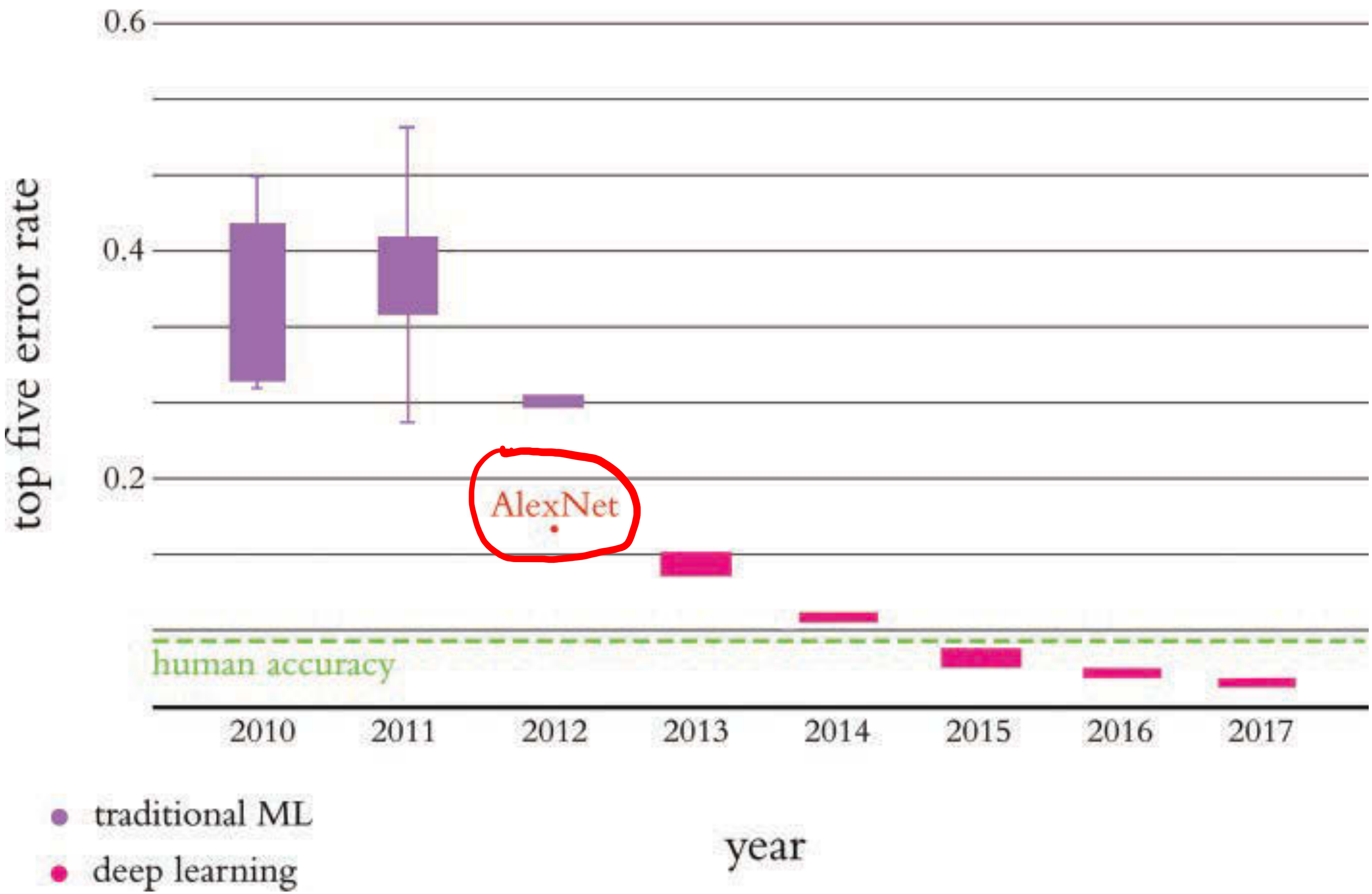
https://www.sli.do/
#073374

Deep Learning for Computer Vision (I)

Learning Objectives

- Learn the vast applications of deep learning for computer vision.
- Learn the basics of digital image representation
- Learn the workhorse: convolutional neural networks (CNNs)
- Learn the basic components and theories behind CNNs.

ILSVRC (the ImageNet Large Scale Visual Recognition Challenge)



沈向洋：以 Deep Learning 為核心的 Computer Vision，十年內將全面取代人眼 (2019.10.31 【與AI大師沈向洋博士對話】)

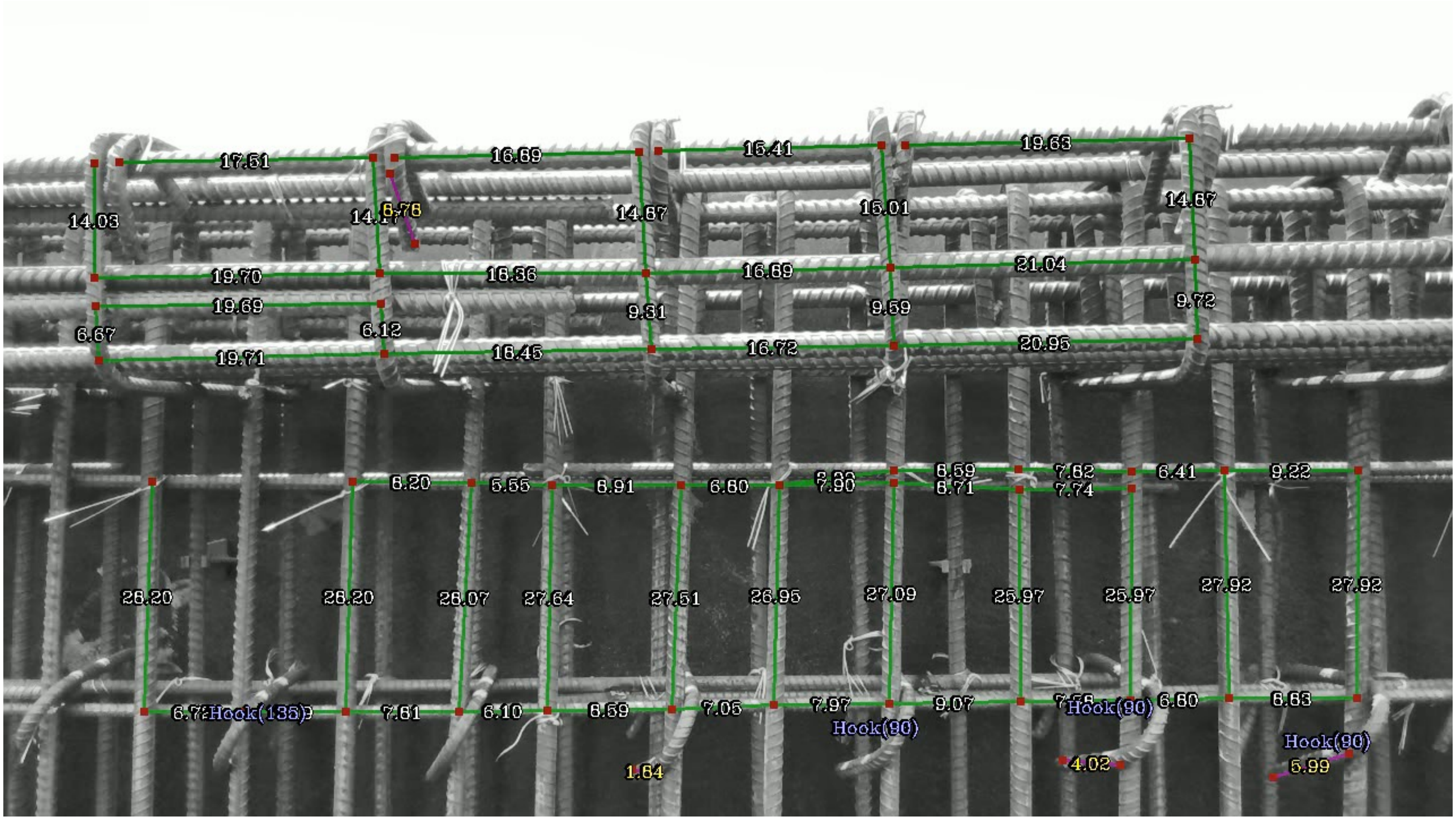
Tesla released what Autopilot's neural net can **see** (2020.01.31)

