EECE-4710 “IoT and TinyML”

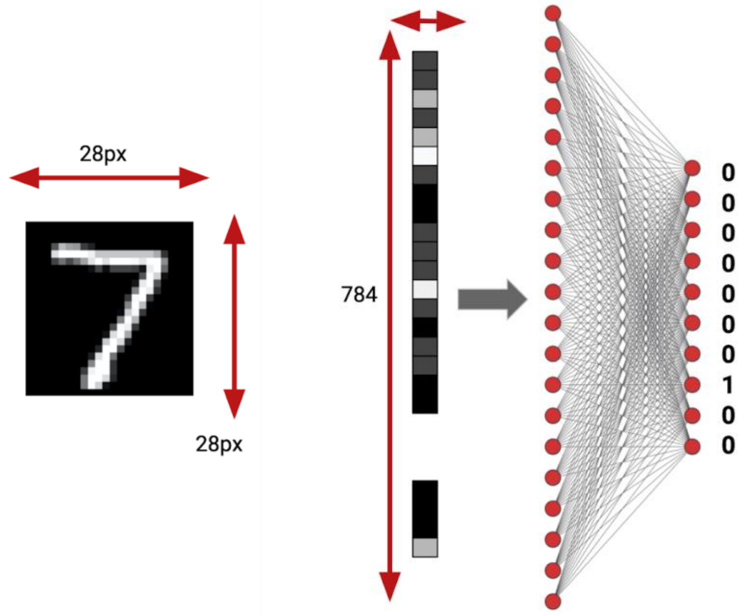
Image Classification with Convolutional Neural Networks (CNN)

Cris Ababei

MARQUETTE UNIVERSITY

BE THE DIFFERENCE.

Introducing Convolutions



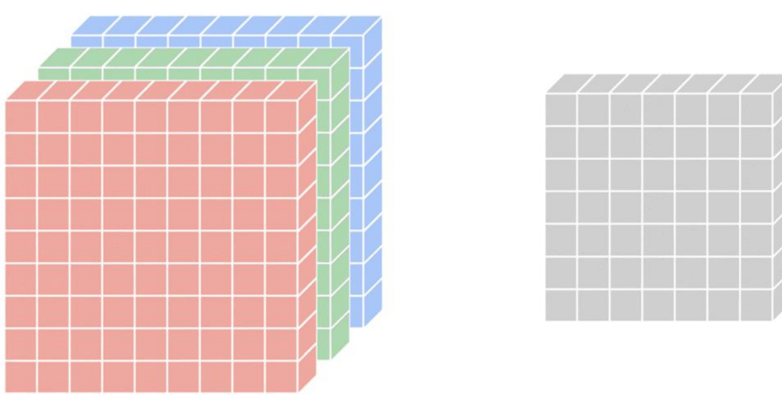
3

4

Standard Convolution (1 Channel)

5

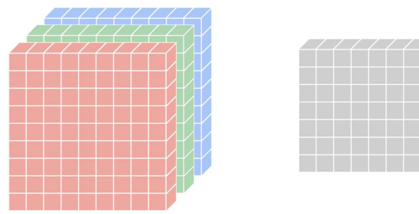
Standard Convolution ***(3 Channel****—e.g., RGB****)***



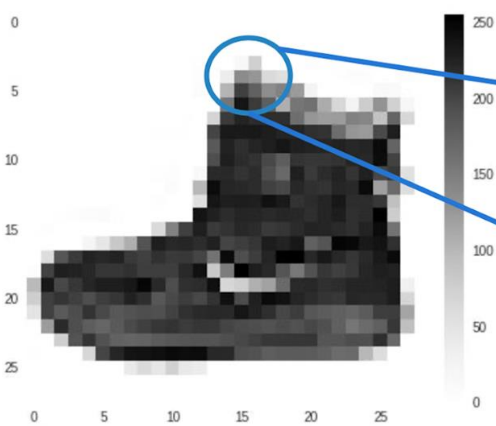
6

Standard Convolution ***(3 Channel****—e.g., RGB)*

- Input Feature Map
  - 8 × 8 × 3
  - Width × Height × Channels
- **Kernel (*1 Filter*)**
  - 3 × 3 × 3



