§1. Introduction

- Computer Vision
  - Biology / Psychology: Optics, Neuron Science, Cognitive Science.
  - Computer Science: Graphics, Algorithms, Theory, Systems,
  - Mathematics / Physics: Information retrieved, ML.
  - Engineering: Robotics, Speech, NLP, Image Processing.

- Camera Obscura: (小孔成像.) / Huble & Wiesel (1959) (视觉皮层的刺激) — Electrical Signal from Brain
  Block World (1963) 边缘检测. / Stages of Visual Representation, David Marr (1970) (视觉信息.)
  Protorial Structure (1973) vs. Generalized Cylinder (1979) / David Lowe (1999) SIFT (1999)
  Normalized Cut (1997) 图像分割 / Face Detection — Viola & Jones (2001) AdaBoost
  - PASCAL VOC (2006-2012) (20 object categories)
  - ImageNet: AlexNet (2012); VGG (2014); GoogleNet (2014); ResNet (2015); SENet (2017);

- Ingradients for Deep Learning:
  ① Algorithms ② Data ③ Computation

§2. Image Classification Pipeline (KNN & Linear Classification)

<Segmantic Gap>: 语义鸿沟.
 └ An image is just a grid of numbers between [0, 255] (像素矩阵: RGB 3 channel — 每个通道为1个矩阵.) ← 256: use 8 bit for 1 pixel

Challenges:
① Background Clutter (背景杂乱) ② illumination (照明) ③ deformation (形变) ④ Occlusion (遮挡) ⑤ intraclass variation (类内变化)

<ML Data-Driven Approach>:

① Collect a dataset of images & labels — training set
② Use ML to train a classifier
③ Evaluate the classifier on new images — testing set.

```
def classify_image(image):
    # Black Box
    return class_label
```

- K-Nearest Neighbour:
  - training step: memorize all the training data and label.
  - prediction step: predict the label of the most similar training image.

<L1 Distance> Manhattan $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$ ⇐ changing the coordinate frame!

Q. With N examples, how fast are training and prediction?
training → O(1); prediction → O(N)

bigger k ⇒ smooth boundary.

* instead of copying label from nearest neighbour, take majority vote from k closest points.

<L2 Distance> Euclidean: $d_2 = \sqrt{\sum_p (I_1^p - I_2^p)^2}$ (平方和开方根)

- Hyperparameters: choices about the algorithms that we set rather than learn.
  - x ① Choose the ones that work best on the data. — k=1 always perfect.
  - x ② Split data into train and test. choose the one that work best on test data. (No idea about new data)
  - ✓ ③ Split data into train, validation and test; choose the ones on validation and evaluate on test!

Ch.
Date /

④ Cross-Validation: split data into folds, try each fold as validation and average the result.

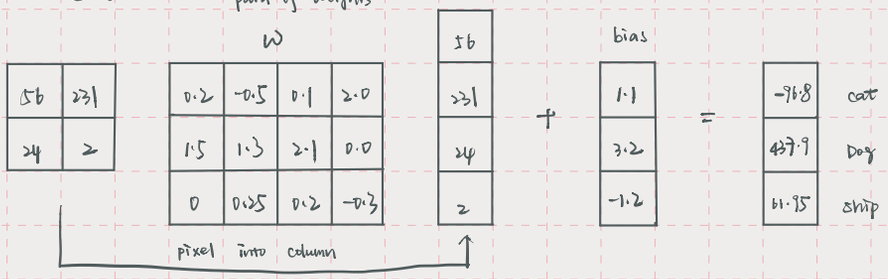
* KNN on images NEVER used! { very slow at test time.
  distance metrics on pixels are not informative.
  ↓
  curse of dimensionality. ⇒ exponential growth

Linear Classification

< Parametric Approach >

image → $f(x, W) = Wx + b$ → class (bias)

(array of num) — para of weights



* image = point in high dimensional space.

  linear classifier = linear decision boundary between categories.