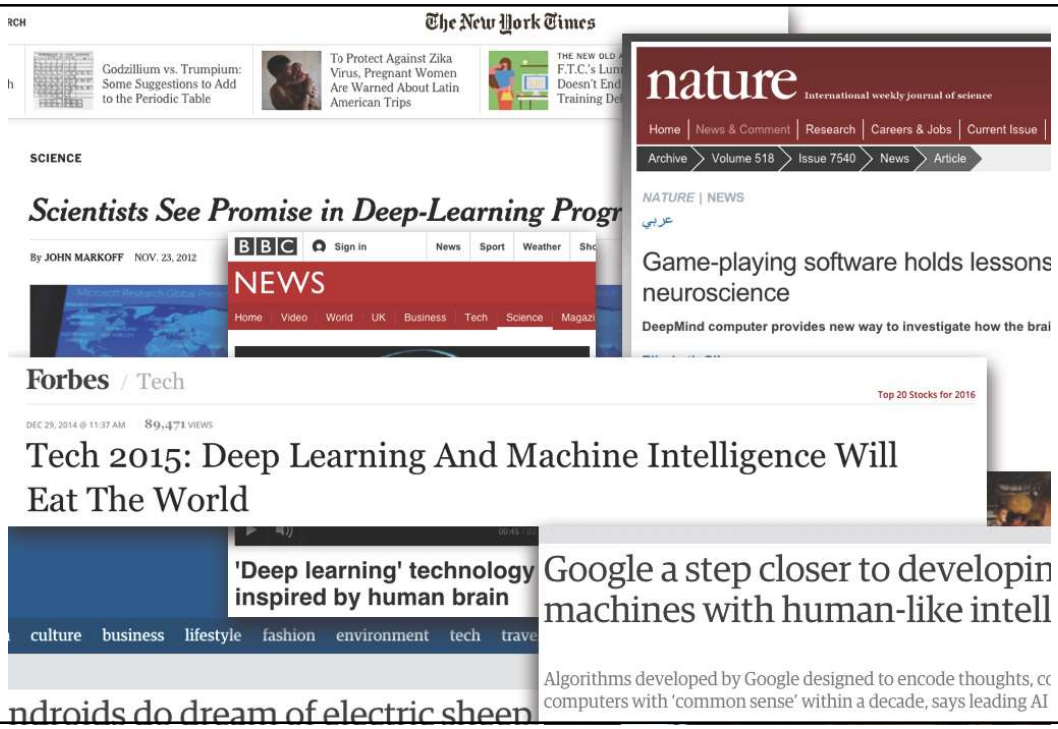
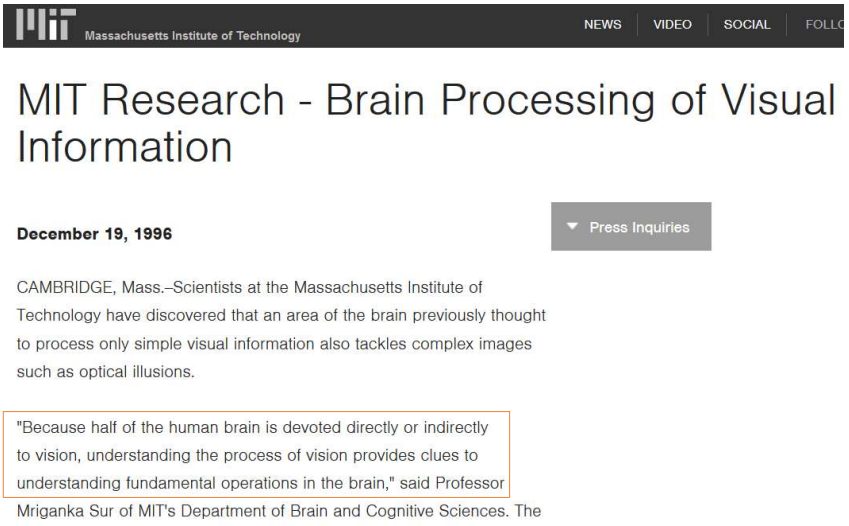
Introduction to Deep Learning for Computer Vision Applications

