Convolutional Neural Networks (CNNs)

ICDSS - Kacper Kazaniecki (kk518)

Overview

- Where are CNNs used? i.e. why should you care?
- How do CNNs work?
- Overview of common architectures
- Live Demo

Where are CNNs used?

Fundamental modern building block of most computer vision systems:

- Classification
- Object Detection
- Image Segmentation
- Pose Estimation
- Image Captioning
- Generative models
- + many many more applications.

Fully connected neural network

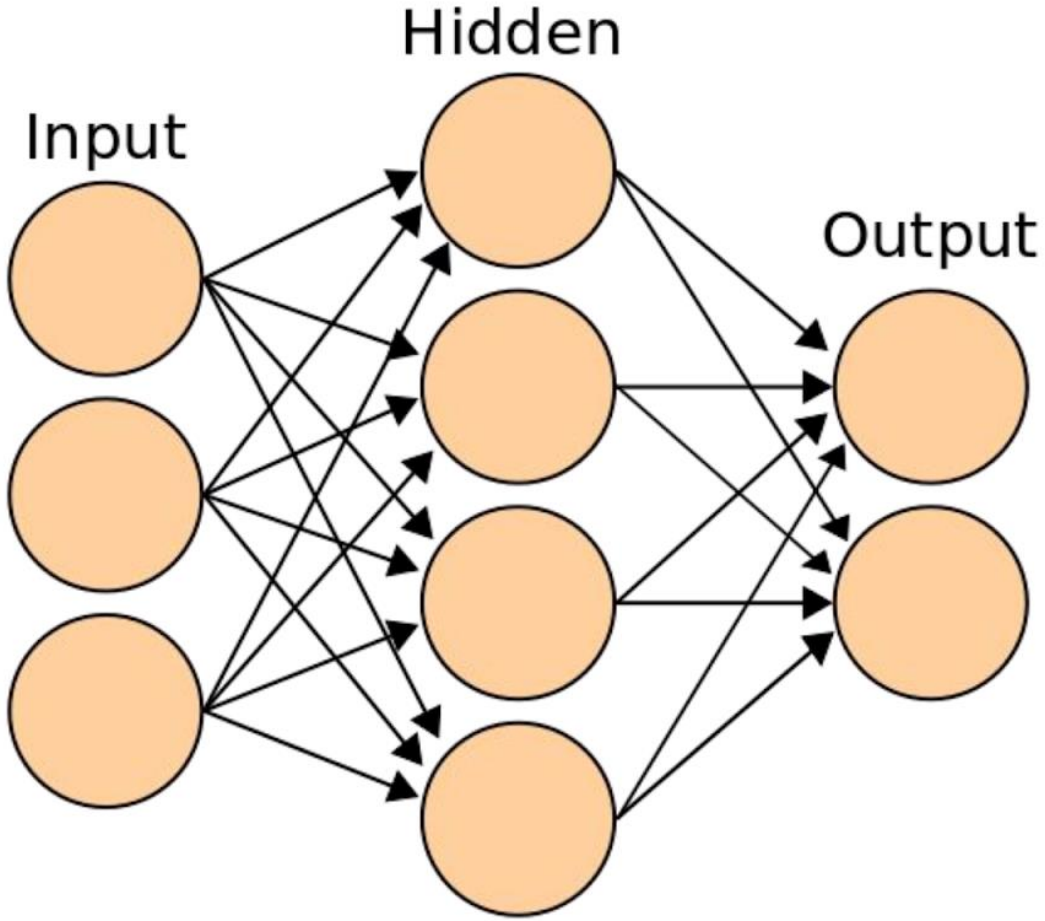
Neural Network

Stack perceptrons into layers

Stack layers into network

Weights now represented by matrix

All parameters (weights) represented by θ



https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Artificial_neural_network.svg

Issue when dealing with images

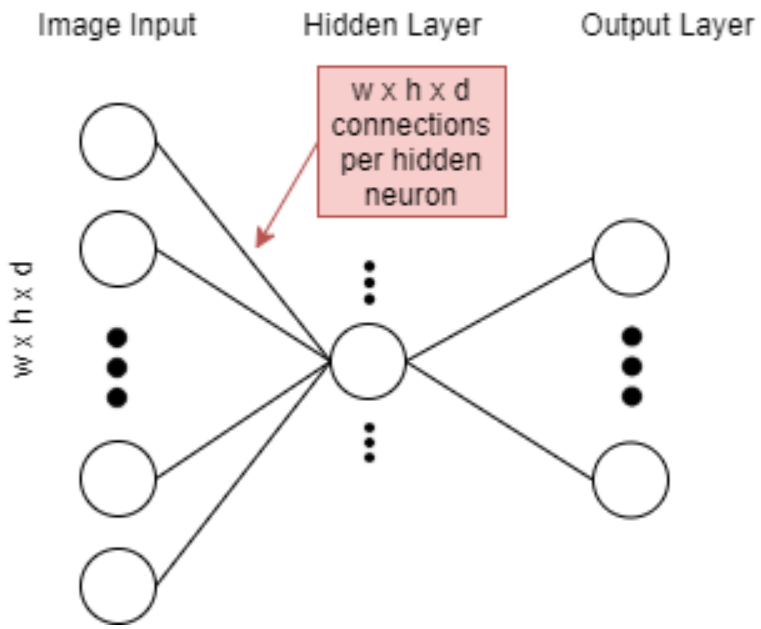
Take an image of size 200x200x3 (width x height x RGB colour depth)

That's 120,000 connections per neuron in 1 hidden layer!

We don't need that many connections and having so many would be very prone to overfitting.

Maybe there's an efficient way we can share weights?

