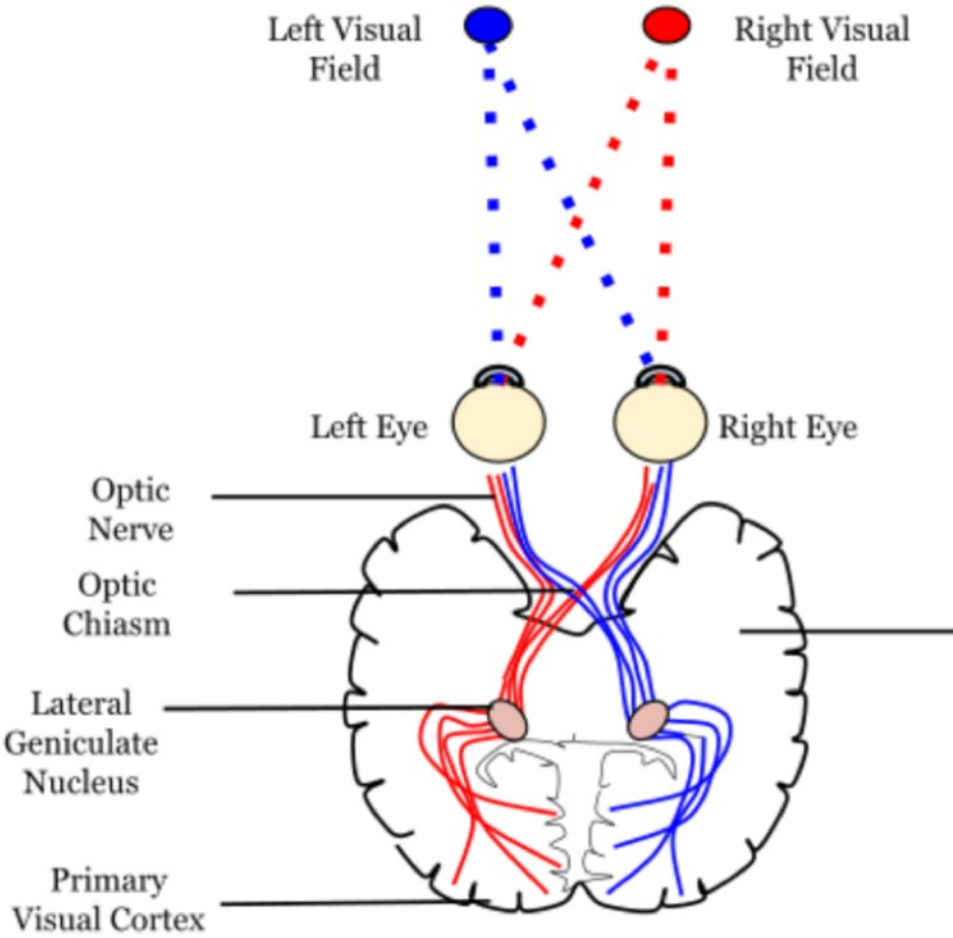
CONVOLUTIONAL NEURAL NETWORKS

Convolutions as feature extraction and network structures

INSPIRATION FROM NATURE

Main Idea

The concept is inspired by the anatomy of the visual cortex and the image representations in the brain.



Visual structure of the human brain

RESEARCH IN PHYSIOLOGY

Seminal Paper:

J. Physiol. (1959) 148, 574–591

RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

By D. H. HUBEL* and T. N. WIESEL*

From the Wilmer Institute, The Johns Hopkins Hospital and University, Baltimore, Maryland, U.S.A.

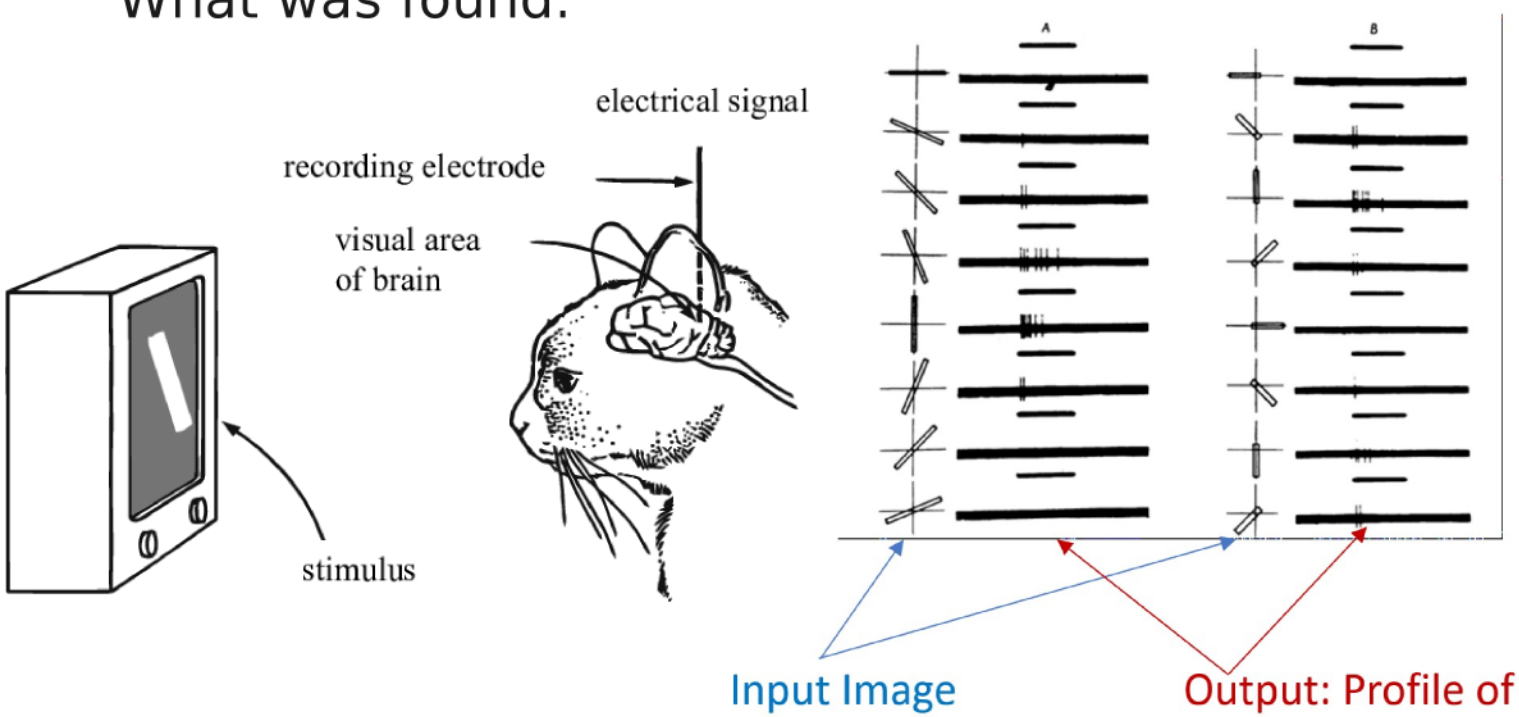
(*Received* 22 *April* 1959)

