McGill **Artificial Intelligence** Society

Image Classification Workshop

By Rick Wu and Cheng Lin

Agenda

- **Introduction** — *6:00pm - 6:30pm*
  - Motivation and intuition
- **Image Classification** — *6:30pm - 7:00pm*
  - Hands-on walkthrough
- **Transfer Learning** — *7:00pm - 7:30pm*
  - Hands-on walkthrough

What is Image Classification?

- Given a image, determine what **category** it belongs to

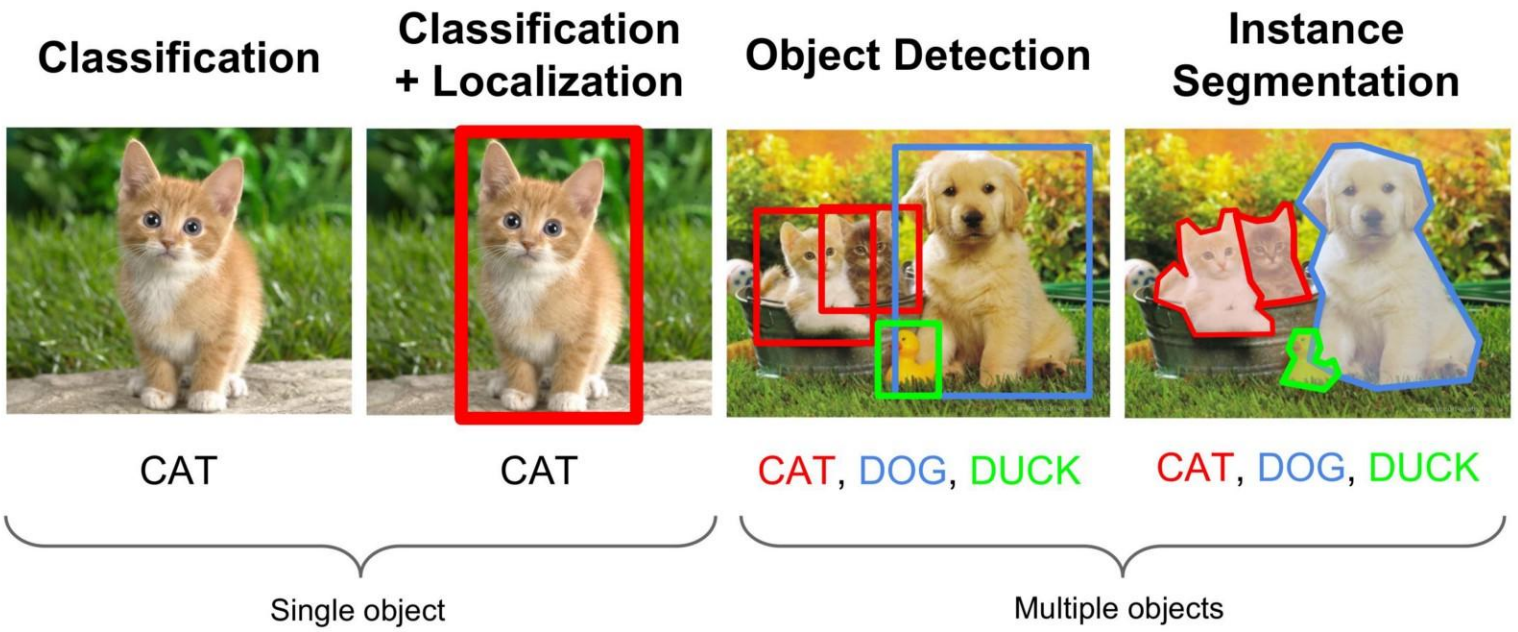


Image Source: https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

How to Build an Image Classifier: **Intuition**

- We might say an image contains a cat if we detect **eyes**, **nose**, **fur**, and **whiskers**
- **Classify** an image by detecting features present within an image
- Abstract these two procedures as **feature extraction** and **classification**