-We’re often looking for the same feature in different parts of an image



Dealing with images

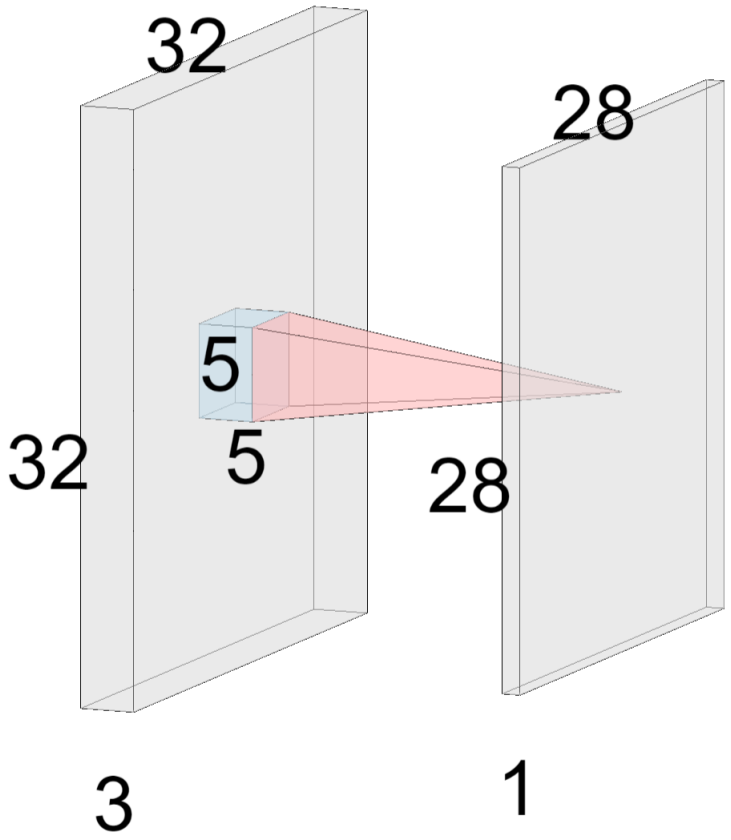
Input of image size 32x32x3 (width x height x depth)

Blue is our filter (our reusable weights)

Same as before, dot product between input and weights (+bias) making up the input to a neuron in the next layer

Note that the **depth** of filter and the input always matches.

Convolution Layer



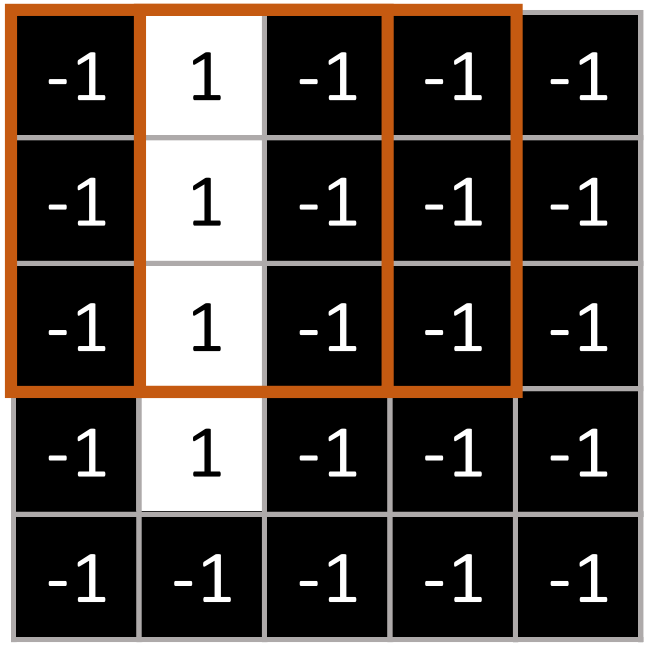
Input of image size 32x32x3 (width x height x depth)

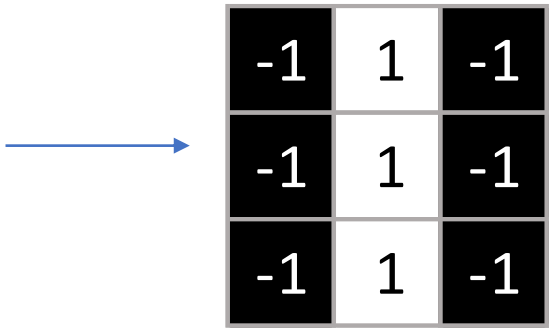
Blue is our filter (our reusable weights)

Same as before, dot product between input and weights (+bias) making up the input to a neuron in the next layer

Note that the **depth** of filter and the input always matches.

Filter (kernel)





