

# Tópicos Avanzados en Analítica

## Maestría en Analítica para la Inteligencia de Negocios

Sergio Alberto Mora Pardo - H2 2024

# **Deep Computer Vision**

# **Convolutional Neural Networks**

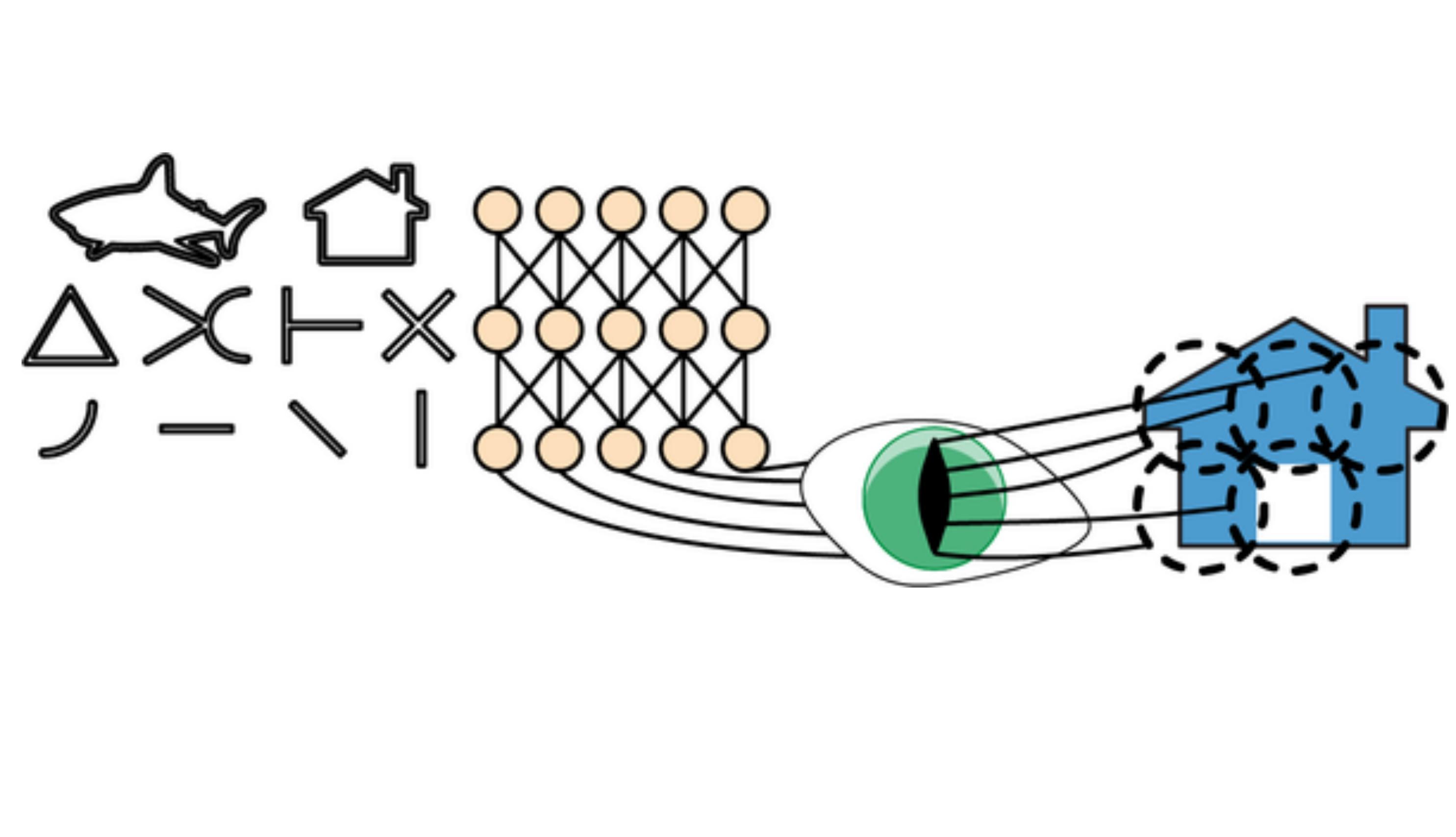


# Deep Computer Vision

## Convolutional Neural Network

### Visual Cortex

*Biological neurons in the visual cortex respond to specific patterns in small regions of the visual field called receptive fields.*



# Deep Computer Vision

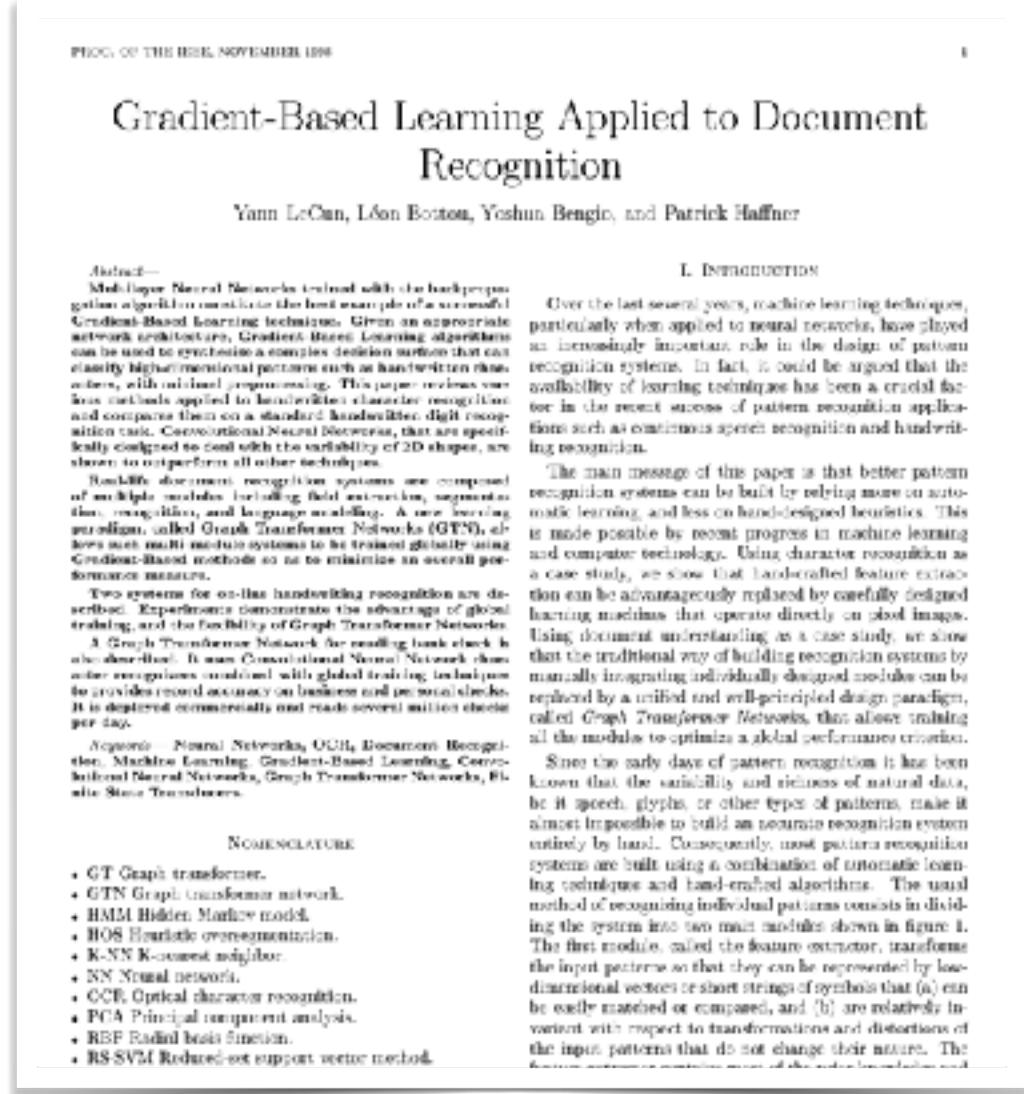
## Convolutional Neural Network

### Visual Cortex

*Biological neurons in the visual cortex respond to specific patterns in small regions of the visual field called receptive fields.*



**Abstract:** A neural network model for a mechanism of visual pattern recognition is proposed in this paper. The network is self-organized by "learning without a teacher", and acquires ability to recognize stimulus patterns based on the geometrical similarity (Euclid) of their shapes without affected by their position. This network is given a nickname "neocognitron". After completion of self-organization, the network has a structure similar to the hierarchy model of the visual nervous system proposed by Hubel and Wiesel. The network consists of an input layer (photoreceptor array), followed by a cascade connection of a number of modular structures, each of which is composed of two layers of cells connected in a cascade. The first layer of each module consists of "S-cell", which shows characteristics similar to simple cells or lower order hypercomplex cells, and the second layer consists of "G-cells" similar to complex cells or higher order hypercomplex cells. The afferent synapses to each S-cell have plasticity and are modifiable. The network has an ability of unsupervised learning. We do not need any "teacher" during the process of self-organization, and it is only needed to present a set of stimulus patterns repeatedly to the input layer of the network. The network has been simulated on a digital computer. After repetitive presentation of a set of stimulus patterns, each stimulus pattern has become to



**Abstract:** Multi-layer Neural Networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. In an attempt to understand the mechanism of Gradient-Based Learning, we can use it to examine a samples decision surface that can classify handwritten patterns such as handwritten digits, letters, and characters. This paper also shows how gradient descent methods applied to handwritten character recognition and compares them on a standard handwriting digit recognition task. Convolutional Neural Networks, that are specific fully connected layers capable of learning features of 2D shapes, are shown to outperform all other methods.

The main message of this paper is that better pattern recognition systems can be built by applying gradient-based learning, and less ad-hoc hand-tailored techniques. This is made possible by recent progress in machine learning and computer technology. Using character recognition as a case study, we show that hand-tailored feature extraction can be advantageously replaced by carefully designed learning machines that operate directly on pixel images. Using document understanding as a case study, we show that the traditional way of building recognition systems by manually ingesting individually designed modules can be replaced by a unified and well-structured design paradigm, called *Graph Transformer Networks*, that allows training all the modules to optimize a global performance criterion.

**NOMENCLATURE**

- GT Graph transformer.
- GTN Graph transformer network.
- HMM Hidden Markov model.
- RBF Radial basis function.
- K-NN K-nearest neighbor.
- NN Neural network.
- OCR Optical character recognition.
- PCA Principal component analysis.
- RBF Radial basis function.
- RS-SVM Reduced-set support vector method.

Kunihiko Fukushima  
Spring 1980

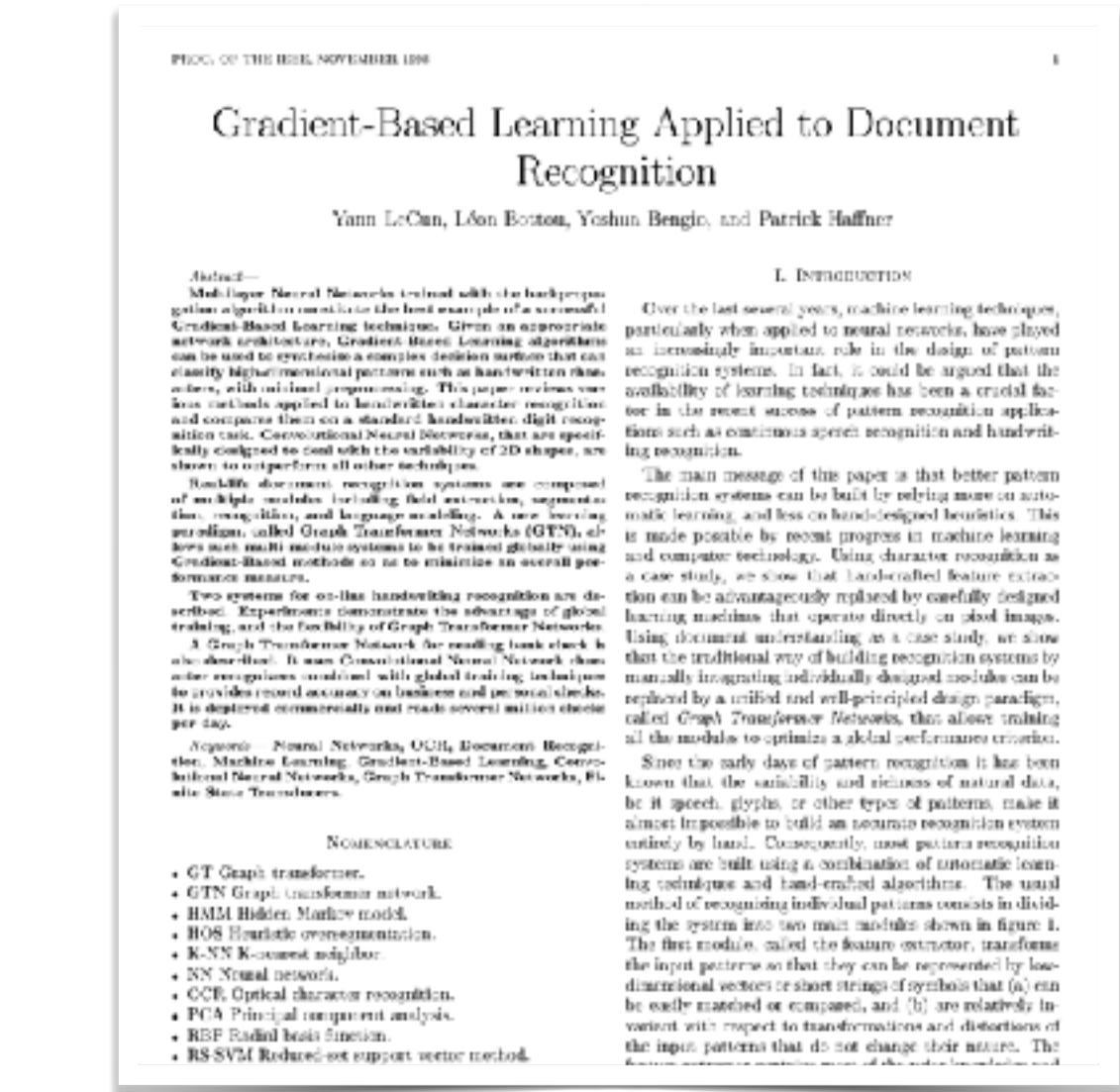
Yann Le Cun et al.  
November 1998

# Deep Computer Vision

## Convolutional Neural Network

### Visual Cortex

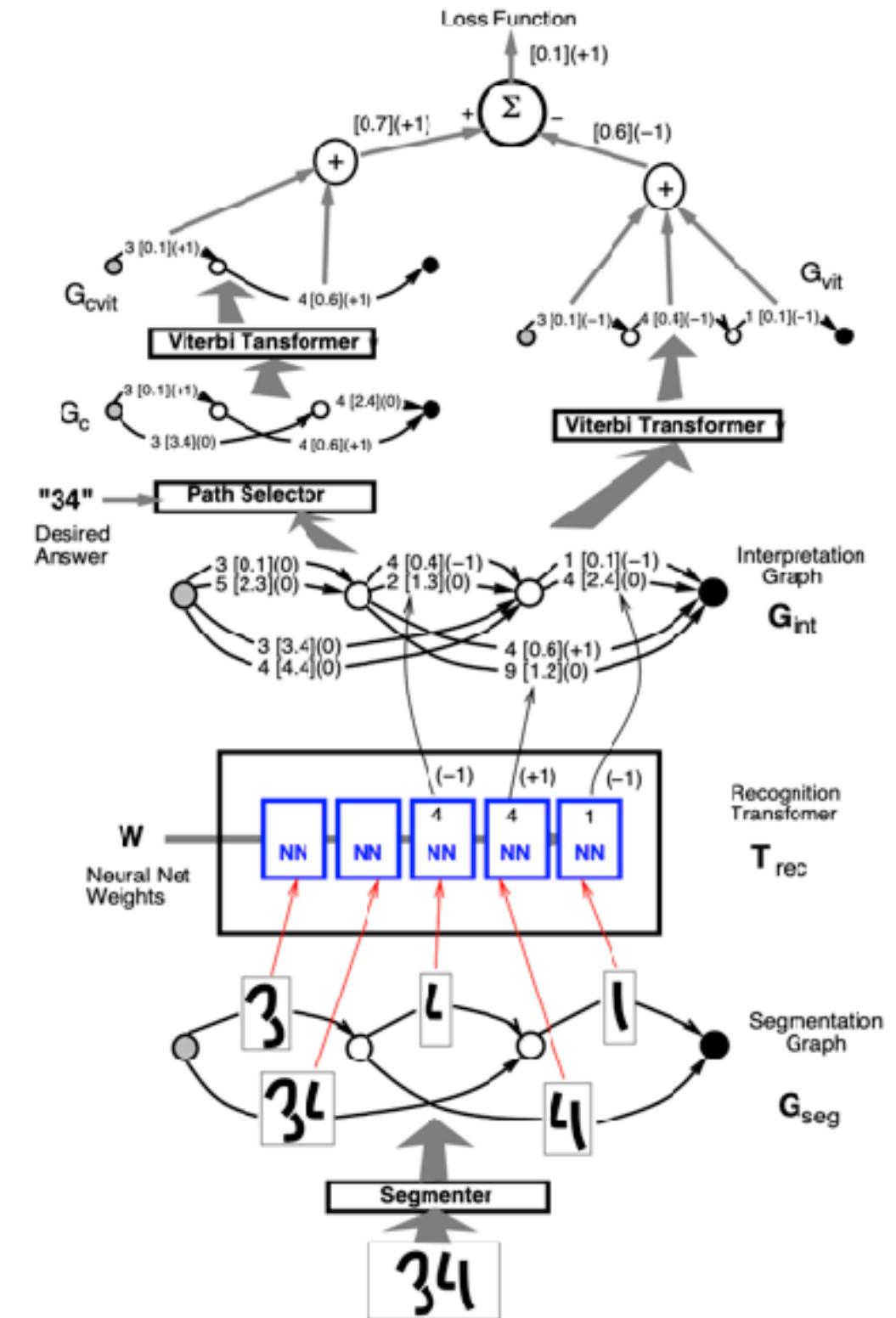
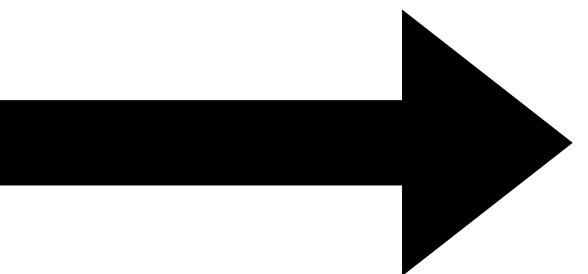
*Biological neurons in the visual cortex respond to specific patterns in small regions of the visual field called receptive fields.*



Yann Le Cun et al.

November 1998

### LeNet-5

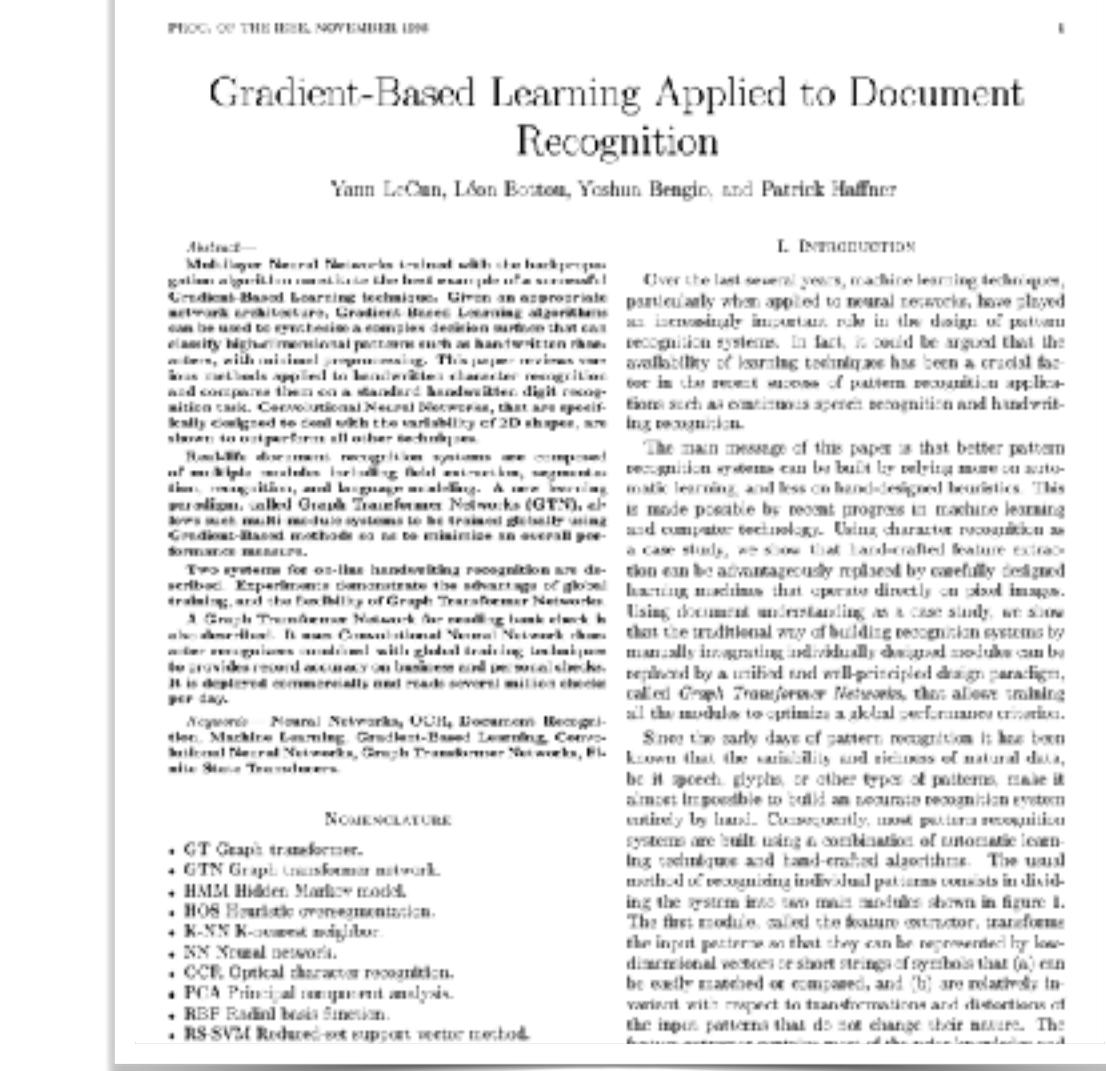


# Deep Computer Vision

## Convolutional Neural Network

### Visual Cortex

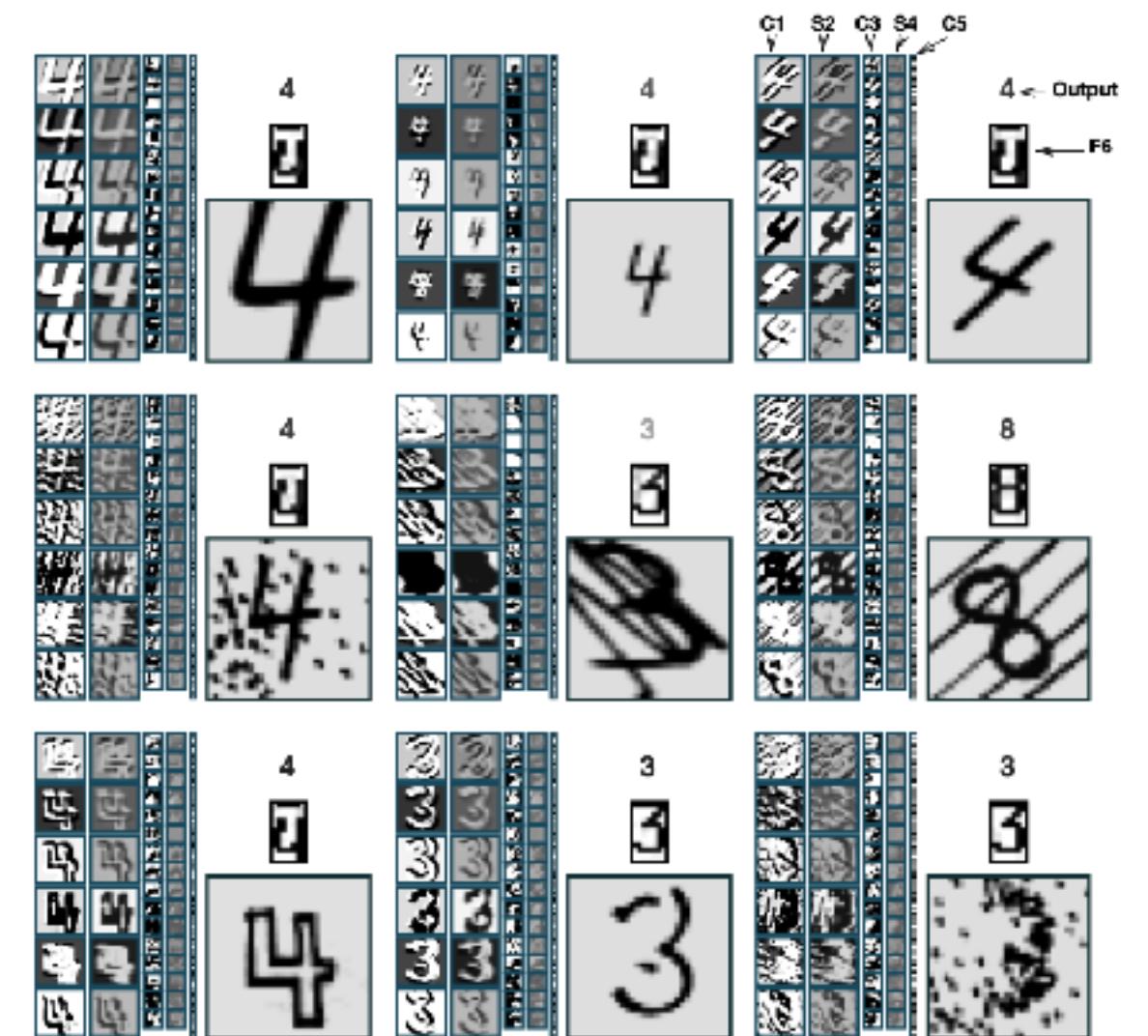
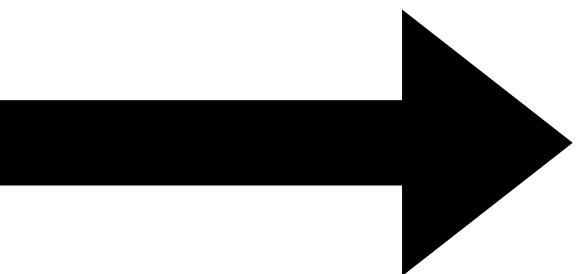
*Biological neurons in the visual cortex respond to specific patterns in small regions of the visual field called receptive fields.*



Yann Le Cun et al.