How to Build an Image Classifier: **Intuition**

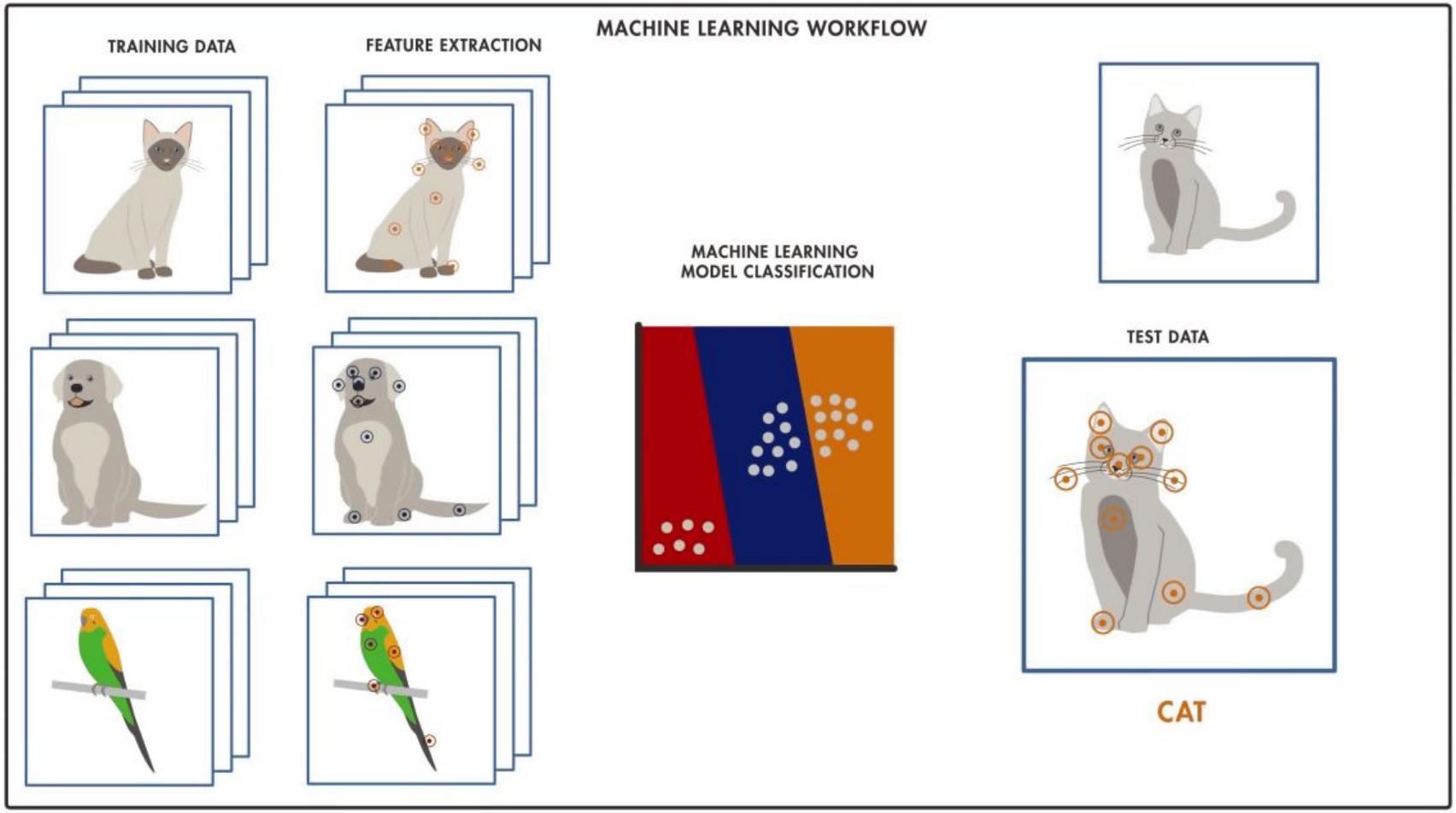
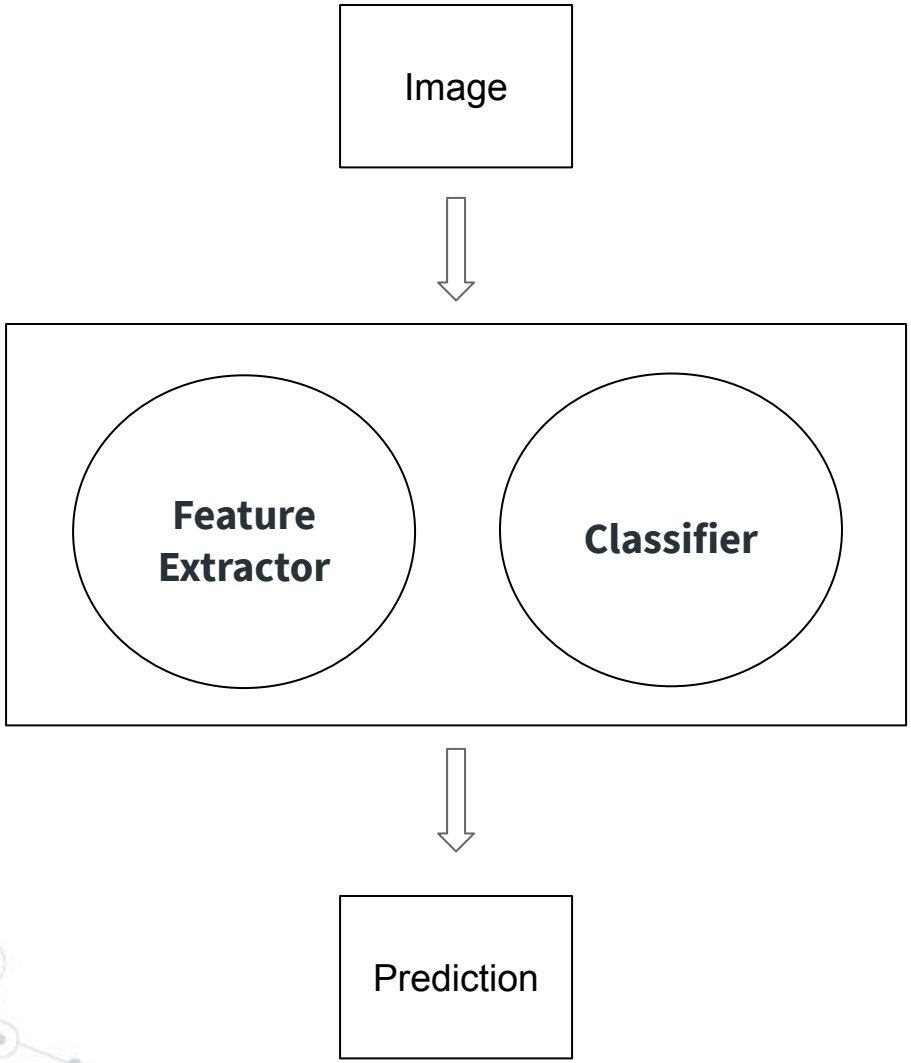