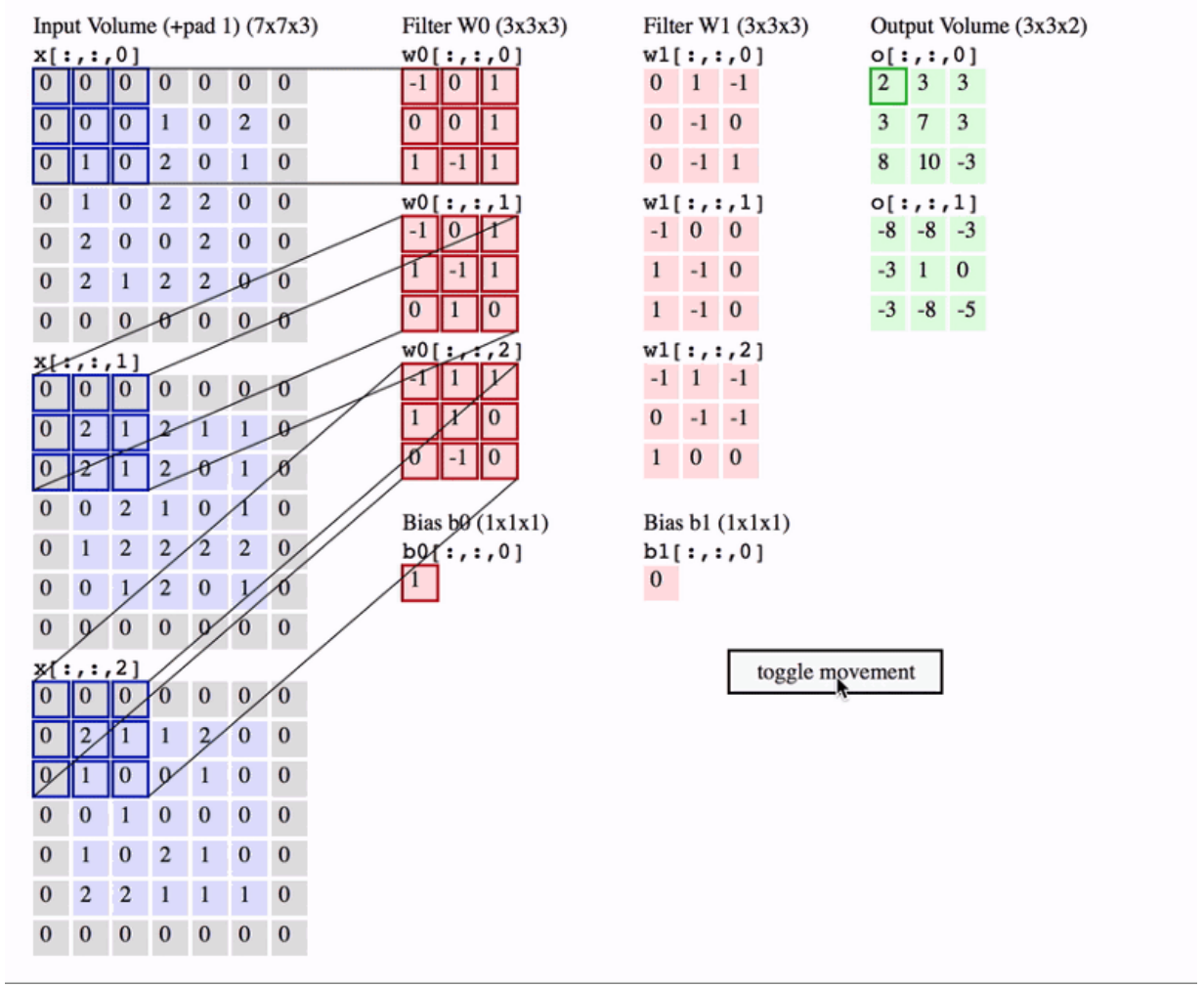
1	**-1/3**	1/3
1	-1/3	1/3
7/9	-1/9	1/3

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

$$\frac{-1-1-1-1-1-1+1+1+1}{9} = -\frac{1}{3}$$

We're using a stride of 1 here.

Extending to multiple filters with depth

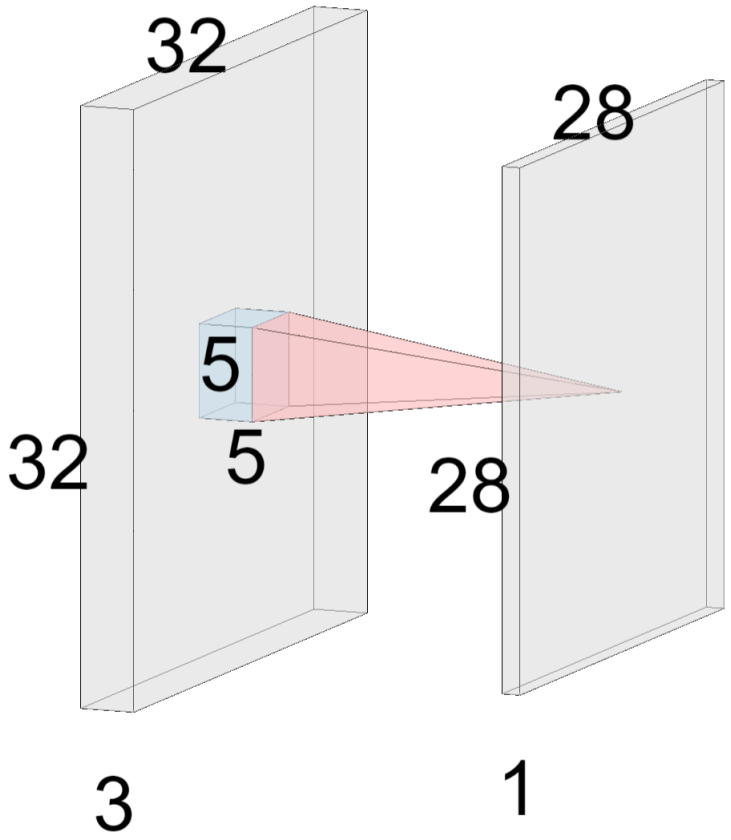


Filter dimensions are equal for width and height (i.e. square filters)

Depth of filter matches depth of input

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

Convolution Layer



Input of image size 32x32x3 (width x height x depth)