In the central nervous system the visual pathway from retina to striate cortex provides an opportunity to observe and compare single unit responses at several distinct levels. Patterns of light stimuli most effective in influencing units at one level may no longer be the most effective at the next. From differences in responses at successive stages in the pathway one may hope to gain some understanding of the part each stage plays in visual perception.

By shining small spots of light on the light-adapted cat retina Kuffler (1953) showed that ganglion cells have concentric receptive fields, with an 'on' centre and an 'off' periphery, or vice versa. The 'on' and 'off' areas within a receptive field were found to be mutually antagonistic, and a spot restricted to the centre of the field was more effective than one covering the whole receptive field (Barlow, FitzHugh & Kuffler, 1957). In the freely moving light-adapted cat it was found that the great majority of cortical cells studied gave little or no response to light stimuli covering most of the animal's visual field, whereas small spots shone in a restricted retinal region often evoked brisk responses (Hubel, 1959). A moving spot of light often produced stronger responses than a stationary one, and sometimes a moving spot gave more activation for one direction than for the opposite.

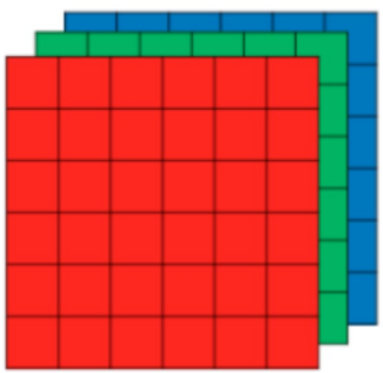
DISCOVERY IN PHYSIOLOGY

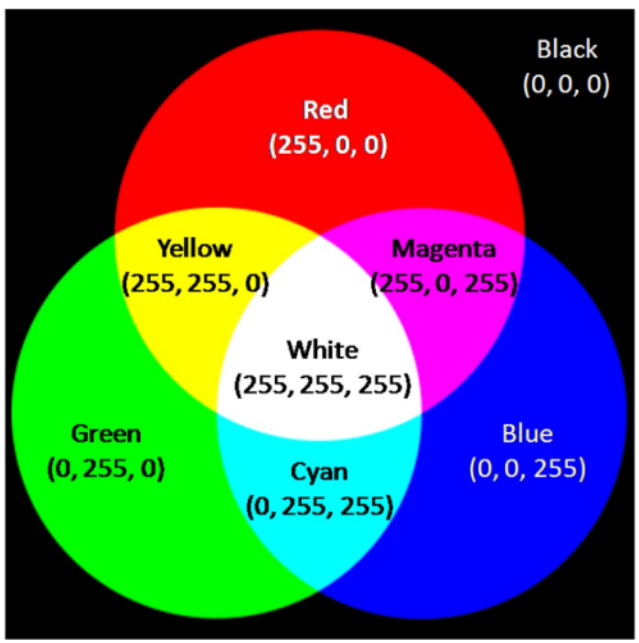
What was found:



COMPUTERS NEED NUMBERS

How can computers represent images?