Image Source: https://gigazine.net/gsc_news/en/20170804-machine-learning-vs-deep-learning

How to Build an Image Classifier: **Building Blocks**



How to Build an Image Classifier: **Building Blocks**

Feature Extractor: A module which extracts features from the image

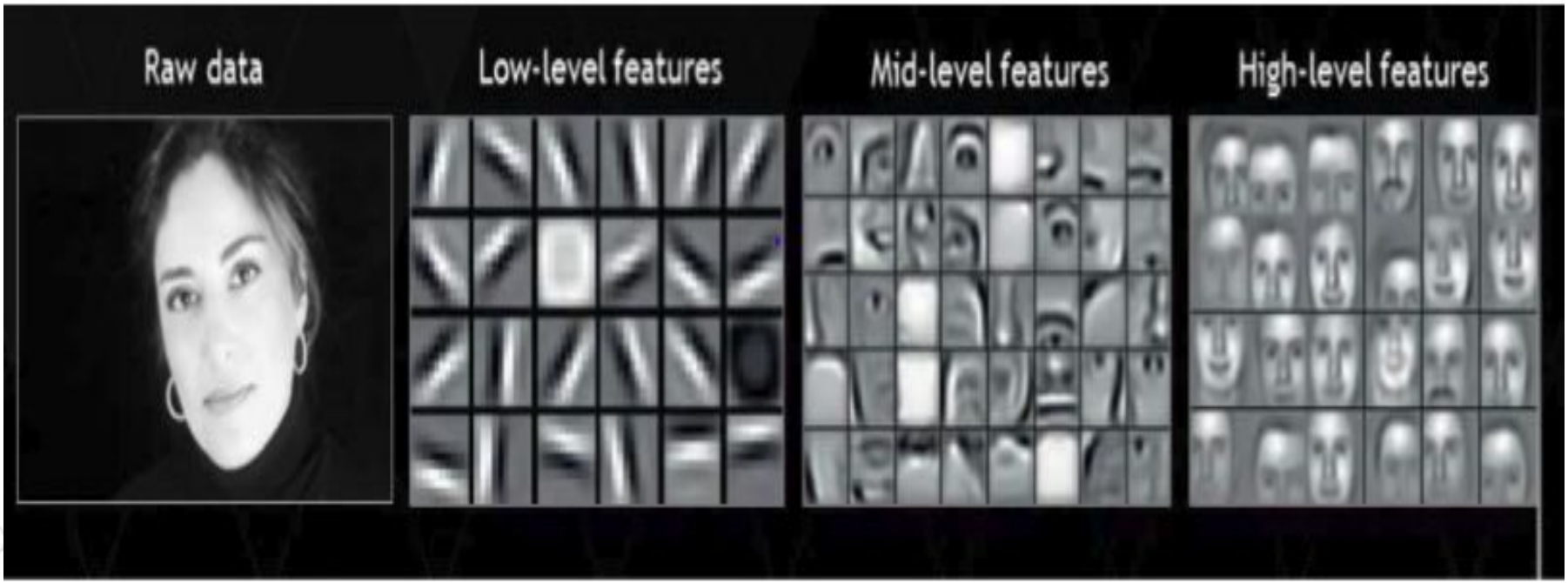


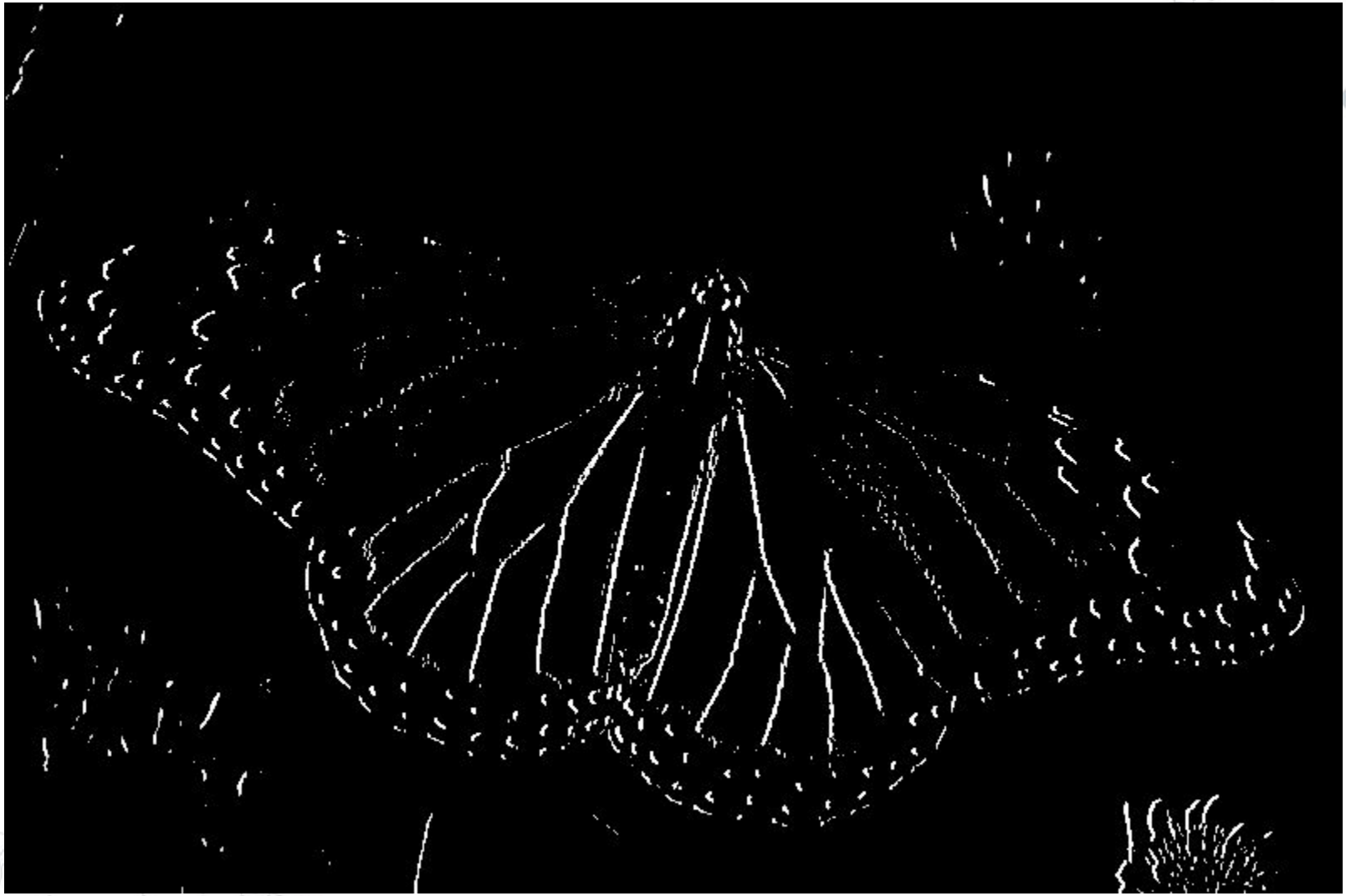
Image Source: https://www.slideshare.net/papisdotio/introduction-to-multi-gpu-deep-learning-with-digits-2-mike-wang