7



0	64	128
48	192	144
142	226	168

Current Pixel Value is 192

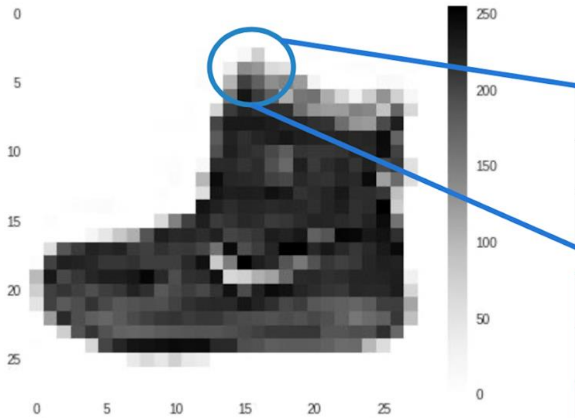
Consider neighbor Values

-1	0	-2
.5	4.5	-1.5
1.5	2	-3

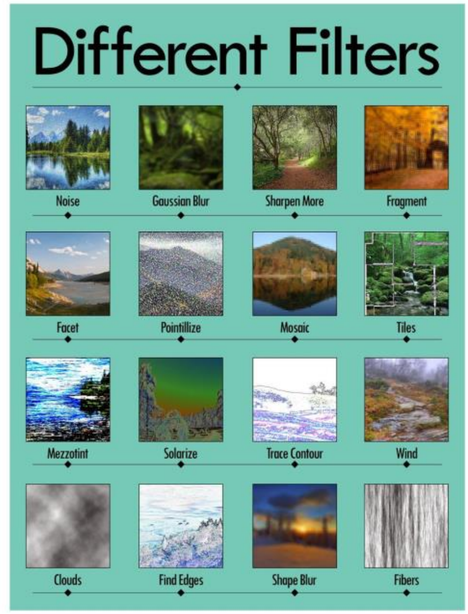
Filter Definition

```
CURRENT_PIXEL_VALUE = 192
NEW_PIXEL_VALUE = (-1 * 0) + (0 * 64) + (-2 * 128) +
                  (.5 * 48) + (4.5 * 192) + (-1.5 * 144) +
                  (1.5 * 42) + (2 * 226) + (-3 * 168)
```