November 1998

### LeNet-5



# Deep Computer Vision

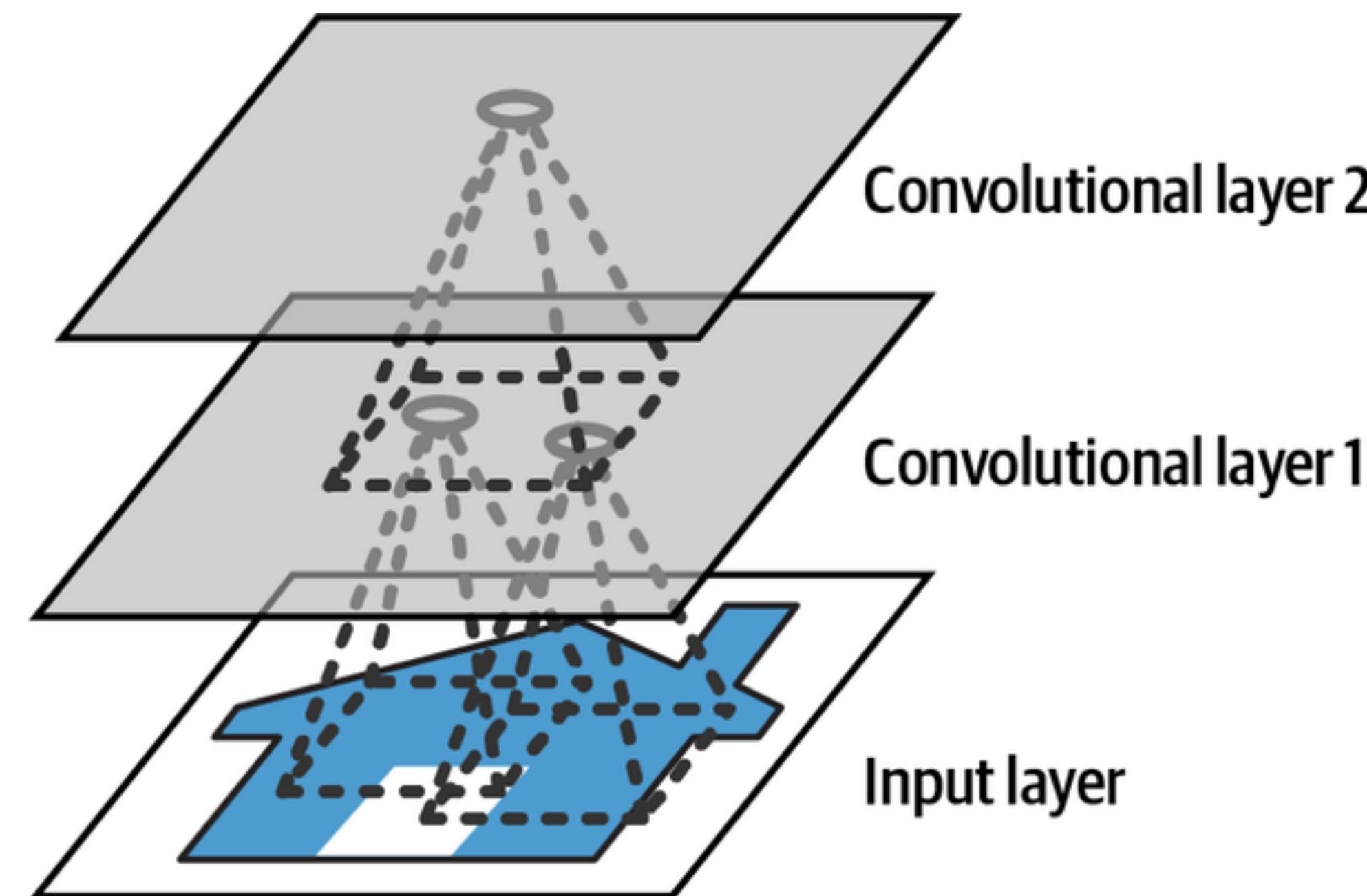
## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*

*This architecture allows the network to focus on small, low-level features in the first hidden layer,*



# Deep Computer Vision

## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*

### Padding:

$f_h$  : height

$i$  to  $i + f_h - 1$

Rows

$f_w$  : width

$j$  to  $j + f_w - 1$

Columns

# Deep Computer Vision

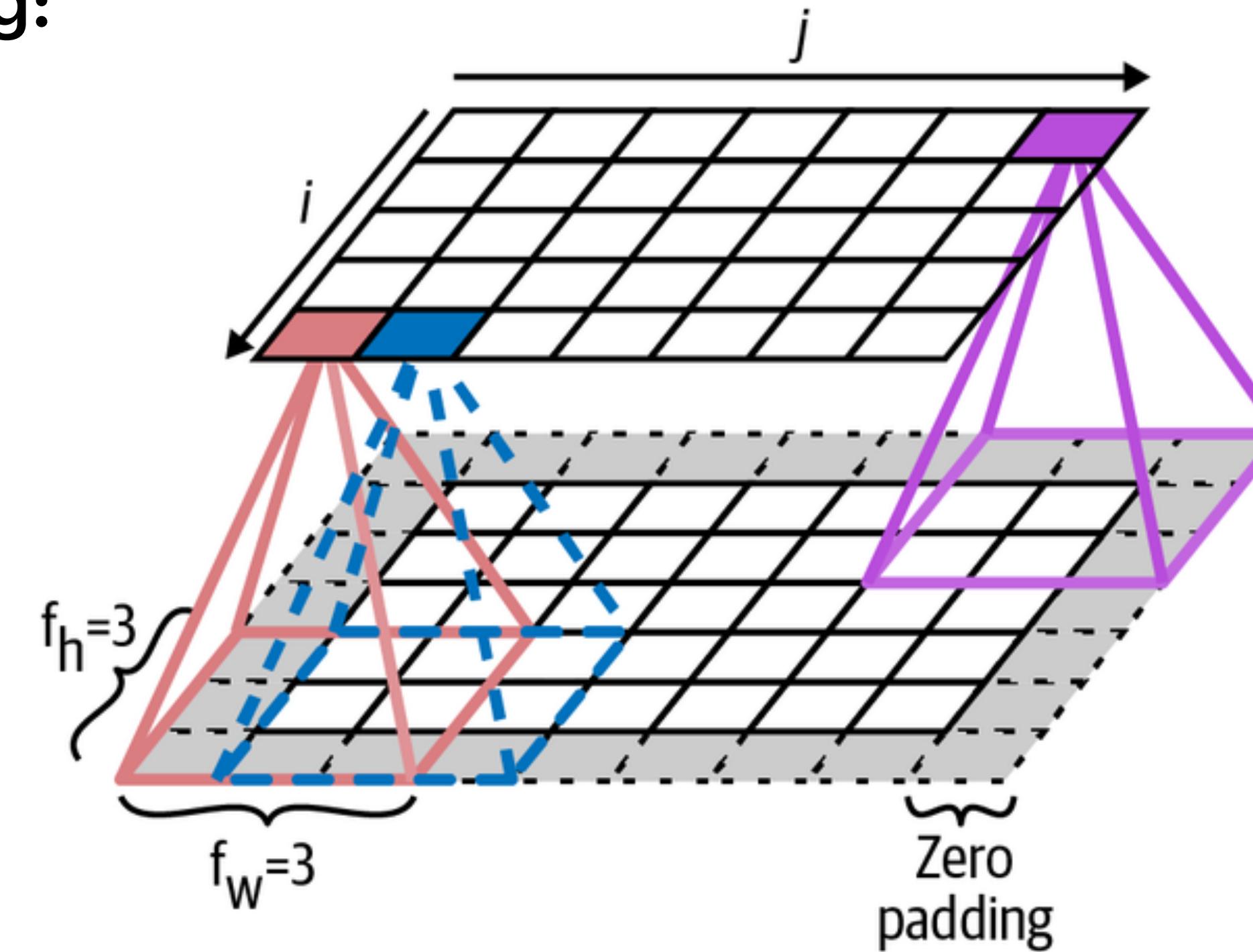
## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*

### Padding:



# Deep Computer Vision

## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer.*

### Stride:

$f_h$  : height

$s_h$  : vertical

$i \cdot s_h$  to  $i \cdot s_h + f_h - 1$

Rows

$f_w$  : width

$s_w$  : horizontal

$j \cdot s_w$  to  $j \cdot s_w + f_w - 1$

Columns

# Deep Computer Vision

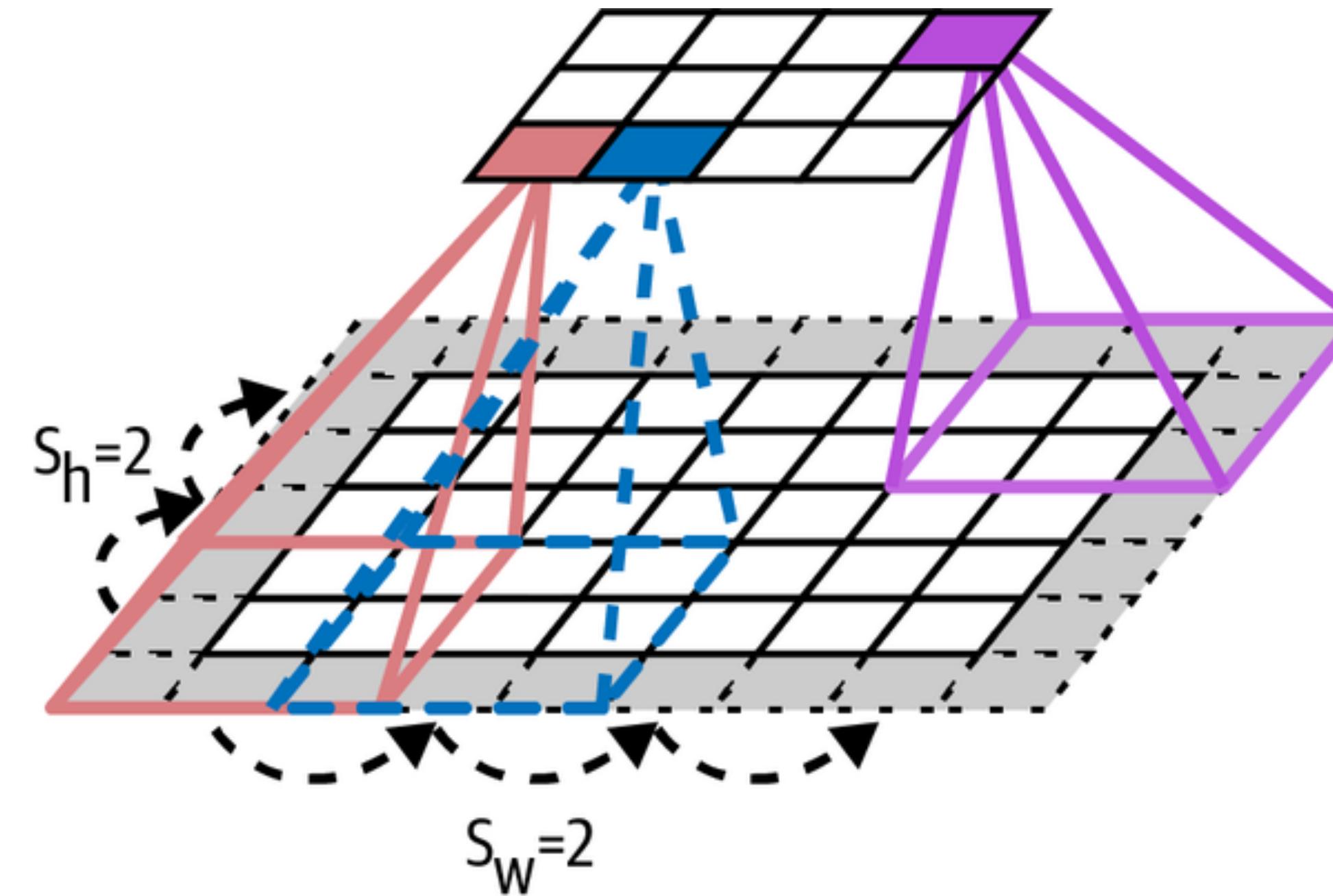
## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*

### Stride (steps):



# Deep Computer Vision

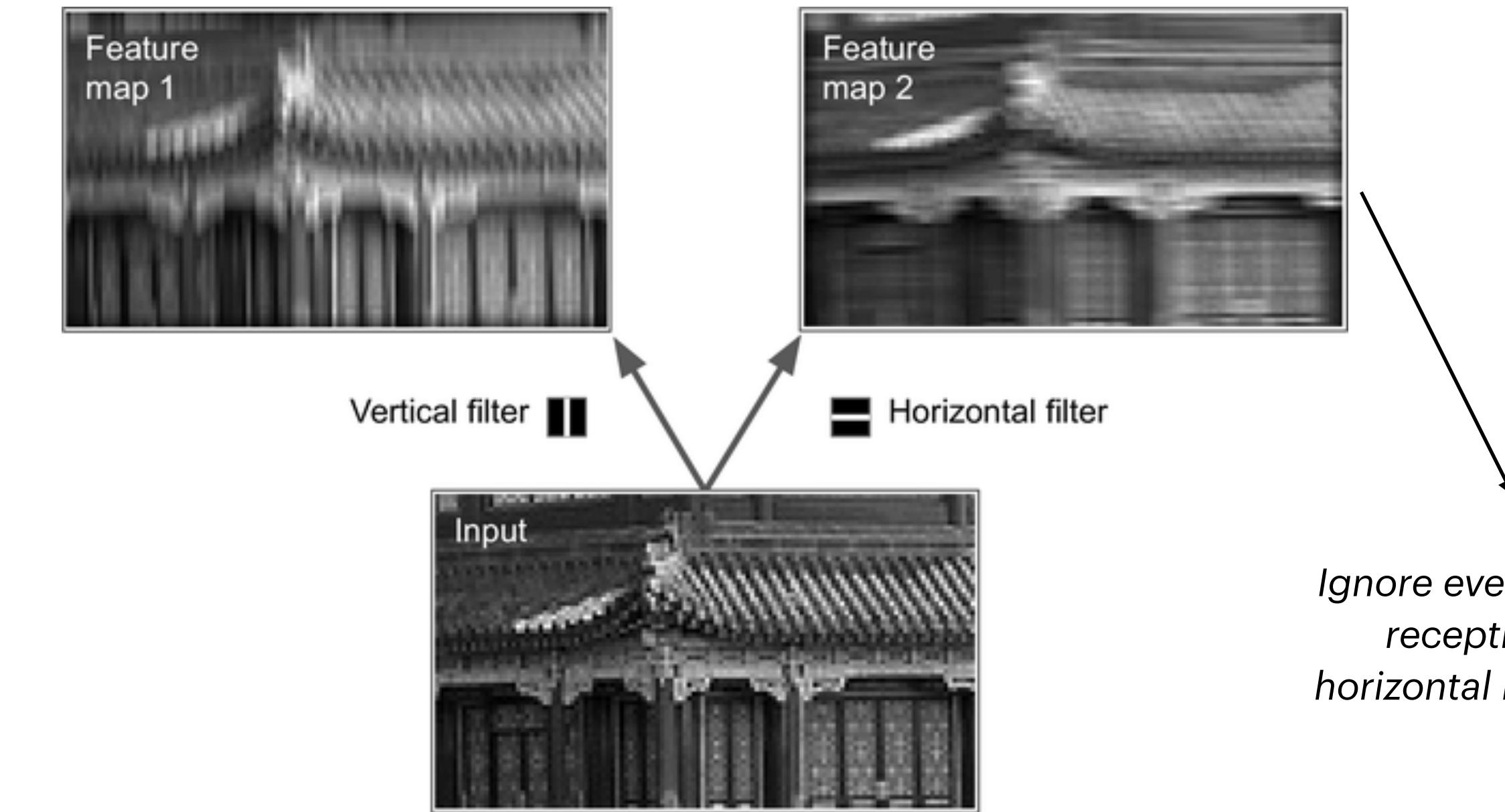
## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*

**Filters:** “Convolution kernels” or “kernels”



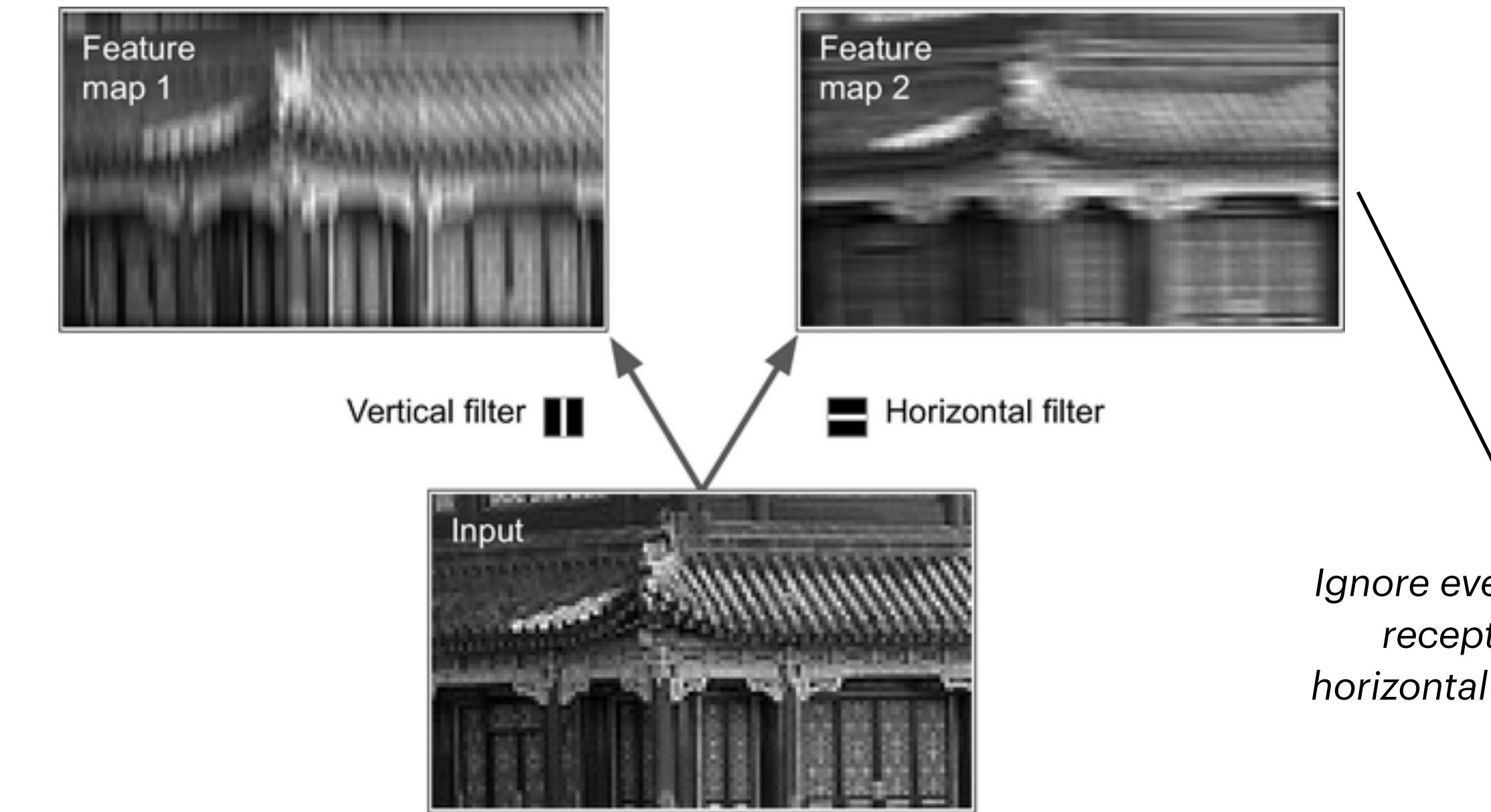
# Deep Computer Vision

## Convolutional Neural Network

### Feature map

A layer full of neurons using the same filter outputs a “feature map”