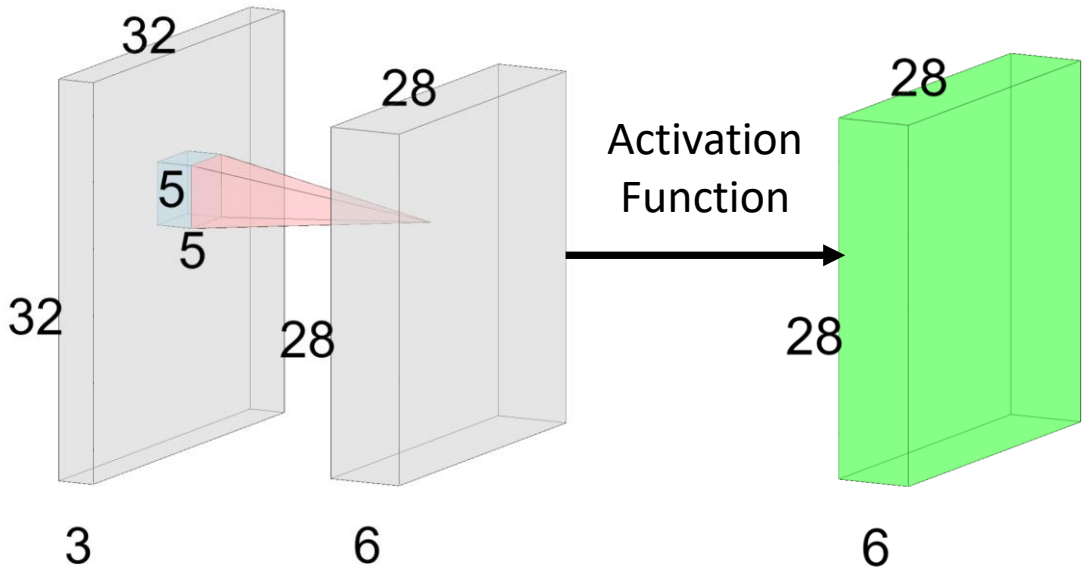
Blue is our filter (our reusable weights)

Same as before, dot product between input and weights (+bias) making up the input to a neuron in the next layer

Note that the **depth** of filter and the input always matches.

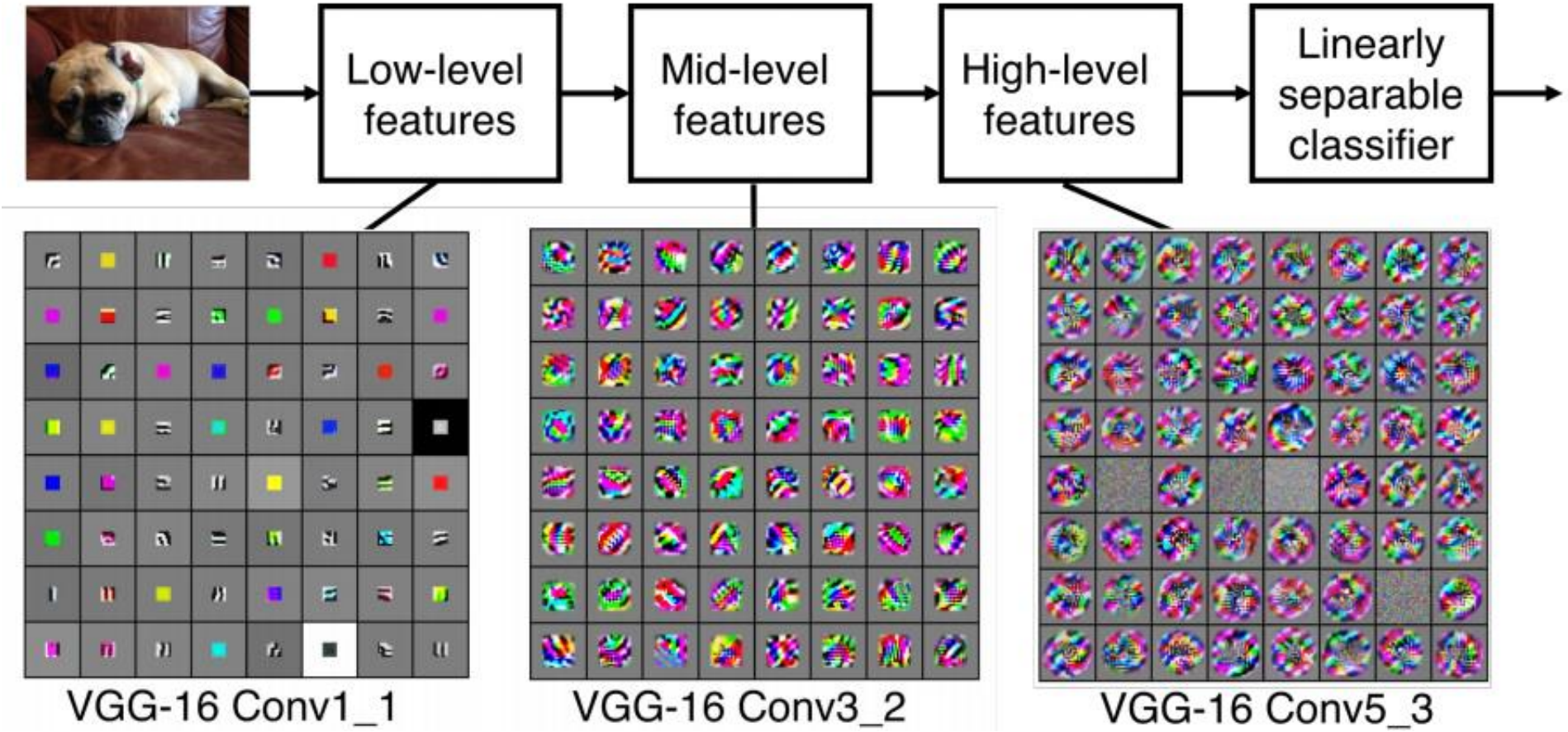
Multiple filters

- Given 6 filters of 5x5*, we get 6 corresponding activation maps
- We'd usually apply an activation function on the activation maps (elementwise). E.g. ReLU