8



Kernels = Filters



9

Image Kernels

-1	0	1
-2	0	2
-1	0	1

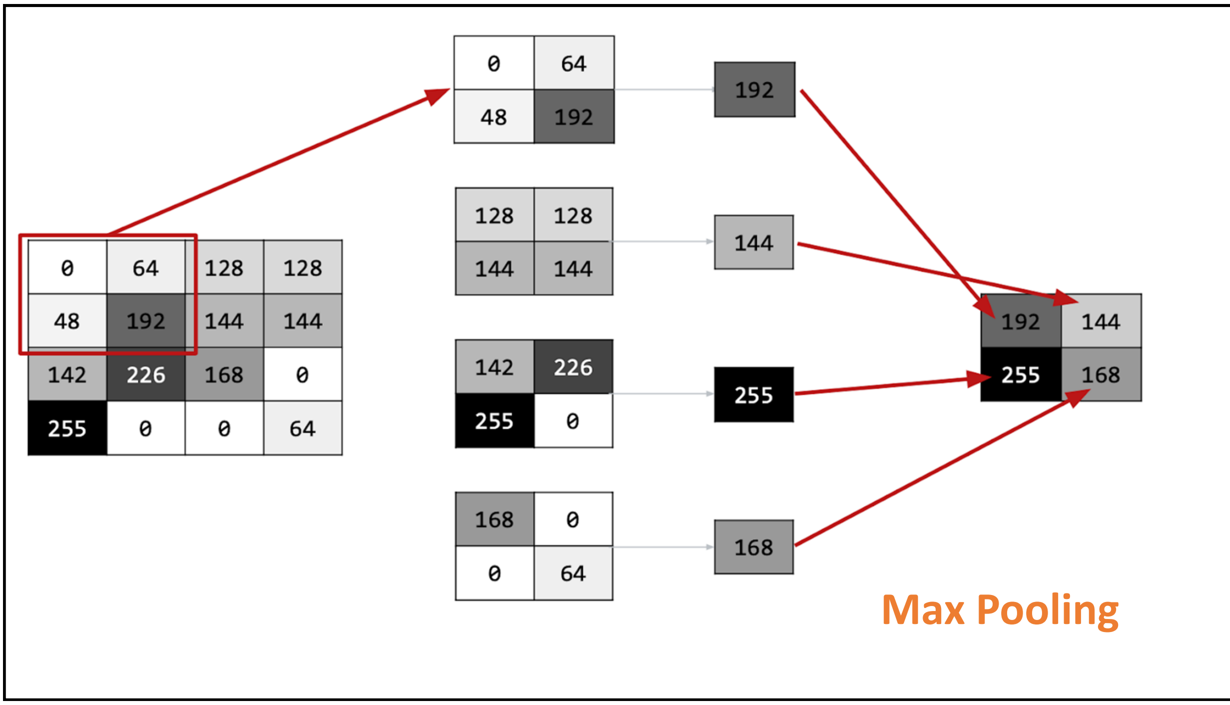
custom

-1	-2	-1
0	0	0
1	2	1

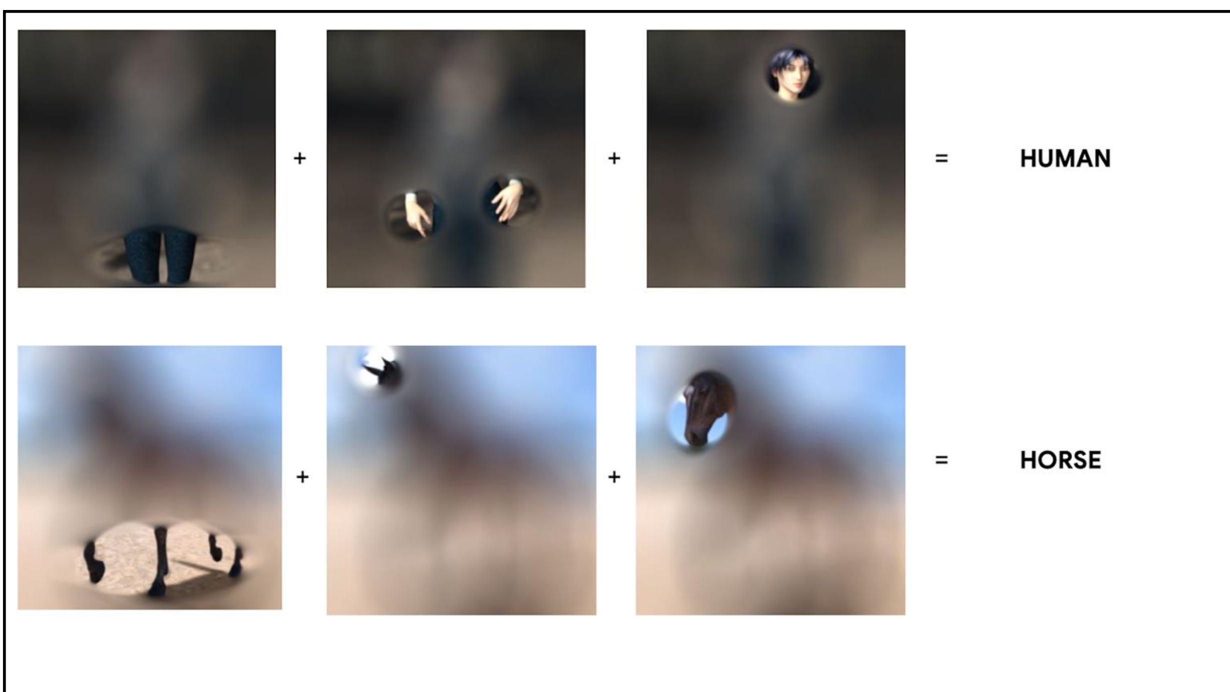
custom

https://setosa.io/ev/image-kernels/