**Filters:** “Convolution kernels” or “kernels”



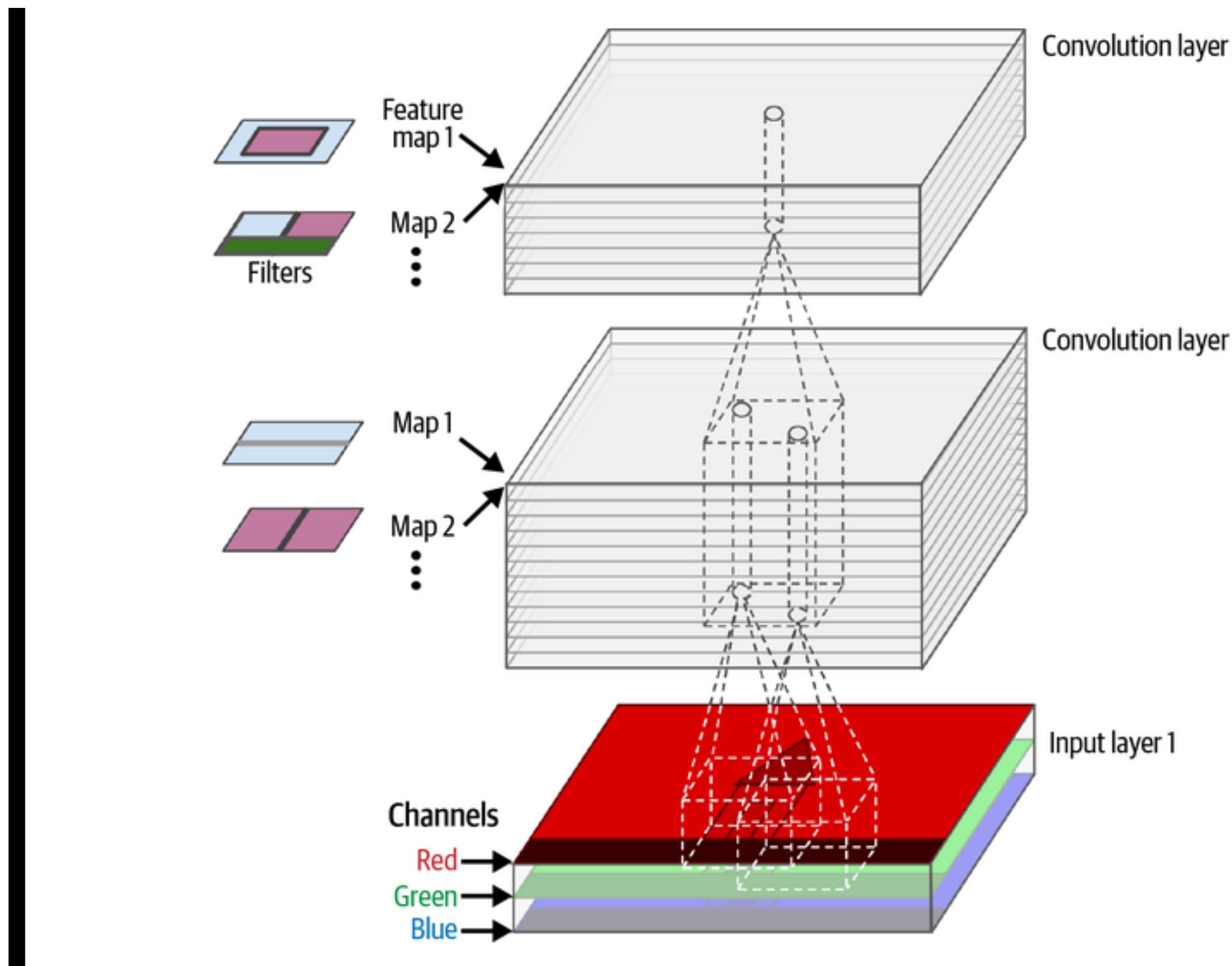
Ignore everything in their receptive field for horizontal line (white line)

# Deep Computer Vision

## Convolutional Neural Network

Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*

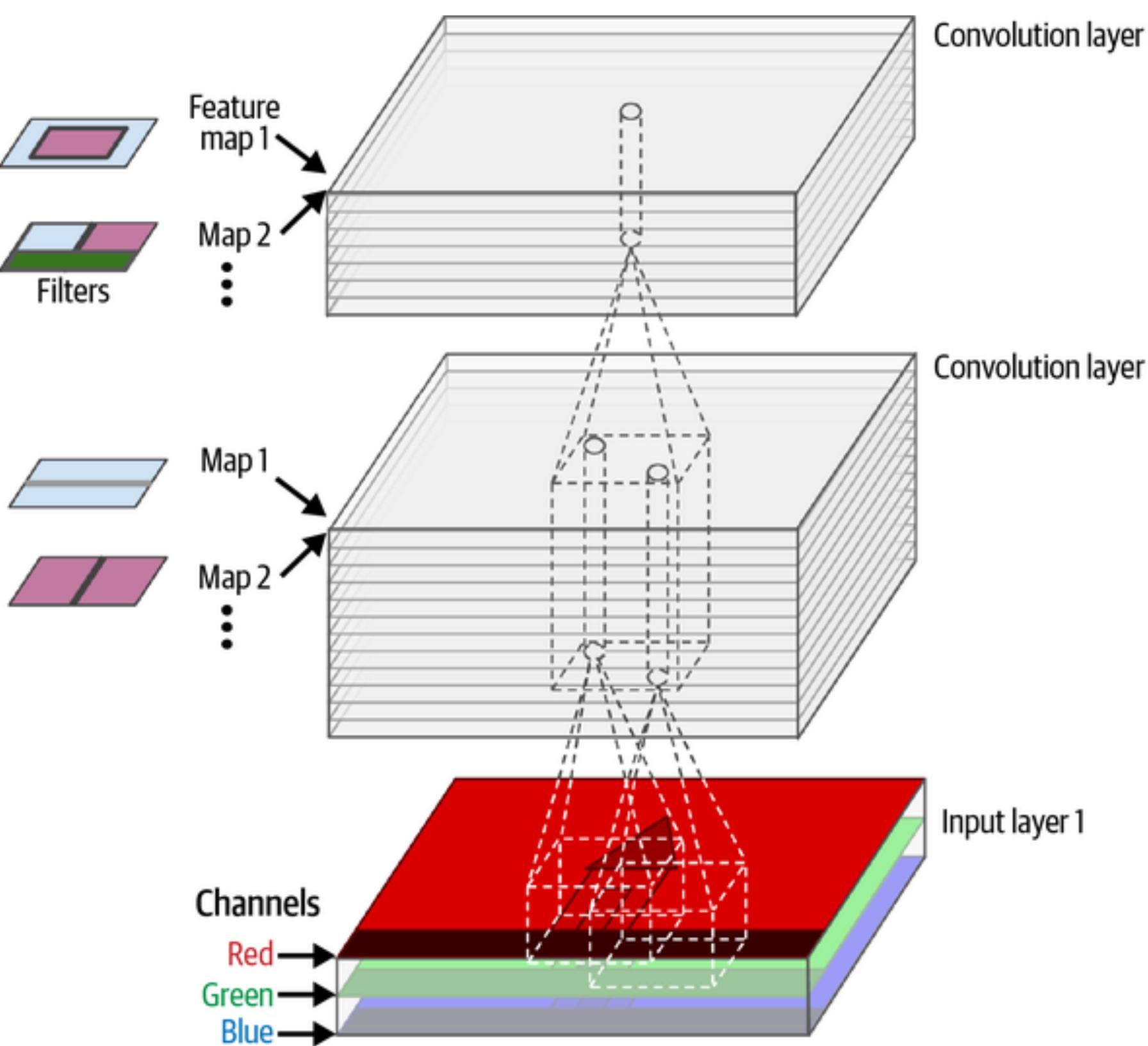


# Deep Computer Vision

## Convolutional Neural Network

Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*

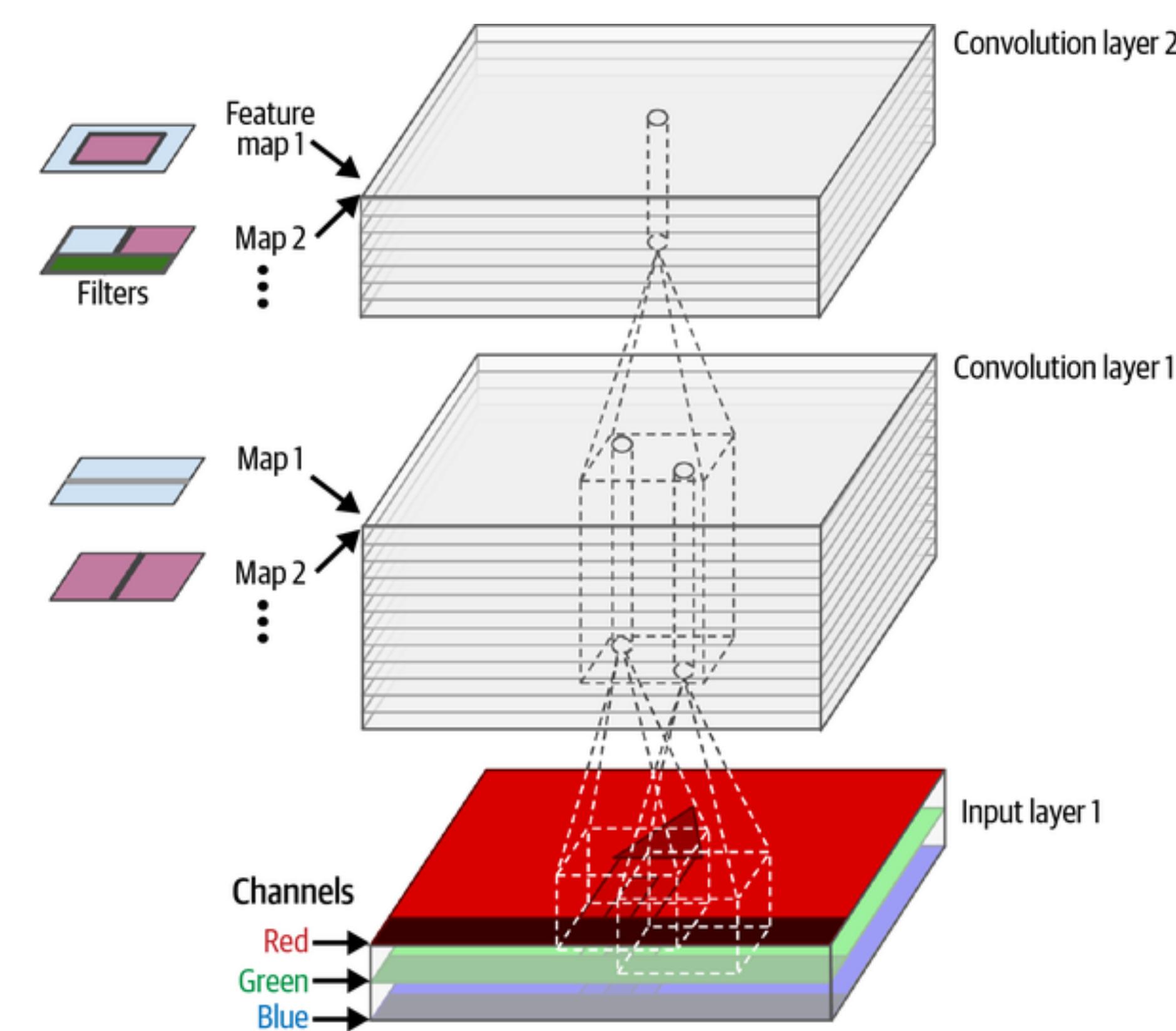


# Deep Computer Vision

## Convolutional Neural Network

### Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*



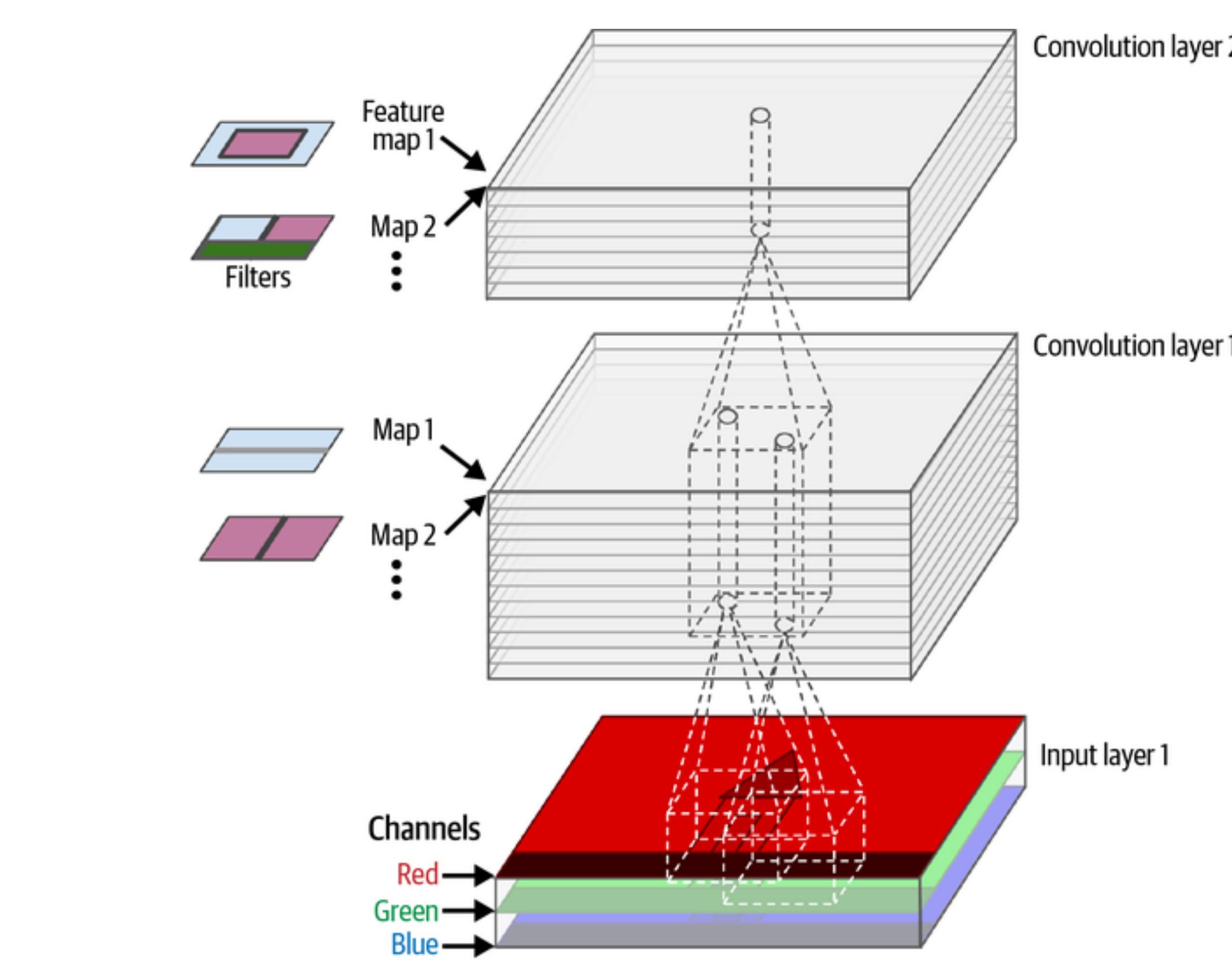
- 1 neuron : 1 Pixel
  - Between feature maps we have different params and bias
- ↓
- The fact that all neurons in a feature map share the same parameters drastically reduces the number of parameters in the model

# Deep Computer Vision

## Convolutional Neural Network

### Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*



- 1 neuron : 1 Pixel
  - Between feature maps we have different params and bias
- ↓
- The fact that all neurons in a feature map share the same parameters drastically reduces the number of parameters in the model
  - Once CNN has learned to recognize a pattern in a location, it can recognize it in any other location.

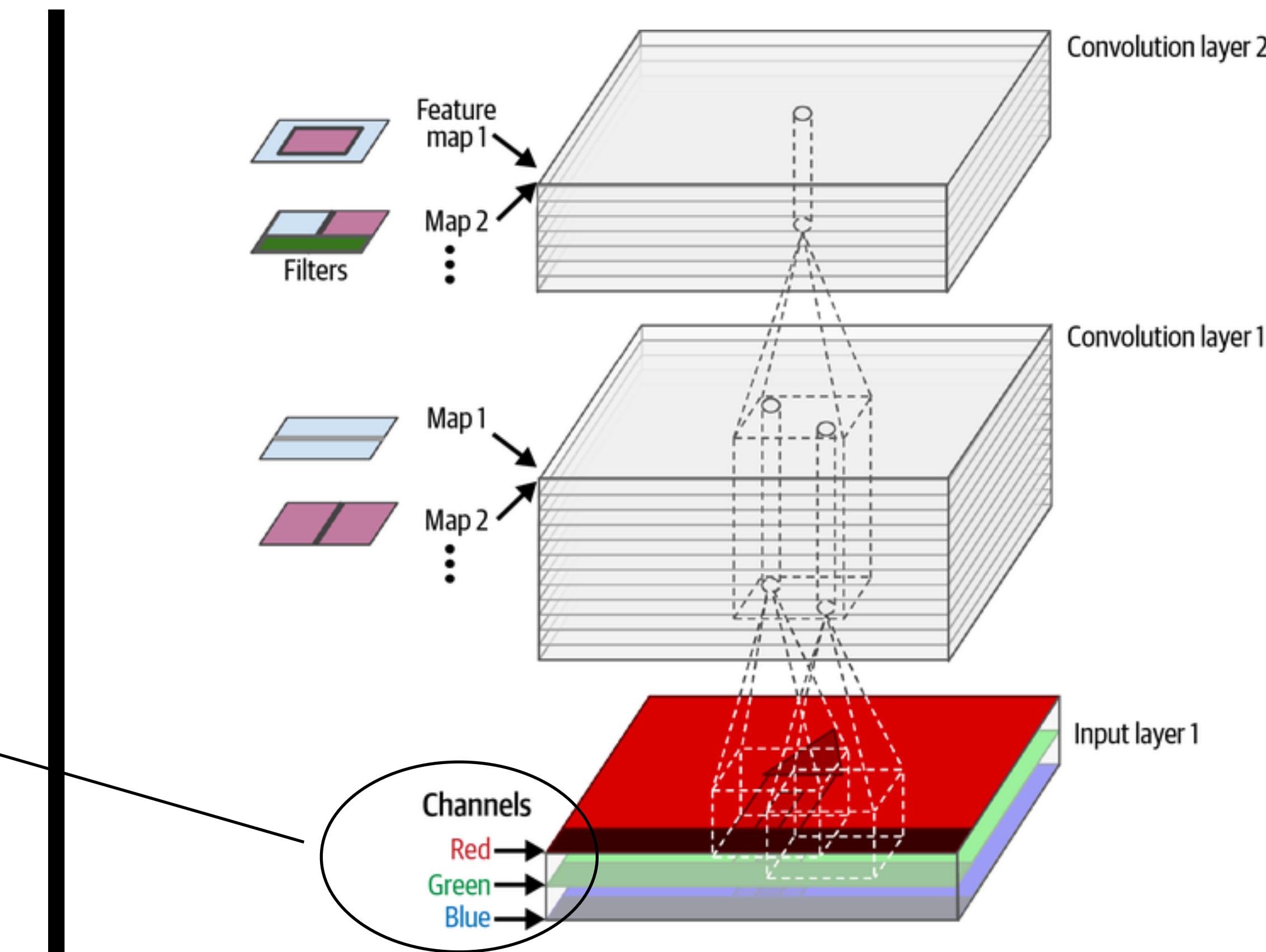
# Deep Computer Vision

## Convolutional Neural Network

### Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*

RGB + Infrared



- 1 neuron : 1 Pixel
  - Between feature maps we have different params and bias
- ↓
- The fact that all neurons in a feature map share the same parameters drastically reduces the number of parameters in the model
  - Once CNN has learned to recognize a pattern in a location, it can recognize it in any other location.

# Deep Computer Vision

## Convolutional Neural Network

**Stacking multiple Feature maps**

A *layer full of neurons using the same filter outputs a “feature map”*

**Calculate the output of a neuron in a convolutional layer**

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} \times w_{u,v,k',k}$$

with  $i' = i \times s_h + u$   
 $j' = j \times s_w + v$

$Z_{i,j,k}$  is the output of the neuron located in row  $i$ , column  $j$  in the map of characteristics  $k$  of the convolutional layer (layer I).

$X_{i',j',k'}$  Is the output of the neuron located in layer  $I - 1$ , row  $i'$ , column  $j'$ , characteristic map  $k'$  (or channel  $k'$  if the previous layer is the input layer).

# Deep Computer Vision

## Convolutional Neural Network

**Stacking multiple Feature maps**

A *layer full of neurons using the same filter outputs a “feature map”*

**Calculate the output of a neuron in a convolutional layer**

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} \times w_{u,v,k',k}$$

with  $i' = i \times s_h + u$   
 $j' = j \times s_w + v$

$b_k$

Is the bias term for the characteristic map k (in layer l). You can think of it as a knob that adjusts the general brightness of the k characteristic map.

$W_{u,v,k',k}$

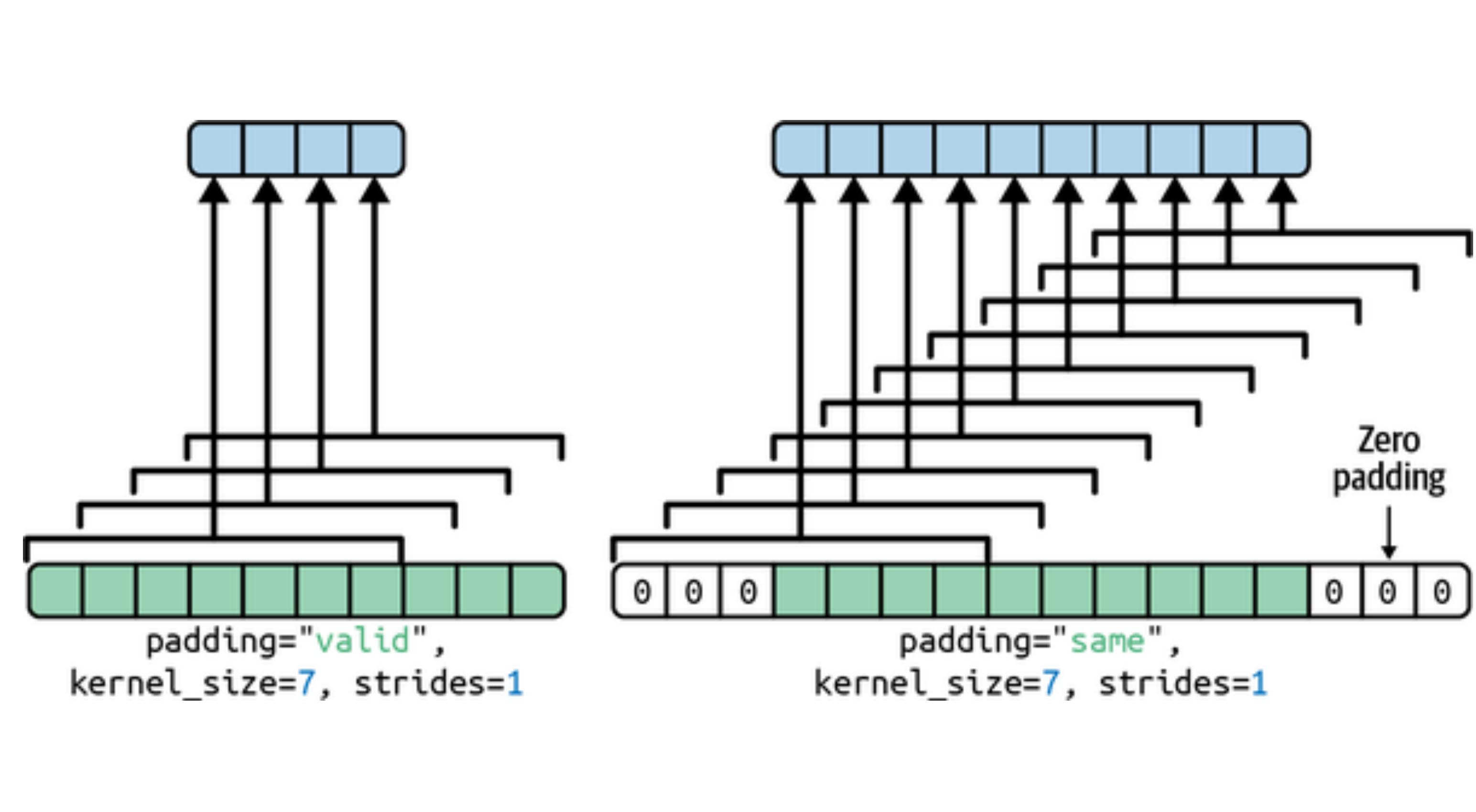
Is the connection weight between any neuron in the characteristic map k of the l layer and its input located in row u, column v (in relation to the receptive field of the neuron) and the map of characteristics k'.