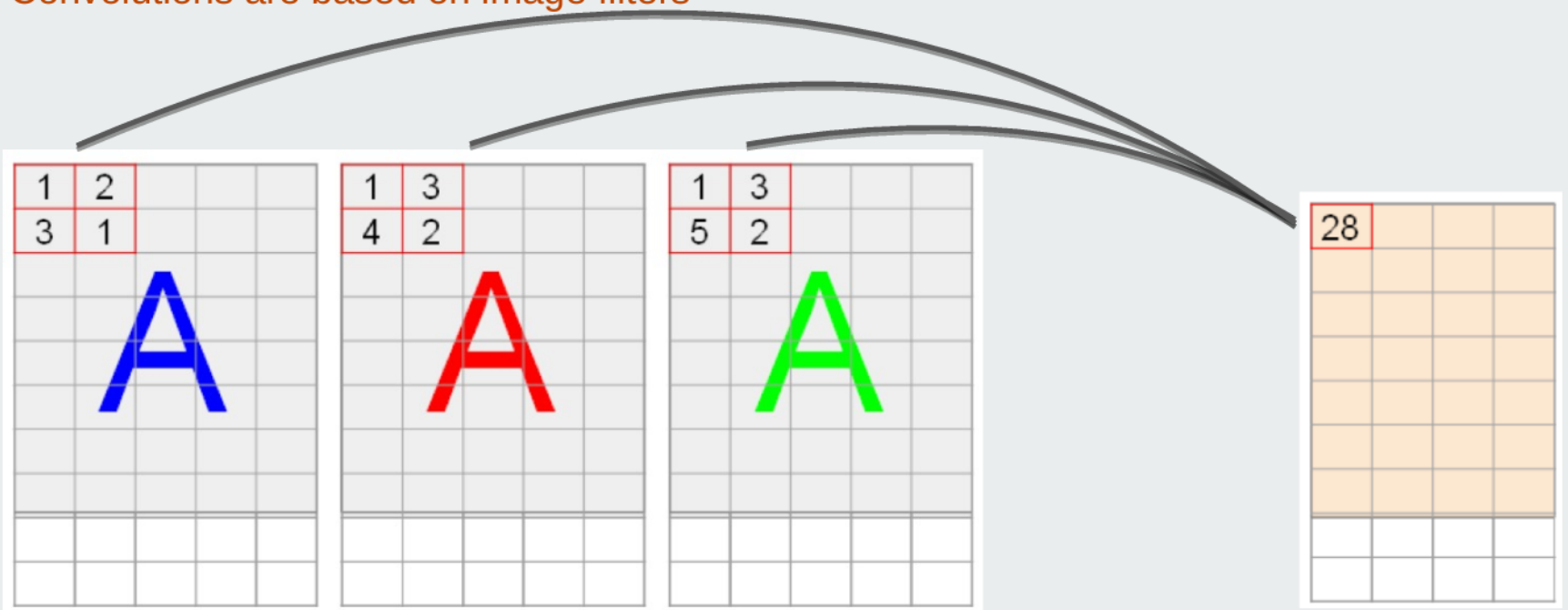


FEATURES AND MAPPINGS

How can we simulate the electrical activity in the brain?

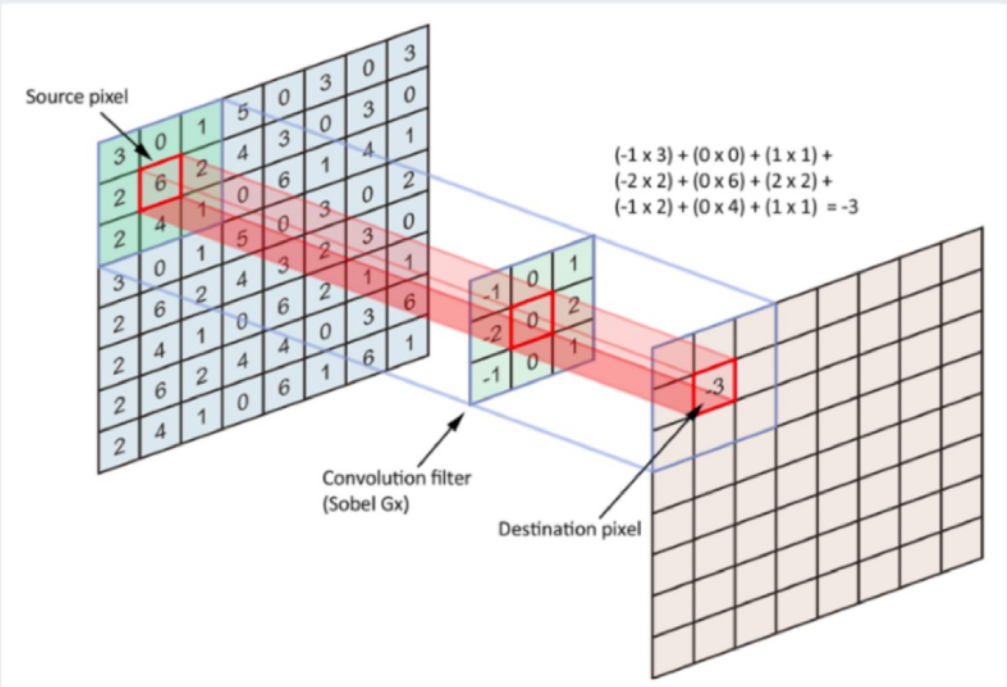
Answer: we can use numbers and special functions called “convolutions.”