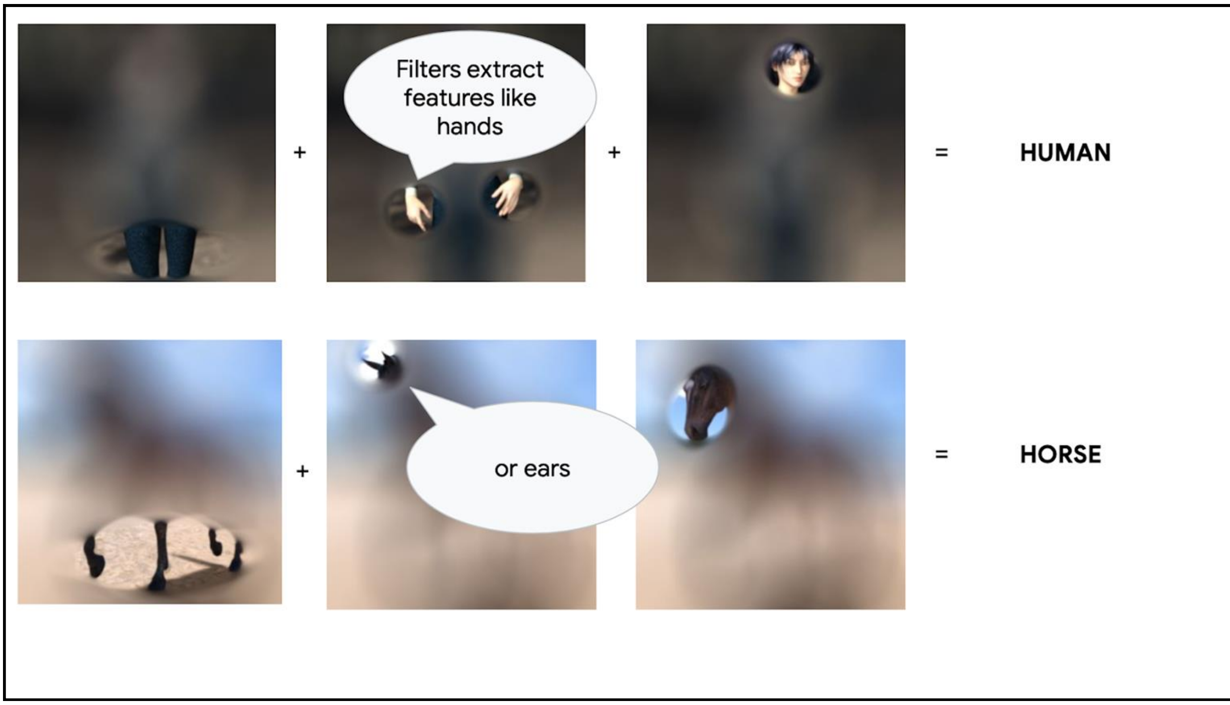
10


0 64 128 128
48 192 144 144
142 226 168 0
255 0 0 64
0 64
48 192
128 128
144 144
142 226
255 0
168 0
0 64
192
144
255
168
192 144
255 168
Max Pooling