# Deep Computer Vision

## Convolutional Neural Network

Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs a “feature map”*

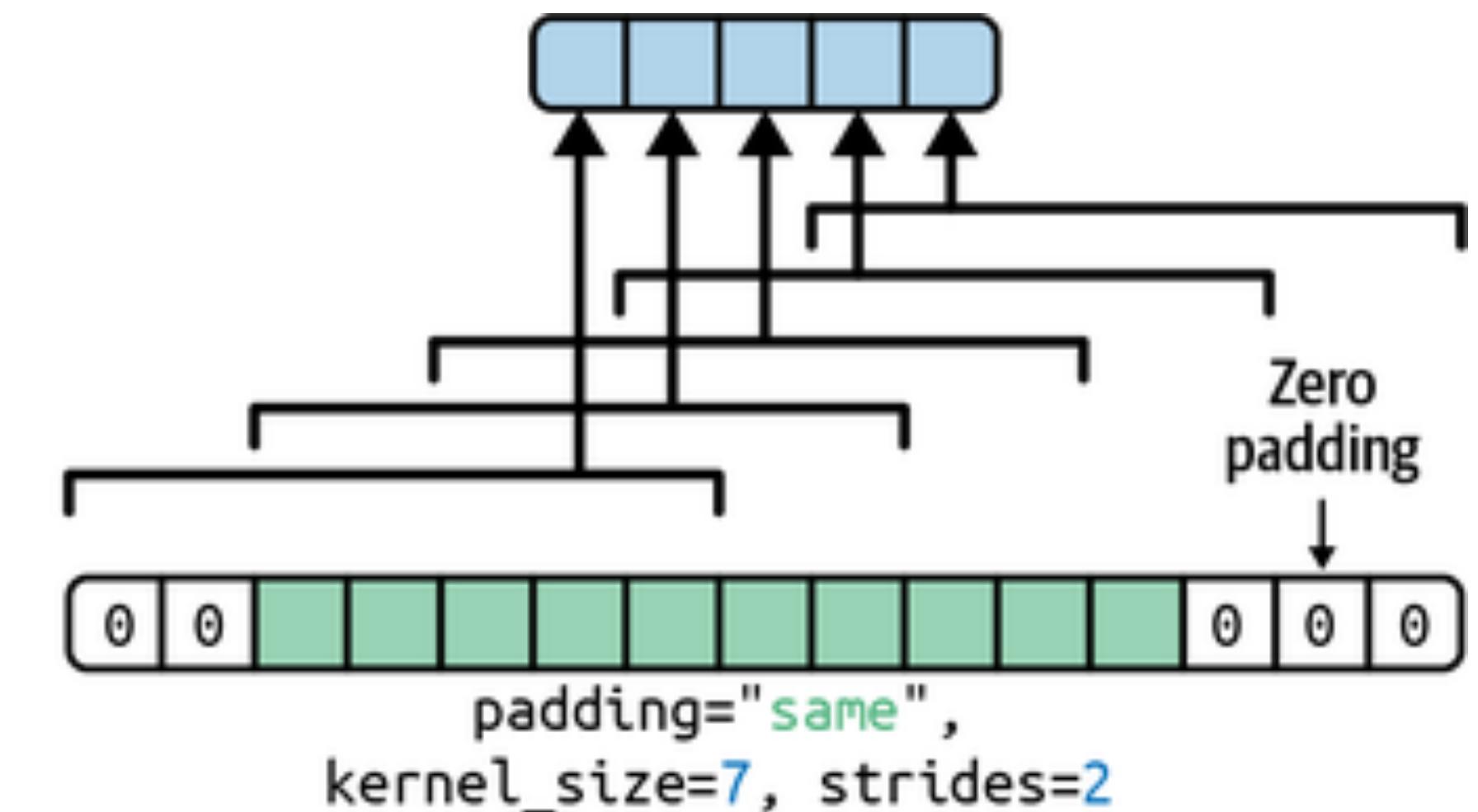
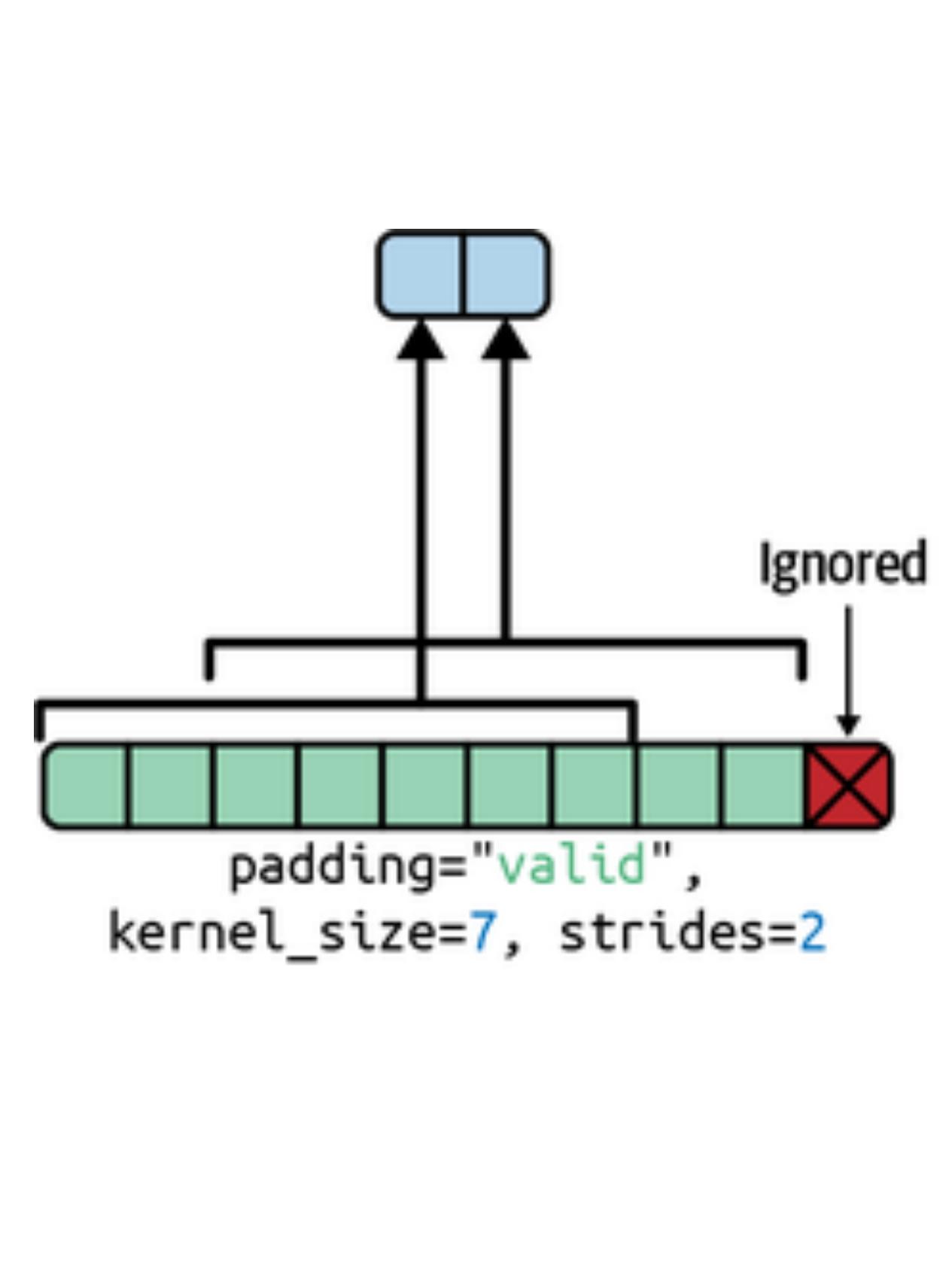


# Deep Computer Vision

## Convolutional Neural Network

Stacking multiple Feature maps

A *layer full of neurons using the same filter outputs* a “feature map”



# Deep Computer Vision

## Convolutional Neural Network

**Stacking multiple Feature maps**

*A layer full of neurons using the same filter outputs a “feature map”*

- Filters
- Strides

Hyperparams

- Kernel Size
- Activation
- Padding
- Kernel Initializer

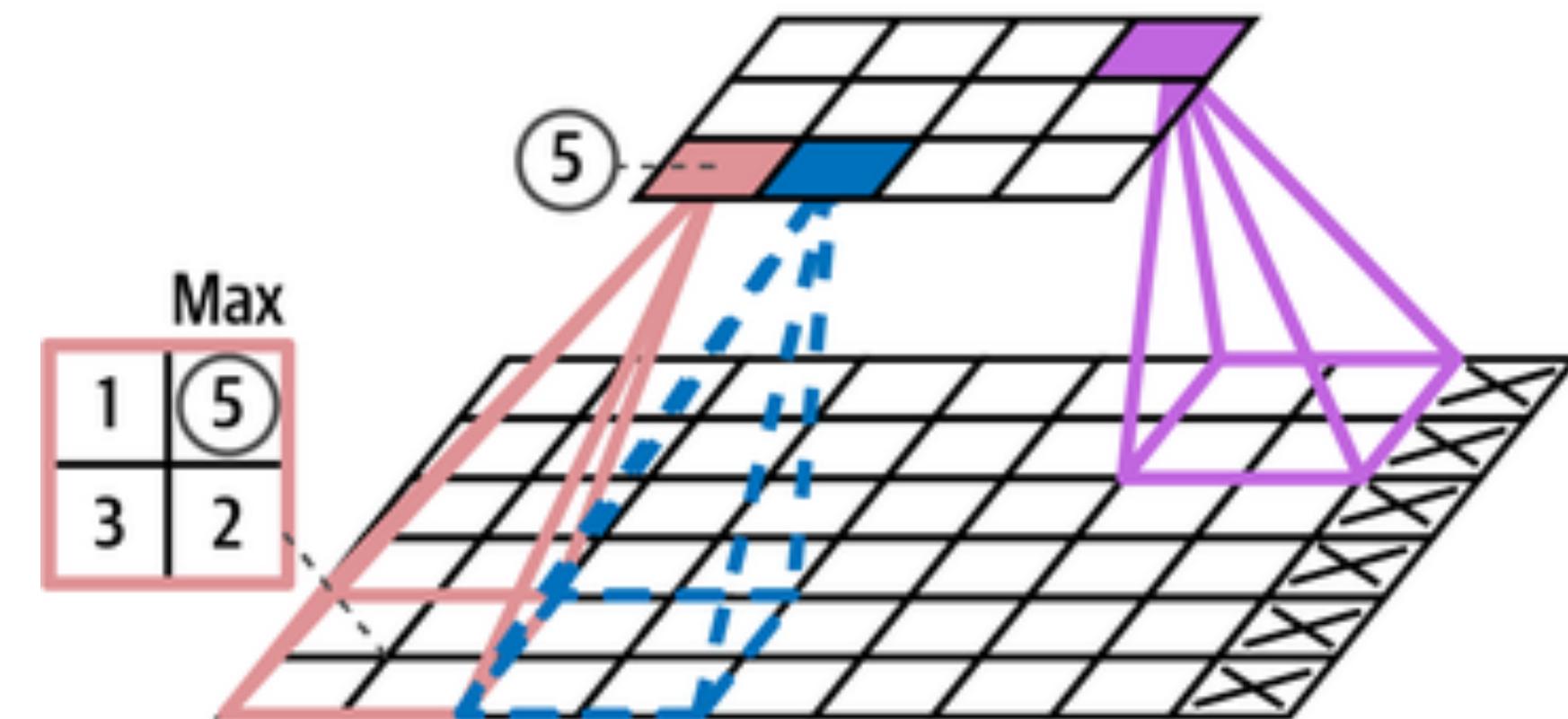
# Deep Computer Vision

## Convolutional Neural Network

### Pooling or Grouping layers

Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).

#### Max pooling



Kernel size:  $2 \times 2$

Stride: 2

Padding: "valid"



# Deep Computer Vision

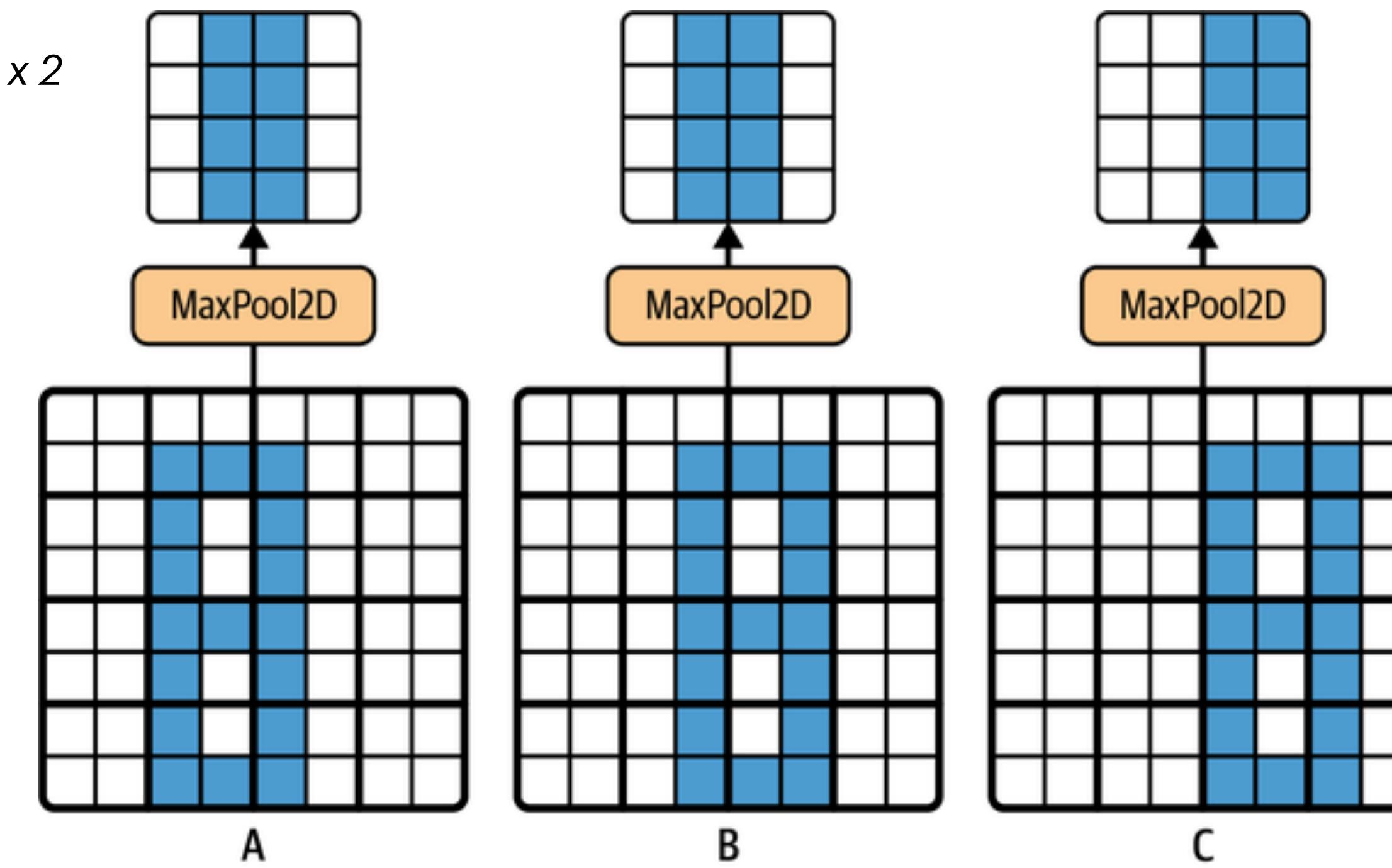
## Convolutional Neural Network

### Pooling or Grouping layers

*Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).*

#### Max pooling

*Kernel size: 2 x 2  
Stride: 2*



*Invariancia a traducciones pequeñas  
Output: 75% de los valores de entrada.*

# Deep Computer Vision

## Convolutional Neural Network

### Pooling or Grouping layers

Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).

#### Max pooling

- Preserves only the strongest characteristics
- The following layers get a cleaner signal to work with
- A stronger translation invariance
- Requires a little less computing

#### Avg. pooling

- Loses less information

# Deep Computer Vision

## Convolutional Neural Network

### Pooling or Grouping layers

Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).

#### Max pooling

- Preserves only the strongest characteristics
- The following layers get a cleaner signal to work with
- A stronger translation invariance
- Requires a little less computing

#### Avg. pooling

- Loses less information

#### Global Avg. pooling

Only generates a single number per feature map.

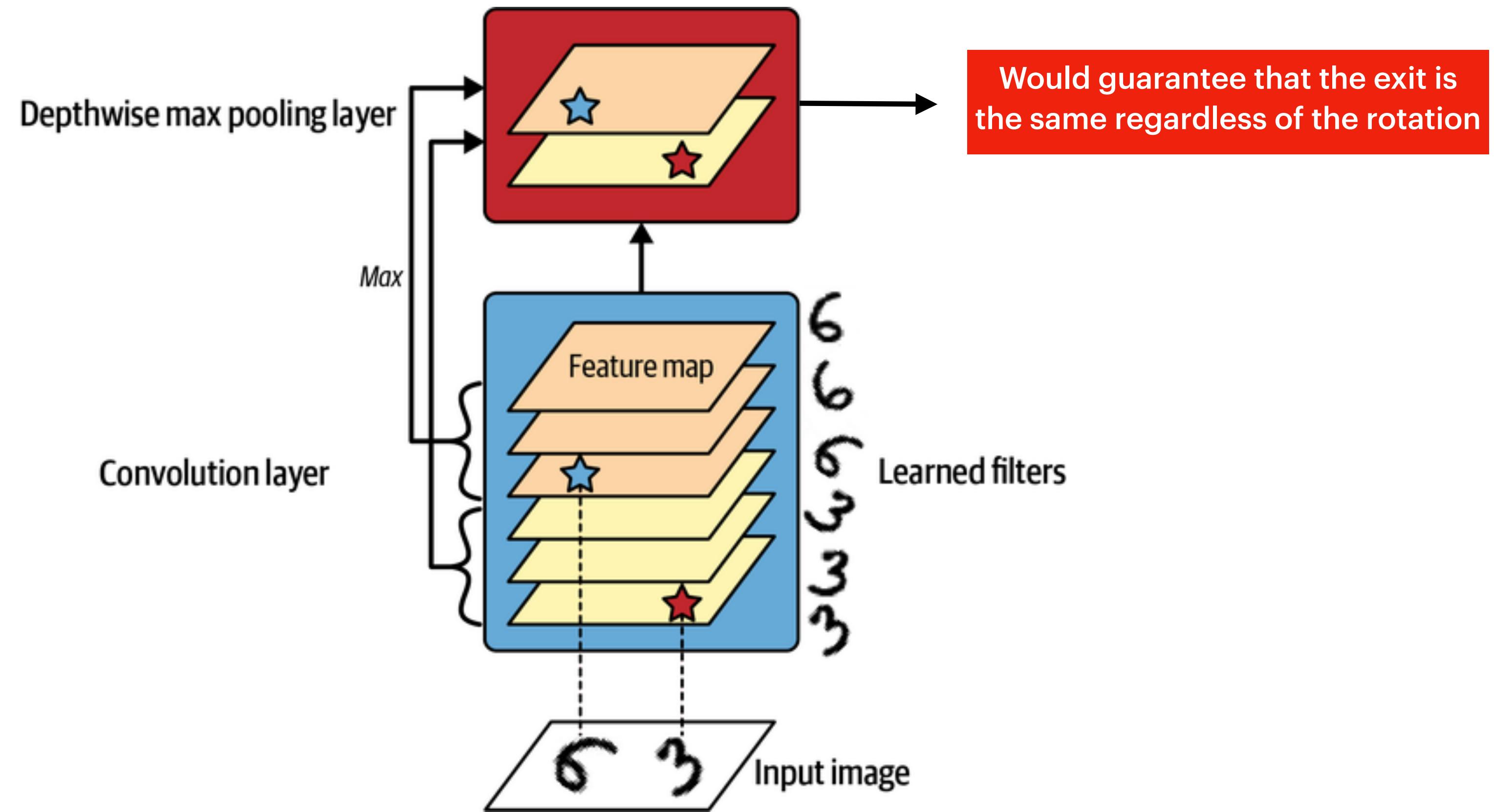
- **Extremely destructive** (most of the information in the feature map is lost)
- It can be useful just before the output layer

# Deep Computer Vision

## Convolutional Neural Network

### Pooling or Grouping layers

Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).



# Deep Convolutional

## Pooling or Grouping layers

Its objective is to subsampling (i.e., reduce) the input image to reduce the computational load, memory usage and the number of parameters (thus limiting the risk of overfitting).

Depthwise max pooling

Convolution



should guarantee that the exit is same regardless of the rotation

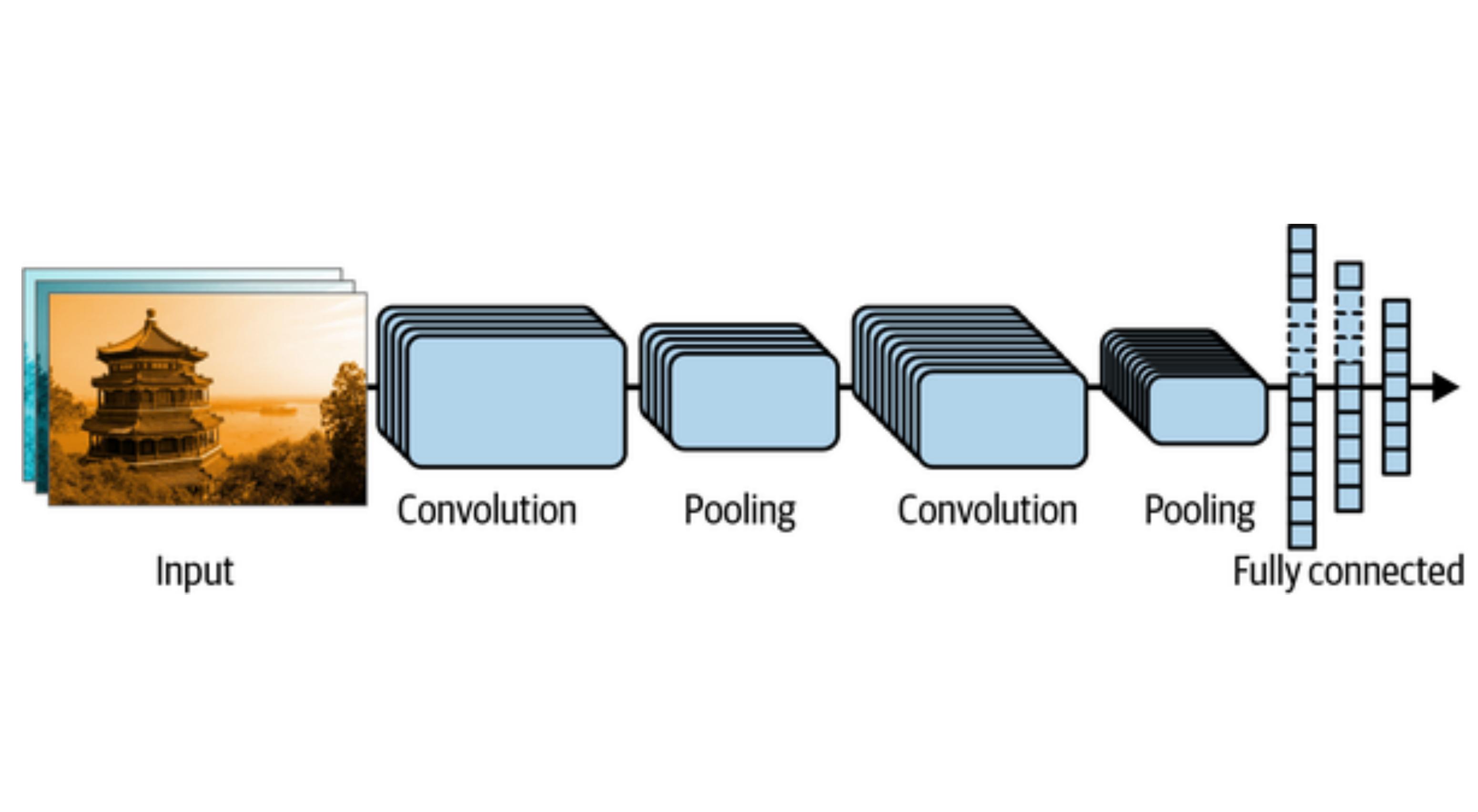
# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*