Kai-Lung Hua (花凱龍)



Vision is an amazing feat of natural intelligence

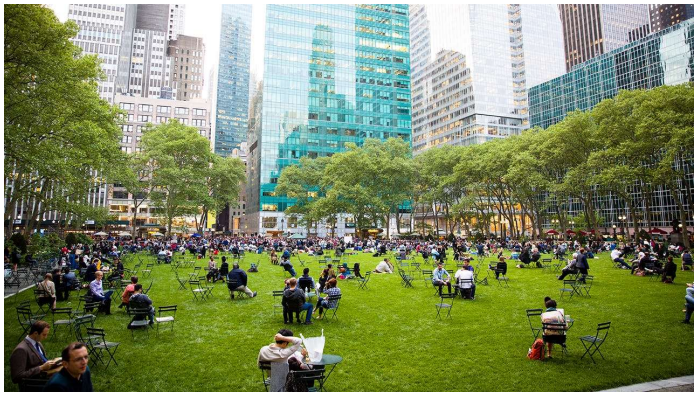
"Half of the human brain is devoted directly or indirectly to vision"



3 of 69

Computer Vision

Deals with how to make computers understand images and video.



What kind of scene is this?

How many people are there?

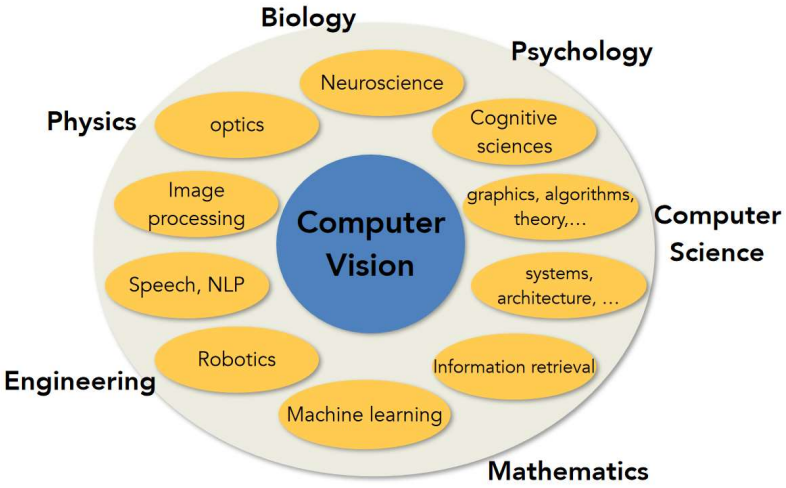
How far are the buildings?

What are they doing?

…

4 of 69

Related disciplines



5 of 69

Source: Stanford's CS231n Fei-Fei Li & Justin Johnson & Serena Yeung

Why is computer vision hard?

6 of 69

What did you see?

- Where was this picture taken?
- How many people are there?
- What are they doing?
- What object is the person on the left standing on?
- Why is this picture funny?