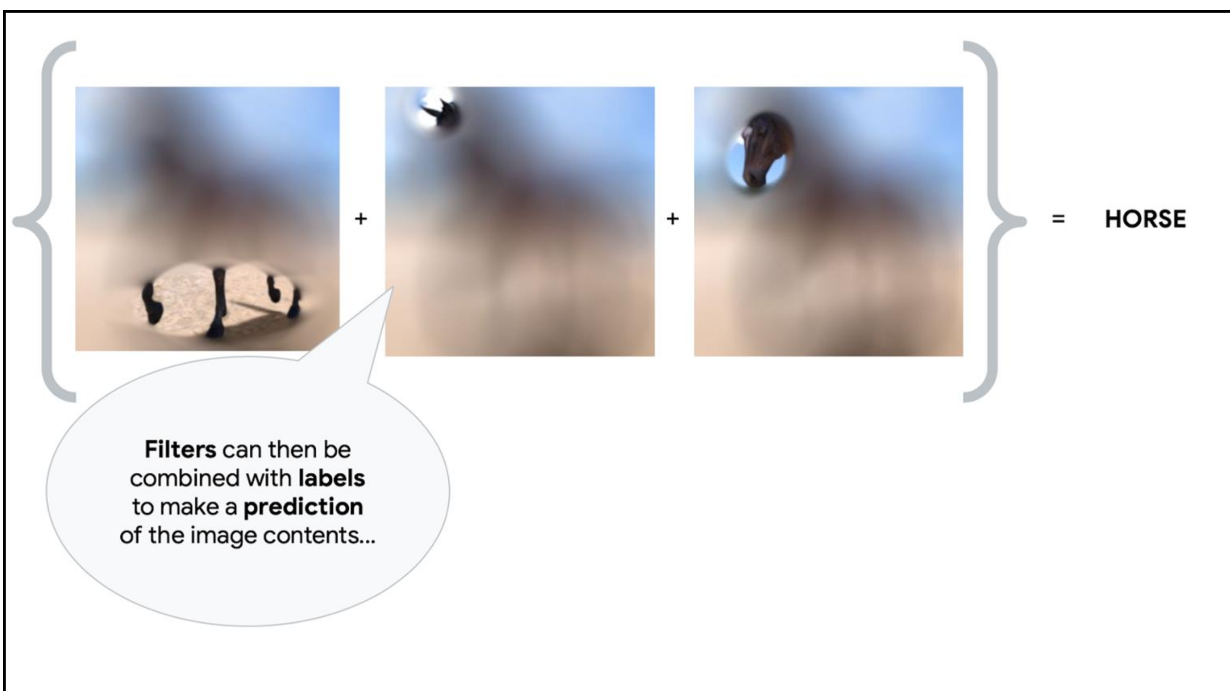
11


+
+
=
HUMAN
+
+
=
HORSE

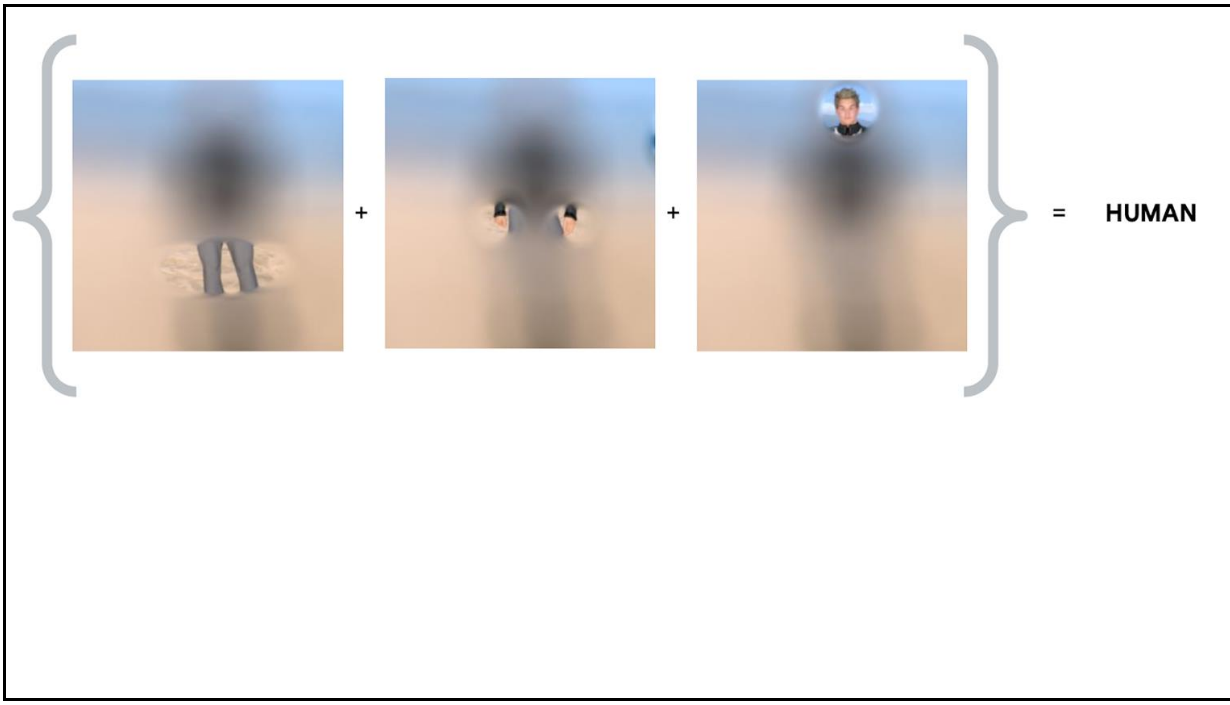
12


+
Filters extract features like hands
+
=
HUMAN
+
or ears
=
HORSE

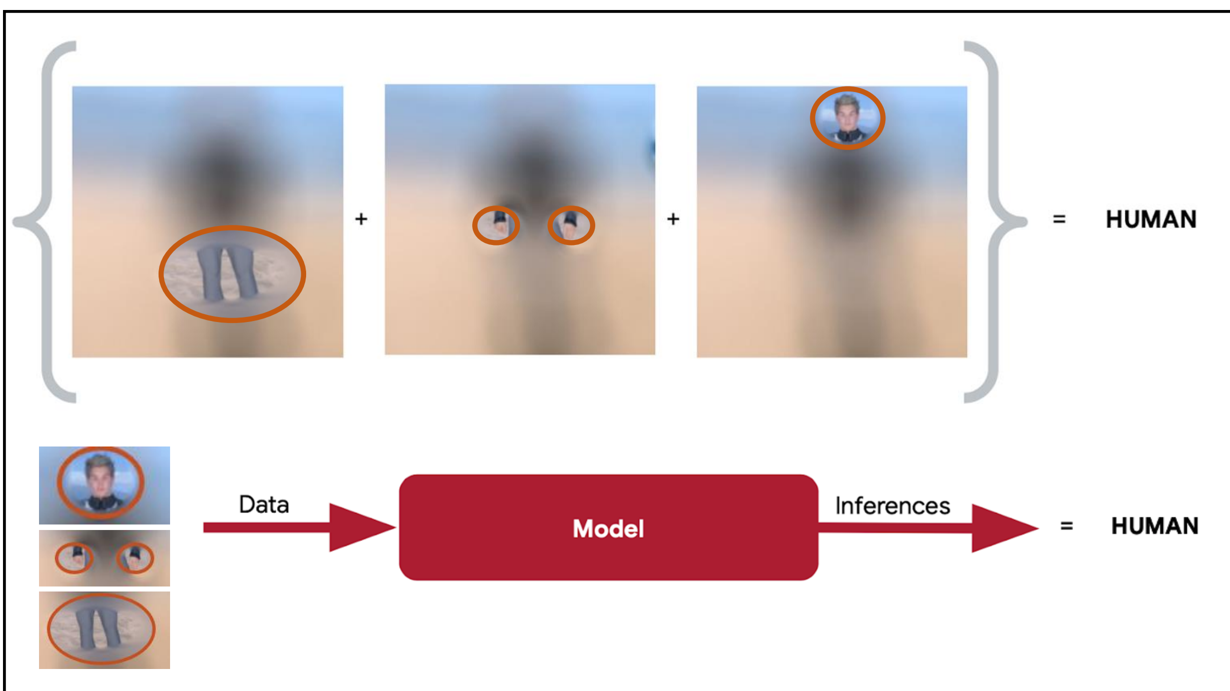
13


+
+
=
HORSE
Filters can then be combined with **labels** to make a **prediction** of the image contents...