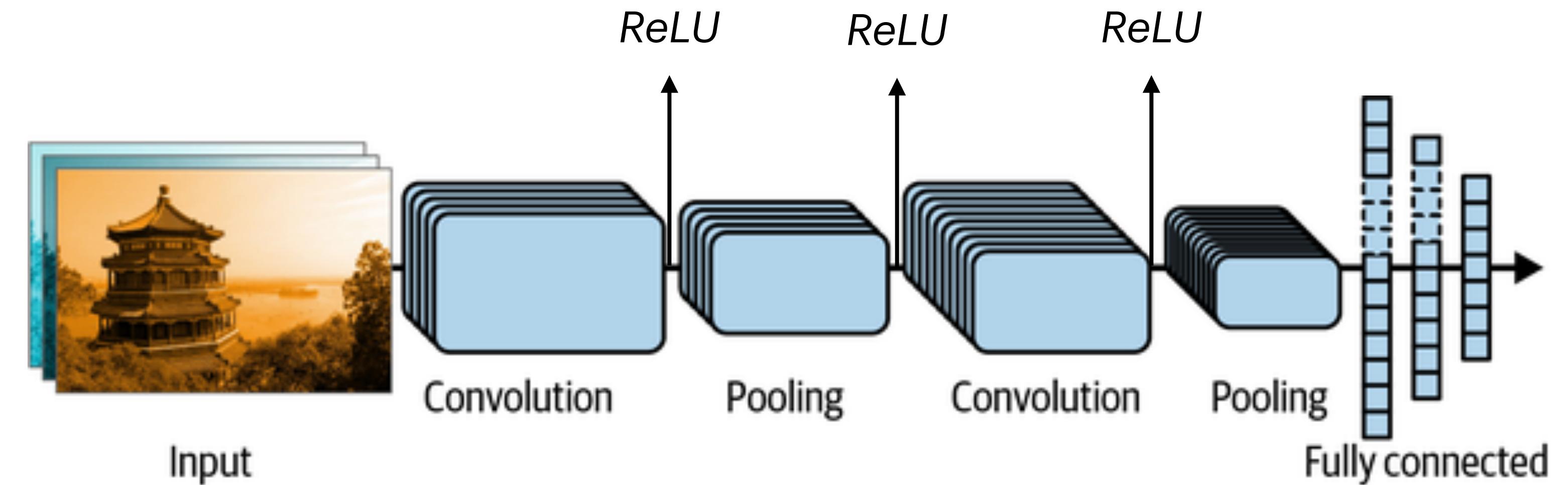
# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*



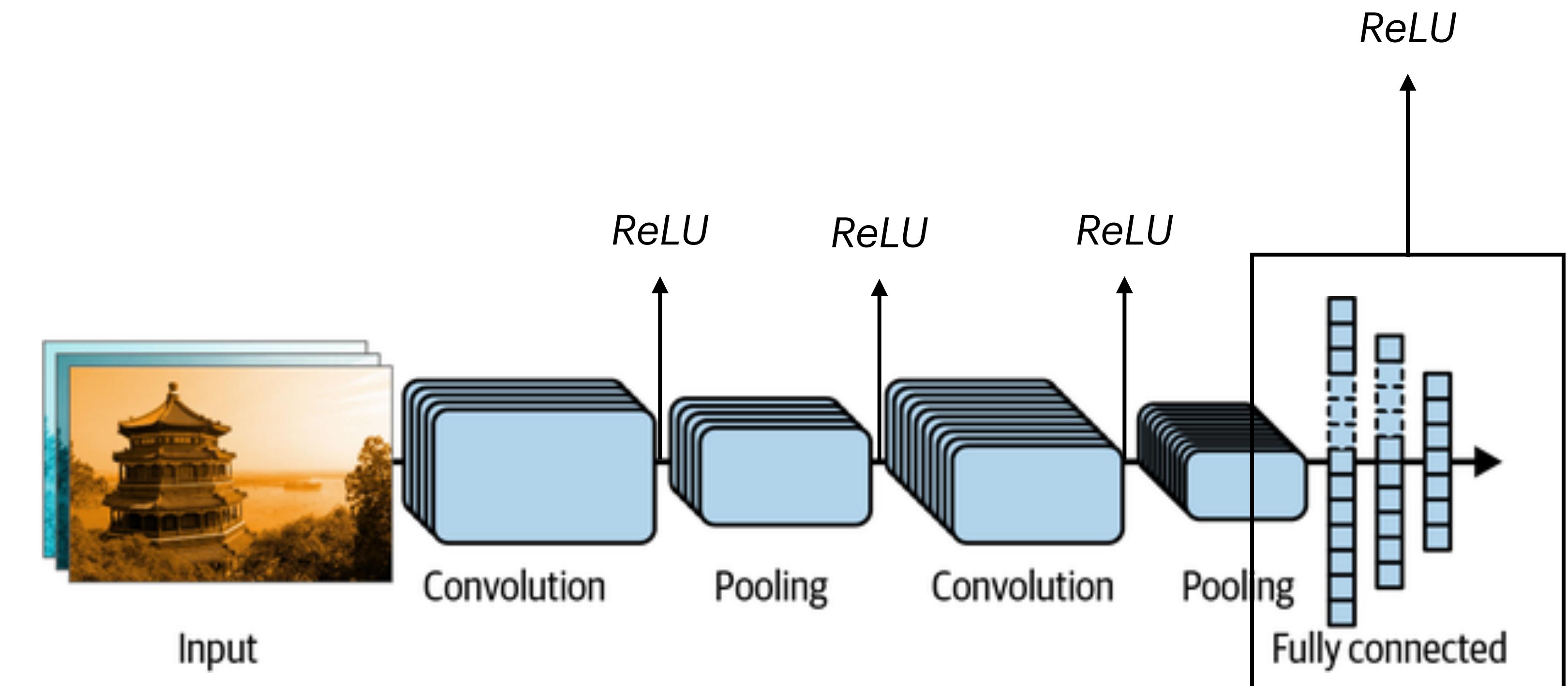
# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*



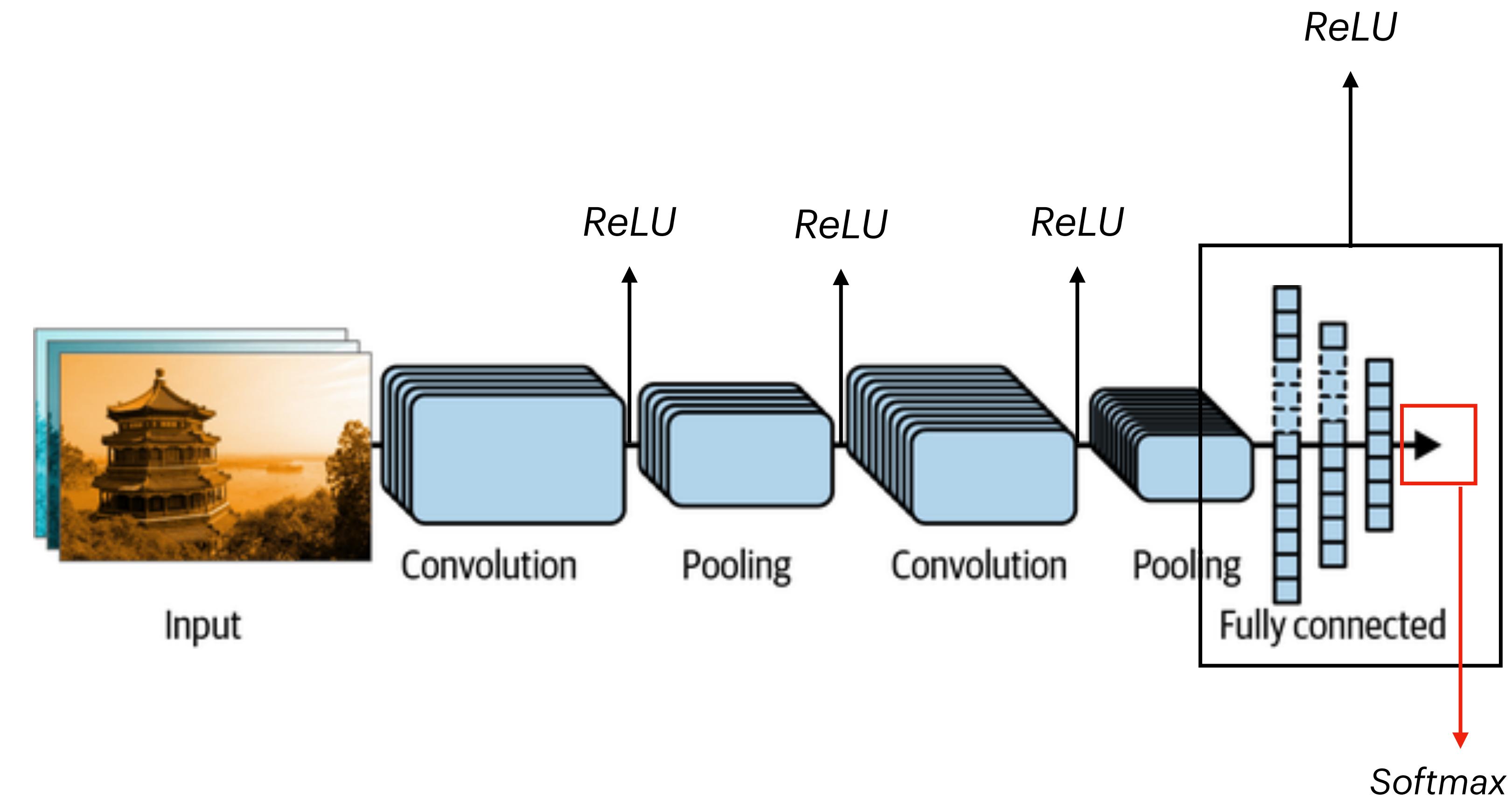
# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*



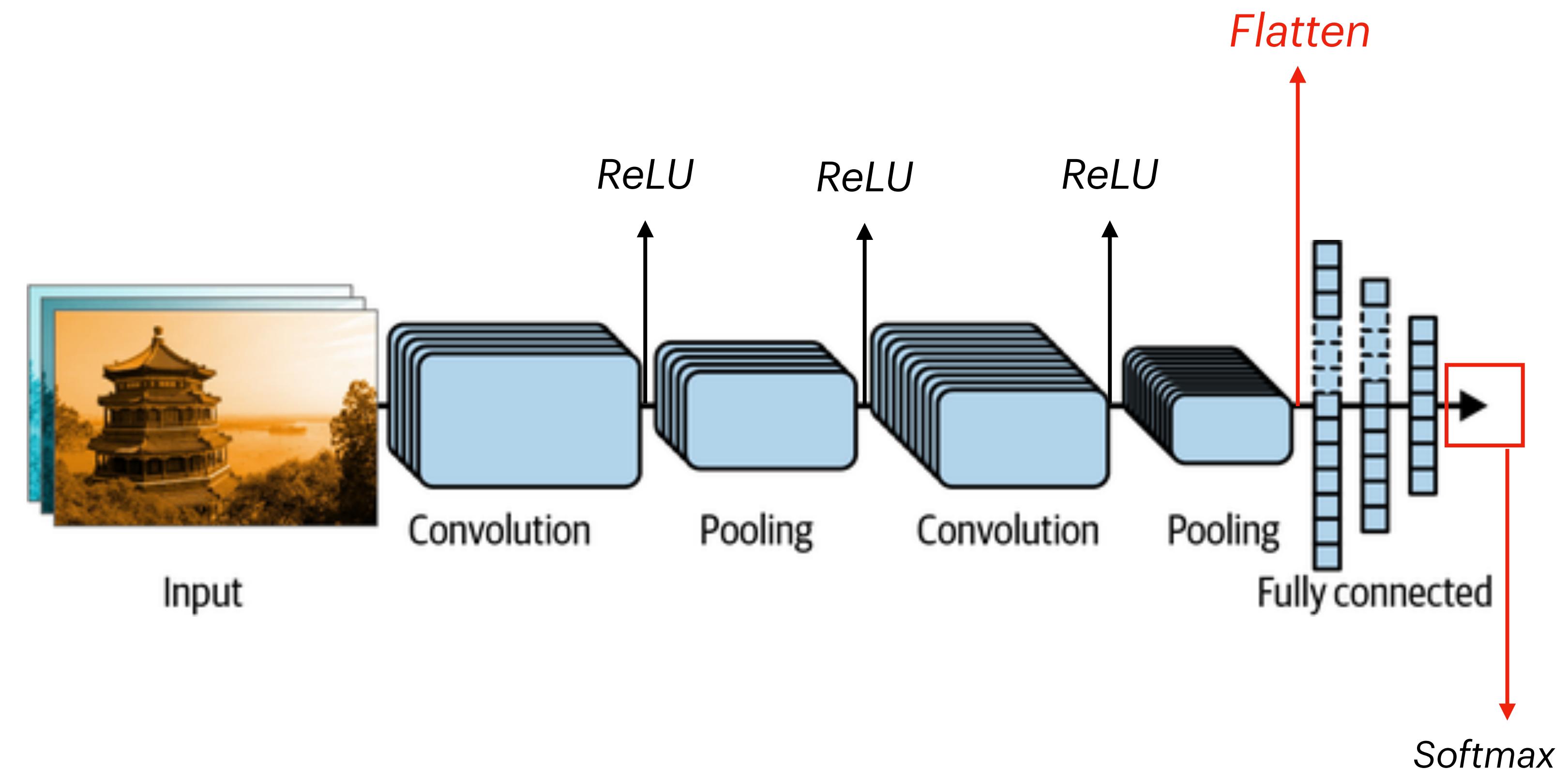
# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*



# Deep Computer Vision

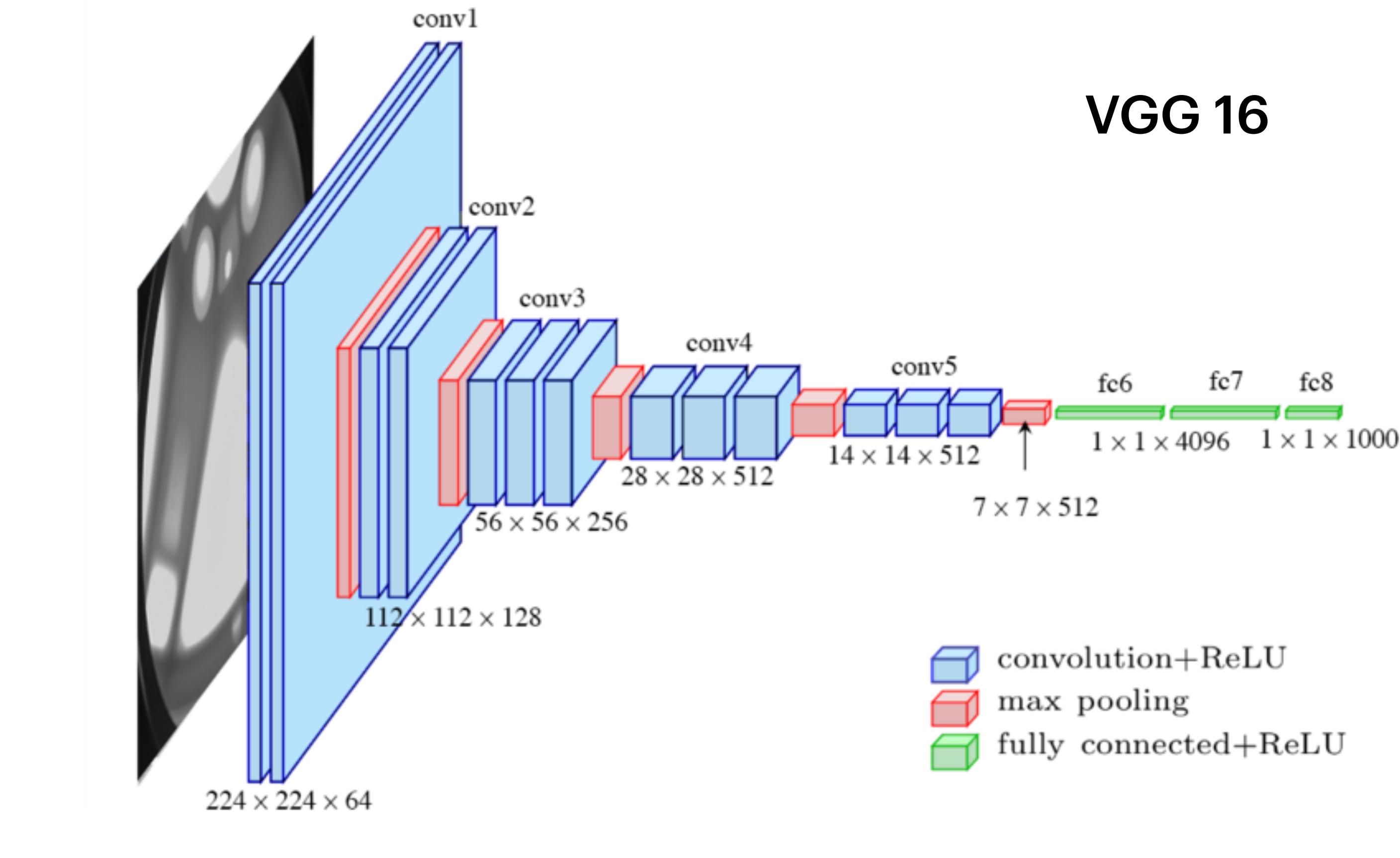
## CNN Architectures

### Typical CNN architecture

The typical architectures of CNN stack some convolutional layers.

The image becomes smaller and smaller as it advances through the network.

Destilation Law



# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*

Destilation Law



14,197,122 images, 21841 synsets indexed

[Home](#) [Download](#) [Challenges](#) [About](#)

Not logged in. [Login](#) | [Signup](#)

### An Update to the ImageNet Website and Dataset

March 11, 2021

We are proud to see ImageNet's wide adoption going beyond what was originally envisioned. However, the decade-old website was burdened by growing download requests. To serve the community better, we have redesigned the [website](#) and upgraded its hardware. The new website is simpler; we removed tangential or outdated functions to focus on the core use case—enabling users to [download the data](#), including the full ImageNet dataset and the [ImageNet Large Scale Visual Recognition Challenge \(ILSVRC\)](#).

Meanwhile, the computer vision community has progressed, and so has ImageNet. The dataset was created to benchmark object recognition—at a time when it barely worked. The problem then was how to collect labeled images at a sufficiently large scale to be able to train complex models in laboratories. Today, computer vision is in real-world systems impacting people's Internet experience and daily lives. An emerging problem now is how to make sure computer vision is fair and preserves people's privacy. We are continually evolving ImageNet to address these emerging needs.

In a [FAT\\* 2020 paper](#), we filtered 2,702 synsets in the "person" subtree that may cause problematic behaviors of the model. We have updated the full ImageNet data on the website to remove these synsets. The update does not affect the 1,000 categories in ILSVRC.

In a [more recent paper](#), we investigate privacy issues in ILSVRC. 997 out of 1000 categories in ILSVRC are not people categories; nevertheless, many incidental people are in the images, whose privacy is a concern. We first annotated faces in the images and then constructed a face-blurred version of ILSVRC. Experiments show that one can use the face-blurred version for benchmarking object recognition and for transfer learning with only marginal loss of accuracy. We release our [face annotations](#) to facilitate further research on privacy-aware visual recognition.

Team members working on these new improvements: [Kaiyu Yang](#) (Princeton), [Jacqueline Yau](#) (Stanford), [Li Fei-Fei](#) (Stanford), [Jia Deng](#) (Princeton), [Olga Russakovsky](#) (Princeton).

# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

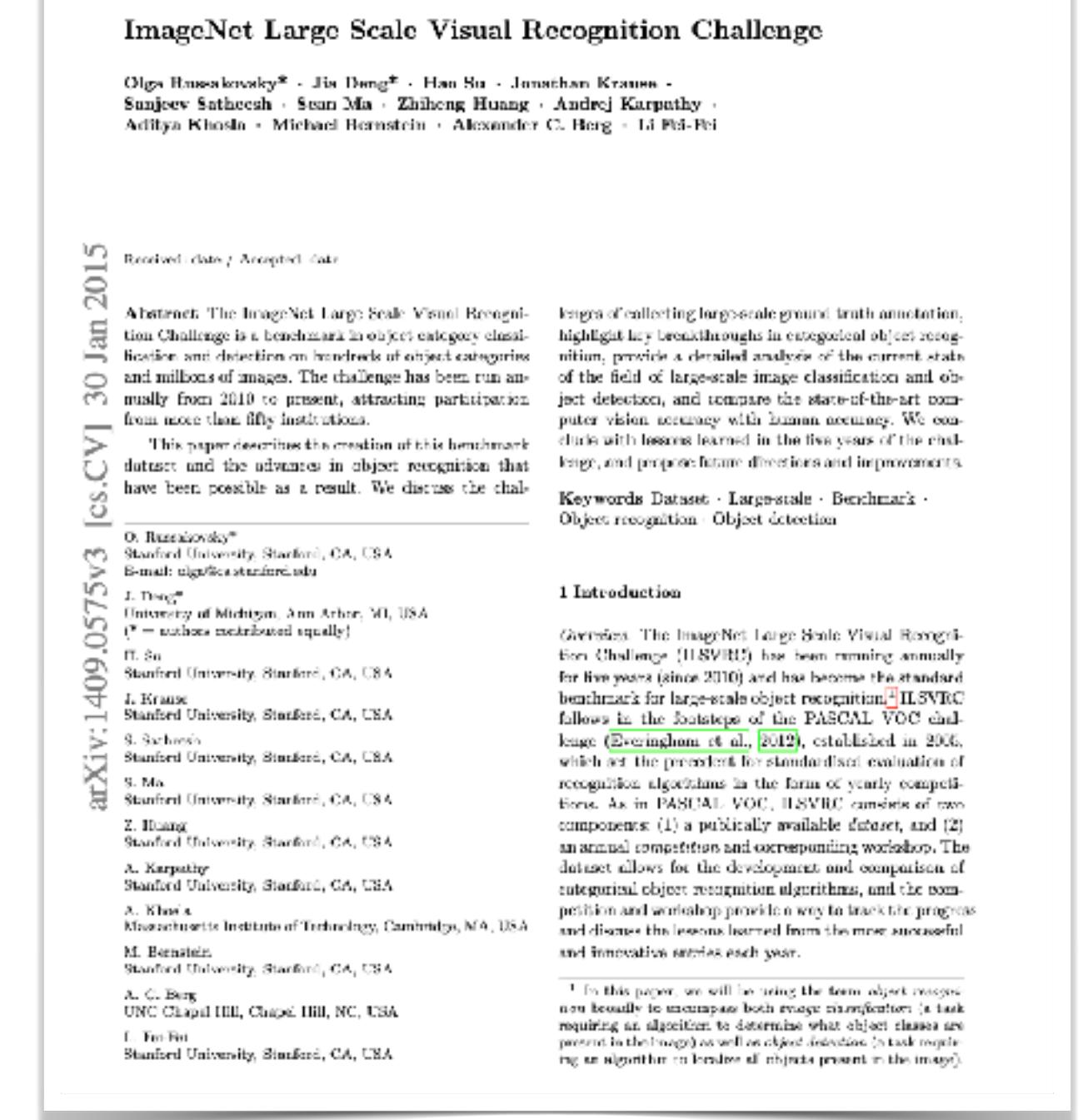
*The image becomes smaller and smaller as it advances through the network.*

Destilation Law

### Error rate improvement: ILSVRC ImageNet



The screenshot shows the ILSVRC competition page. It features a header with the IMGENET logo and navigation links for Home, Download, Challenges, and About. Below the header, there's a brief introduction to the competition, mentioning 14,197,122 images and 21,841 synsets indexed. A 'Competition' section details the challenge's purpose of comparing progress in detection across various objects and measuring computer vision for large-scale image indexing. It lists past years from 2017 to 2010. A 'Workshop' section notes the annual workshop at premier conferences. A 'Download' section links to the Kaggle dataset. An 'Evaluation Server' section provides instructions for evaluating results. An 'Updates' section tracks recent changes. A 'Citation' section includes a reference to a paper by Olga Russakovsky et al. (2015).