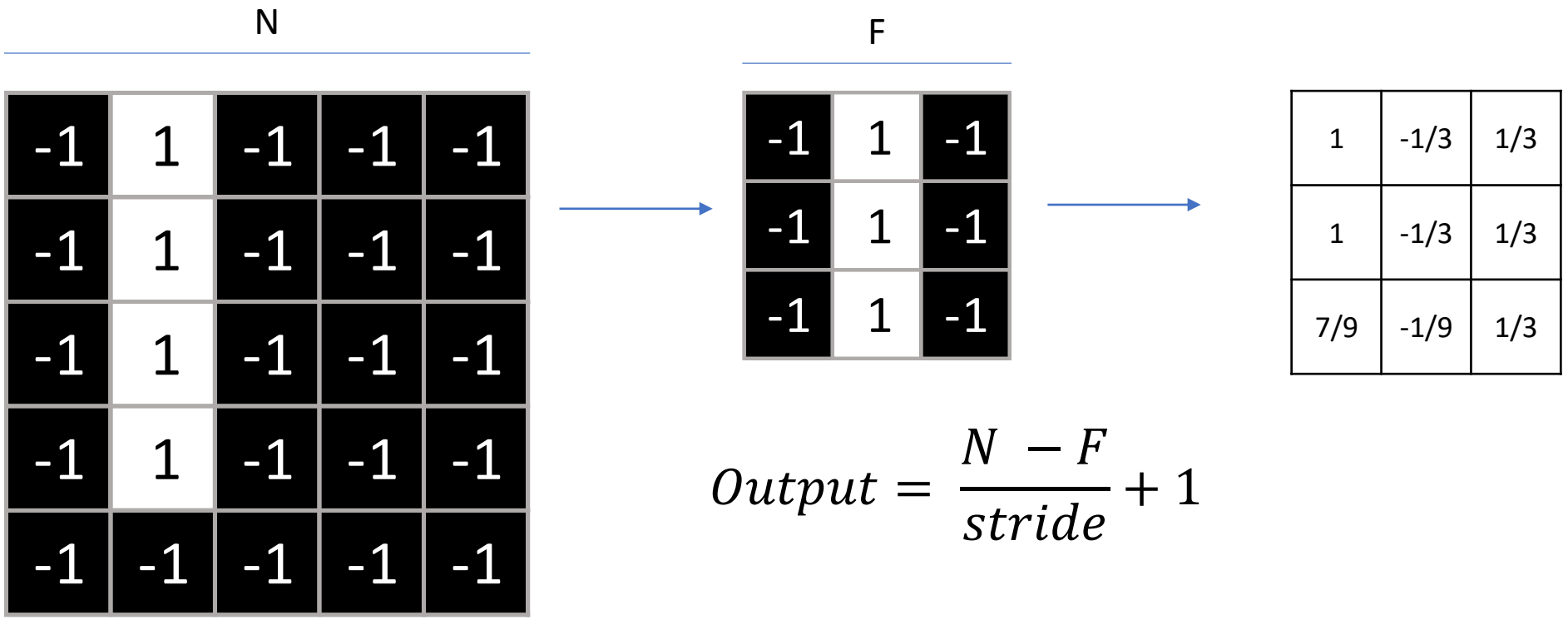
* x3 depth implied

Visualisation



Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].

Dimensions



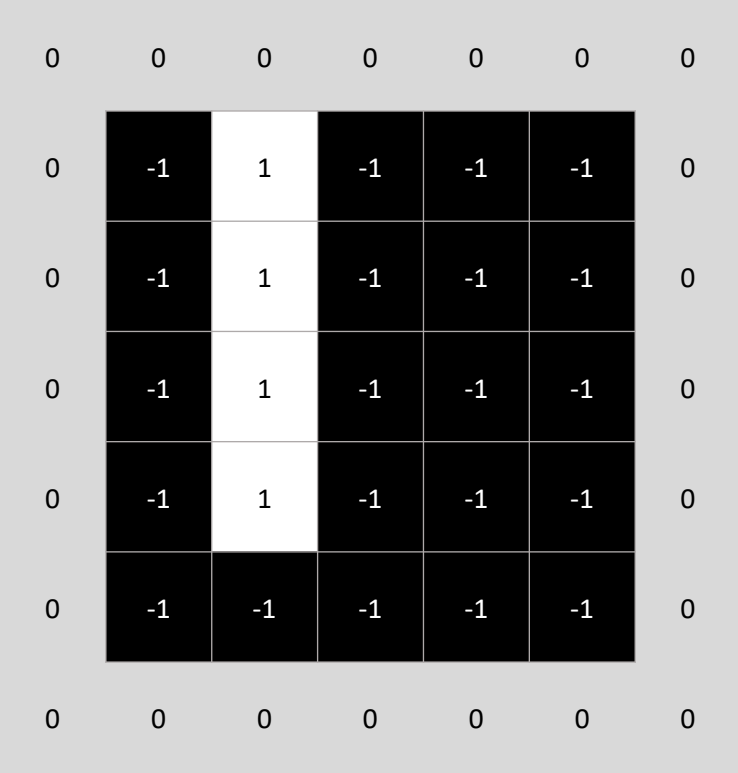
$$Output = \frac{N - F}{stride} + 1$$

Examples: For N = 5, F = 3

Stride 1: (5 - 3)/1 + 1 = 3
Stride 1: (5 - 3)/2 + 1 = 2
Stride 1: (5 - 3)/3 + 1 = 1.6666 < ---- doesn't fit

