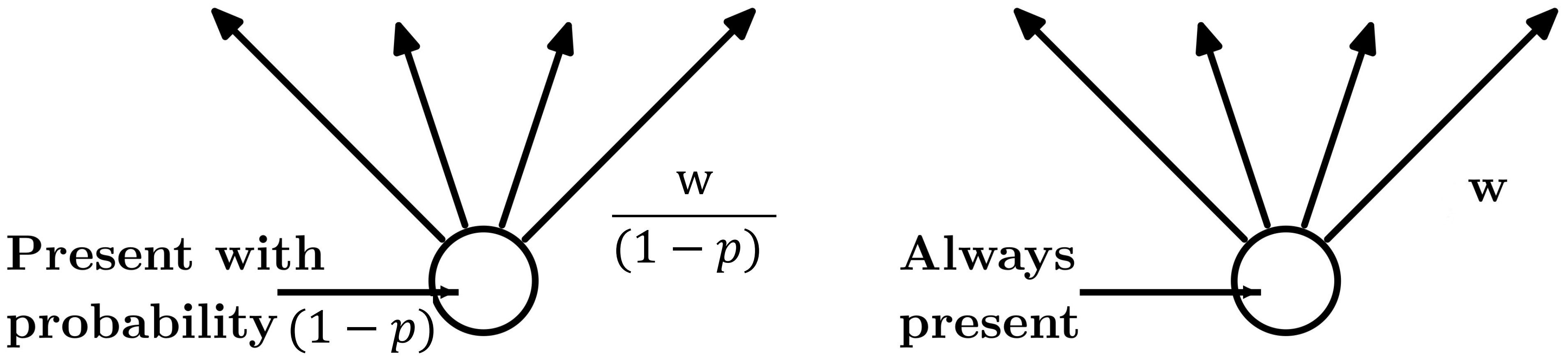
Max pooling operation



Why max or average pooling ?

- extract dominant feature
- reduce the computation power by reducing dimension

(Optional Layers) Dropout Layer



(a) At training time

(b) At test time

Dropout layer

- makes a certain node zero with a probability of p during training
- helps the "*overfitting*" and co-adaptation problems
- only during training