The screenshot shows the arXiv preprint titled 'ImageNet Large Scale Visual Recognition Challenge'. It includes the authors' names (Olga Russakovsky\*, Jia Deng\*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhifeng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei), the submission date (30 Jan 2015), and the accepted date. The abstract discusses the creation of the benchmark dataset and advances in object recognition. The paper is categorized under Dataset, Large-scale, Benchmark, Object recognition, and Object detection. The introduction section highlights the goal of collecting large-scale ground truth annotation, significant breakthroughs in object recognition algorithms, and the challenges of large-scale image classification and object detection. The paper concludes with lessons learned and future directions.

<sup>1</sup> In this paper, we will be using the term object recognition broadly to encompass both image classification (a task requiring an algorithm to determine what object classes are present in the image) as well as object detection (a task requiring an algorithm to locate all objects present in the image).

# Deep Computer Vision

## CNN Architectures

### Typical CNN architecture

*The typical architectures of CNN stack some convolutional layers.*

*The image becomes smaller and smaller as it advances through the network.*

**Destilation Law**

### Error rate improvement: ILSVRC ImageNet

**Error rate from 26% to 2.3% in 6 years!!**

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

# Deep Computer Vision

## Convolutional Neural Network

### Convolutional Layer

*Each neuron in the second convolutional layer is connected only to neurons located within a small rectangle in the first layer.*

*Then assemble them into larger, higher-level features in the next hidden layer,*





# Deep Computer Vision

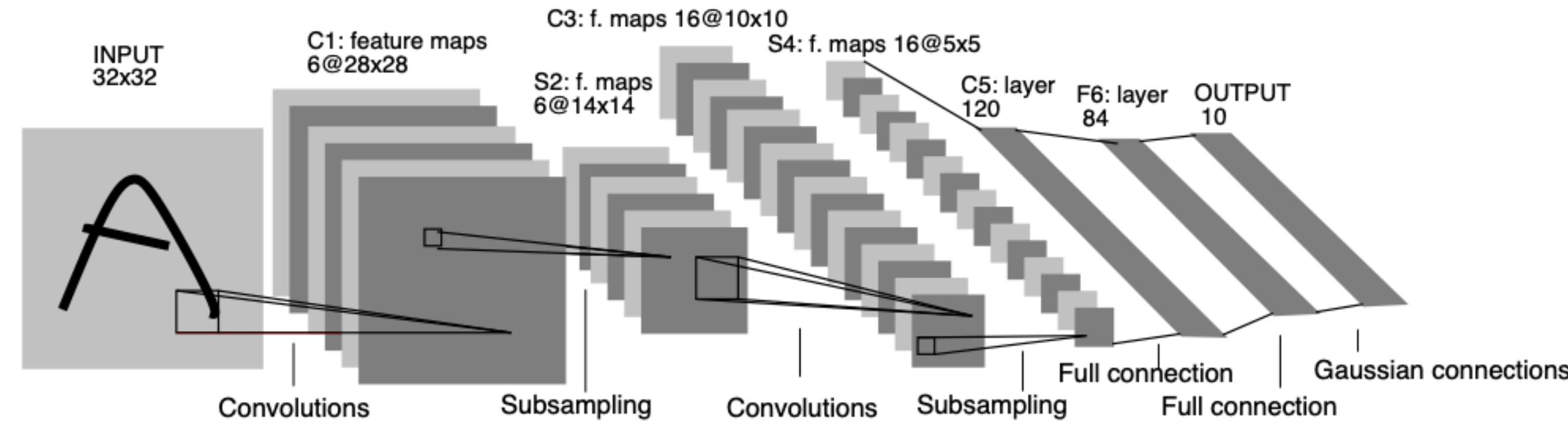
## CNN Architectures

**ILSVRC ImageNet**

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

**LeNet-5 (1998)**



Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

### I. INTRODUCTION

**Abstract—** Multilayer neural networks trained with the backpropagation algorithm constitute the best example of a successful Gradient-Based Learning technique. Given an appropriate architecture, such networks can learn algorithms that can be used to recognize a complex decision surface that can classify high-dimensional patterns such as handwritten characters, spoken digits, and faces. A paper review has been conducted to describe the major contributions of early work on a standard handwritten digit recognition task. Convolutional Neural Networks, that are specifically designed to deal with the variability of 2D shapes, are also discussed.

Recently, document recognition systems are composed of multiple modules including field extraction, segmentation, optical character reading, reading. A new learning paradigm, namely Graph Transformer Networks (GTN), allows such multi-module systems to be trained globally using Gradient-Based methods so as to minimize an overall performance metric.

The systems for outlier handwriting recognition are described. Experiments demonstrate the advantage of global training and the flexibility of Graph Transformer Networks. A Graph Transformer Network for reading handwritings is also described. It uses Convolutional Neural Networks whose neurons combined with global reading techniques provides robustness to local noise and personal checks. It is implemented commercially and reads several million digits per day.

**Keywords:** Neural Networks, DCT, Document Recognition, Machine Learning, Gradient-Based Learning, Convolutional Neural Networks, Graph Transformer Networks, Feature Space Transformations.

### NOTATION

- GT: Graph Transformer
- GIN: Graph transformer network
- HMM: Hidden Markov model
- HOS: Higher-order summation
- K-NN: K-nearest neighbor
- NN: Neural network
- OCR: Optical character recognition
- PCA: Principal component analysis
- RBF: Radial basis function
- RS-SVM: Refined-set support vector method

Over the last several years, machine learning techniques, particularly when applied to neural networks, have played an increasingly important role in the design of pattern recognition systems. In fact, it could be argued that the availability of existing techniques has been a crucial factor in the recent success of pattern recognition applications such as continuous speech recognition and handwriting recognition.

The main message of this paper is that better pattern recognition systems can be built by relying more on automatic learning, and less on hand-designed heuristics. The paper is organized by topic, except for the first section which describes the traditional way of building recognition systems by manually integrating individually designed modules. This provides context to the global training paradigm.

The systems for outlier handwriting recognition are described. Experiments demonstrate the advantage of global training and the flexibility of Graph Transformer Networks. A Graph Transformer Network for reading handwritings is also described. It uses Convolutional Neural Networks whose neurons combined with global reading techniques provides robustness to local noise and personal checks. It is implemented commercially and reads several million digits per day.

Since the early days of pattern recognition, it has been known that the variability and richness of natural data, be it speech, digits, or other types of patterns, make it almost impossible to build an accurate recognition system entirely by hand. Consequently, most pattern recognition systems are built using combinations of automatic learning techniques and hand-coded algorithms. The usual method of recognizing individual patterns consists in dividing the system into two main modules shown in figure 1. The first module takes a raw image of a handwritten pattern and performs tasks that can be expressed by low-dimensional vectors or short strings of symbols that (a) can be easily matched or compared, and (b) are relatively invariant with respect to transformations and distortions of the input patterns that do not change their nature. The feature extractor contains most of the prior knowledge and

# Deep Computer Vision

## CNN Architectures

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky  
University of Toronto  
kriz@cs.utoronto.ca

Ilya Sutskever  
University of Toronto  
ilya@cs.utoronto.ca

Geoffrey E. Hinton  
University of Toronto  
hinton@cs.utoronto.ca

### Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high resolution images in the ImageNet LSVRC 2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 25.2% achieved by the second-best entry.

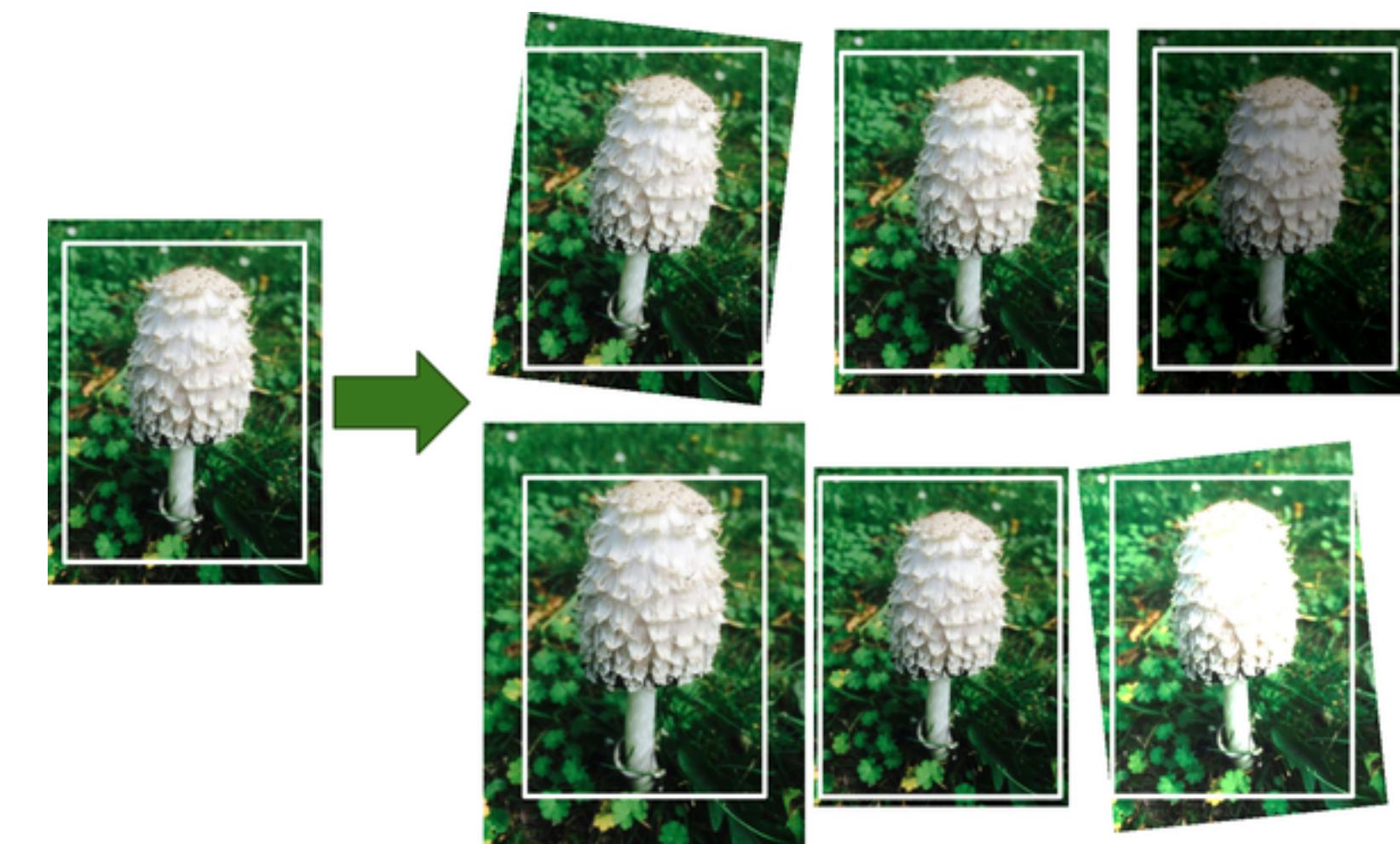
### 1 Introduction

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting. Until recently, datasets of labeled images were relatively small – on the order of tens of thousands of images (e.g., NORB [16], Caltech-101/256 [8, 9], and CIFAR-10/100 [12]). Simple recognition tasks can be solved quite well with datasets of this size, especially if they are augmented with label-preserving transformations. For example, the current-best error rate on the MNIST digit-recognition task (<0.3%) approaches human performance [4]. But objects in realistic settings exhibit considerable variability, so to learn to recognize them it is necessary to use much larger training sets. And indeed, the shortcomings of small image datasets have been widely recognized (e.g., Pinto et al. [21]), but it has only recently become possible to collect labeled datasets with millions of images. The new larger datasets include LabelMe [23], which consists of hundreds of thousands of fully-segmented images, and ImageNet [5], which consists of over 15 million labeled high-resolution images in over 22,000 categories.

## ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning



INCREASE IN DATA

Local Response Normalization (LRN)

$$b_i = a_i k + \alpha \sum_{j=j_{\text{low}}}^{j_{\text{high}}} a_j^2 \right)^{-\beta} \quad \text{with} \quad j_{\text{high}} = \min i + \frac{r}{2}, f_n - 1 \Big) \\ j_{\text{low}} = \max 0, i - \frac{r}{2} \Big)$$

# Deep Computer Vision

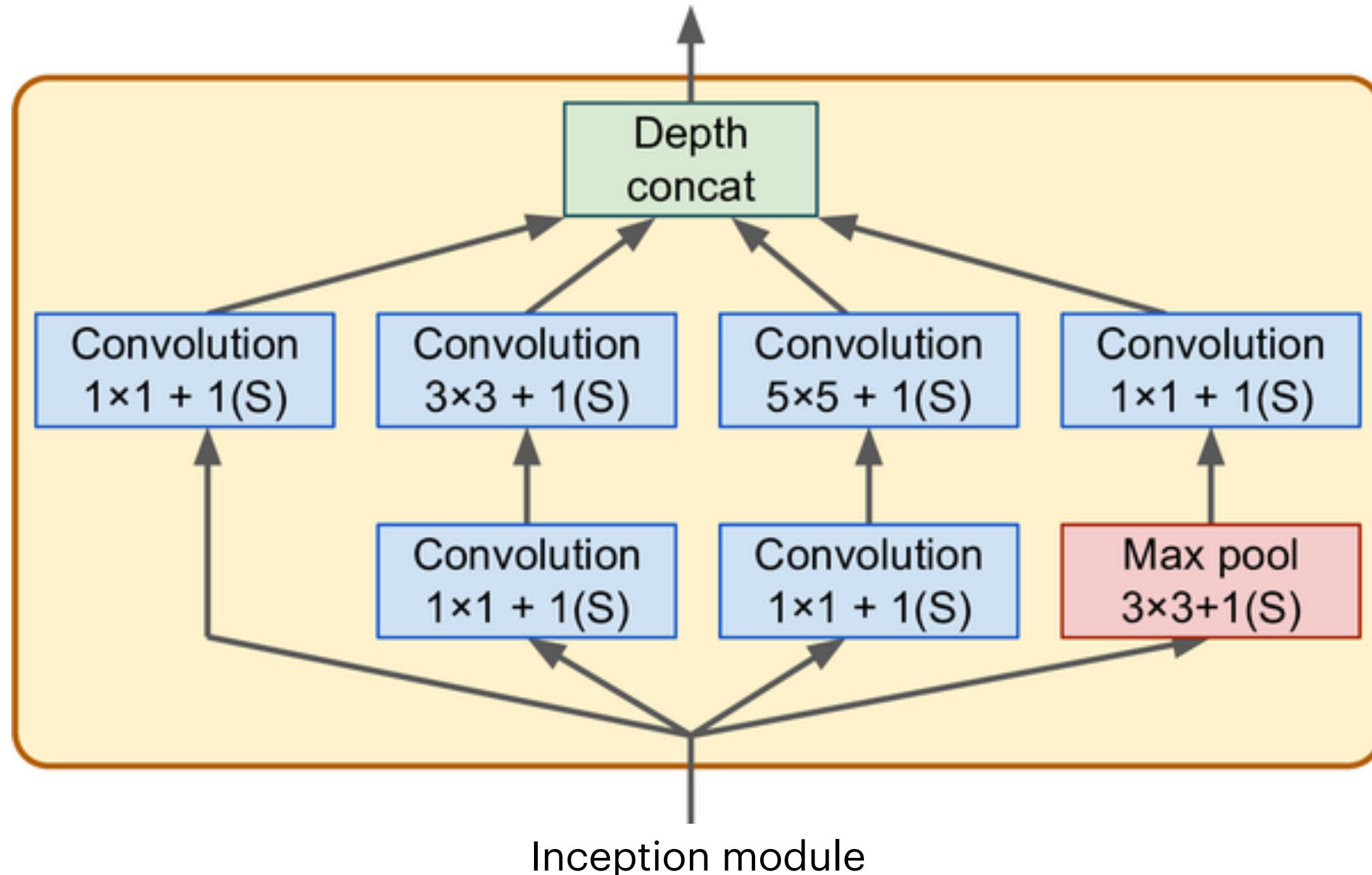
## CNN Architectures

### ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

### GoogleNet (ILSVRC winner 2014)



- Has 10 times fewer parameters than AlexNet
- Approximately 6 million instead of 60 million

#### Going Deeper with Convolutions

Christian Szegedy<sup>1</sup>, Wei Liu<sup>2</sup>, Yangqing Jia<sup>1</sup>, Pierre Sermanet<sup>1</sup>, Scott Reed<sup>2</sup>, Dragomir Anguelov<sup>3</sup>, Dumitru Erhan<sup>1</sup>, Vincent Vanhoucke<sup>1</sup>, Andrew Rabinovich<sup>4</sup>

<sup>1</sup>Google Inc. <sup>2</sup>University of North Carolina, Chapel Hill

<sup>3</sup>University of Michigan, Ann Arbor <sup>4</sup>Magic Leap Inc.

{szegedy, liu, jia, sermanet, reed, erhan, vanhoucke}@google.com

szegedy@google.com, jia@csail.mit.edu, reed@csail.mit.edu, vanhoucke@csail.mit.edu

#### Abstract

We propose a deep convolutional neural network architecture—“Inception” that achieves the new state-of-the-art for classification and detection in the ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC14). The main driving force of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the computational cost constant. To optimize quality, the architectural decisions revolve around the “Inception module” and its two-scale parallelism. One “Inception module” for classification and one for estimation for ILSVRC14 is called “GoogLeNet”, a 22-layer deep network, the quality of which is measured in the context of classification and detection.

#### 1. Introduction

In the last three years, our object classification and detection capabilities have dramatically improved due to advances in deep learning and convolutional networks [10]. Our encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models. It is also a consequence of new ideas, algorithms and improved architectures. No new data sources were used, for example, by the top teams in the ILSVRC 2014 competition besides the classification

and bigger deep networks, but from the synergy of deep architectures and visual computer vision, like the R-CNN algorithm by Girshick et al [6].

A rather notable factor is that with the ongoing focus of mobile and embedded computing, the efficiency of our algorithms—especially their power and memory use—gains importance. It is noteworthy that the core decisions leading to the design of the deep architecture presented in this paper included this factor rather than having a focus driven on accuracy numbers. From all of the experiments, the model was designed to keep a computation budget of 1 billion multiply-adds at inference time, so that the model could fit in memory and be deployed on mobile devices. Our model works well, even on large datasets, at a reasonable cost.

In this paper, we will focus on an efficient deep neural network architecture for computer vision, called “Inception”, which derives its name from the “Inception” network paper by Jia et al [12] in conjunction with the famous “two roads to go deeper” comment made [1]. In our case, the word “deep” is used in two different meanings: first of all, in the sense that we introduce a new level of organization in the form of the “Inception module”, and also in the more direct sense of “deep network” [9]. In general, we can view the Inception model as a logical extension of [12] while taking inspiration and guidance from the theoretical work by Arora et al [3]. The details of the architecture are experimentally verified in the ILSVRC 2014 classification and detection challenges, where it significantly outperforms the current state-of-the-art.

# Deep Computer Vision

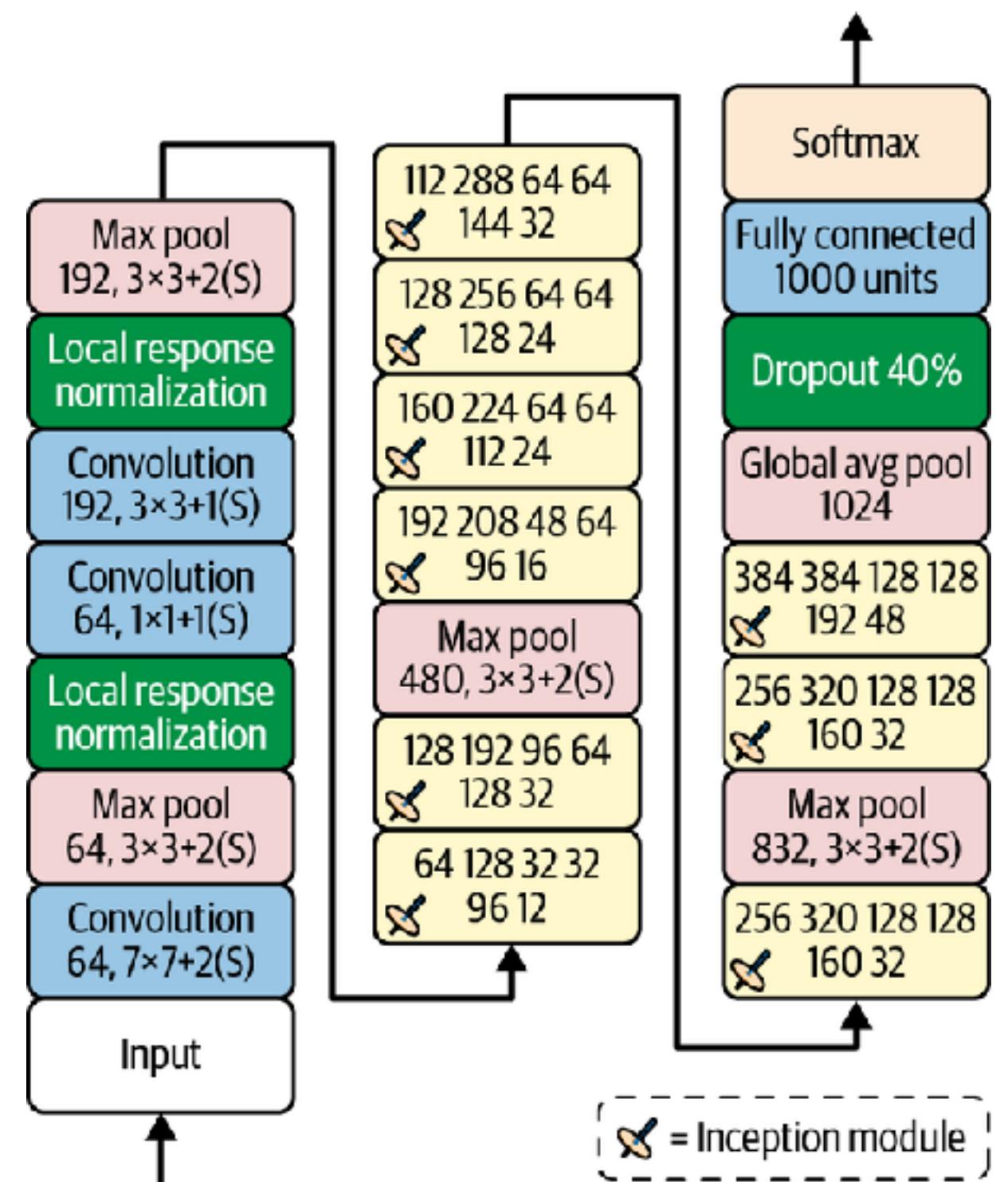
## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

### GoogleNet (ILSVRC winner 2014)



- Has 10 times fewer parameters than AlexNet
- Approximately 6 million instead of 60 million

#### Abstract

Christian Szegedy<sup>1</sup>, Wei Liu<sup>2</sup>, Yangqing Jia<sup>1</sup>, Pierre Sermanet<sup>1</sup>, Scott Reed<sup>2</sup>, Dragomir Anguelov<sup>3</sup>, Dumitru Erhan<sup>1</sup>, Vincent Vanhoucke<sup>1</sup>, Andrew Rabinovich<sup>4</sup>

<sup>1</sup>Google Inc. <sup>2</sup>University of North Carolina, Chapel Hill

<sup>3</sup>University of Michigan, Ann Arbor <sup>4</sup>Magic Leap Inc.

{szegedy, liu, jia, sermanet, reed, erhan, vanhoucke}@google.com

{dumitru.anguelov, andrew.rabinovich}@microsoft.com

http://arxiv.org/abs/1409.1556 | http://arxiv.org/pdf/1409.1556.pdf | http://arxiv.org/abs/1409.1556v3

We propose a deep convolutional neural network architecture—“Inception” that achieves the new state-of-the-art for classification and detection in the ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC14). The main driving force of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the computational cost constant. To improve quality, the architectural decisions revolve around the Inception module and its variants, scale and reuse. One “Inception” module performs all the operations needed in our architecture for ILSVRC14. Our GoogLeNet, a 22-layer deep network, its quality of which is measured in the context of classification and detection, is

#### 1. Introduction

In the last three years, our object classification and detection capabilities have dramatically improved due to advances in deep learning and convolutional networks [16]. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models. It also leads a consequence of new ideas, algorithms and improved architectures. No new data sources were used, for example, by the top teams in the ILSVRC 2014 competition besides the classification

and bigger deep networks, but from the synergy of deep architectures and visual computer vision, like the R-CNN algorithm by Girshick et al [16].