Convolutions are based on image filters

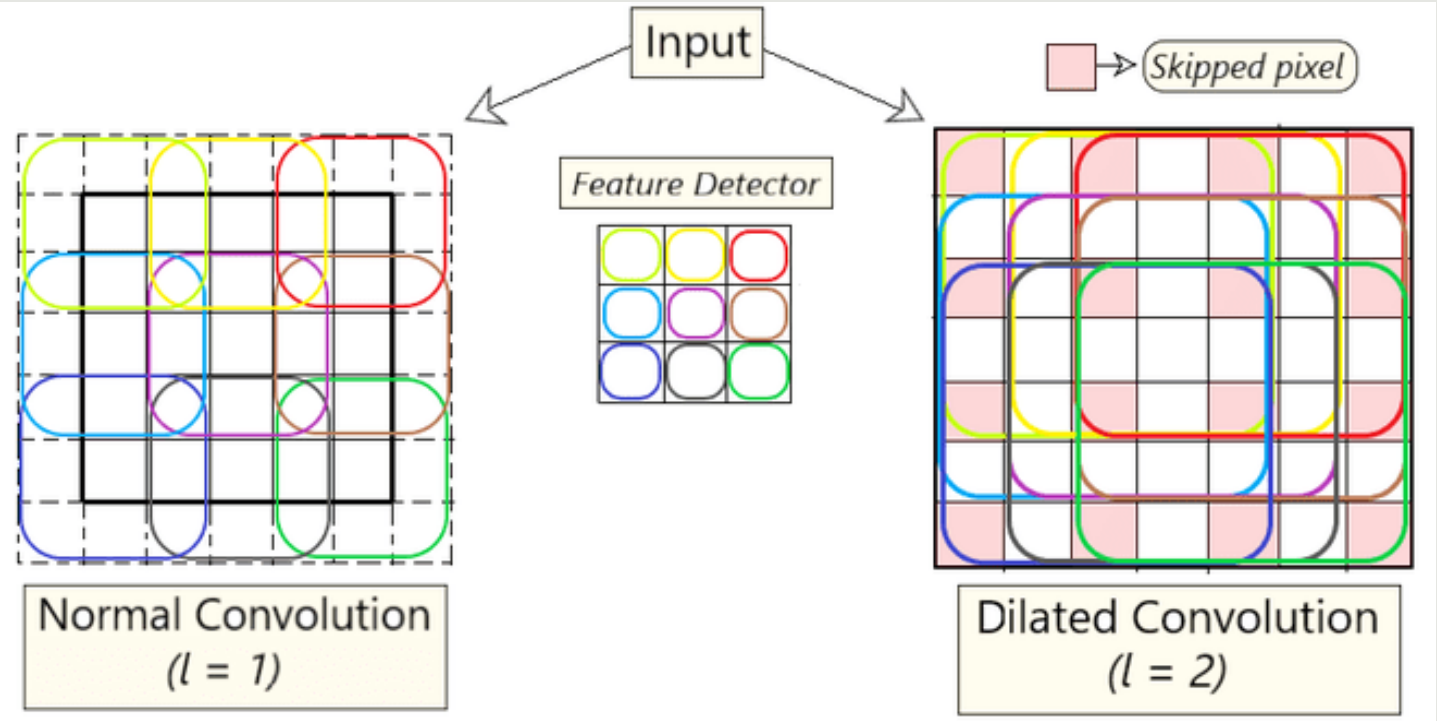


FEATURE EXTRACTION

How convolutions work:



DILATED CONVOLUTIONS



Dilated convolutions, also known as **atrous convolutions**, are a type of convolution operation where the convolutional kernel is expanded by inserting spaces (zeros) between its elements. This allows the convolution to have a larger receptive field without increasing the number of parameters or the computational cost significantly.

DILATED CONVOLUTIONS

In a standard convolution operation, a kernel (filter) slides over an input feature map, and at each position, it computes the dot product between the kernel weights and the corresponding input values. The kernel elements are applied to adjacent input pixels.

- **Kernel Size (K)**: The dimensions of the kernel (e.g., 3×3).
- **Stride (S)**: How many pixels the kernel moves after each operation.
- **Padding (P)**: Adding zeros around the input to control the output size.
- **Dilation (D)**: In standard convolution, $D = 1$.

In a dilated convolution, the kernel is "dilated" by inserting spaces between its elements. This is controlled by the **dilation rate (D)**, which specifies the spacing between kernel elements.

- **Dilation Rate (D)**: The factor by which the kernel is dilated. A dilation rate of $D = 1$ means no dilation (standard convolution).
- **Effective Kernel Size**: The actual area of the input covered by the dilated kernel.

DILATED CONVOLUTIONS

- **Kernel Size (K)**: The dimensions of the kernel (e.g., 3×3).
- **Stride (S)**: How many pixels the kernel moves after each operation.
- **Padding (P)**: Adding zeros around the input to control the output size.
- **Dilation (D)**: In standard convolution, $D = 1$.

In a dilated convolution, the kernel is "dilated" by inserting spaces between its elements. This is controlled by the **dilation rate (D)**, which specifies the spacing between kernel elements.

Mathematically, for a 2D input, the output $y[i, j]$ of a dilated convolution is calculated as:

$$y[i, j] = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} x[i + D \cdot m, j + D \cdot n] \cdot w[m, n]$$

- x: Input feature map.
- w: Kernel weights.
- K: Kernel size.
- D: Dilation rate.

DILATED CONVOLUTIONS IN PYTORCH

```python
import torch
import torch.nn as nn

# Define a dilated convolutional layer
conv_layer = nn.Conv2d(
    in_channels=16,
    out_channels=32,
    kernel_size=3,
    stride=1,
    padding=2,   # Adjust padding to maintain output size
    dilation=2   # Dilation rate
)

# Input tensor with shape (batch_size, in_channels, height, width)
input_tensor = torch.randn(1, 16, 32, 32)

# Output tensor
output_tensor = conv_layer(input_tensor)

print(f"Output shape: {output_tensor.shape}")
```