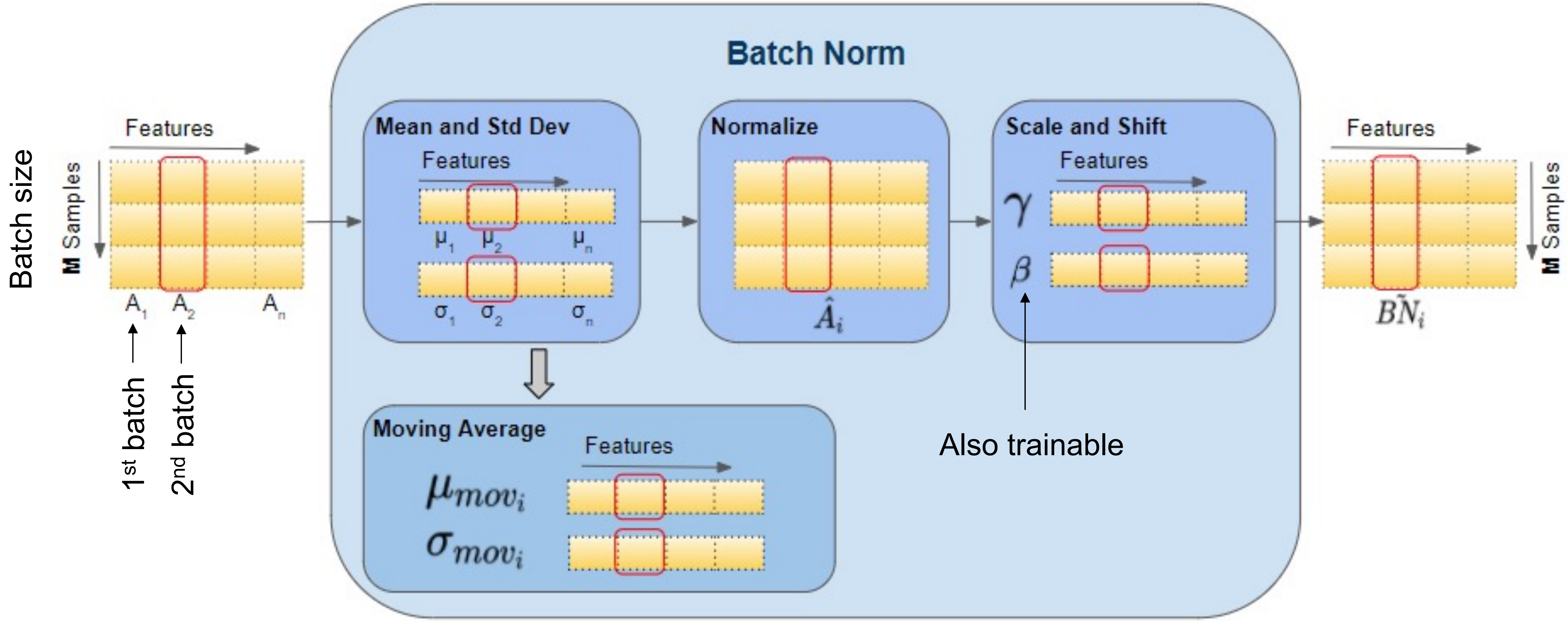
(Optional Layers) Batch Normalization

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

Batchnorm layer

- The mean and standard-deviation are calculated per-channel over the mini-batches during training
- γ (default: 1) and β (default: 0) are learnable parameter
- does not update during inference, but just calculate with fixed mean and var

Batchnorm Visualization



- The last value during the training is used for inference

https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739

[CODE] Batch-normalization Demo (Example 1)

- Input is 2D data
- Check the mean with manual calculation
- Difference during training vs. inference

[CODE] CNN for MNIST (Example 2)

- Introducing GPU use-case
- padding option in conv
- dropout
- Check point save and load
- Network size calculation

  - 1^{st} conv output size = 28 – (3 – 1) = 26

  - 2^{nd} conv output size = 26 – (3 – 1) = 24 → max pool → 12
- fc1 input = 12^2 * input channel (64) = 9216 (Analyze page 5 as well in the same way)
- Pre-hook use-case

https://github.com/iam-mhaseeb/Multi-Layer-Perceptron-MNIST-with-PyTorch/blob/master/mnist_mlp_exercise.ipynb