Apply cutting-edge research to train **deep neural networks** on problems ranging from perception to control. Our per-camera networks analyze raw images to perform **semantic segmentation**, **object detection** and **monocular depth estimation**. Our birds-eye-view networks take video from all cameras to output the **road layout**, **static infrastructure** and **3D objects** directly in the top-down view.

Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train 🔥. Together, they **output 1,000 distinct tensors (predictions)** at each timestep.

https://www.tesla.com/autopilotAI

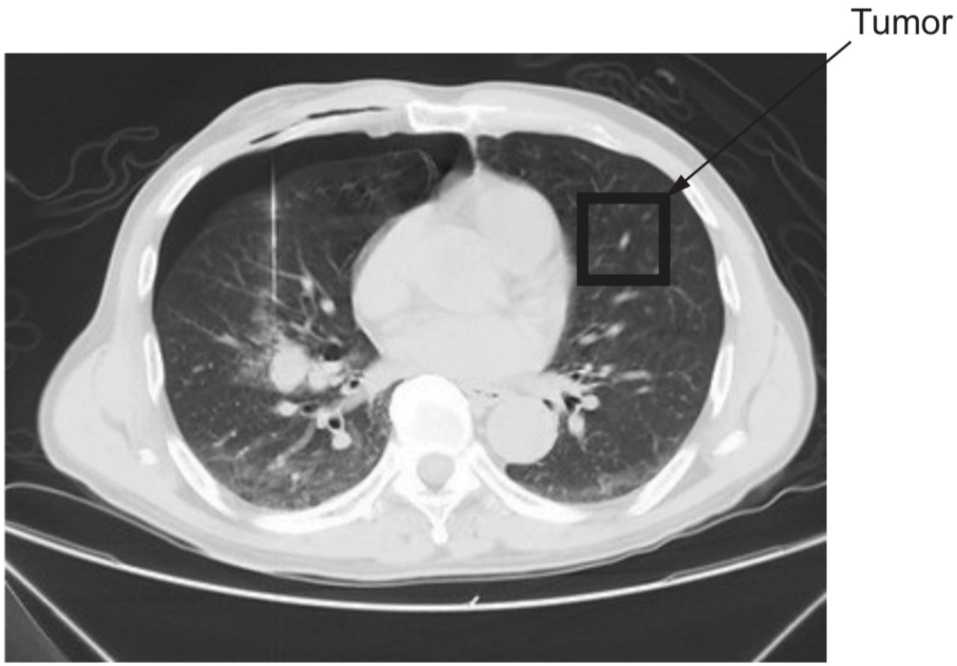
工地鋼筋全檢測


17.51
16.69
15.41
19.63
14.03
6.78
14.67
15.01
14.67
19.70
16.36
16.69
21.04
19.69
9.31
9.59
9.72
6.67
6.12
19.71
16.45
16.72
20.95
6.20
5.55
6.91
6.80
7.90
6.59
6.71
7.62
7.74
6.41
9.22
26.20
26.20
26.07
27.64
27.51
26.95
27.09
25.97
25.97
27.92
27.92
Hook(135)
7.61
6.10
6.59
7.05
7.97
9.07
Hook(90)
6.60
6.63
Hook(90)
1.64
4.02
Hook(90)
5.99

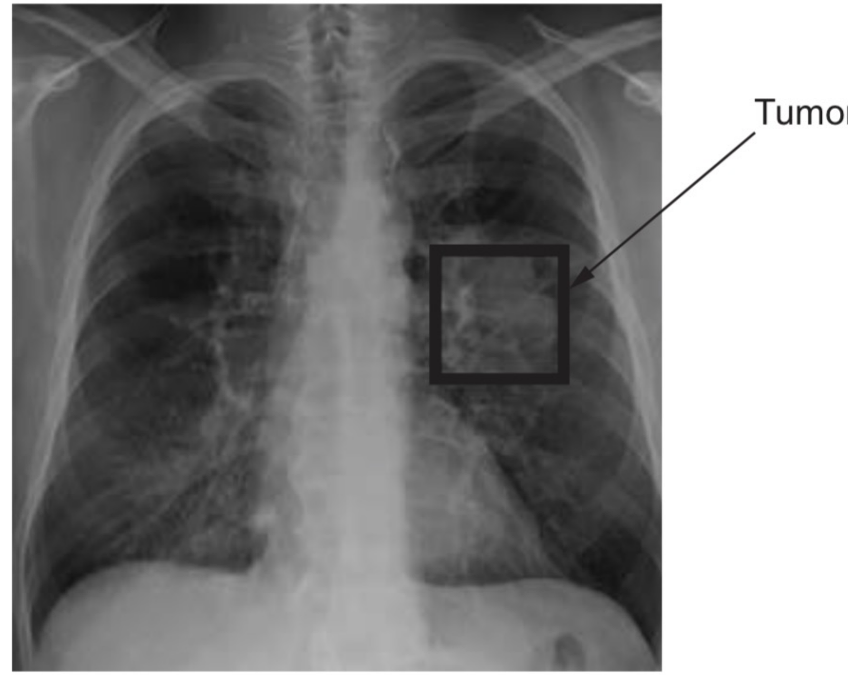
監造影像智慧加值：運用AI輔助現場工安管理

Source：趙志偉 研發工程師 中興工程顧問研發及資訊部

Applications of Computer Vision



CT scan



X-ray

Figure 1.5 Vision systems are now able to learn patterns in X-ray images to identify tumors in earlier stages of development.