Padding

- Usually to preserve dimensions



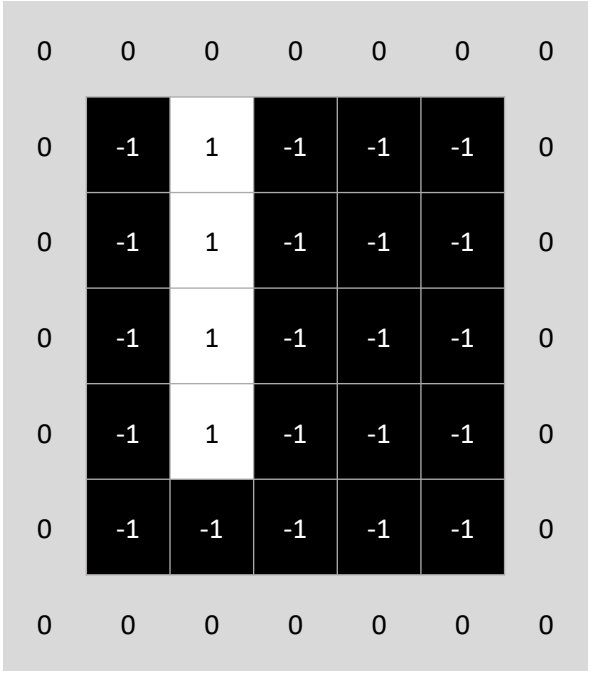
$$Output = \frac{N - F}{stride} + 1$$

Our previous example with stride 1 but we want to keep the dimensions

$$Output = \frac{7 - 3}{1} + 1 = 5$$

Dimensions of convolutional layer in general

- Inputs shape: $W_1 \; x \; H_1 \; \mathrm{x} \; D_1$
- Given:
  - Number of filters K
  - Size of filters F,
  - Stride S,
  - Padding of P
- Output shape:
  - $W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
  - $H_2 = \frac{(H_1 - F + 2P)}{S} + 1$
  - $D_2 = K$