CIFAR10 Dataset

airplane

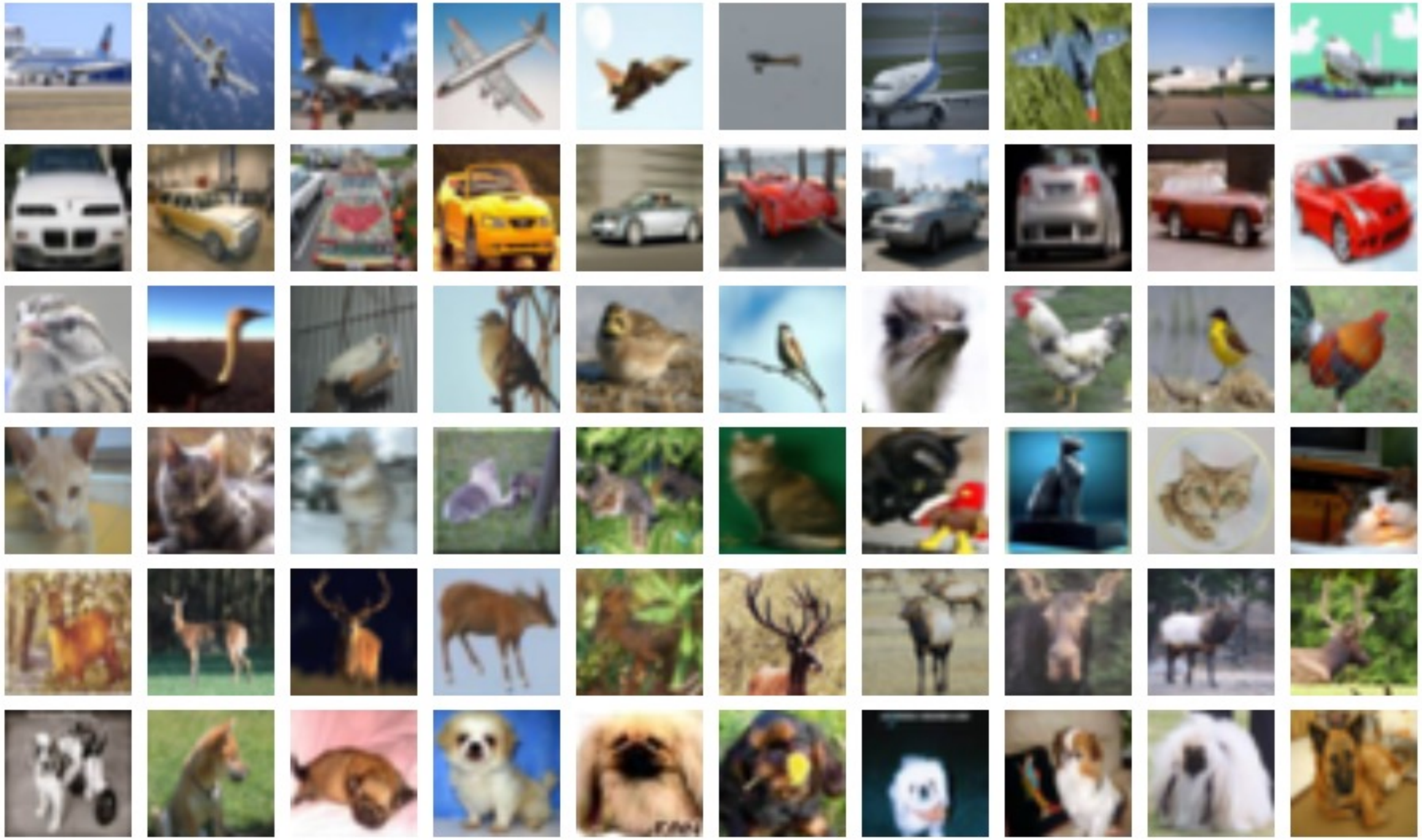
automobile

bird

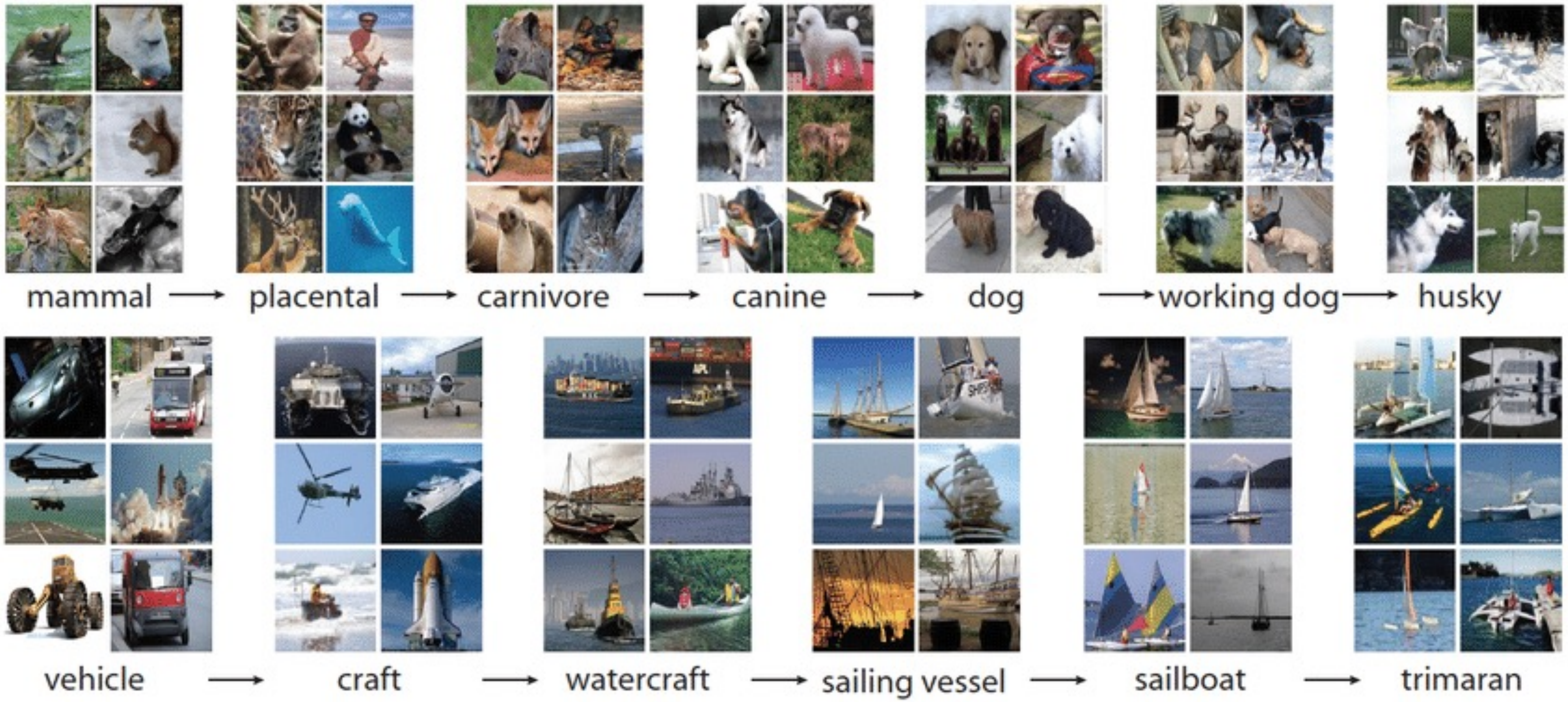
cat

deer

dog

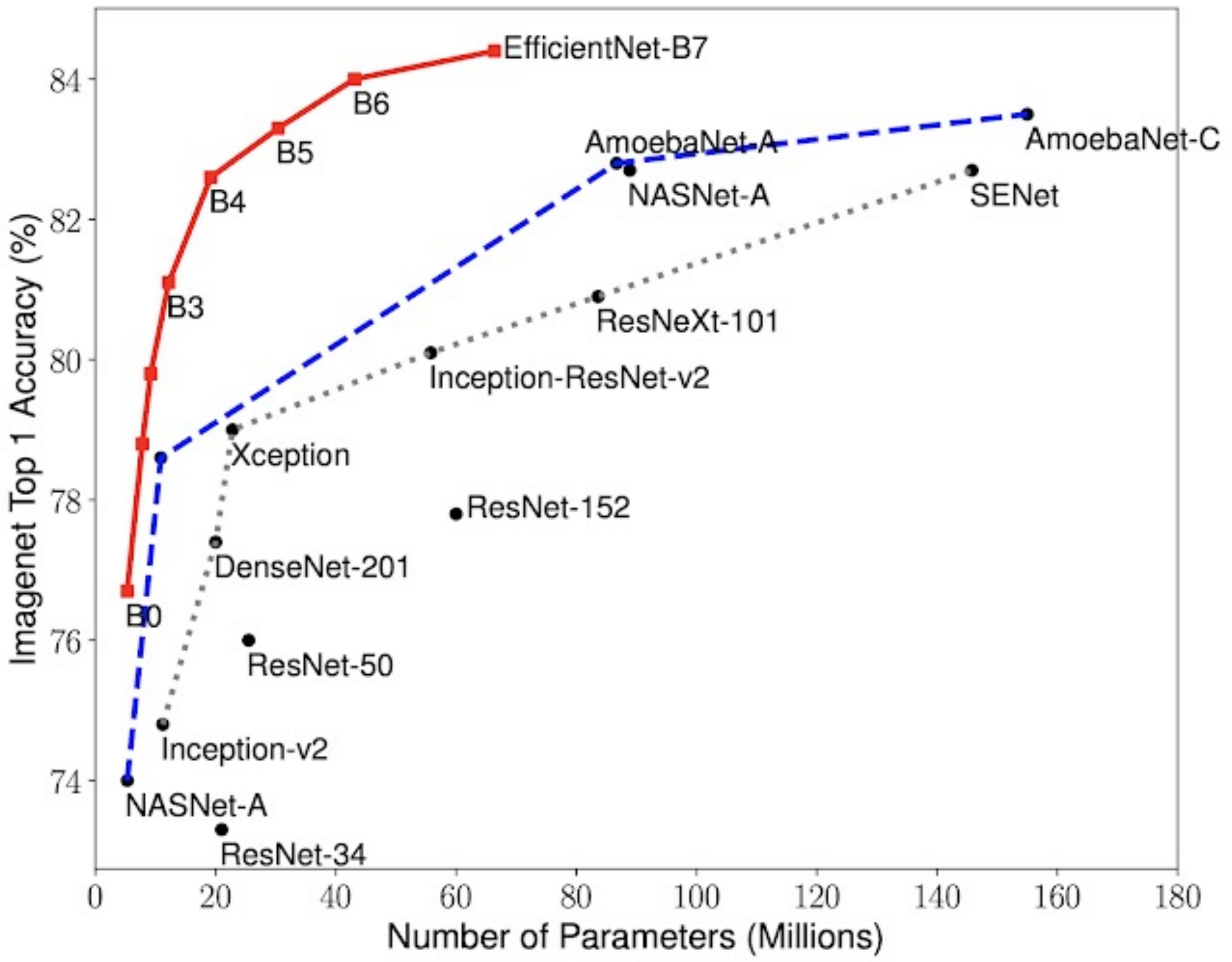


ImageNet Dataset



https://devopedia.org/imagenet

ImageNet Accuracy Trend



- Total 1000 classes:

  Full list of classes [Link]

[CODE] CNN Training for CIFAR10 (Example3)