Another notable factor is that with the ongoing focus of mobile and embedded computing, the efficiency of our algorithms—especially their power and memory consumption—is important. It is noteworthy that the research leading to the design of the deep architecture presented in this paper included this factor rather than having a focus on raw accuracy numbers. From the experiments, the model was designed to keep a computation budget of 1 billion multiply-adds at inference time, so that the model does not need to be trained on a GPU, but can run on a mobile device, even on large datasets, at a reasonable speed.

In this paper, we will focus on an efficient deep neural network architecture for computer vision, called Inception, which derives its name from the “Inception” in a well-known paper by Jia et al. [12] in comparison with the famous “two heads to go deeper” Inception module [1]. In our case, the word “Inception” is used in two different meanings: first of all, in the sense that we introduce a new level of organization in the form of the “Inception module”, and also in the more direct sense of “Inception network” [9]. In general, we can view the Inception model as a logical extension of [12] while taking inspiration and guidance from the theoretical work by Arora et al [13]. The details of the architecture are experimentally verified in the ILSVRC 2014 classification and detection challenges, where it significantly outperforms the current state-of-the-art.

# Deep Computer Vision

## CNN Architectures

 VERY DEEP CONVOLUTIONAL NETWORKS  
FOR LARGE-SCALE IMAGE RECOGNITION

 Karen Simonyan\* & Andrew Zisserman\*  
Visual Geometry Group, Department of Engineering Science, University of Oxford  
{karen, az} @robots.ox.ac.uk

## ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small ( $3 \times 3$ ) convolution filters, which shows that a significant improvement on the previous configurations can be achieved by pushing the depth to 15–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the recognition and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made all the best performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

## 1 INTRODUCTION

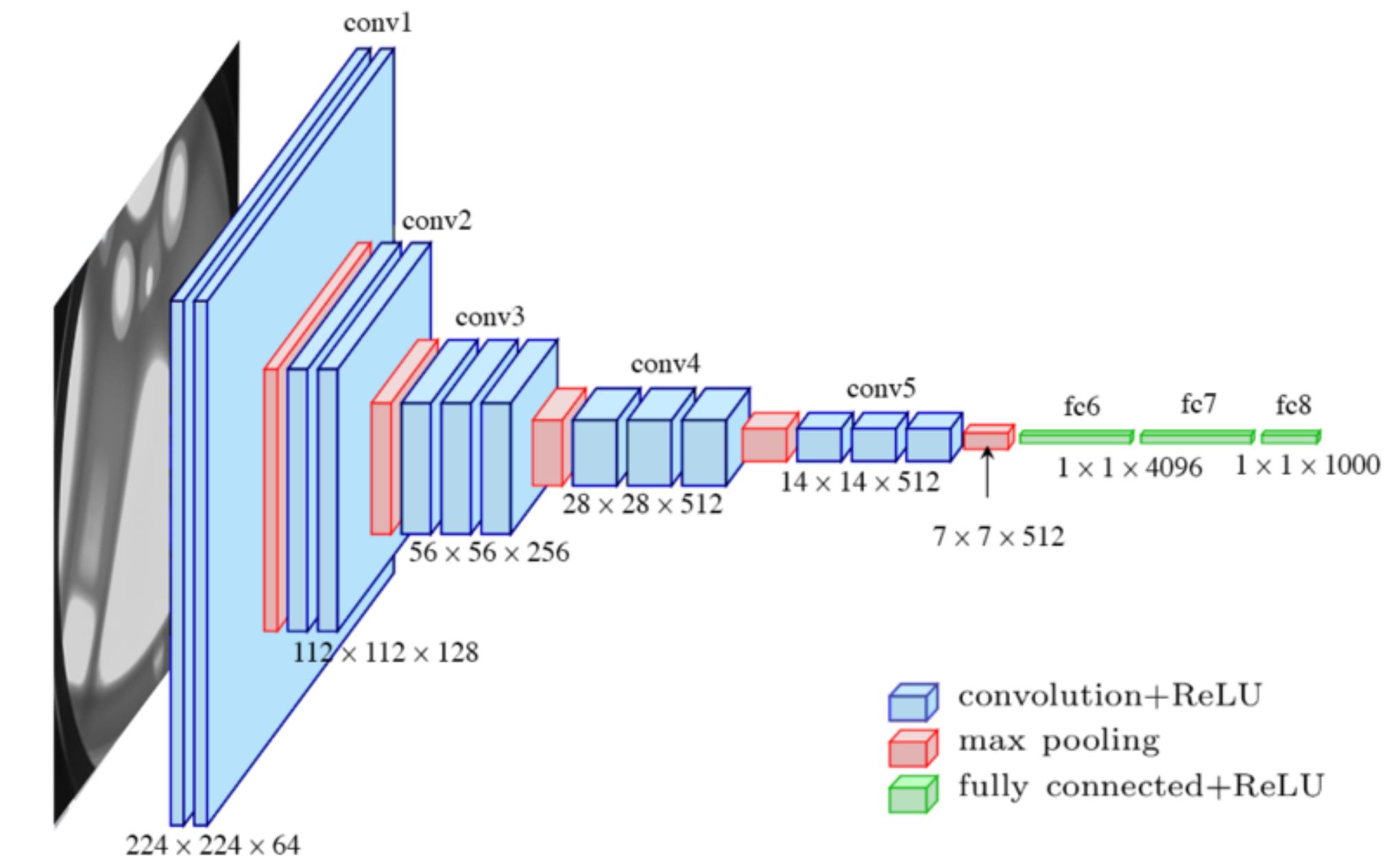
Convolutional networks (CNNs) have recently enjoyed a great success in large-scale image and video recognition (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Simonyan et al., 2014; Srivastava & Zisserman, 2014), which has become possible due to the large public image repositories, such as ImageNet (Deng et al., 2009), and high performance computer systems, such as GPUs or large scale distributed clusters (Brent et al., 2012). In particular, an important role in the advance of deep visual recognition methods has been played by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krauseková et al., 2014), which has served as a testbed for a few generations of large scale image classification systems, from hierarchical shallow feature matching (Fergus et al., 2010) to the winner of ILSVRC 2014 (Zagoruyko & Komodakis et al., 2015) to the winner of ILSVRC 2015.

With ConvNets becoming more of a commodity in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky et al. (2012) as well as derive better ones. For example, the winning solution of ILSVRC 2015 (Zagoruyko & Komodakis et al., 2015; Simonyan et al., 2014) addressed receptive fields and smaller strides of the first convolutional layers. Another line of improvements deal with mapping and scaling the network densely over the whole image and over multiple scales (Simonyan et al., 2013; He et al., 2014). In this paper, we address another important aspect of ConvNet architecture design – its depth. To this end, we fix other parameters of the architecture and steadily increase the depth of the network by adding more convolutional layers, which is feasible due to the use of very small ( $3 \times 3$ ) convolution filters in all layers.

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning



**VGGNet - Visual Geometry Group - Oxford  
University (ILSVRC sub-champion 2014)**



Pontificia Universidad  
JAVERIANA  
Colombia

# Deep Computer Vision

## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

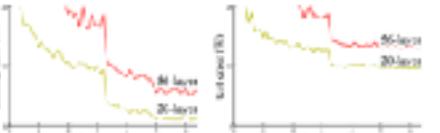
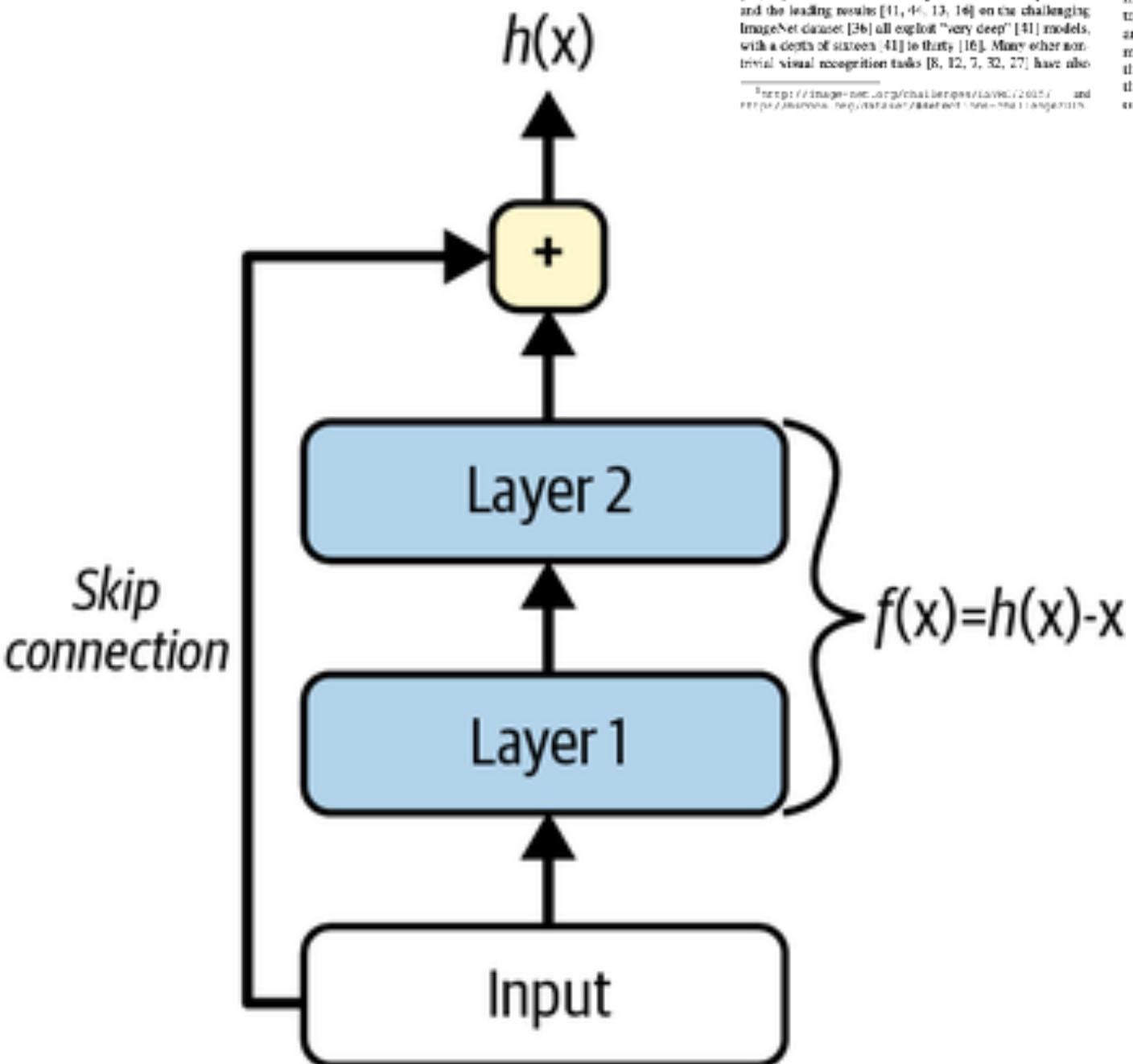
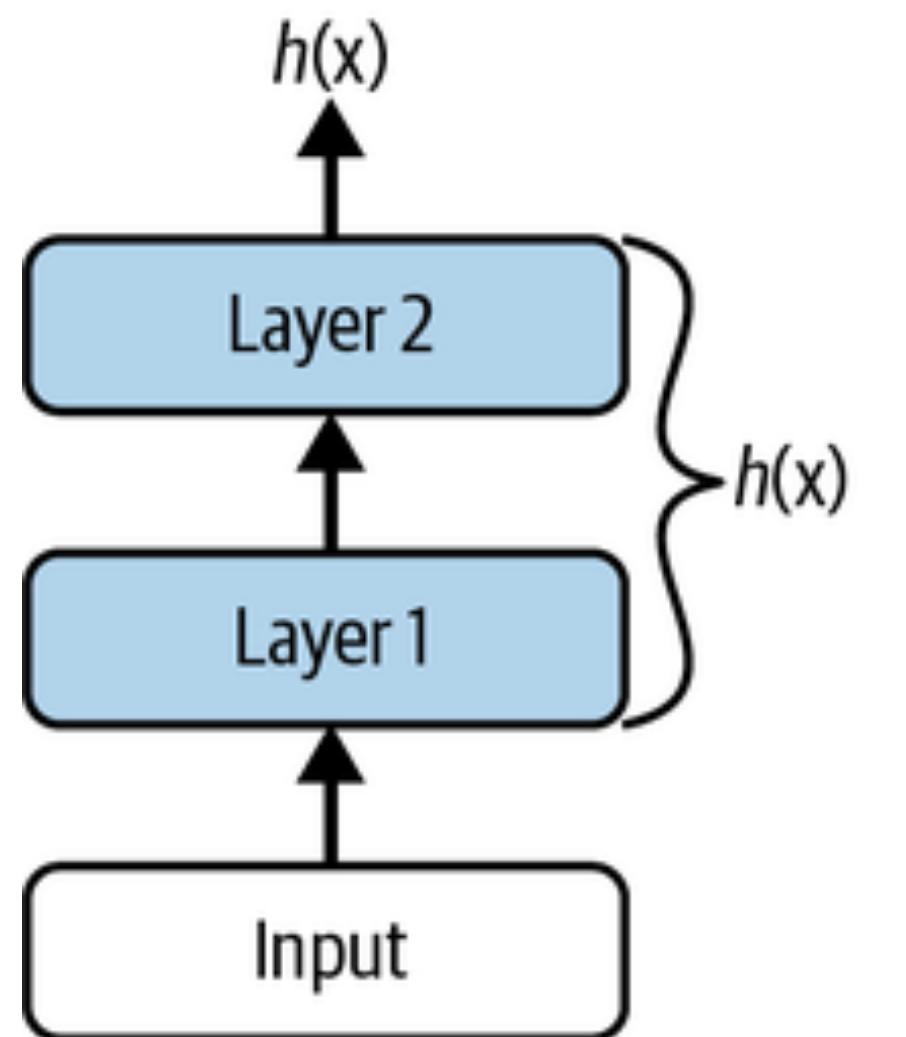


Figure 1: Training error (left) and test error (right) on CIFAR-10 with 20-layer and 35-layer "plain" networks. The deep network has higher training error and lower test error, similar phenomena on ImageNet is presented in Fig. 4.

### Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs instead of learning unparameterized functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate models with a depth of up to 152 layers— $\times$  deeper than VGG nets [44] but still having fewer parameters. An ensemble of three residual networks achieves 5.37% error on the ImageNet test set. This result was the *1st place* on the ILSVRC 2015 classification task. We also present analysis on ImageNet with 100 and 1000 layers.

The depth of representation is of crucial importance for many visual recognition tasks. Solely due to our extremely deep representation, we achieve a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundation of our submission to ILSVRC & COCO 2015 competitions<sup>1</sup>, where we also won the *1st place* on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [51, 50, 49]. Deep networks naturally integrate low-level features [50] and classifiers in an end-to-end multi-layer fashion, and the “levels” of features can be measured by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [11, 4, 13, 16] on the challenging ImageNet dataset [36] all employ “very deep” [31] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also placed their bets on deep learning.

<sup>1</sup><http://image-net.org/challenges/LSVRC/2015/> and <http://msra-coco.csail.mit.edu/coco2015.html>

Given the significance of depth, a question arises: *Is training deeper networks as easy as stacking more layers?*

An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [25, 9, 37, 13] and intermediate scale normalization [16], which make networks with tens of layers in fact converging for stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been reported: while the network depth increases, accuracy gets saturated (which might be unsatisfying) and then degrades rapidly. Unsurprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower such system and its deeper counterpart that adds more layers onto it. There exists a solution  $h(x)$  corresponding to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers in hand are unable to find solutions that



# Deep Computer Vision

## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

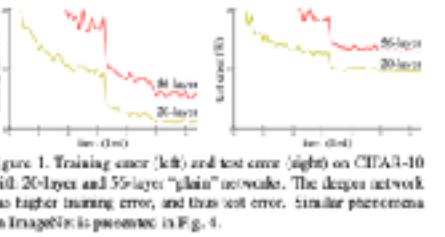
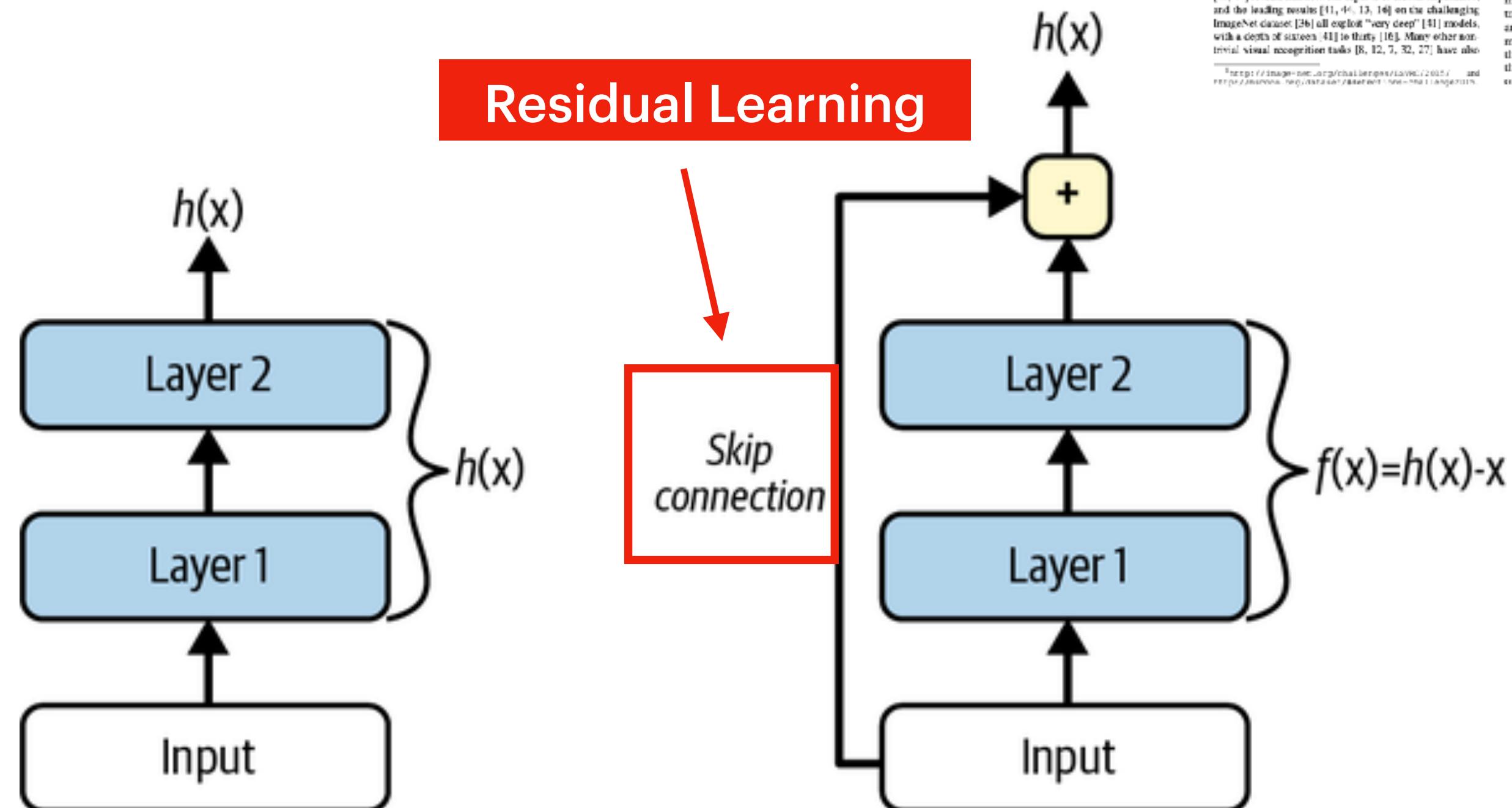


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layers and 35-layers "plain" networks. The deep network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

### Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs instead of learning the original functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate models with a depth of up to 152 layers—8 times deeper than VGG nets [44] but still having fewer parameters. An ensemble of three residual networks achieves 5.37% error on the ImageNet test set. This result was the 1<sup>st</sup> place on the ILSVRC 2015 classification task. We also present analysis on ImageNet with 100 and 1000 layers.

The depth of representation is of crucial importance for many visual recognition tasks. Solely due to our extremely deep representation, we achieve a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundation of our submission to ILSVRC & COCO 2015 competitions<sup>1</sup>, where we also won the 1<sup>st</sup> places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 49]. Deep networks naturally integrate low-level features [50] and classifiers in an end-to-end multi-layer fashion, and the "levels" of features can be measured by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [11, 41, 13, 16] on the challenging ImageNet dataset [36] all employ "very deep" [31] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 5, 32, 27] have also shown the benefits of deep learning.