Challenges: Semantic Gap

What we see

```
array([[115, 211, 182, 173, 107, 166, 172, 213, 200, 133, 111, 145],
       [110,  92, 217, 156, 102, 124, 165,  96, 126, 167, 149, 157],
       [204, 184, 144, 105, 208, 191, 218, 102, 182, 211, 197, 132],
       [ 96, 161, 142, 114, 129,  99, 123, 167, 128, 137, 226, 151],
       [198, 195, 172, 169, 202,  93, 111, 223,  97, 123, 196, 103],
       [137, 130, 159, 111, 190, 176, 108, 153,  93, 149, 126, 173],
       [226, 113, 186, 211, 206, 112, 130, 136, 208, 213, 137, 132],
       [ 98, 118, 130, 137, 135, 222, 158, 226, 223, 212, 208, 104],
       [195, 171,  97, 163, 128, 104, 186, 186, 102, 212, 228, 155],
       [127, 165, 184, 186, 190, 122, 186, 157, 216, 117,  99, 127],
       [229, 125, 214, 131, 103, 137, 151,  97, 137, 192, 150, 172],
       [221, 171, 153, 137, 219, 116, 147, 100, 143, 181, 200, 109],
       [173, 150, 106,  92, 111, 229, 196, 207, 221, 125, 146, 113],
       [211, 151, 120, 227, 157, 173, 120, 158, 101, 174, 182,  92],
       [225, 197, 100, 103, 190, 214, 169, 224, 170, 209, 167, 209],
       [184, 151, 143, 219, 107, 104, 108, 138, 184, 135, 175,  95],
       [190, 186, 162, 118, 162,  91, 218, 210,  89, 203, 226, 157],
       [108, 226, 161, 116, 186, 116, 186, 137, 158, 130, 101, 103],
       [217, 195, 188, 109, 225, 156, 117, 143, 165, 191, 119, 181],
       [203, 206, 203, 220, 121, 103, 128, 121, 222, 141, 206, 227]])
```

What the computer sees

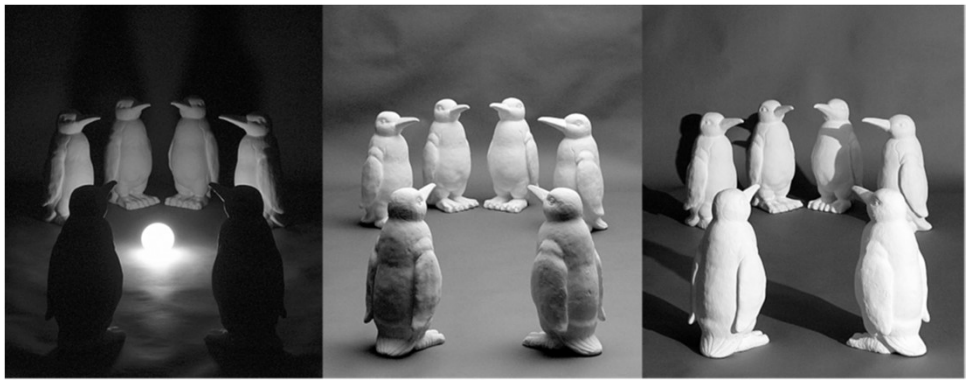
An image is just a big grid (matrix) of numbers

Challenges: Viewpoint Variation

9 of 69

Slide credit: S. Lazebnik

Challenges: illumination



10 of 69

Image credit: J. Koenderink

Challenges: Scale

11 of 69

Slide credit: Fei-Fei Li & Antonio Torralba

Challenges: Deformation

12 of 69

Slide credit: Fei-Fei Li & Antonio Torralba

Challenges: Object intra-class variation






13 of 69

Slide credit: Fei-Fei Li & Antonio Torralba

Challenges: Occlusion, Clutter



14 of 69

Slide credit: Fei-Fei Li & Antonio Torralba

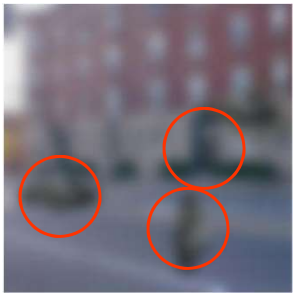
Challenges: Motion

15 of 69

Slide credit: S. Lazebnik

Challenges: Ambiguity

What is this patch?