M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision

Figure 1.6 Vision systems can detect traffic signs with very high performance.

M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision

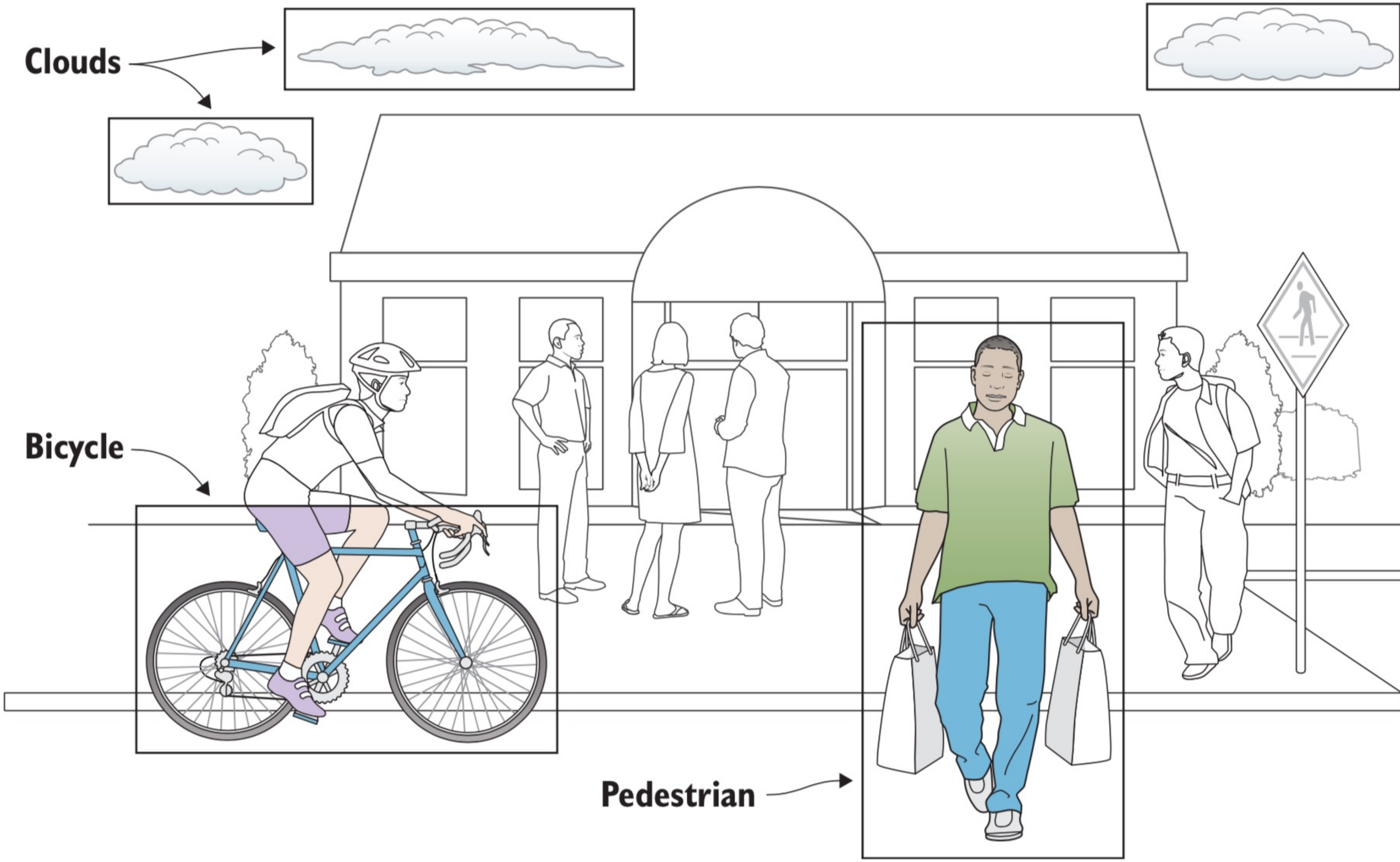


Figure 1.7 Deep learning systems can segment objects in an image.

M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision

Figure 1.8 Style transfer from Van Gogh's *The Starry Night* onto the original image, producing a piece of art that feels as though it was created by the original artist

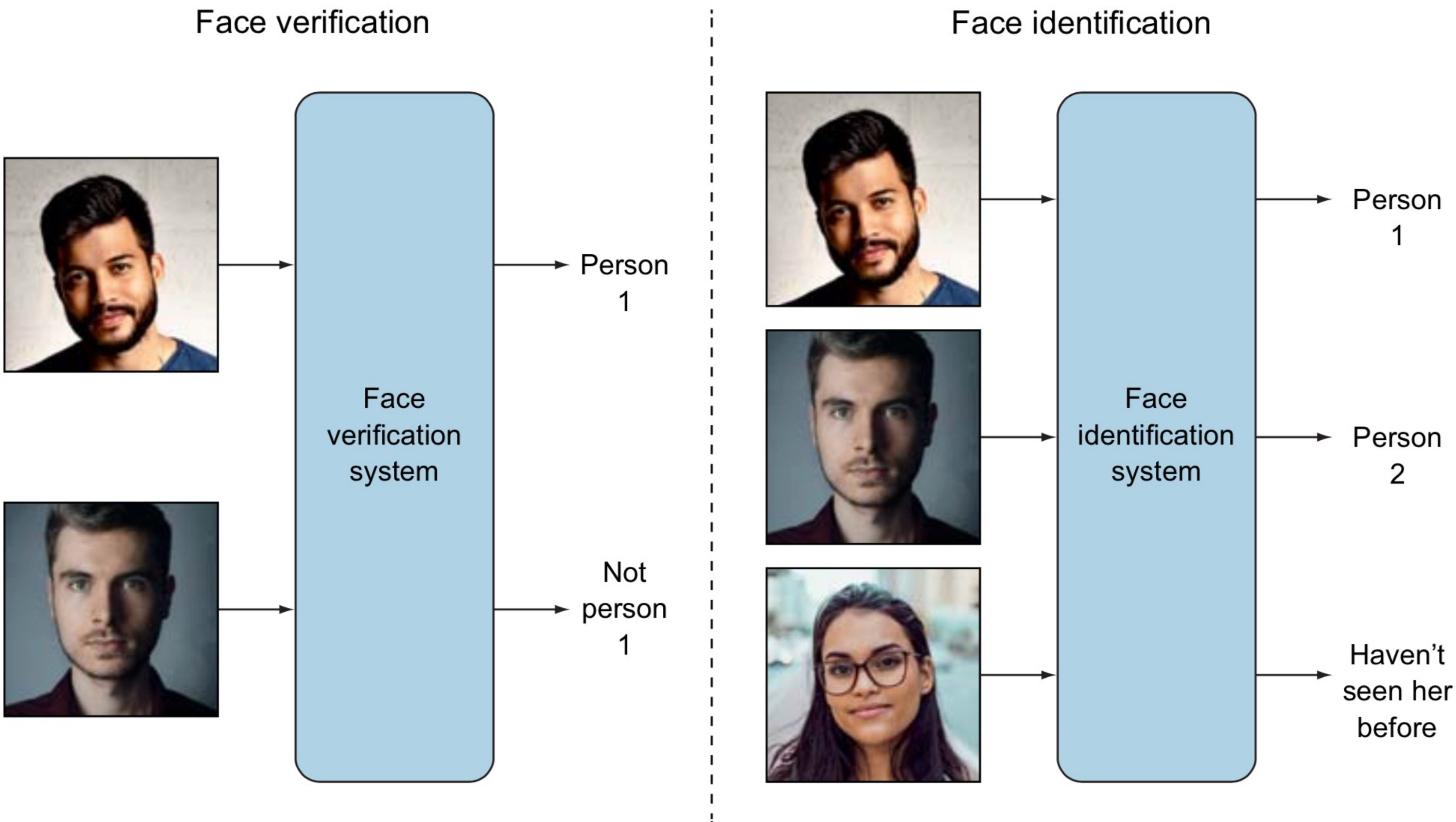
M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision

Figure 1.9 Generative adversarial networks (GANS) can create new, “made-up” images from a set of existing images.

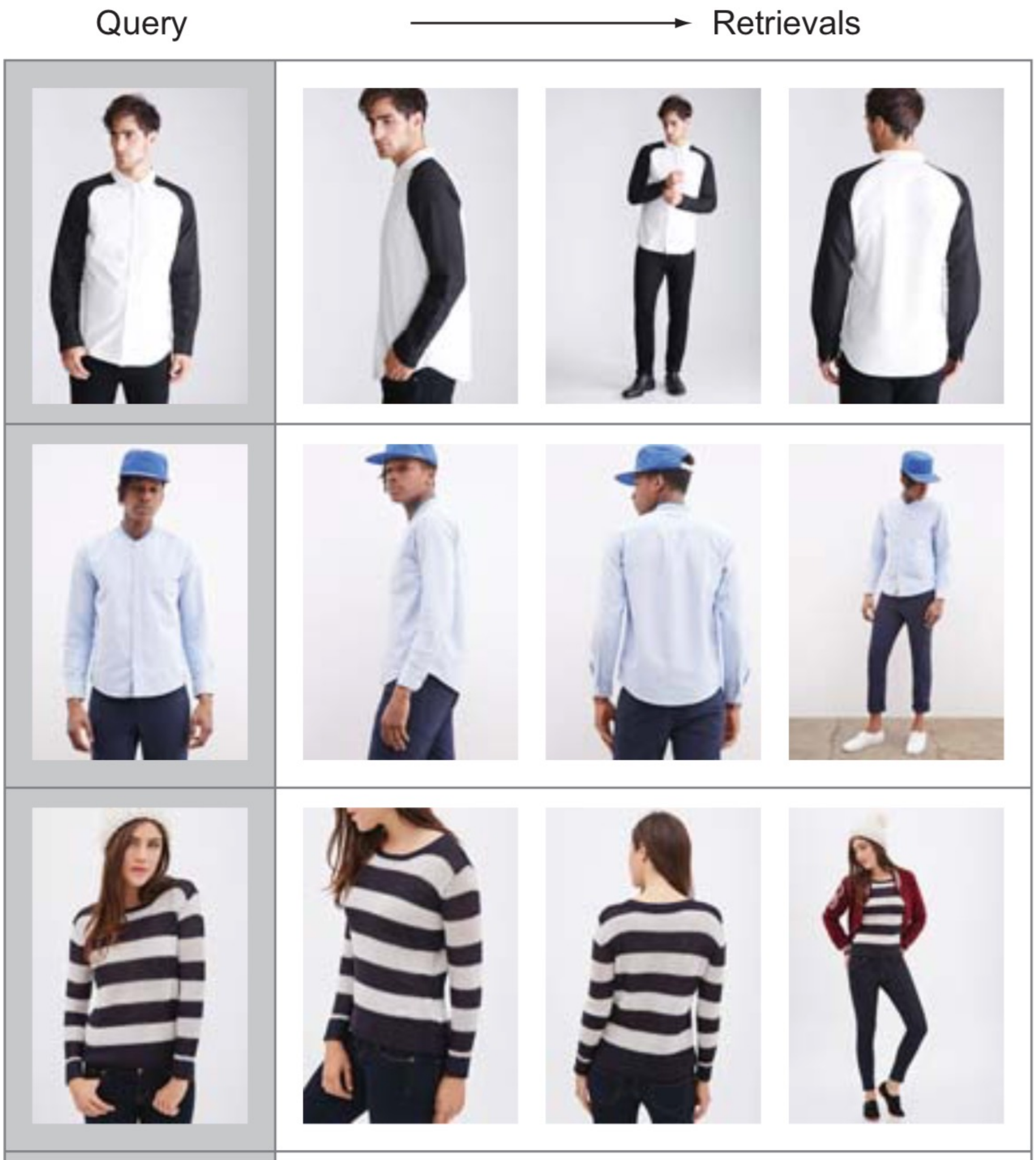
M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision



M. Elgendy (2020) Deep Learning for Vision System.

Applications of Computer Vision



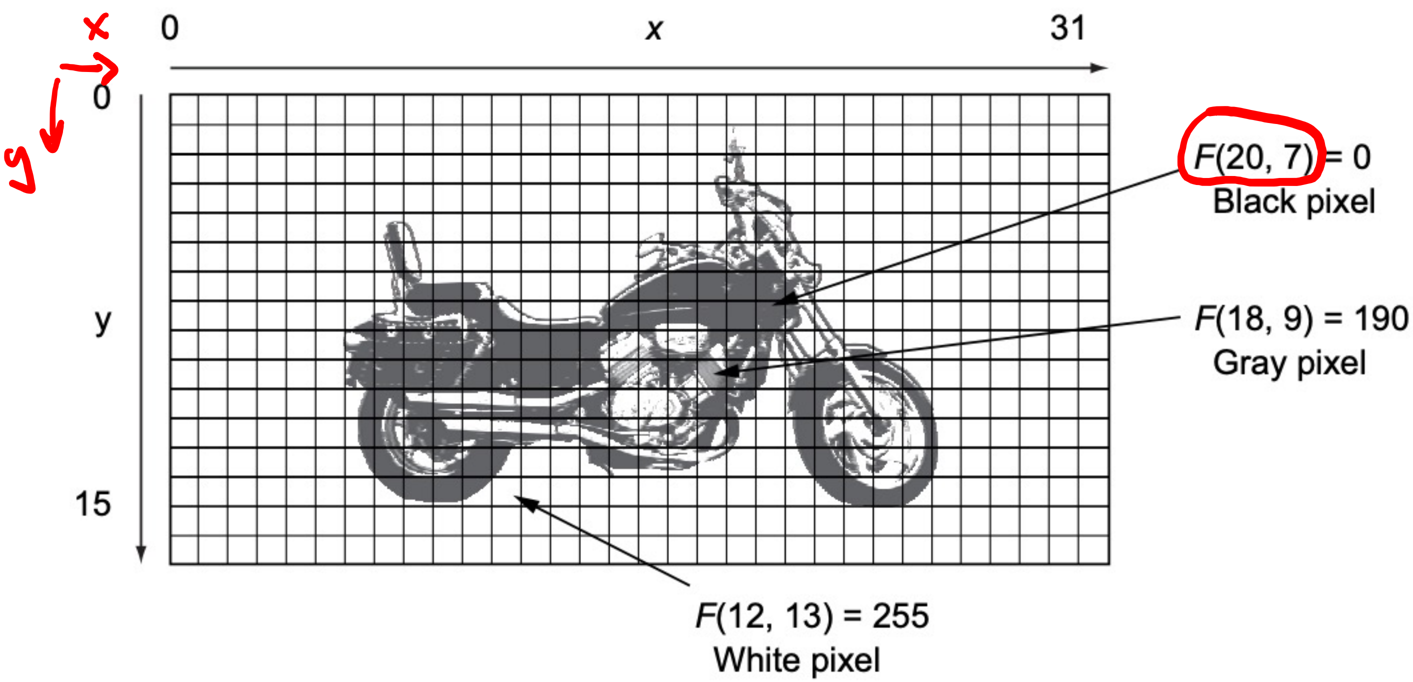
M. Elgendy (2020) Deep Learning for Vision System.

Image Representation 101

Image as a Function

- An image can be represented as a function of two variables *x* and *y*, which define a two- dimensional area. A digital image is made of a grid of pixels.
- The ***pixel*** is the raw building block of an image. Every image consists of a set of pixels in which their values represent the ***intensity*** of light that appears in a given place in the image.

0.. 255



How Computer See Image

- To a computer, the image looks like a 2D matrix of the pixels' values, which represent intensities. There is no context here, just a massive pile of data.

28 × 28
= 784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	87	157	156	187	215	81	0	0	0	0	0
0	0	0	0	0	0	0	0	5	25	54	25	96	140	215	215	251	254	254	254	254	254	163	0	0	0	0	0
0	0	0	0	0	0	0	0	157	254	254	254	0	254	254	254	254	254	254	254	254	241	100	0	0	0	0	0
0	0	0	0	0	0	0	0	156	117	184	214	214	156	117	49	19	19	141	254	241	158	23	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	173	254	246	70	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	90	247	254	219	58	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	43	203	254	250	121	15	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	43	145	254	254	229	111	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	4	137	245	254	254	238	40	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	130	254	254	254	254	254	235	106	23	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	55	196	196	114	194	223	254	254	216	23	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	75	245	253	55	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	150	254	134	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	200	254	134	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	246	253	59	0	0	0	0	0	0	0	0	0
0	0	0	17	11	18	0	0	0	0	0	0	0	0	67	254	254	170	0	0	0	0	0	0	0	0	0	0
0	0	0	128	254	159	0	0	0	0	0	0	20	140	248	254	254	9	0	0	0	0	0	0	0	0	0	0
0	0	0	154	254	229	80	79	79	161	176	218	254	254	254	104	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	29	203	254	254	254	215	254	254	254	250	95	11	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	5	80	155	155	156	111	58	58	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

How Computer See Color Image

- Color images have three channels (red, green, and blue) and are often represented by three matrices: one represents the intensity of red in the pixel, one represents green, and one represents blue

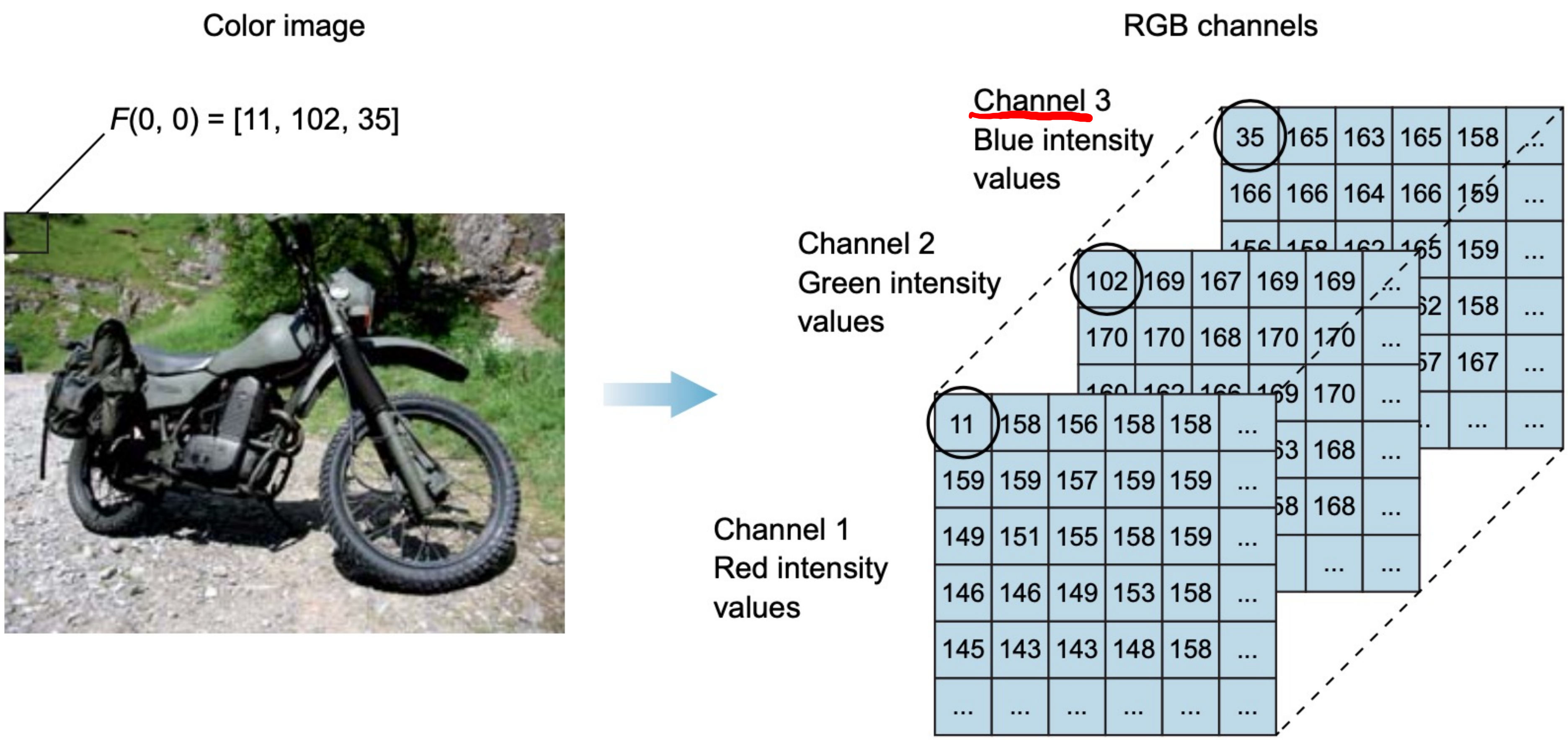
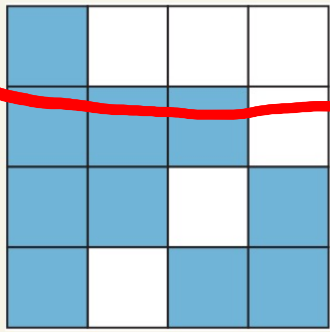