<sup>1</sup><http://image-net.org/challenges/LSVRC/2015/> and <http://msrazone.net/datasets/detection.html>

Training the significance of depth, a question arises: "Is training deeper networks as easy as stacking more layers?" An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [25, 9, 37, 13] and intermediate scale normalization [16], which make networks with tens of layers in fact converging for stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been reported: while the network depth increasing, accuracy gets saturated (which might be unsatisfying) and then degrades rapidly. Unsurprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution  $h(x)$  corresponding to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers in hand are unable to find solutions that



Pontificia Universidad  
JAVERIANA  
Colombia

# Deep Computer Vision

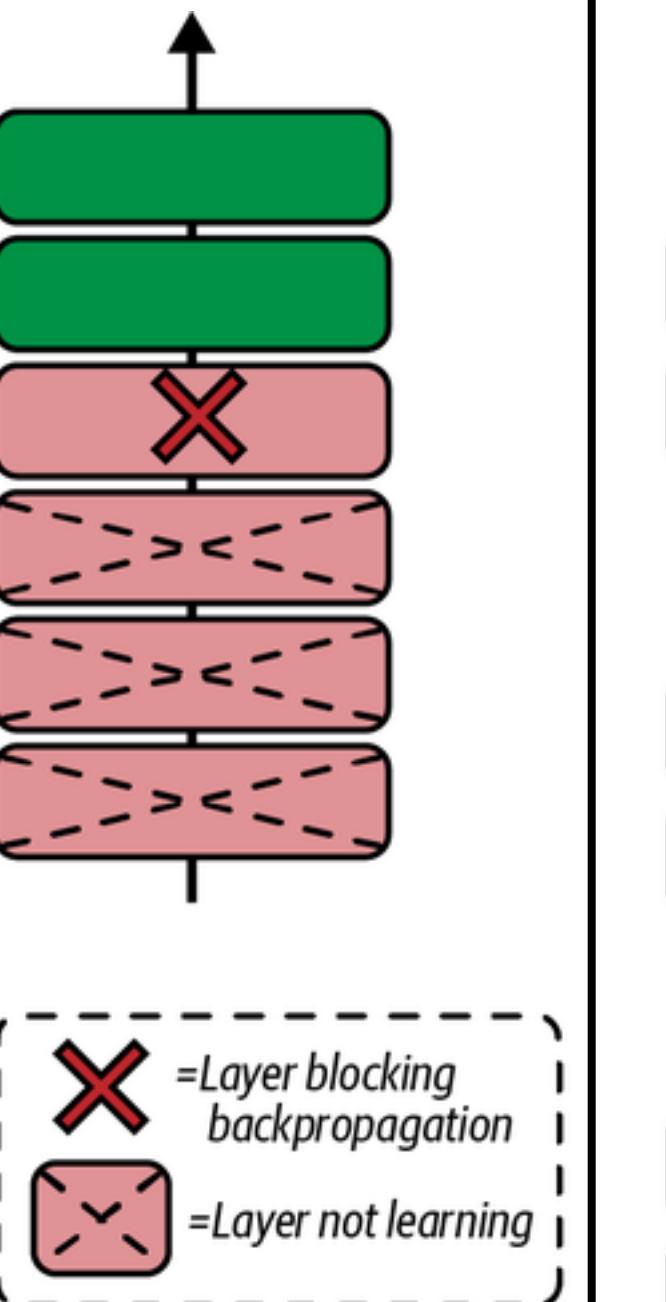
## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

Regular Neural Network



Residual Neural Network

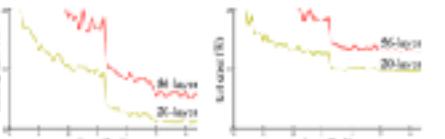
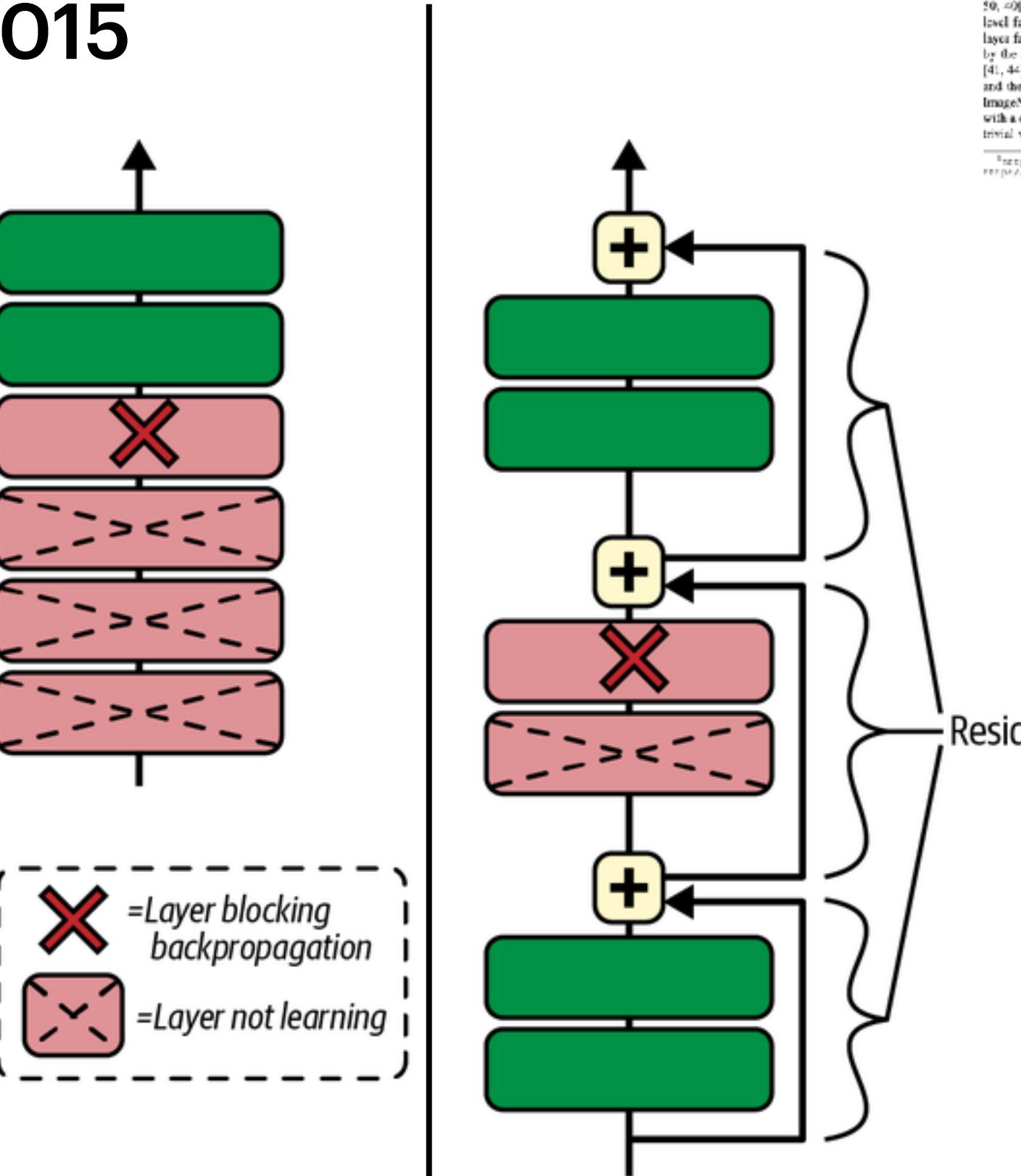


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layers and 35-layers "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

### Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs instead of learning unparameterized functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate models with a depth of up to 152 layers— $\times$  deeper than VGG nets [44] but still having fewer parameters. An ensemble of three residual networks achieves 5.3% error on the ImageNet test set. This result beats the *In-place* on the ILSVRC 2015 classification task. We also present analysis on ImageNet with 100 and 1000 layers.

The depth of representation is of crucial importance for many visual recognition tasks. Solely due to our extremely deep representation, we achieve a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundation of our submission to ILSVRC & COCO 2015 competitions<sup>1</sup>, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low-level features [50] and classifiers in an end-to-end multi-layer fashion, and the "levels" of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [11, 4, 13, 16] on the challenging ImageNet dataset [36] all exploit "very deep" [41] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also shown the power of deep learning.

<sup>1</sup><http://image-net.org/challenges/lsvc09/2015/> and <http://msrazone.net/datasets/imagenet-2015/>

Training by the significance of depth, a question arises: *Is training deeper networks as easy as stacking more layers?*

An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hampers convergence from the beginning. This problem, however, has been largely addressed by normalization layers [25, 9, 37, 13] and intermediate scale normalization layers [16], which make networks with tens of layers tractable using stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been reported: while the network depth increases, accuracy gets saturated (which might be unsatisfying) and then degrades rapidly. Unsurprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *In-place* constraining to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers in hand are unable to find solutions that



Pontificia Universidad  
JAVERIANA

Colombia

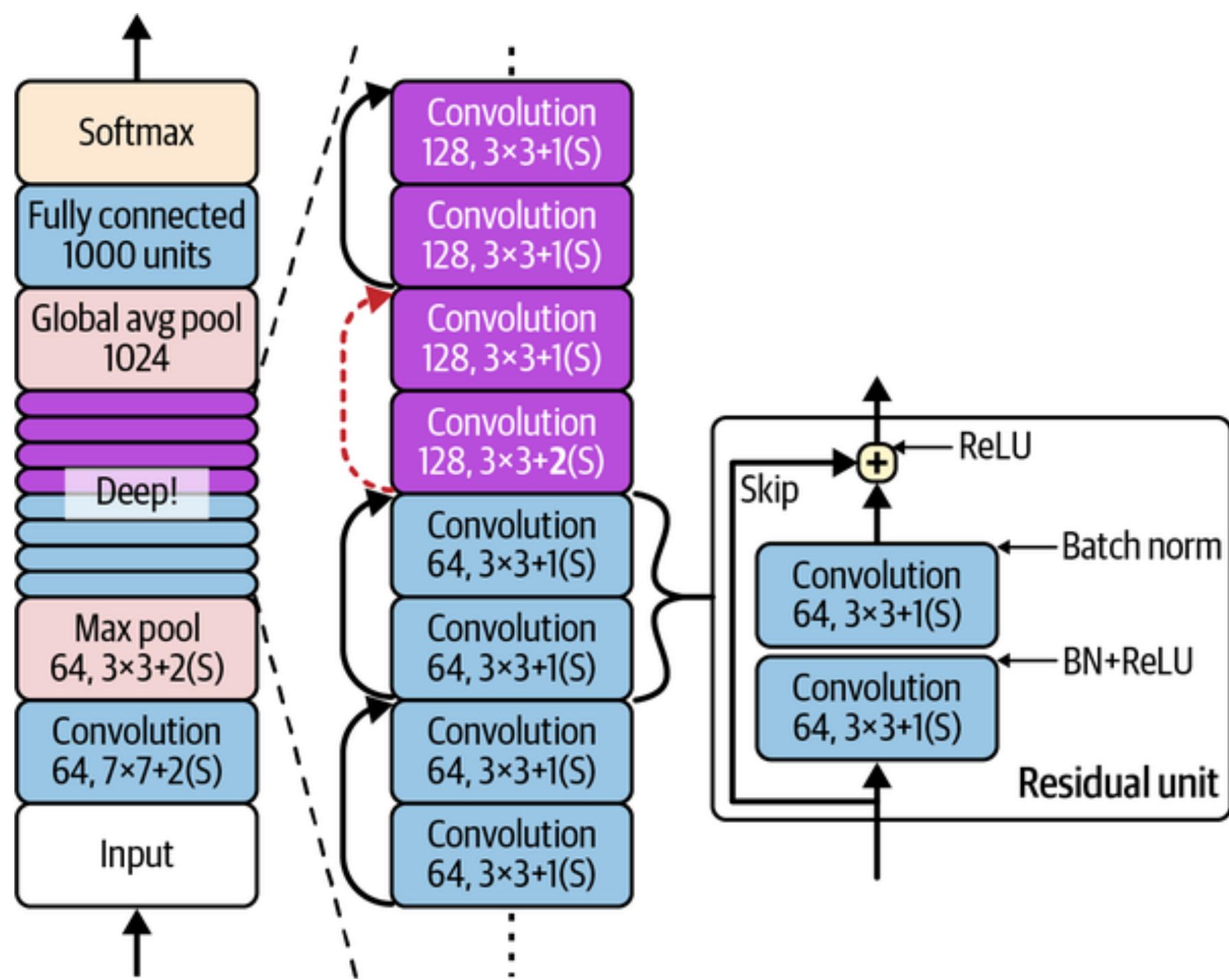
# Deep Computer Vision

## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning



Residual Neural Network

### Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs instead of learning unparameterized functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers— $\times$  deeper than VGG nets [44], but still having lower complexity. An ensemble of three residual nets achieves 5.37% error on the ImageNet test set. This result won the ILSVRC 2015 classification task. We also present analysis on ILSVRC 10 with 100 and 1000 layers.

The depth of representation is of crucial importance for many visual recognition tasks. Solely due to our extremely deep representation, we achieve a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundation of our submission to ILSVRC & COCO 2015 competitions<sup>1</sup>, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low-level features [50] and classifiers in an end-to-end multi-layer fashion, and the “levels” of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [11, 4, 13, 16] on the challenging ImageNet dataset [36] all employ “very deep” [31] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also placed their bets on deep learning.

<sup>1</sup><http://image-net.org/challenges/lsvc15/> and <http://msra-cv.tistory.com/entry/ResNet-DeepImageNet>

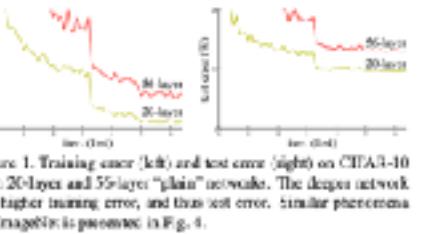


Figure 1: Training error (left) and test error (right) on CIFAR-10 with 20-layers and 35-layers. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models

Training by the significance of depth, a question arises: Is training deeper networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hampers convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [25, 9, 37, 13] and intermediate scale normalization [16], which make networks with tens of layers in a challenging task converge by stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been reported: while the network depth increases, accuracy gets saturated (which might be unsatisfying) and then degrades rapidly. Unsurprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *by construction* to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers in hand are unable to find solutions that

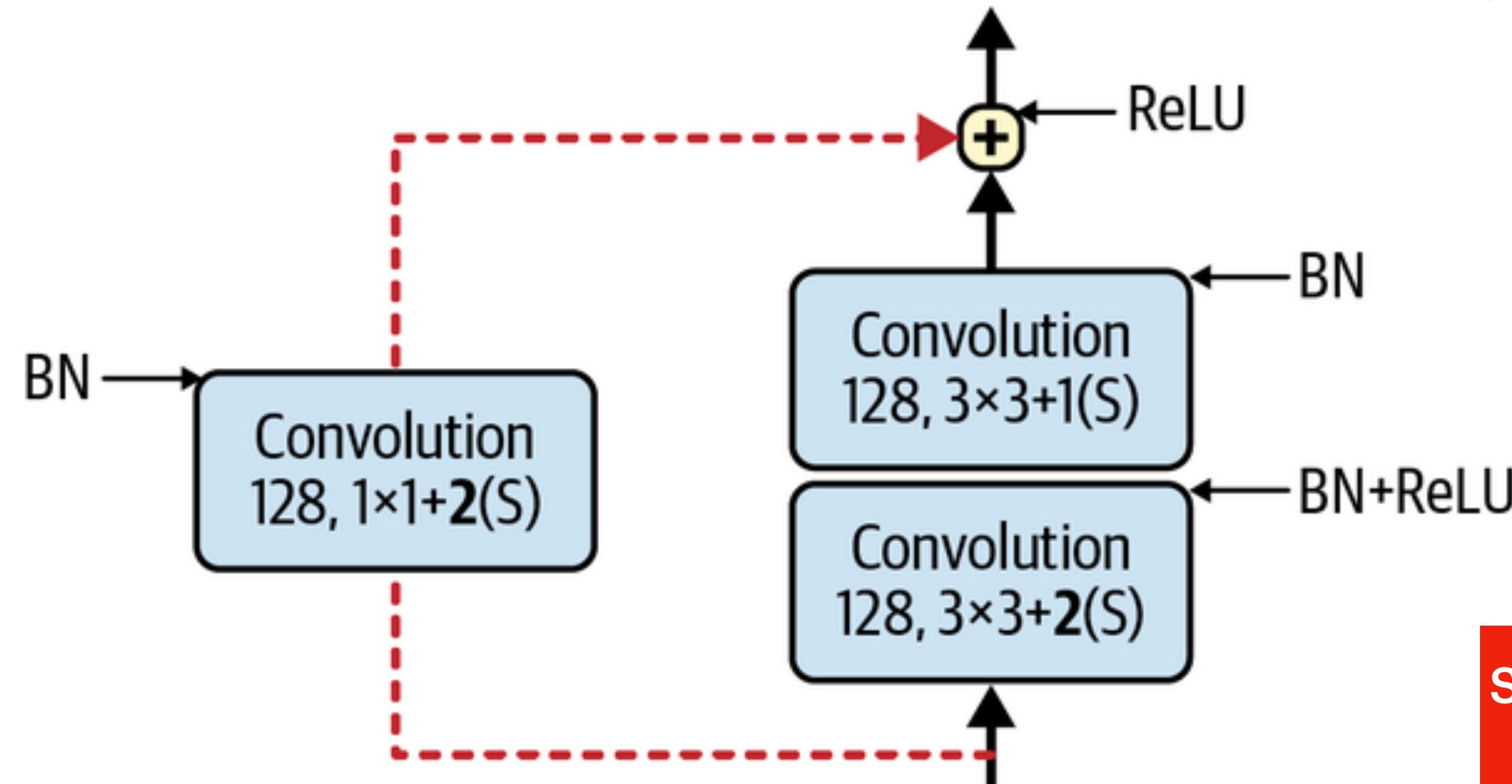
# Deep Computer Vision

## CNN Architectures

**ILSVRC ImageNet**

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning



Skip the connection when  
changing the size and  
depth of the feature map

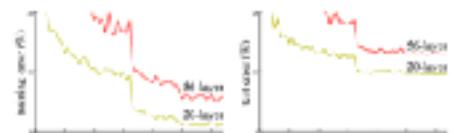


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layers and 35-layers. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

### Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs instead of learning unparameterized functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers— $\times$  deeper than VGG nets [44], but still having lower complexity. An ensemble of three residual nets achieves 5.37% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on ImageNet with 100 and 1000 layers.

The depth of representation is of crucial importance for many visual recognition tasks. Solely due to our extremely deep representation, we achieve a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundation of our submission to ILSVRC & COCO 2015 competitions<sup>1</sup>, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

### 1. Introduction

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low-level features [50] and classifiers in an end-to-end multi-layer fashion, and the ‘levels’ of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [11, 41, 13, 16] on the challenging ImageNet dataset [36] all employ “very deep” [41] models, with a depth of sixteen [41] to thirty [16]. Many other non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also

<sup>1</sup><http://image-net.org/challenges/LSVRC/2015/> and <http://msra-coco.csail.mit.edu/>

greatly benefited from very deep models.

Training the significance of depth, a question arises: Is training deeper networks as easy as stacking more layers? An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hampers convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [25, 9, 37, 13] and intermediate scale normalization [16], which make networks with tens of layers in fact converging for stochastic gradient descent (SGD) with back-propagation [22].

When deeper networks are able to start converging, a degradation problem has been reported: as the network depth increases, accuracy gets saturated (which might be unsatisfying) and then degrades rapidly. Unsurprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *by construction* to the deeper model: the added layers are identity mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers in hand are unable to find solutions that

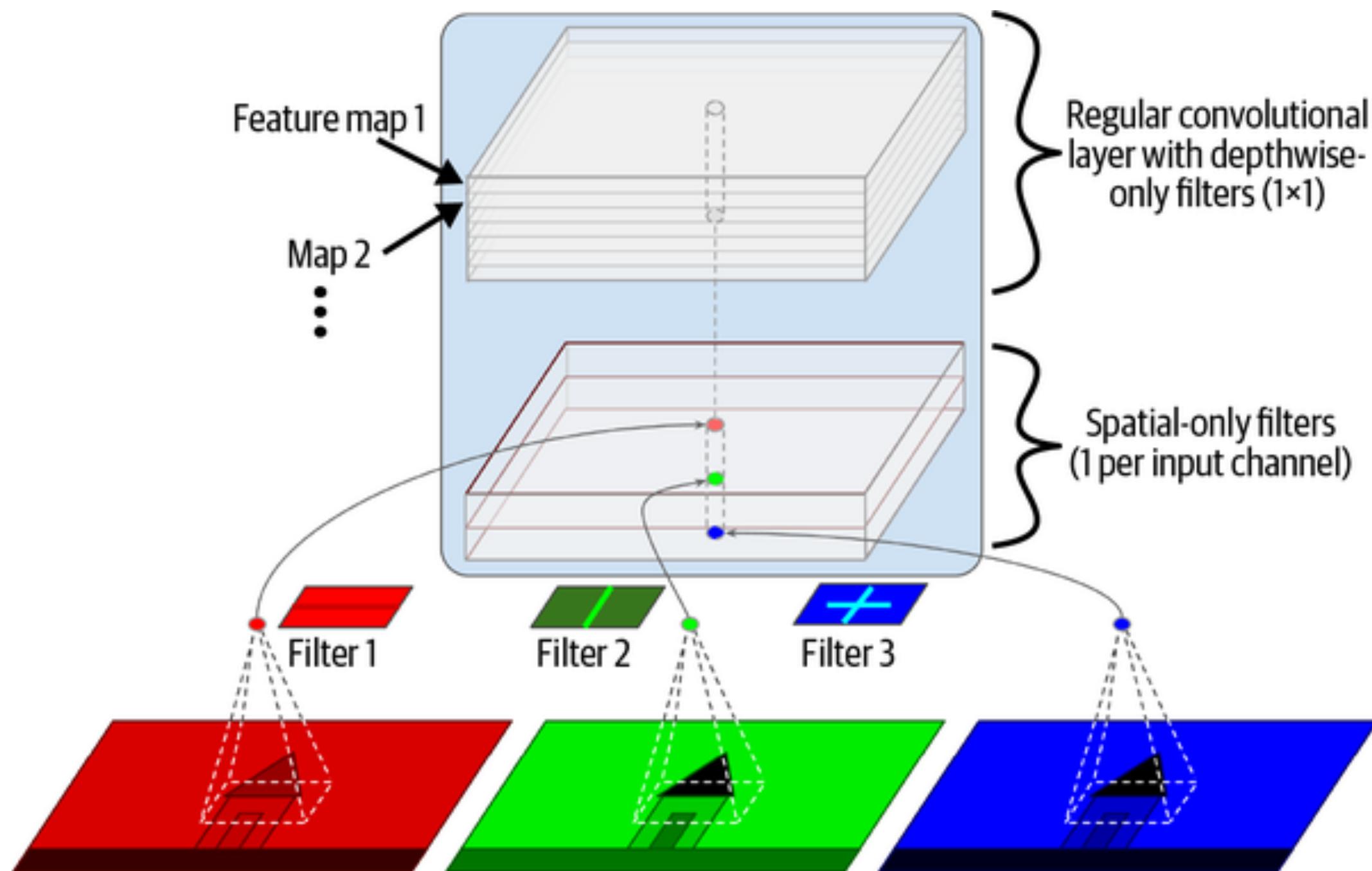
# Deep Computer Vision

## CNN Architectures

ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning



### Separable Convolutional Layers

- use fewer parameters
- less memory
- fewer computations
- they often perform better

#### Abstract

We present an interpretation of Inception modules as convolutional neural networks using depthwise separable convolutions instead of standard convolutions. We show that depthwise separable convolutions can be understood as Inception modules with an extremely large number of filters. This observation leads us to propose a new deep convolutional neural network architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions. We show that the new Inception module, closely equivalent to Inception V2 as we consider ImageNet Inception V3 has improved f1 and significantly outperforms Inception V3 on a large ImageNet validation dataset comprising 350 million images and 17,000 classes. Since the Inception architecture has the same number of operations as Inception V3, the performance gains are not due to increased capacity but rather to a more efficient use of model resources.

#### 1. Introduction

Convolutional neural networks have emerged as the main algorithm in computer vision in recent years, and developing recipes for designing them has been a subject of considerable attention. The history of convolutional neural networks, designated with LeNet-style models [13], which were simple stacks of convolution, local feature extraction and max pooling operations for spatial sub-sampling. In 2012, these ideas were refined into the AlexNet architecture [9], where convolution operations were being repeated multiple times between max pooling operations, allowing the network to learn more features at every spatial scale. What followed was a trend to make this style of network increasingly deeper, mostly driven by the yearly ILSVRC competition, first with Zeiler and Fergus in 2013 [25] and then with the VGG architecture in 2014 [26].

At Inception, a new style of network emerged, the Inception architecture, introduced by Szegedy et al. in 2014 [29].

Inception (Inception V1), Inception V2 [21], Inception V3 [22], and most recently Inception-ResNet [23]. Inception itself was inspired by the earlier Newell-DeNero architecture [11]. Since its first introduction, Inception has been one of the best performing family of models in the ImageNet dataset [16], as well as animal datasets in ImageNet Google, in particular IFT [5].

The fundamental building block of Inception cells used in the Inception module, of which several different versions exist (Figure 1), is shown by the general form of an Inception module, as found in the Inception V3 architecture. An Inception module can be understood as a stack of such modules. This is a departure from earlier VGG-style networks which were stacks of simple consecutive layers.

While Inception modules are conceptually similar to convolutional blocks they are considerably more complex; they originally appear to be capable of learning inter-relationships with less parameters. How they work, and how in they other than regular convolutions? What design strategies come into play?

#### 1.1. The Inception hypothesis

A convolutional hypothesis function takes a 3D space with 3 spatial dimensions (width and height) and a channel dimension. Thus a single convolution kernel is tested with an extremely large cross-channel correlations and spatial correlations.

The idea behind the Inception module is to make this process easier and more efficient by explicitly factoring it into a series of operations that would independently look at cross-channel correlations and at spatial correlations. More precisely, the typical Inception module will take a cross-channel convolution, via a set of local convolutions, mapping the input feature space, and then a sum of convolutions in the output feature space, via a set of 1x1 convolutions. This is illustrated in Figure 1. It offers the fundamental hypothesis behind Inception that cross-channel correlations and spatial correlations are sufficiently decoupled that it is possible to factorize them jointly.

As a result of this process it is independently looking at the two components.

# Deep Computer Vision

## CNN Architectures

### ILSVRC ImageNet

Error rate from 26% to 2.3%  
in 6 years!!

- 256 pixels height
- 1000 class (120 dog race)
- Outputs describe CNN evolution
- Research progresses in deep learning

### Other Architectures:

ResNeX

MobileNet

EfficientNet

DenseNet

CSPNet

YOLO