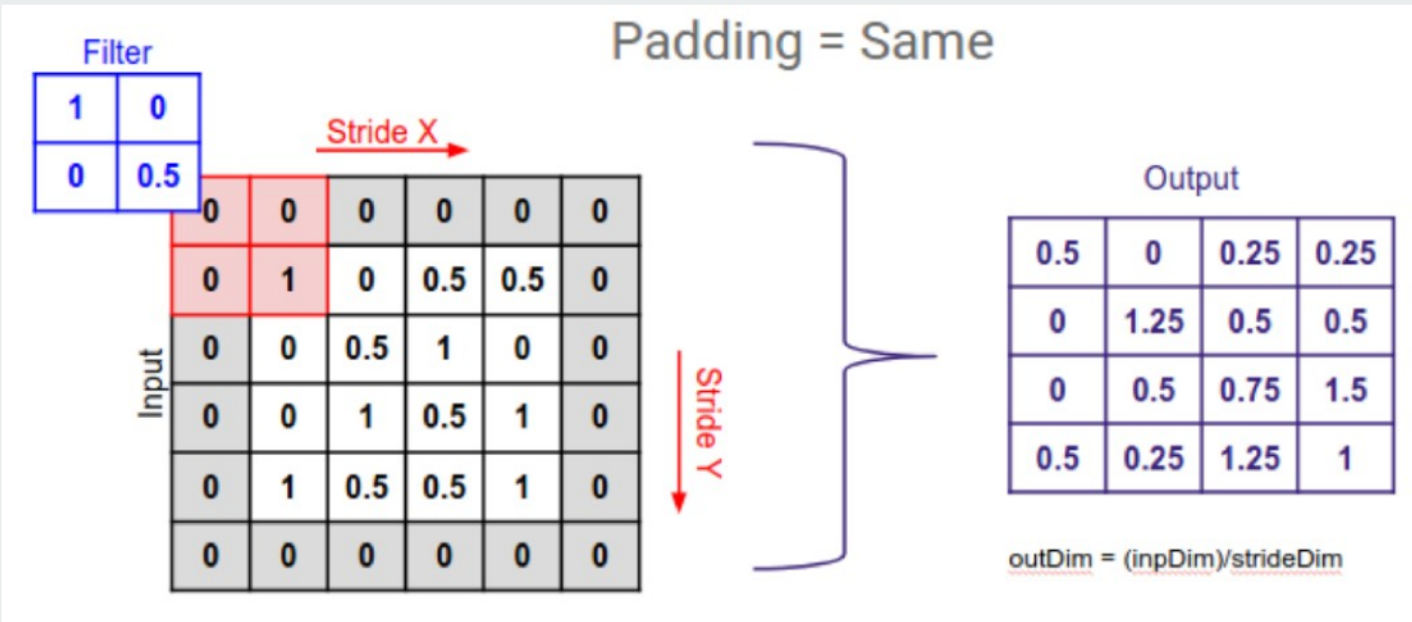
Note on Padding:

- To maintain the output size, padding should be set to $P = D$.
- If you want to compute the padding needed to maintain the output size:

$$P = D \times \frac{K - 1}{2}$$

PADDING

Padding needs to be considered:



BASICS OF 2D CONVOLUTIONS

In PyTorch, a 2D convolutional layer is created using **`nn.Conv2d`**. The key parameters are:

- **`in_channels`**: Number of channels in the input image.
- **`out_channels`**: Number of channels produced by the convolution.
- **`kernel_size`**: Size of the convolving kernel (filter).
- **`stride`**: Stride of the convolution (how the filter moves over the input).
- **`padding`**: Zero-padding added to both sides of the input.
- **`dilation`**: Spacing between kernel elements.

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} + 2 \times \text{Padding} - \text{Dilation} \times (\text{Kernel Size} - 1) - 1}{\text{Stride}} + 1 \right\rfloor$$

CALCULATING OUTPUT DIMENSIONS

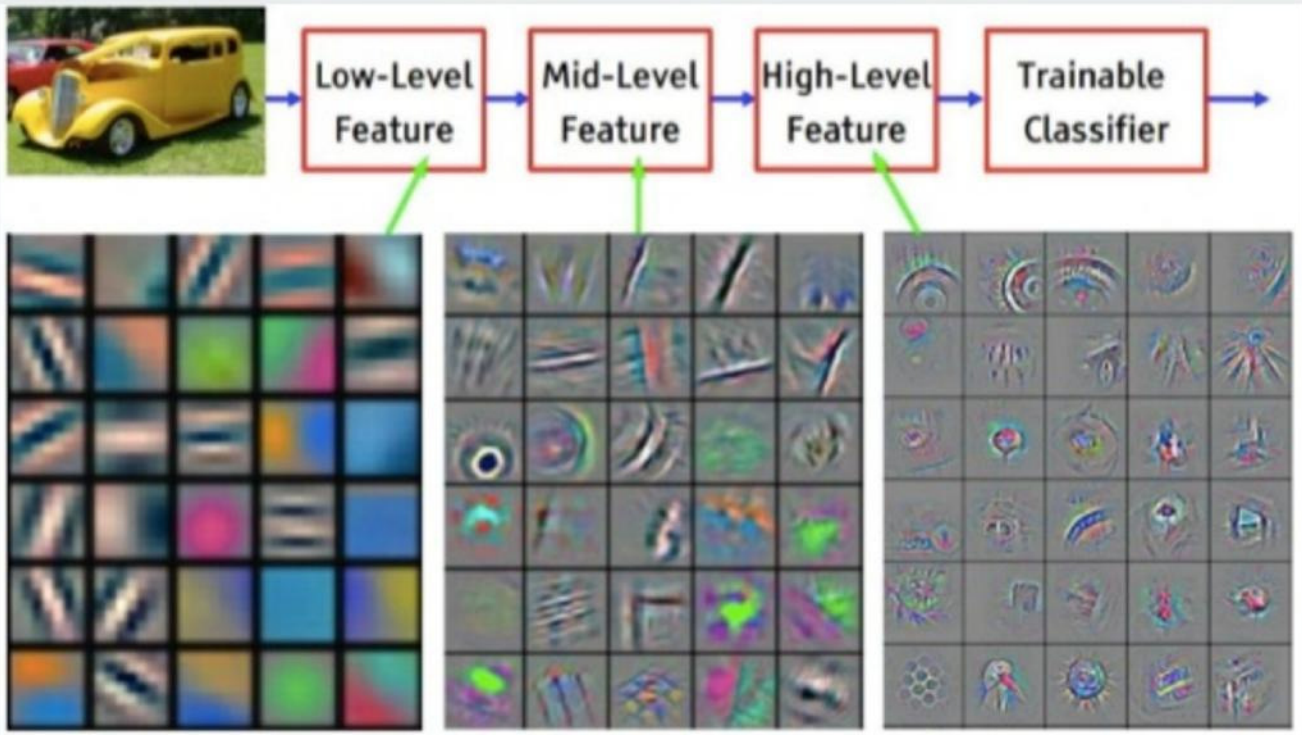
The output dimensions of a convolutional layer can be calculated using the following formula for each spatial dimension (height and width):

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} + 2 \times \text{Padding} - \text{Dilation} \times (\text{Kernel Size} - 1) - 1}{\text{Stride}} + 1 \right\rfloor$$

- **Input Size (H_{in} or W_{in})**: The size of the input tensor along height or width.
- **Padding (P)**: The number of pixels added to each side of the input.
- **Kernel Size (K)**: The size of the convolution kernel.
- **Stride (S)**: The step size with which the kernel moves over the input.
- **Dilation (D)**: The spacing between elements in the kernel.