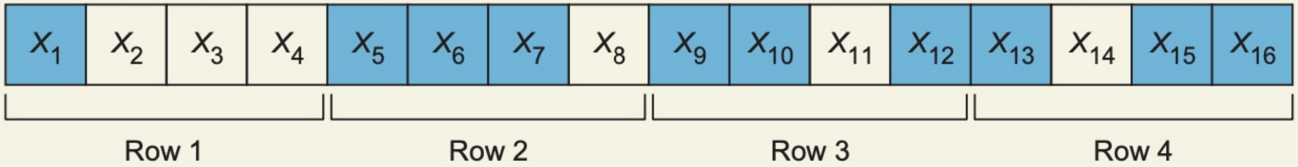
Image Flattening

- We often need to do image flattening when we use MLP for classification.

To help visualize the flattened input vector, let's look at a much smaller matrix (4, 4):



The input (x) is a flattened vector with the dimensions (1, 16):

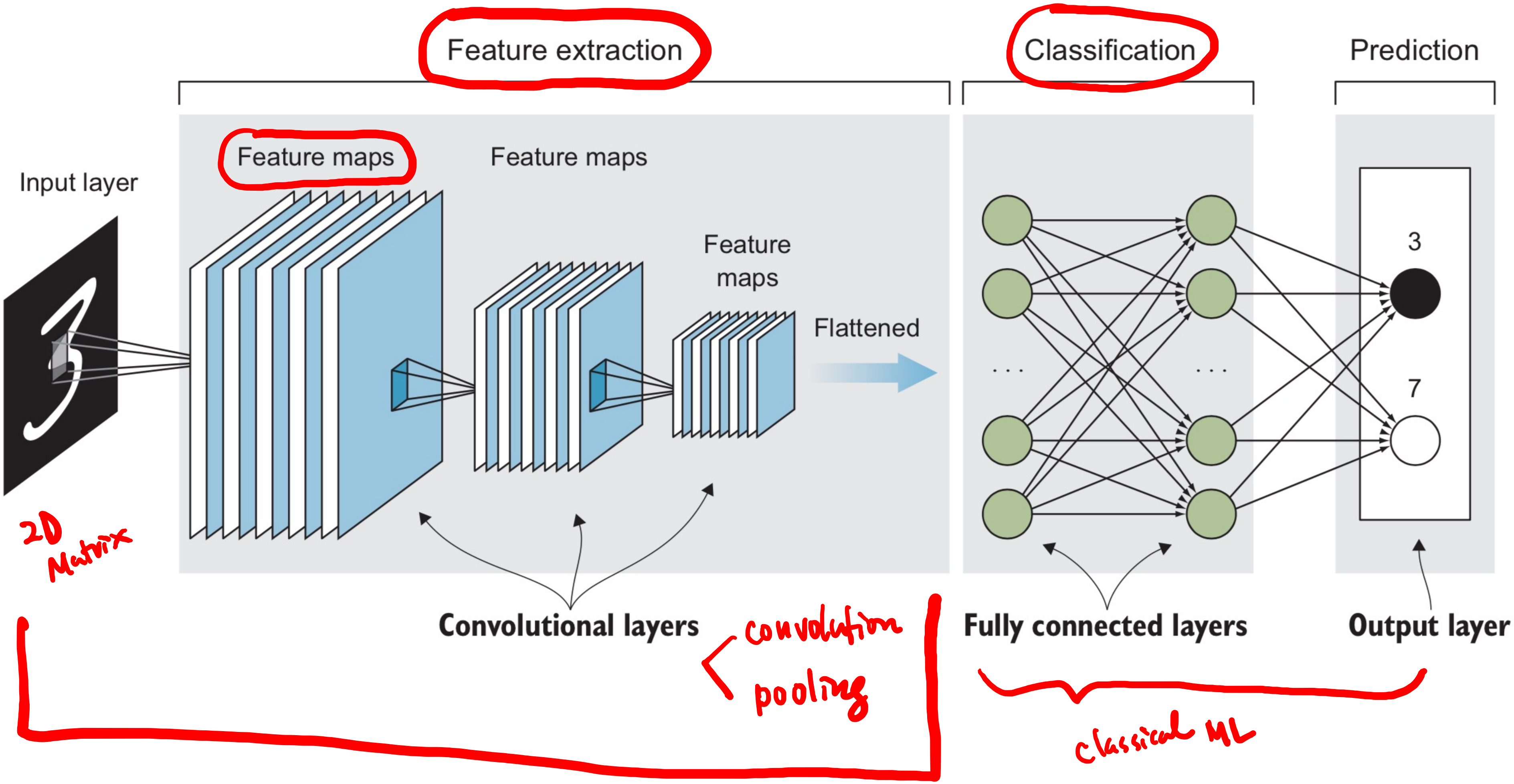


So, if we have pixel values of 0 for black and 255 for white, the input vector will be as follows:

Input = [0, 255, 255, 255, 0, 0, 0, 255, 0, 0, 255, 0, 0, 255, 0, 0]

Convolutional Neural Networks (CNNs)：An Overview

High-level architecture of CNNs: **input layer, convolutional layers, fully connected layers, and output prediction**



DNNs vs. CNNs: See CNN_MNIST.ipynb

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 784)               615440
_________________________________________________________________
dense_1 (Dense)              (None, 10)                7850
=================================================================
Total params: 623,290
Trainable params: 623,290
Non-trainable params: 0
_________________________________________________________________
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                36928
_________________________________________________________________
dense_5 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
```

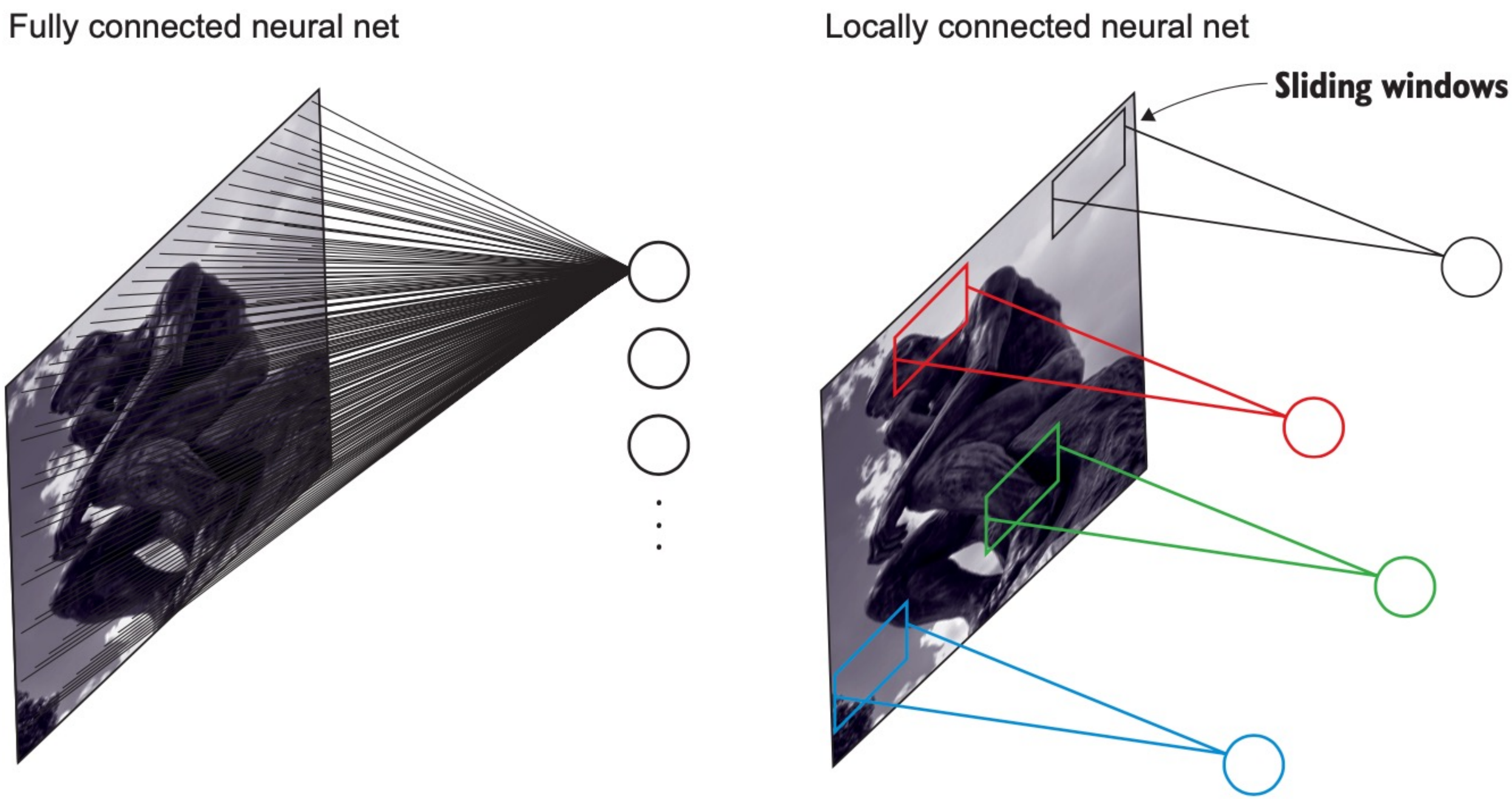
784

* CNNs yield better results with less training parameters