How to Build an Image Classifier: **Building Blocks**

2D Convolution: A mathematical operation which can be used to **filter** images for specific **shapes** or **patterns**

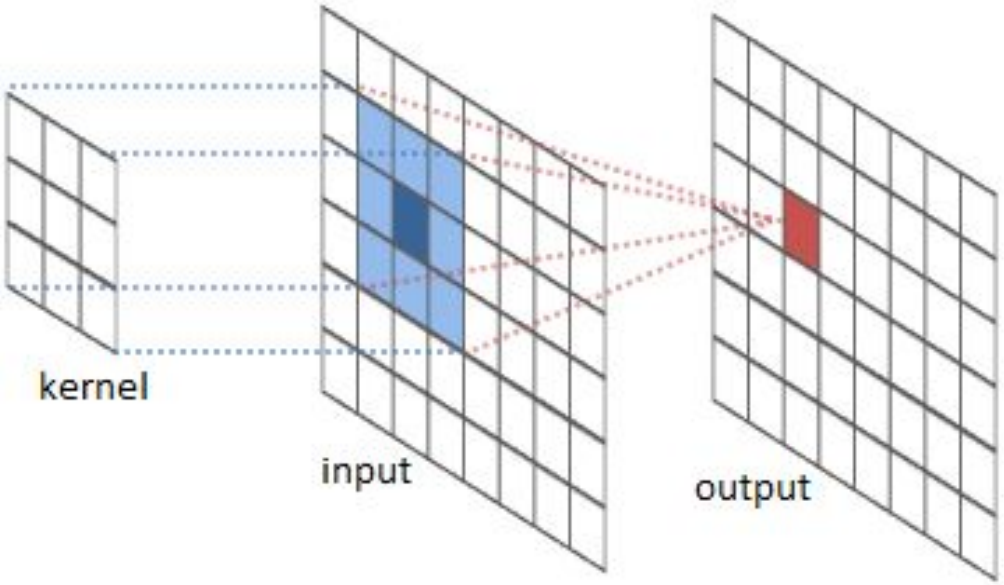
Image Source: https://meta.wikimedia.org/wiki/File:Monarch_In_May.jpg

Vertical Edges (x filter)



Horizontal Edges (y filter)

How to Build an Image Classifier: **Building Blocks**



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

x filter

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

y filter

Image Source: https://colah.github.io/posts/2014-07-Understanding-Convolutions/

Demo

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

How to Build an Image Classifier: **Building Blocks**

- Takes an image as an input and outputs another **filtered** image
- Works by taking a matrix (kernel) and sliding it across an image. At each point the values are multiplied and summed.
- The **kernel** matrix contains **parameters** which can be adjusted to extract the information we are looking for.