14

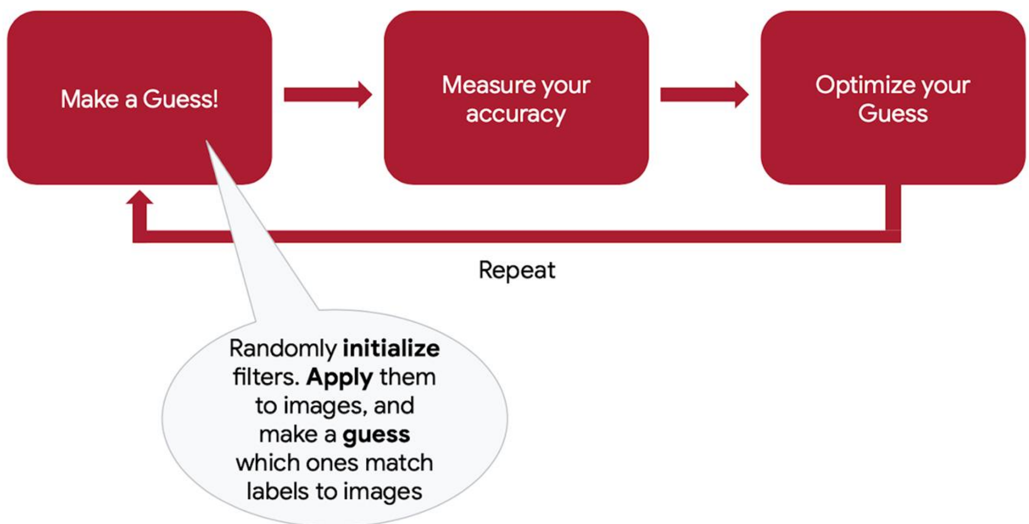


15



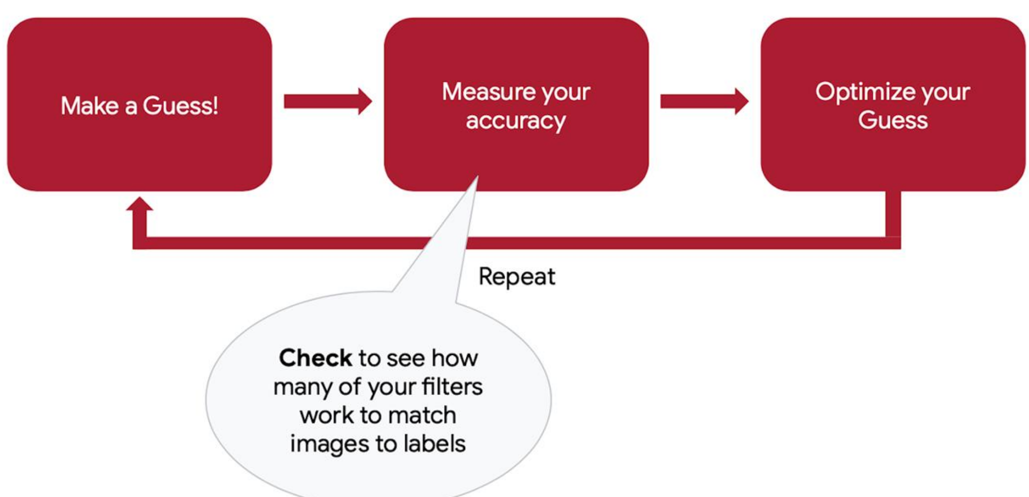
16

The Machine Learning Paradigm



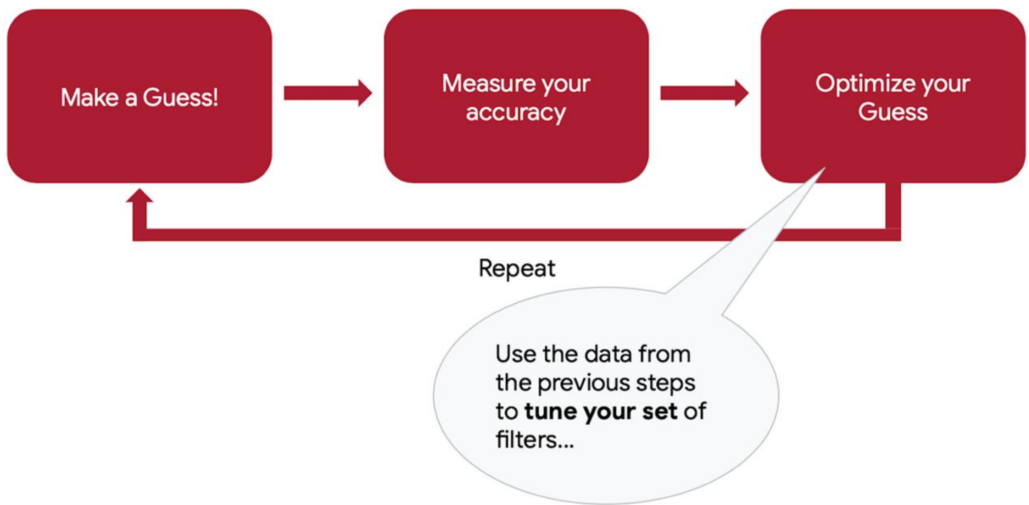
17

The Machine Learning Paradigm



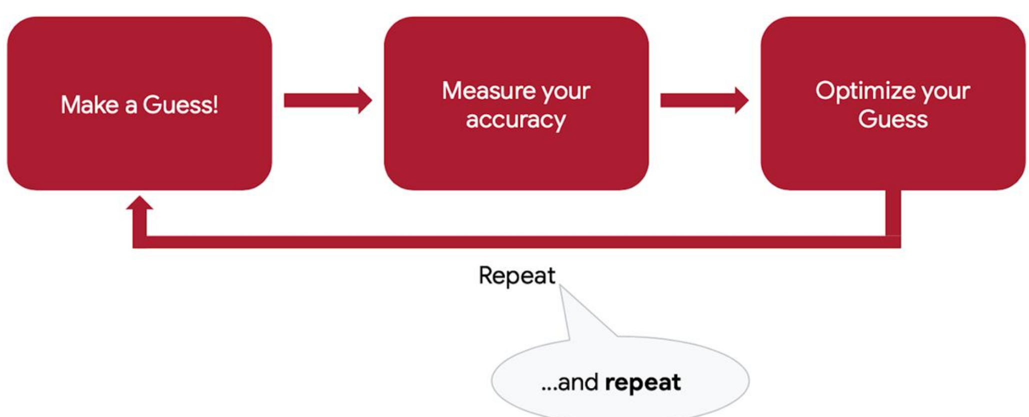
18

The Machine Learning Paradigm

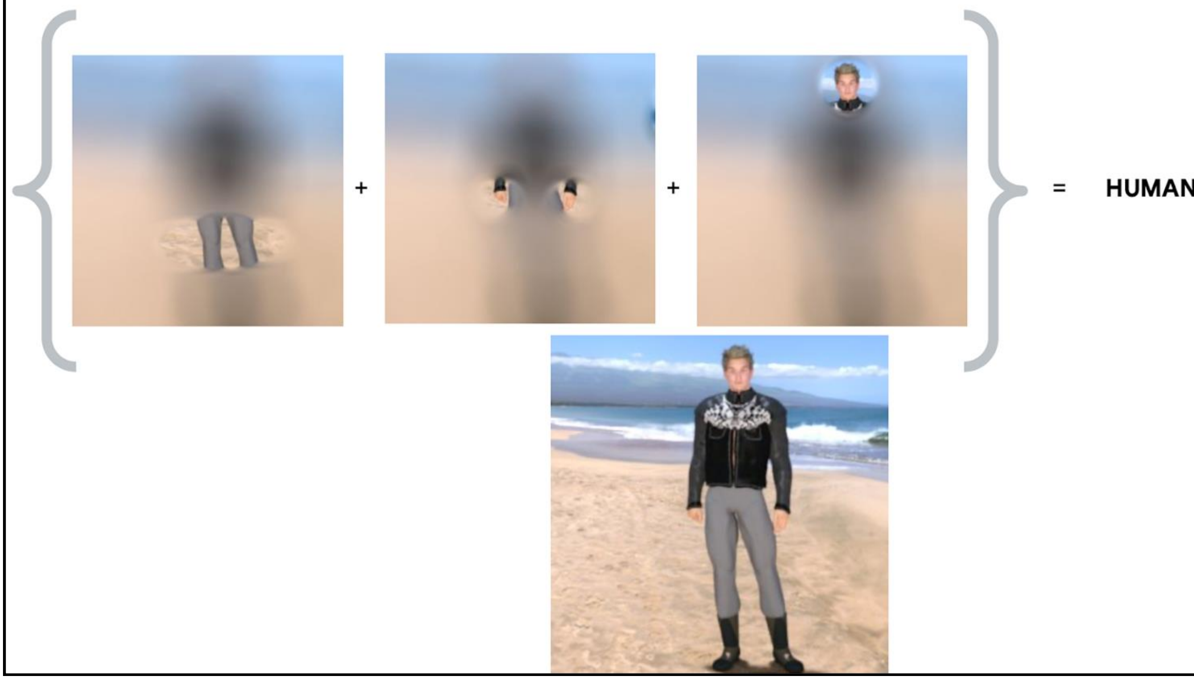


19

The Machine Learning Paradigm



20



21

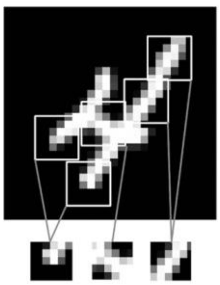
Image Classification With Convolutions

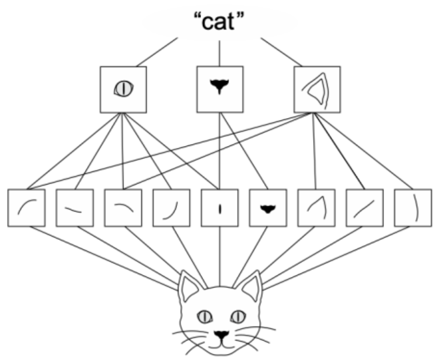
22

22

The Convolution Operation