* DNNs and CNNs do not usually yield comparable results; MNIST dataset is an exception. In messy real-world image data, CNNs truly outshine DNNs.

Fundamental Difference between DNNs and CNNs

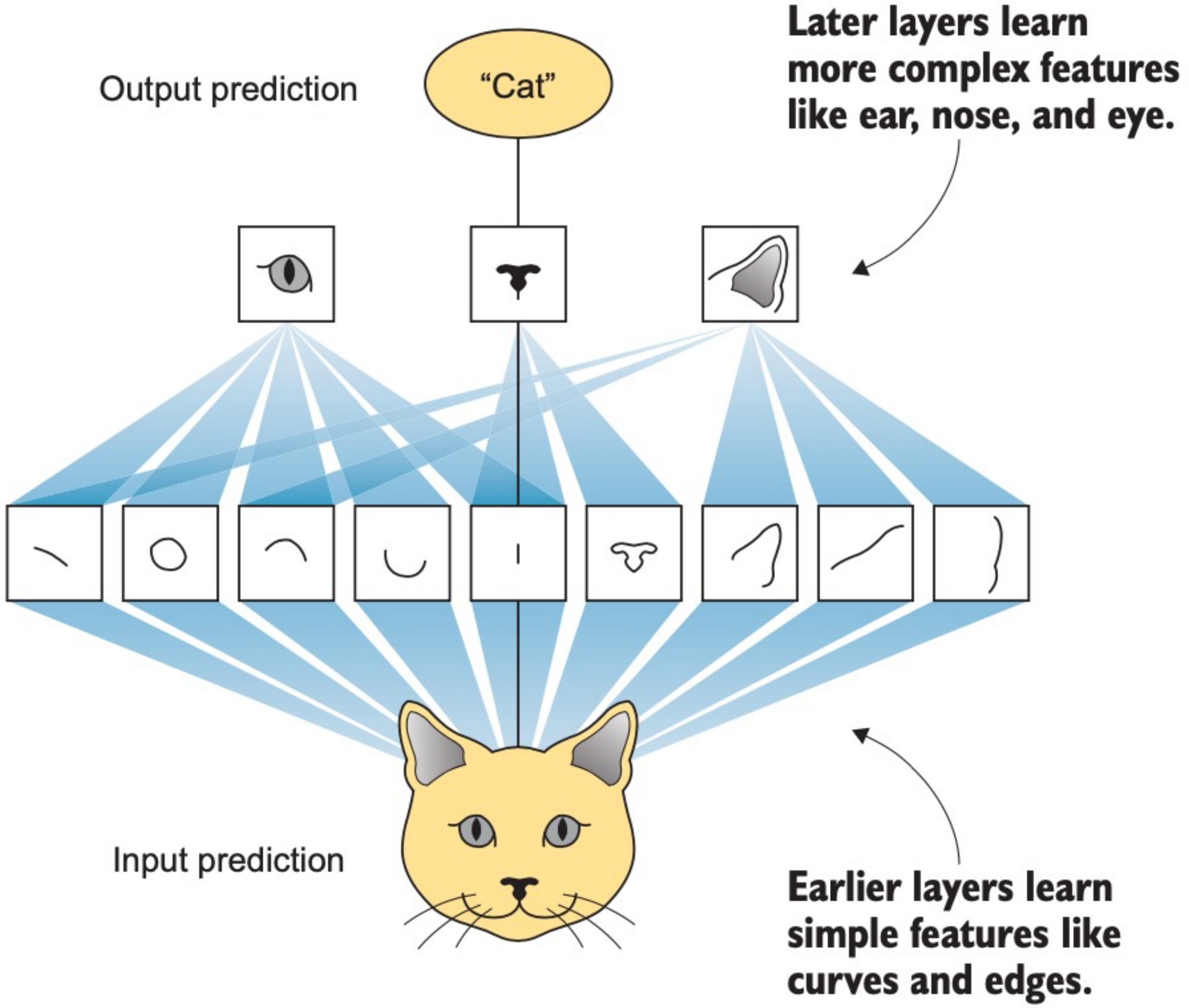
Global vs. Local



The patterns CNNs learn are translation invariant.
CNNs can learn spatial hierarchies of patterns.

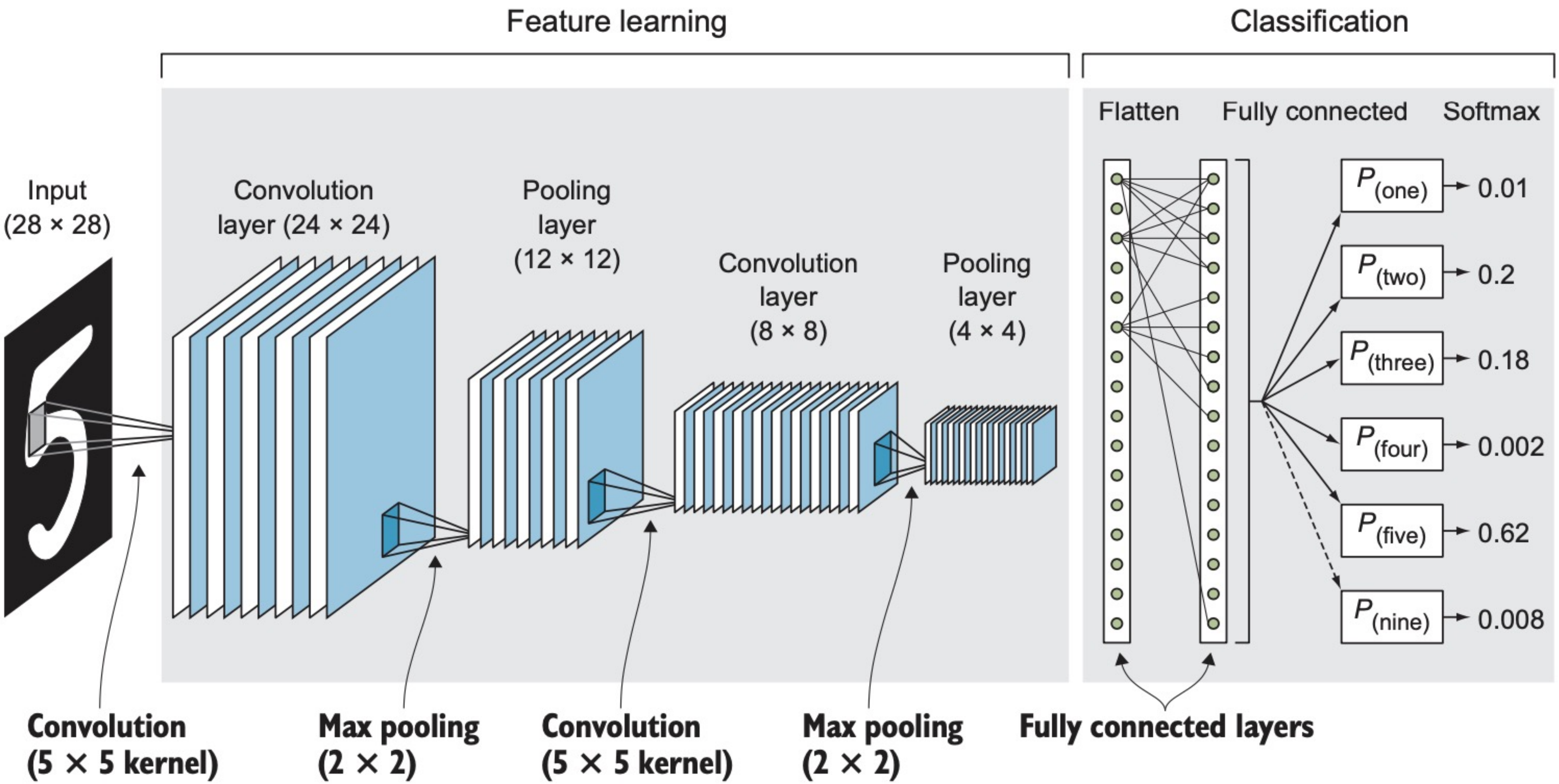
CNNs learn the image features through its layers.

The patterns CNNs learn are translation invariant.
CNNs can learn spatial hierarchies of patterns.



Basic Components of Convolutional Neural Networks (CNNs)

The basic components of convolutional networks are convolutional layers and pooling layers to perform feature extraction, and fully connected layers for classification



Basic components of CNNs: theoretical minimum and example

- The phrase “theoretical minimum” is taken from a very successful book series written by Leonard Susskind, a great physicist at Stanford University.
- “Theoretical minimum” means just the minimum theories and equations you need to know in order to proceed to the next level.
- See CNN_Basics.pdf