How to Build an Image Classifier: **Building Blocks**

Classifier: A module which takes a set of features and predicts a category

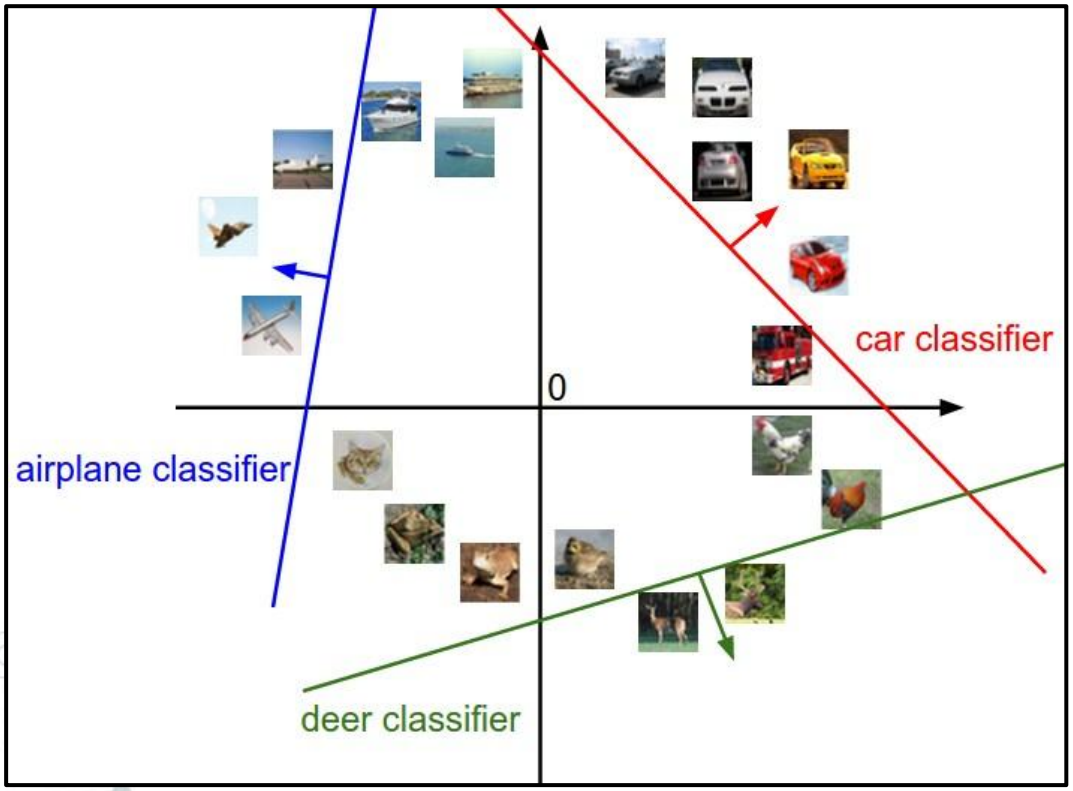


Image Source: http://cs231n.github.io/linear-classify/

How to Build an Image Classifier: **Machine Learning**

- We can construct the feature extractor and image classifier using **machine learning**
- Given a set of **training data**, we can train an image classifier by **penalizing** the model when it makes an error

How to Build an Image Classifier: **Machine Learning**

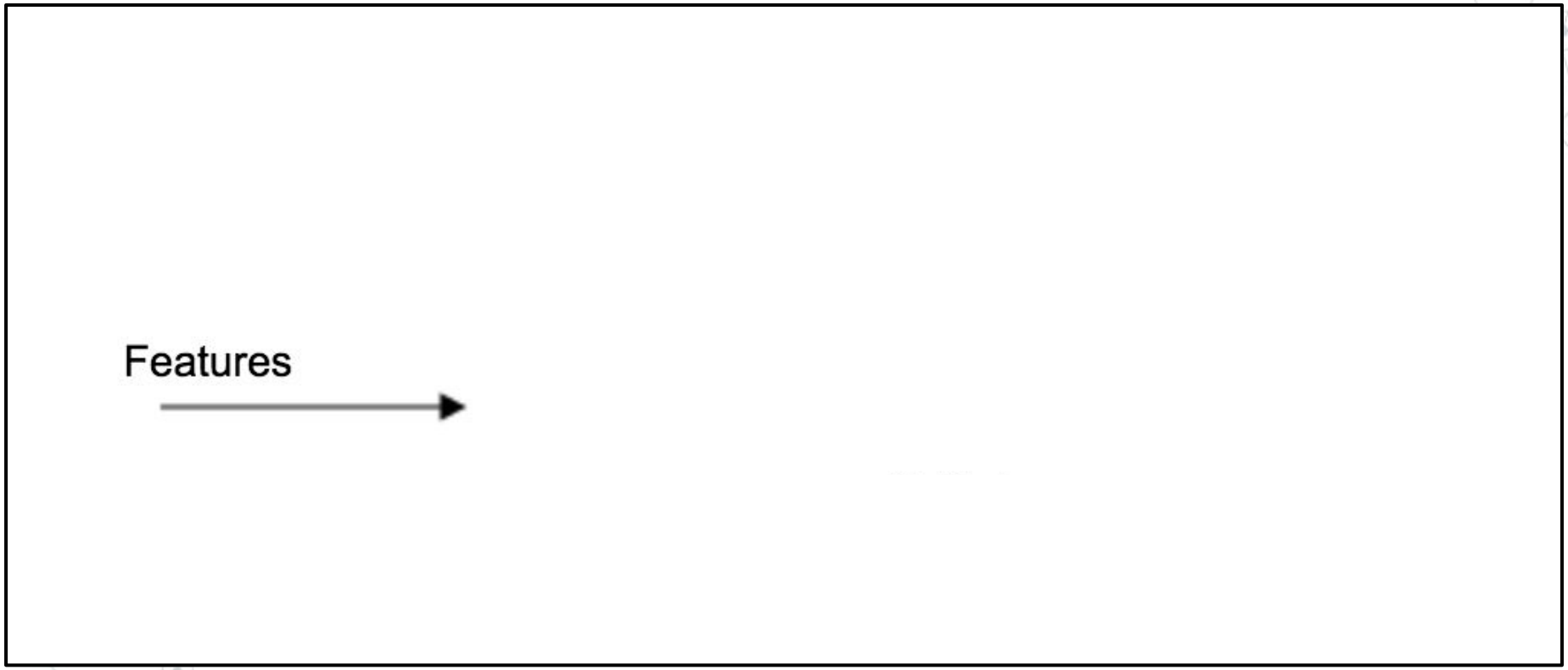


Image Source: a-practical-overview-of-backpropagation

How to Build an Image Classifier: **Machine Learning**

- In summary, the complete image classifier:

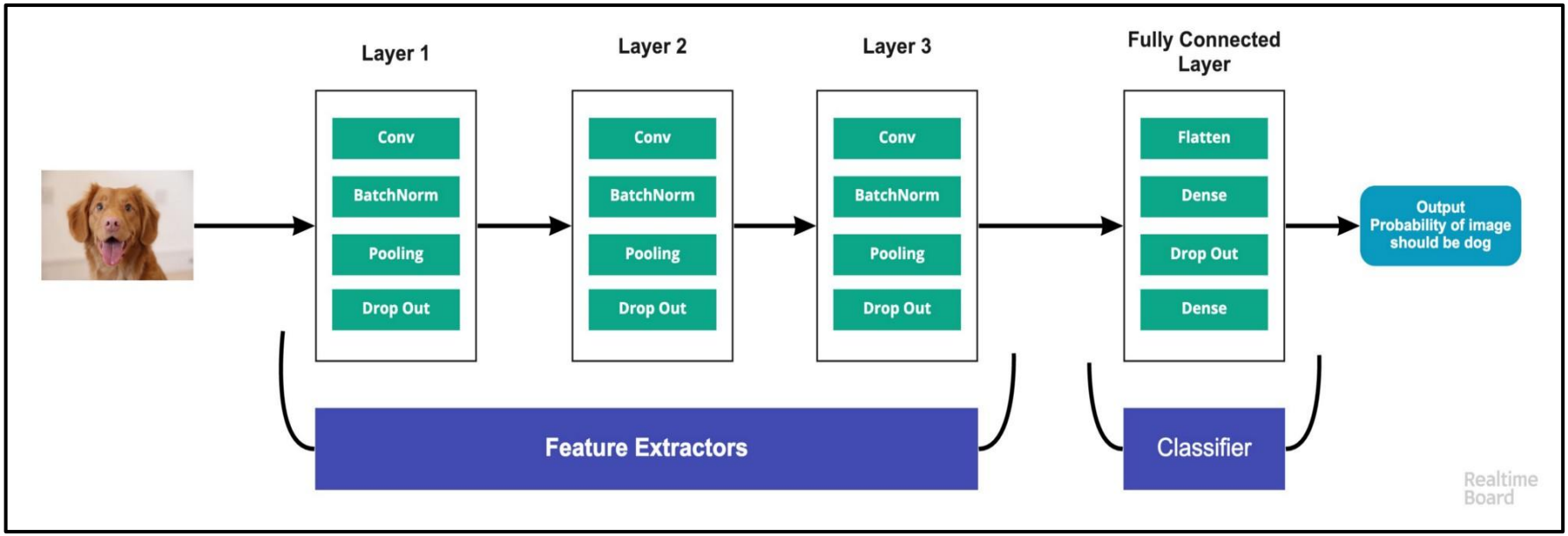


Image Source: https://www.kaggle.com/uysimty/keras-cnn-dog-or-cat-classification

How to Build an Image Classifier: **Implementation**

- Lastly, in order to implement our image classifier, we will be using the **Keras** library
- **Keras** provides a simple interface for implementing neural networks and is great for beginners!
- Documentation: https://keras.io/

Image Source: https://www.kaggle.com/uysimty/keras-cnn-dog-or-cat-classification

Demo

http://scs.ryerson.ca/~aharley/vis/conv/

Worksession #1

https://colab.research.google.com/drive/1A0yrGhh_G8iOownG62kgqwcL5re4hv1Z

Solutions: https://colab.research.google.com/drive/1hxsJltPixt7eOI5qvyL6f8ugw7bsRjiN

What is Transfer Learning?

- Informally, transfer learning is the concept of taking a model trained to perform one task and **repurposing** the **knowledge** for another task

- For instance, an image classifier trained to identify **vehicles** may be repurposed to identify **cats and dogs**

What can we use Transfer Learning for?