*"The fundamental difference between a densely connected layer and a convolution layer is this: **Dense layers learn global patterns** in their input feature space (for example, for an MNIST digit, patterns involving all pixels), whereas **Convolution layers learn local patterns**—in the case of images, patterns found in small 2D windows of the inputs In the previous example, these windows were all 3 × 3."*



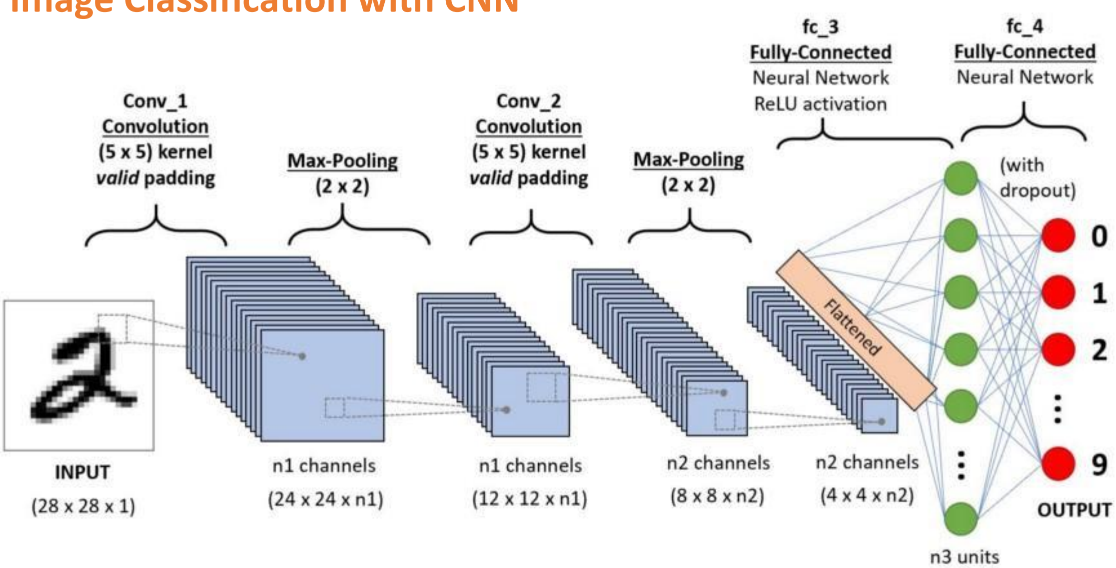


"They can learn spatial hierarchies of patterns. A first convolution layer will learn small local patterns such as edges, a second convolution layer will learn larger patterns made of the features of the first layers, and so on."