INTUITION

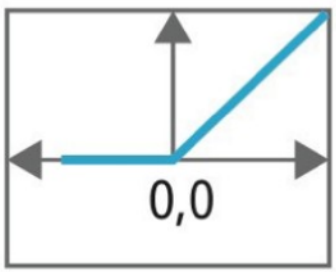
Intuitive example:



RELU

What happens in an activation layer:

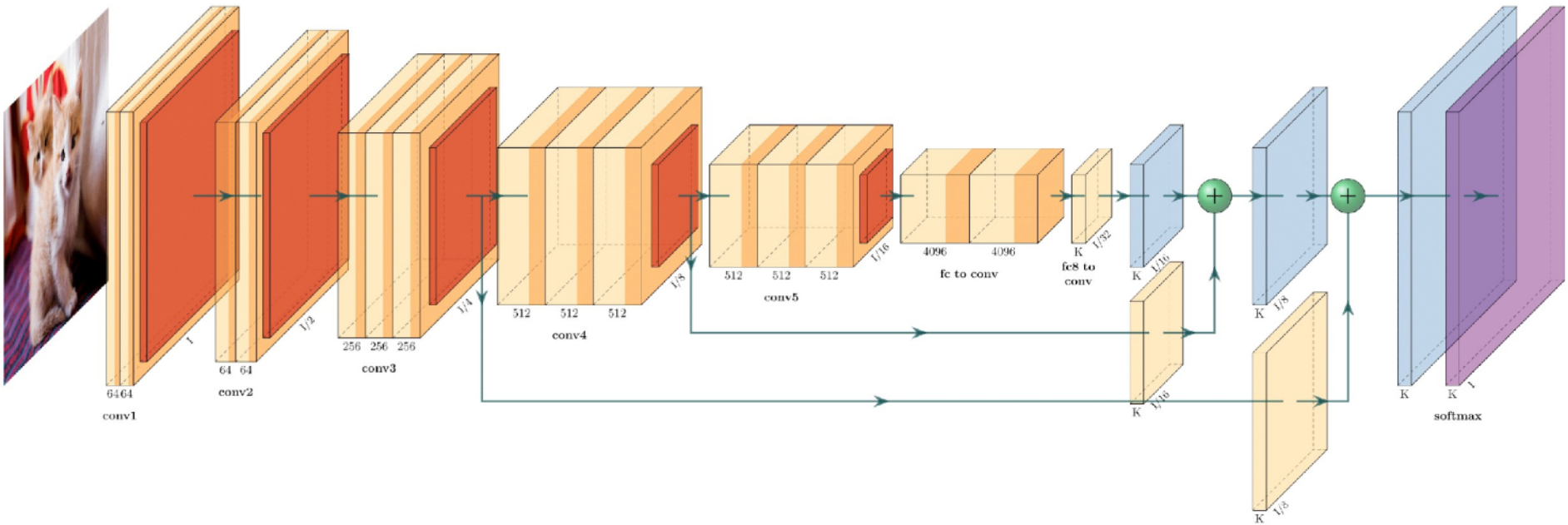
15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23



15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

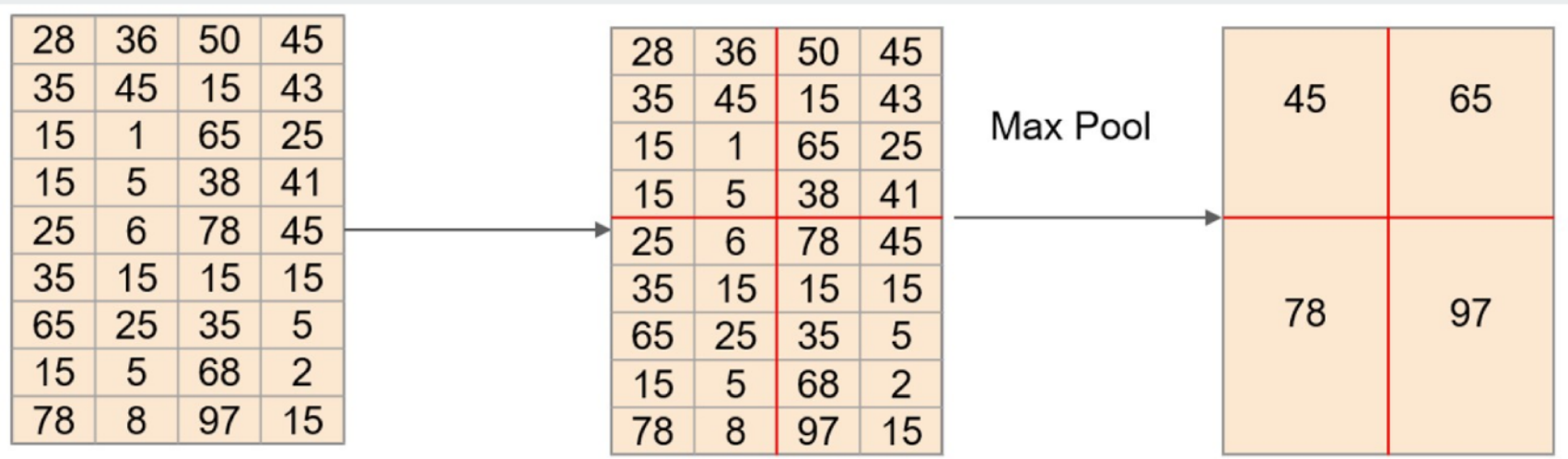
NETWORK STRUCTURE

Example for the neural network architecture:

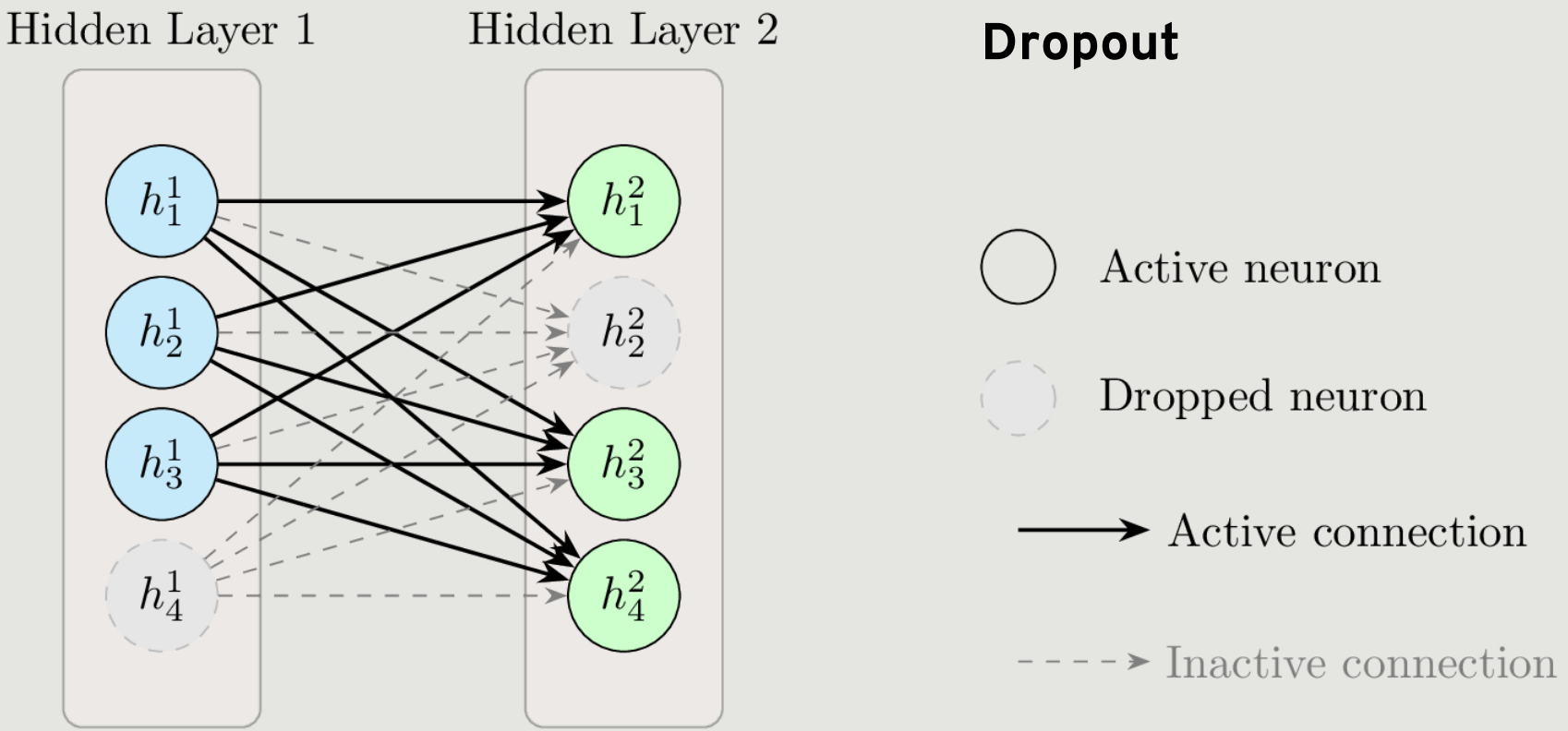


MEMORY EFFICIENCY

Pooling:



HEURISTICS AGAINST OVERFITTING

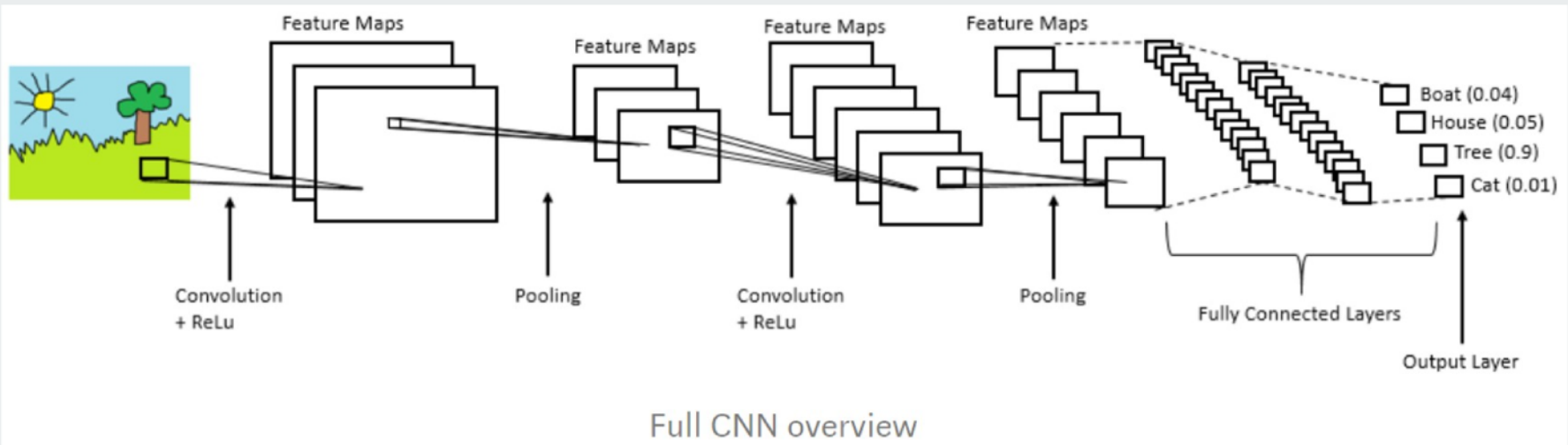


Dropout temporarily disables random neurons (not connections) during each training iteration. During training, each neuron has some probability (often 0.5 for hidden layers) of being "dropped out" - meaning its output is temporarily set to zero for that specific training pass. The neurons and their connections remain intact, but their contribution is zeroed out for that particular forward and backward pass.

This forces the network to learn more robust features because it can't rely on any single neuron always being present.

PUTTING THINGS TOGETHER

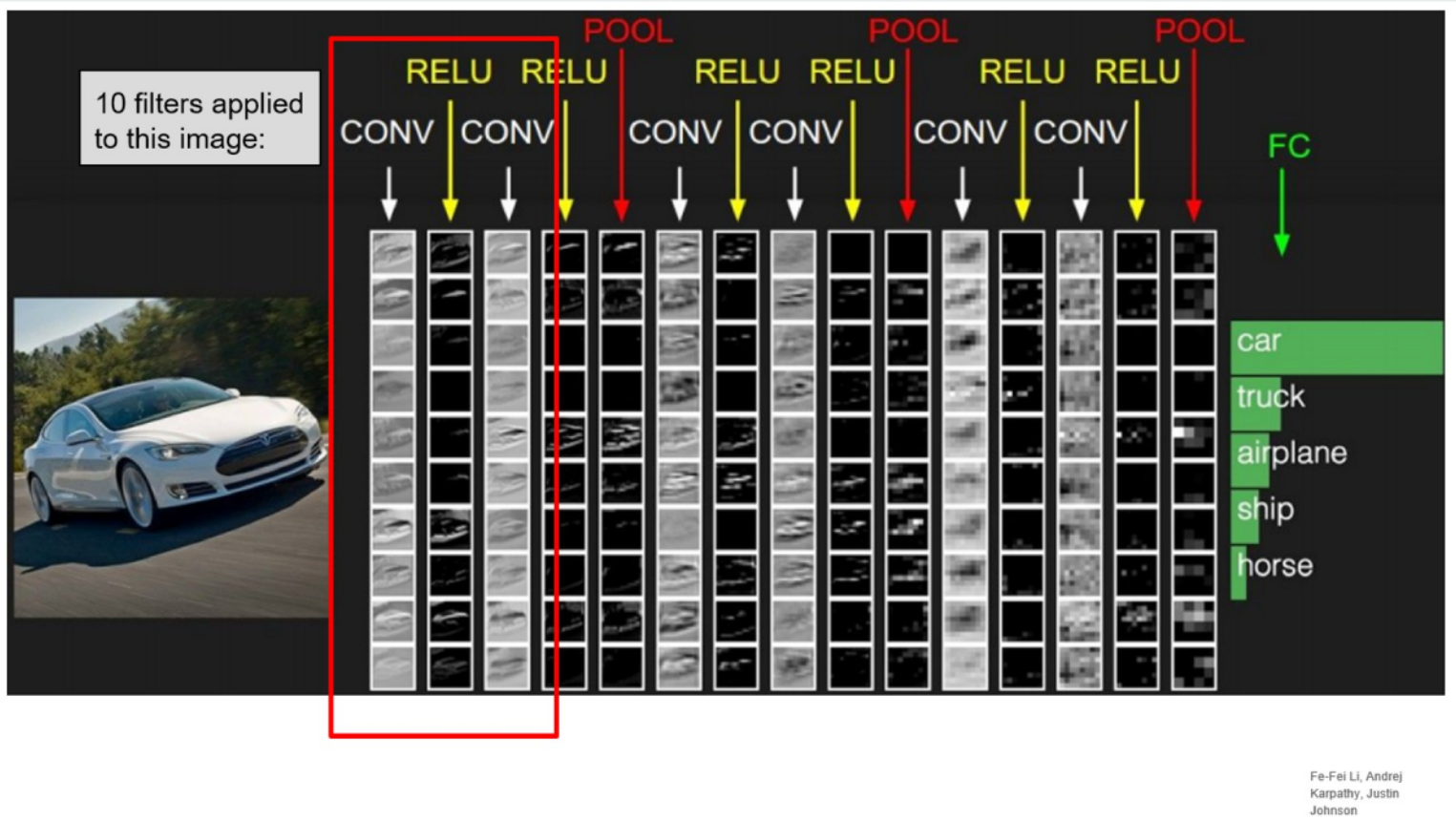
The anatomy of a convolutional neural network:



Full CNN overview

PUTTING THINGS TOGETHER

More intuition:



APPLICATIONS

Applications include:

- Self-driven cars and industrial robots.
- Colorization (reference)
- Sound recognition (reference).
- Games (reference).
- Image analysis (reference).
- Art: (reference).