Slide credit: Fei-Fei Li & Antonio Torralba

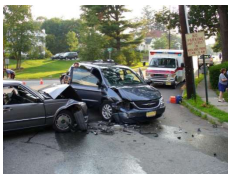
16 of 69

Challenges: Semantic Context

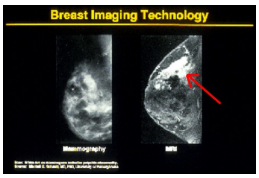
Slide credit: Fei-Fei Li & Antonio Torralba

Applications of Computer vision

As image sources multiply, so does computer vision applications



Safety



Health

Security

Convenience

Mocap for *Pirates of the Carribean*, Industrial Light and Magic
Source: S. Seitz

Visual Effects

Assisted Driving

Human joystick, NewsBreaker Live

Video-based interfaces

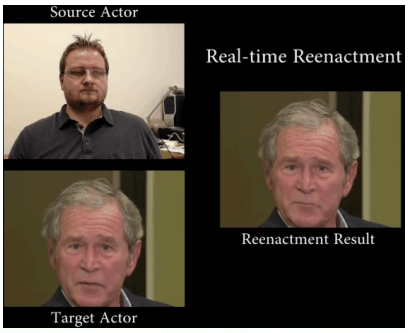
Some state-of-the-art examples of how deep learning for computer vision is being used today

Face Recognition

Facebook's auto-tagging feature

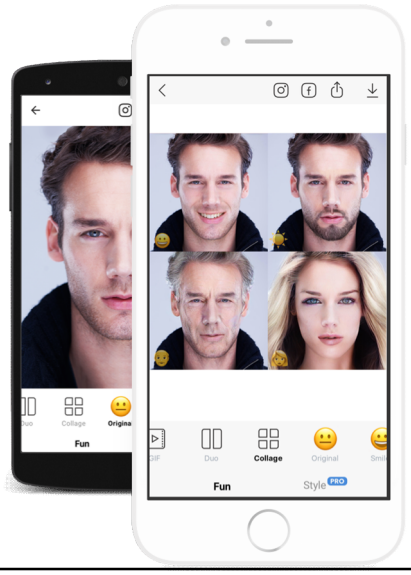
21 of 69

Face2Face: Real-time Face capture and Reenactment of RGB Videos



22 of 69

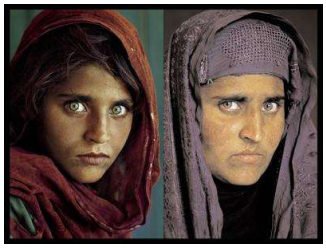
FaceApp: Face aging, Gender Swap



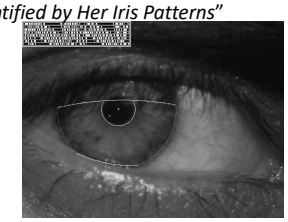
https://www.faceapp.com/

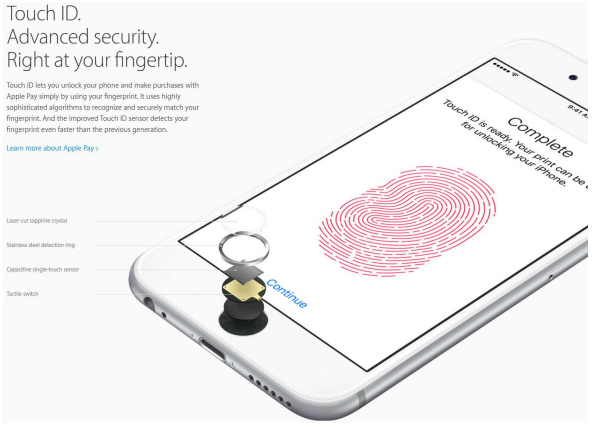
23 of 69

Vision-based Biometrics



"How the Afghan Girl was Identified by Her Iris Patterns"

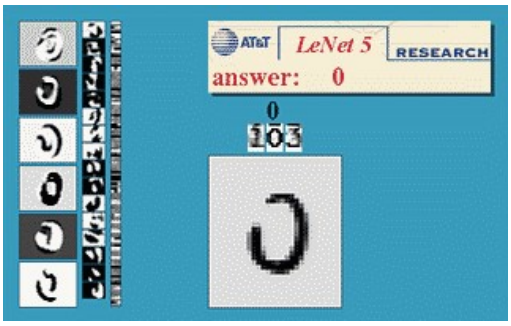




24 of 69

Optical Character Recognition (OCR)