"Deep Learning with Python" by François Chollet

23

Image Classification with CNN

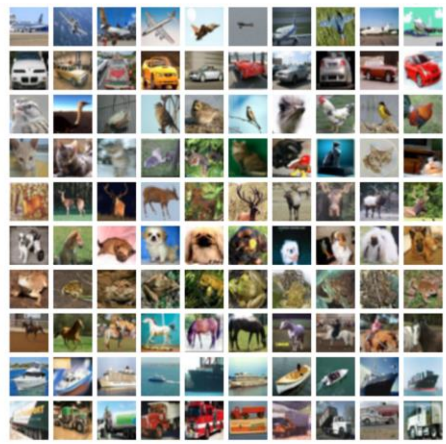


Source: Image from Analytics Vidhya

24

Cifar-10

0 airplane
1 automobile
2 bird
3 cat
4 deer
5 dog
6 frog
7 horse
8 ship
9 truck



https://www.tensorflow.org/datasets/catalog/cifar10

25

So far,

We saw how to build Neural Networks (Dense NN and Convolutional NN) that classify images of digits (**MNIST**).

Now,

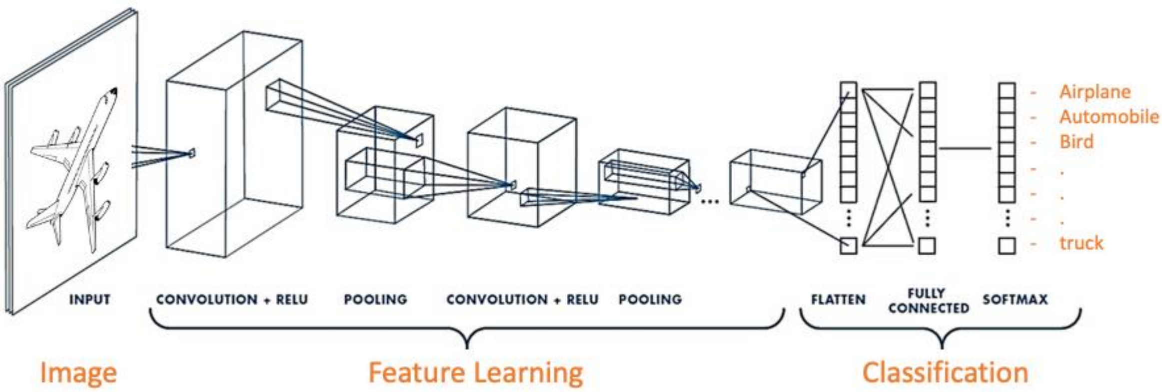
We will instead, recognize the 10 classes of **CIFAR** ('airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship' and 'truck').

There are some key differences between these image datasets that we need to take into account:

- While MNIST has 28x28 monochrome images (1 color channel), CIFAR is 32x32 color images (3 color channels).
- Besides, MNIST images are simple, containing just the object centered in the image, with no background. Conversely, CIFAR ones are not centered and can have the object with a background, such as airplanes that might have a cloudy sky behind them!

Those differences are the main reason to use a Convolutional NN (CNN) instead of a Dense NN!

26



27

Image Classification using CNNs
Code Time!

CNN_Cifar_10.ipynb

jupyter

28

28

Credits

- A previous edition of this course was developed in collaboration with Dr. Susan C. Schneider of Marquette University.
- We are very grateful and thank all the following professors, researchers, and practitioners for jump-starting courses on TinyML and for sharing their teaching materials:
- Prof. Marcelo Rovai - TinyML - Machine Learning for Embedding Devices, UNIFEI
  - https://github.com/Mjrovai/UNIFEI-IESTI01-TinyML-2022.1
- Prof. Vijay Janapa Reddi - CS249r: Tiny Machine Learning, Applied Machine Learning on Embedded IoT Devices, Harvard
  - https://sites.google.com/g.harvard.edu/tinyml/home
- Prof. Rahul Mangharam – ESE3600: Tiny Machine Learning, Univ. of Pennsylvania
  - https://tinyml.seas.upenn.edu/#
- Prof. Brian Plancher - Harvard CS249r: Tiny Machine Learning (TinyML), Barnard College, Columbia University
  - https://a2r-lab.org/courses/cs249r_tinyml/

29

29

References

- Additional references from where information and other teaching materials were gathered include:
- Applications & Deploy textbook: "TinyML" by Pete Warden, Daniel Situnayake
  - https://www.oreilly.com/library/view/tinyml/9781492052036/
- Deploy textbook "TinyML Cookbook" by Gian Marco Iodice
  - https://github.com/PacktPublishing/TinyML-Cookbook
- Jason Brownlee
  - https://machinelearningmastery.com/
- TinyMLedu
  - https://tinyml.seas.harvard.edu/
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
  - https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/introduction-to-embedded-machine-learning
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/computer-vision-with-embedded-machine-learning

30

30