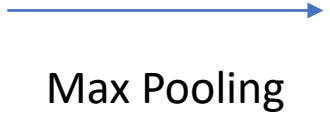
Max Pooling

Given a 3x3 max pooling filter with stride 2

$$Output = \frac{N - F}{stride} + 1$$

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

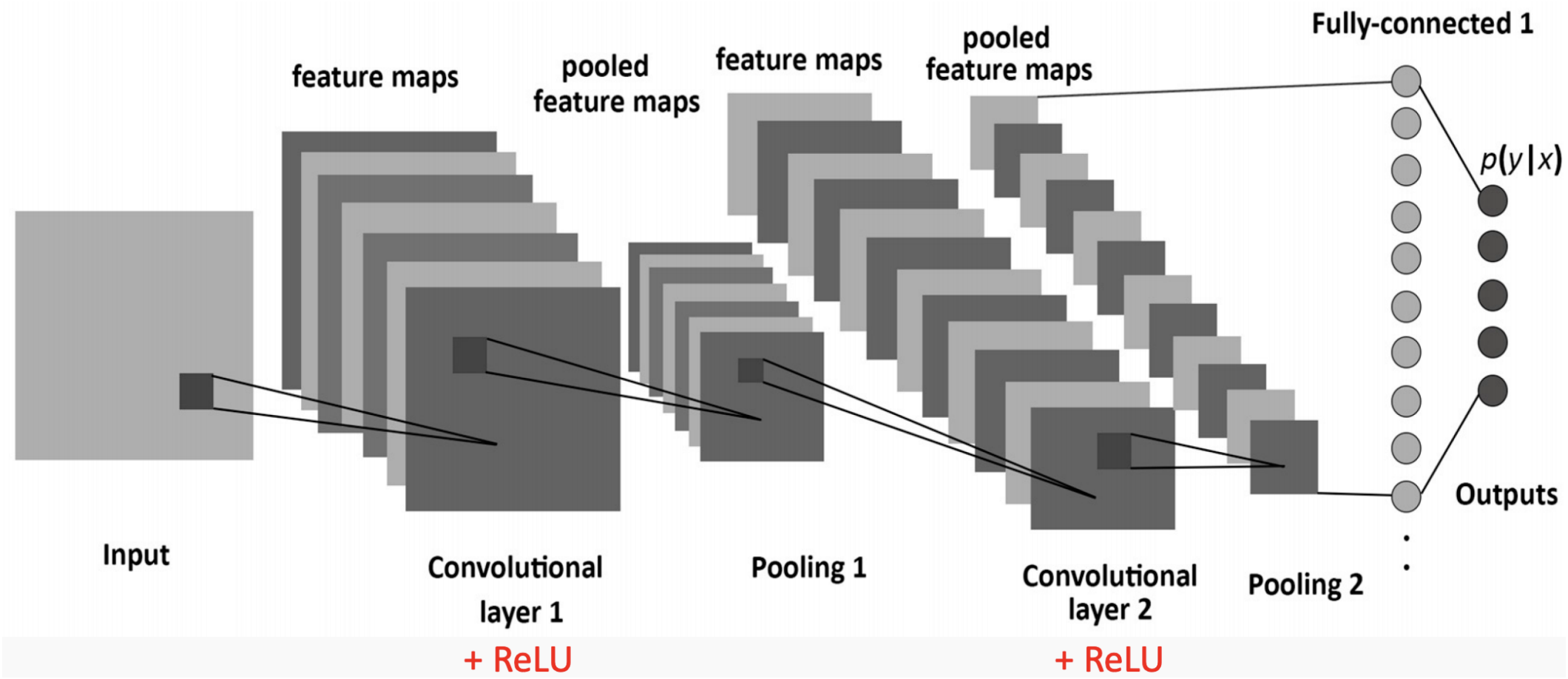
1	-1/3	1/3
1	-1/3	1/3
7/9	-1/9	1/3

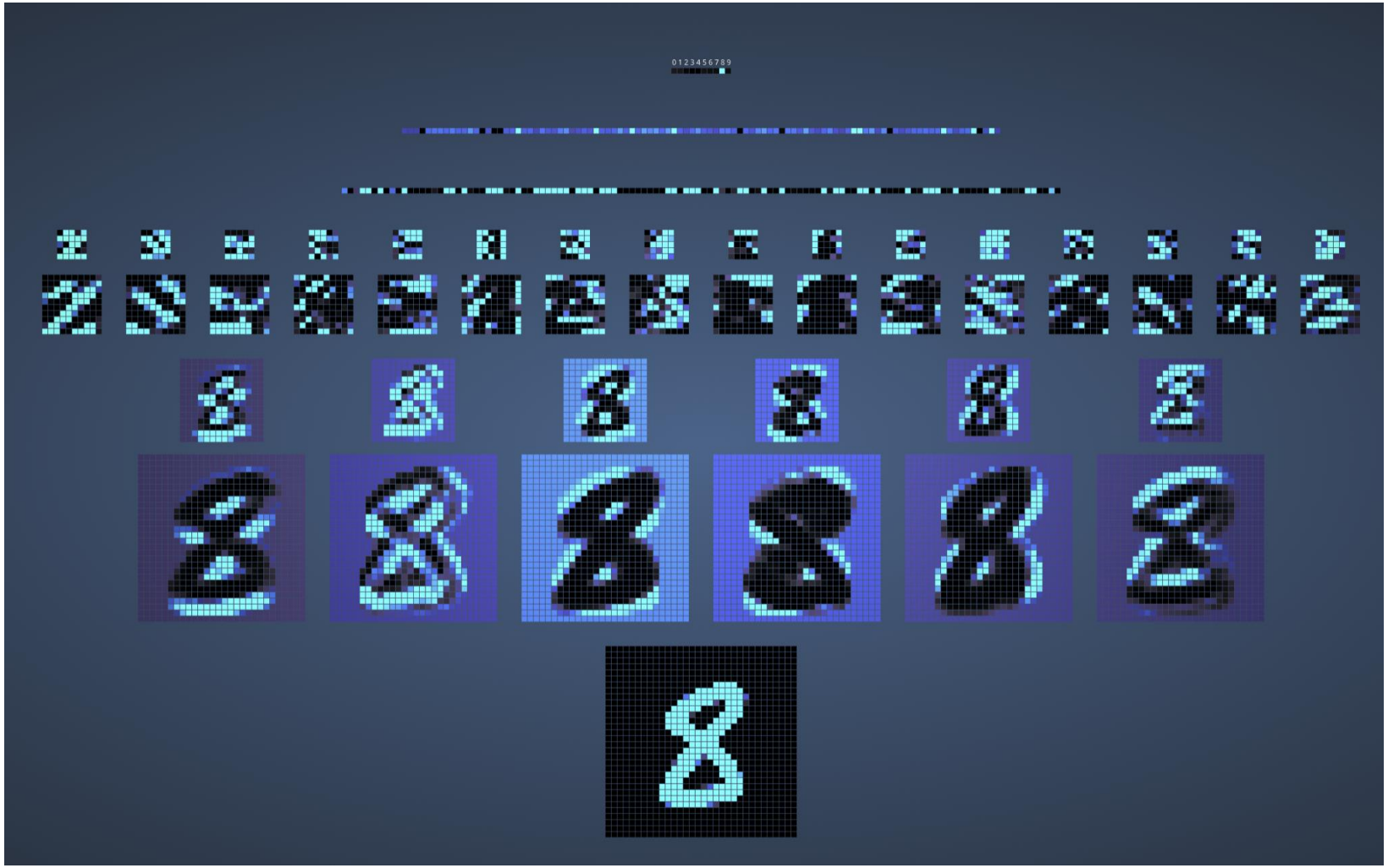


Max Pooling

1	1/3
1	1/3

Example of a CNN





https://www.cs.ryerson.ca/~aharley/vis/conv/flat.html

VGG