- Technology to convert scanned docs to text
  - If you have a scanner, it probably came with OCR software



Digit recognition, AT&T labs

Plate Number = 4YCH428

License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

25 of 69

Computer vision in sports

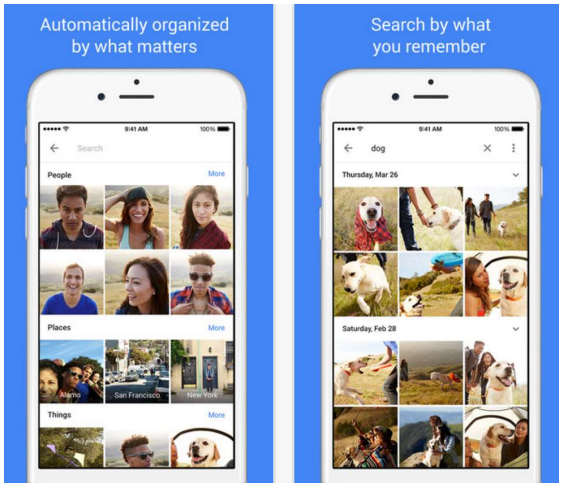
SportVision: improving viewer experiences

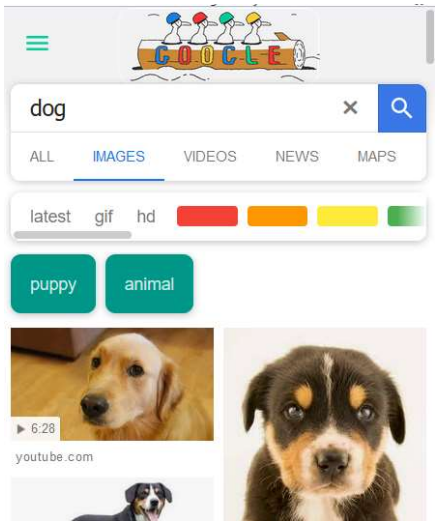
Hawk-Eye: helping/improving referee decisions

Player Tracking

26 of 69

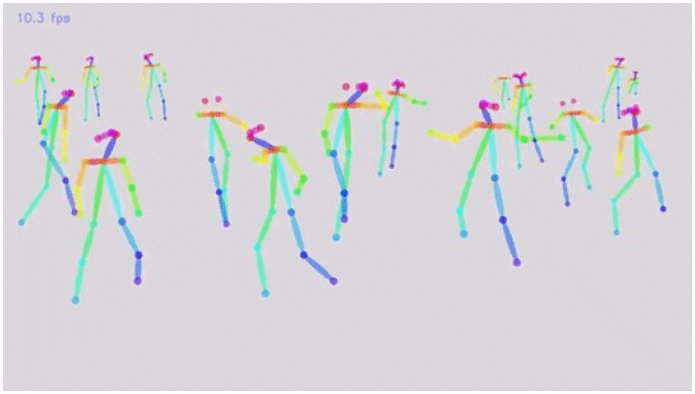
Google Photos / Google Image search





27 of 69

Kinect / Pose Estimation



OpenPose

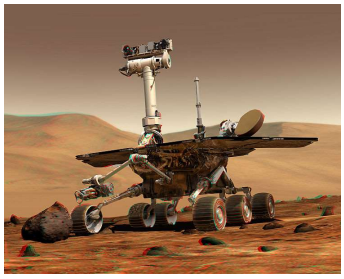
28 of 69

Industrial robots

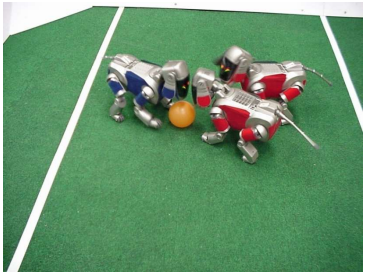
Vision-guided robots position nut runners on wheels