- We can **reuse** the feature extractor from **another image classifier** instead of training our own from scratch
- This saves us a lot of time and effort. We can take a model **“off-the-shelf”** and tailor it for our use-case!
- Very useful for when **training data** is **limited**

How to apply Transfer Learning for Image Classification

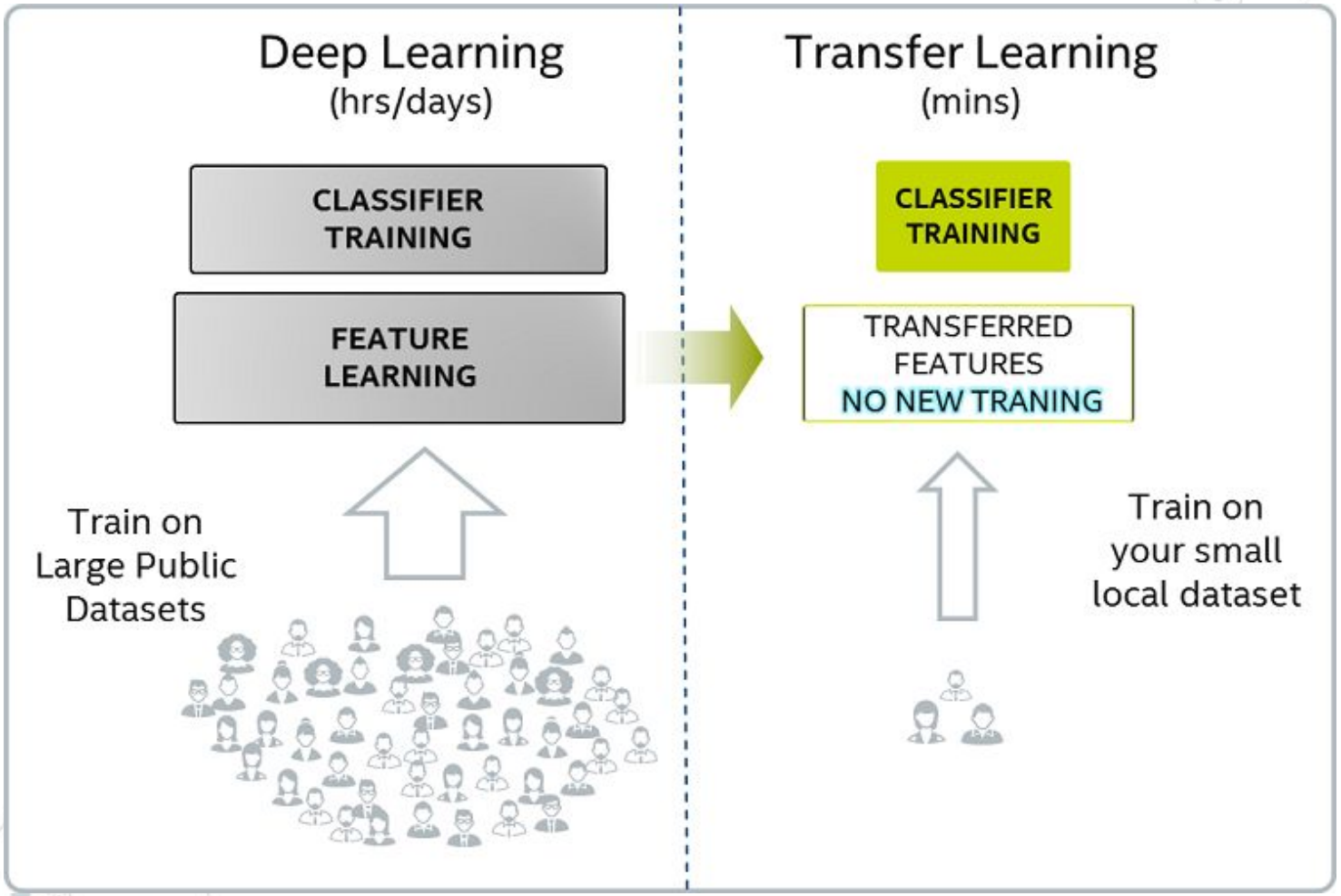


Image Source: https://software.intel.com/sites/default/files/managed/58/24/transfer-learning-fig1-fast-training.png

Worksession #2

https://colab.research.google.com/drive/1A0yrGhh_G8iOownG62kgqwcL5re4hv1Z#scrollTo=QrL9qBV5DOVh
https://colab.research.google.com/drive/1hxsJltPixt7eOI5qvyL6f8ugw7bsRjiN#scrollTo=QrL9qBV5DOVh

Thanks!

Any questions?

You can find us at:
https://mcgillai.com/

Please also give feedback at:
tiny.cc/MAIS-F2019-W4-feedback