- 1st place @ ILSVRC-2014 localisation
- 2nd place @ ILSVRC-2014 classification

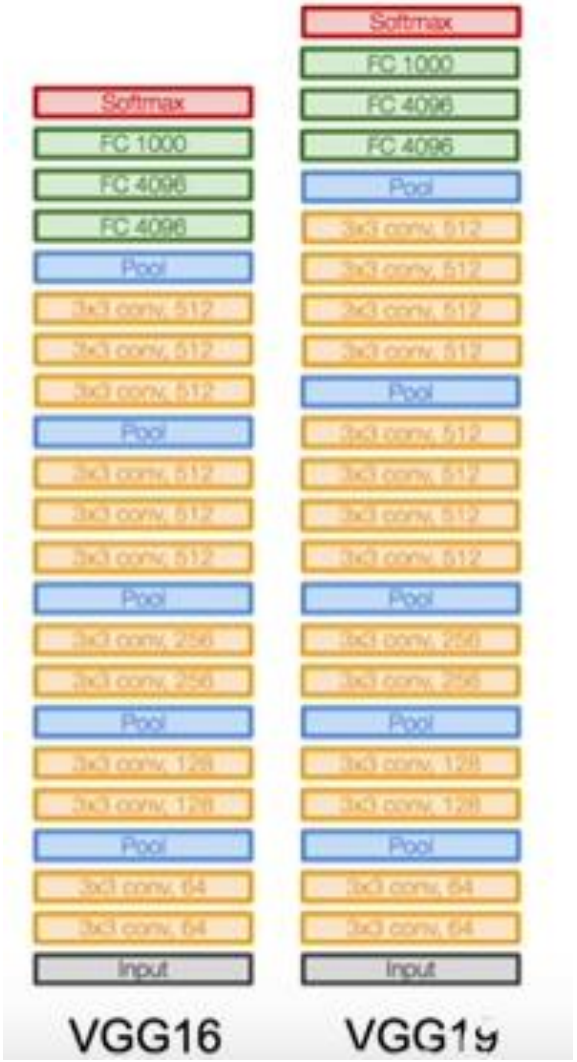
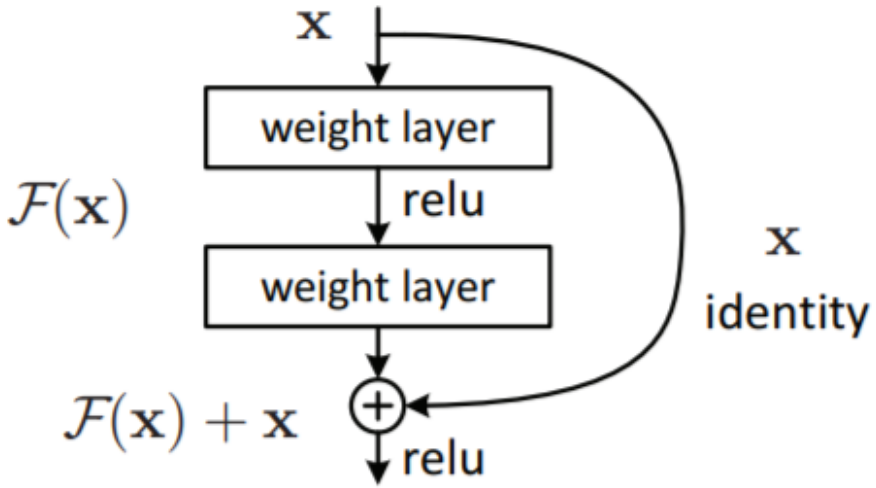


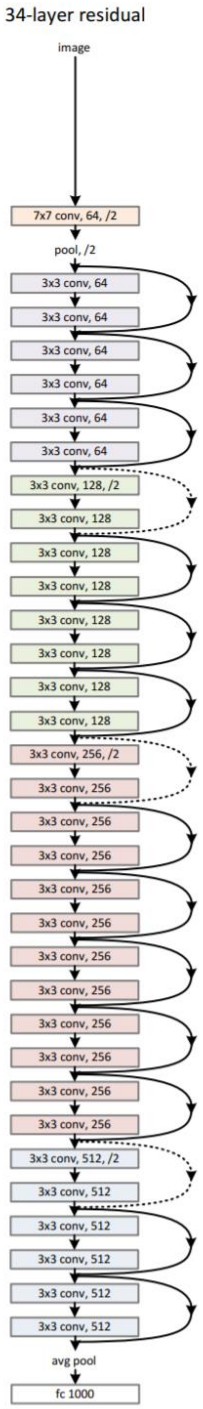
Image Source: http://cs231n.stanford.edu/

ResNet

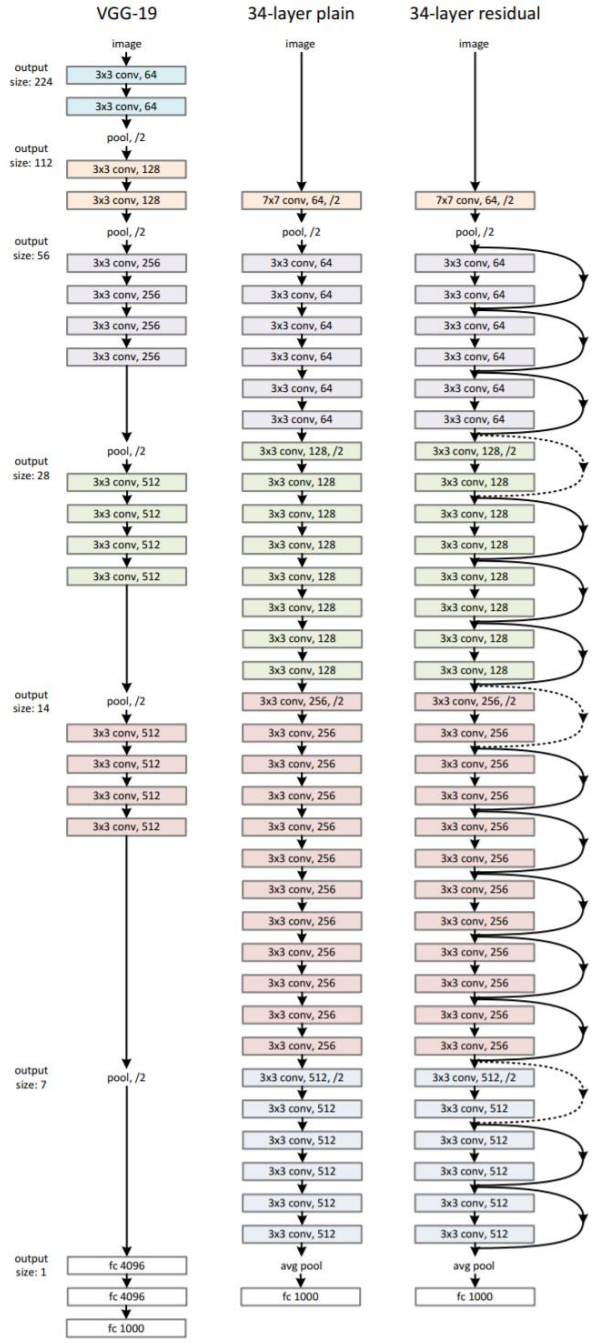
- 1st place @ ILSVRC-2014 classification



Source: https://arxiv.org/abs/1512.03385



VGG vs ResNet



Source: https://arxiv.org/abs/1512.03385