29 of 69

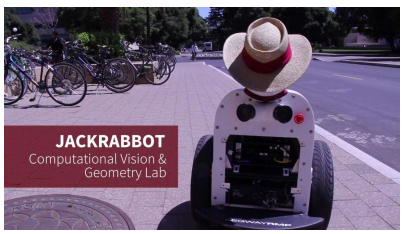
Mobile Robots



NASA's Mars Spirit Rover



http://www.robocup.org/

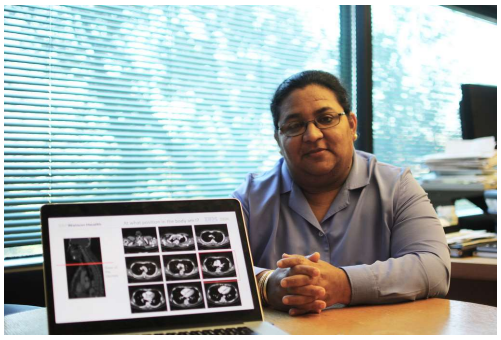


JackRabbot

http://www.youtube.com/watch?v=DF39Ygp53mQ

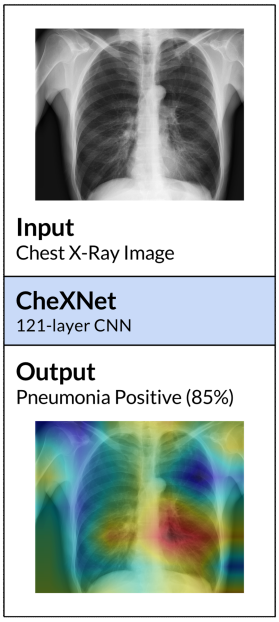
30 of 69

Medical imaging



Radiology

Identifying Skin Cancer

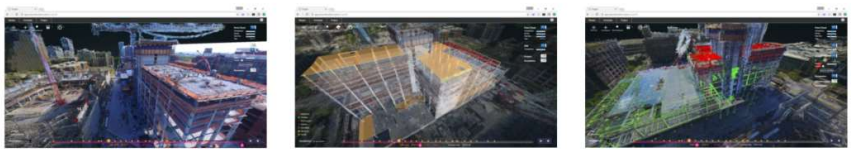


CheXNet

31 of 69

Vision in Construction

RECONSTRUCT INTEGRATES REALITY AND PLAN



Visual Asset Management

Reconstruct 4D point clouds and organize images and videos from smartphones, time-lapse cameras, and drones around the project schedule. View, annotate, and share anywhere with a web interface.

4D Visual Production Models

Integrate 4D point clouds with 4D BIM, review "who does what work at what location" on a daily basis and improve coordination and communication among project teams.

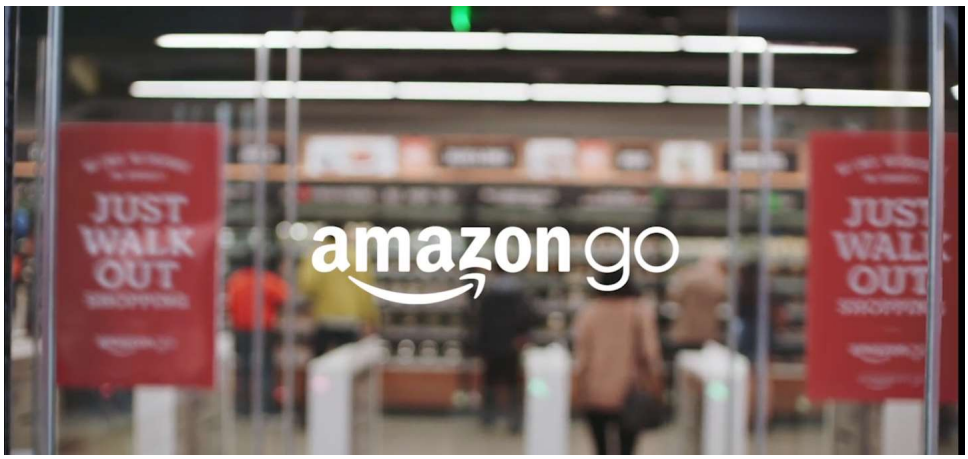
Predictive Visual Data Analytics

Analyze actual progress deviations by comparing Reality and Plan and predict risk with respect to the execution of the look-ahead schedule for each project location, to offer your project team with an opportunity to tap off potential delays before they surface on your jobsite.

reconstructinc.com

32 of 69

Shopping without checkout



https://www.youtube.com/watch?v=NrmMk1Myrxc

33 of 69

Smart Cars: Tesla, Google Cars

https://www.youtube.com/watch?v=VG68SKoG7vE

34 of 69

Current state of the art

- Many of these are less than 5 years old
- Very active and exciting research area!
- To learn more about vision applications and companies
  - David Lowe maintains an excellent overview of vision companies
    - http://www.cs.ubc.ca/spider/lowe/vision.html

35 of 69

What to expect from this course?

"Deep Neural Networks require an interplay between intuitive insights, theoretical modeling, practical implementations, empirical studies, and scientific analyses" – Yann Lecun, Director of AI Research, Facebook

- This will be a heavy course!
- We will go through lots of Math!
- Practical Experience!
  - 6 Programming Assignments
- Final Project

36 of 69

Academic Integrity

- Can discuss HW with peers, but don't copy and/or share code
- Carefully document any sources within HW
- Do not use code from Internet unless you have permission
  - If you're not sure, ask

37 of 69

Prerequisites

- Linear Algebra
- (Multivariate) Calculus
- Probability and Statistics
- Machine Learning basics

- **Python 3** (All programming assignments will be in python)
  - Please Install **Anaconda 3.6** you will need it in all the assignments
  - https://www.anaconda.com/download/
- **TensorFlow 1.5.0**
  - https://www.tensorflow.org/

38 of 69

Tentative Schedule

- March 3– Introduction / k-Nearest Neighbors
  - Programming Assignment 1
- March 10 – Linear Models
  - Programming Assignment 2
- March 17 – Regularization / Bias-Variance Analysis
  - Programming Assignment 3
- March 24 – Logistic Regression (Binomial and Multinomial)
  - Programming Assignment 4
- March 31 – Neural Networks (Multilayered Perceptron)
  - Programming Assignment 5
- April 7 – Practical Tips for Real World Machine Learning Tasks

39 of 69

Tentative Schedule

- April 14 – Mid-term Exams
- April 21 – Convolutional Neural Networks
  - Programming Assignment 6
- April 28– Intel OpenVINO
- May 5 – Final Project Proposal
- May 19 – Object Localization / Detection
- May 12 – Recurrent Neural Networks
- May 26 – Generative Adversarial Networks
- June 2 – Deep Reinforcement Learning
- June 9-23 – Final Project Presentations

40 of 69

Teaching Assistant

- Name:
  John Jethro Virtusio
- Email:
  jetvirtusio@hotmail.com
- Consultation hours:
  10AM-1PM Monday/Wednesday
- Office:
  RB308-2

41 of 69

This class will focus on the intersection of machine learning / deep learning with computer vision.

"Before we can run, we must first learn how to walk"

So we will start with the simplest learning algorithm:
k-Nearest Neighbors

42 of 69

k Nearest Neighbor

43 of 69

Our data

X y

temperature	humidity	weather
1	24	snowy
8	30	snowy
7	21	snowy
22	30	snowy
5	14	sunny
20	10	sunny
16	4	sunny
26	223	rainy
21	25	rainy
17	14	rainy
34	29	rainy

Given X we want to know y

Given features
dimensions
attributes
variables
to know y
labels
prediction

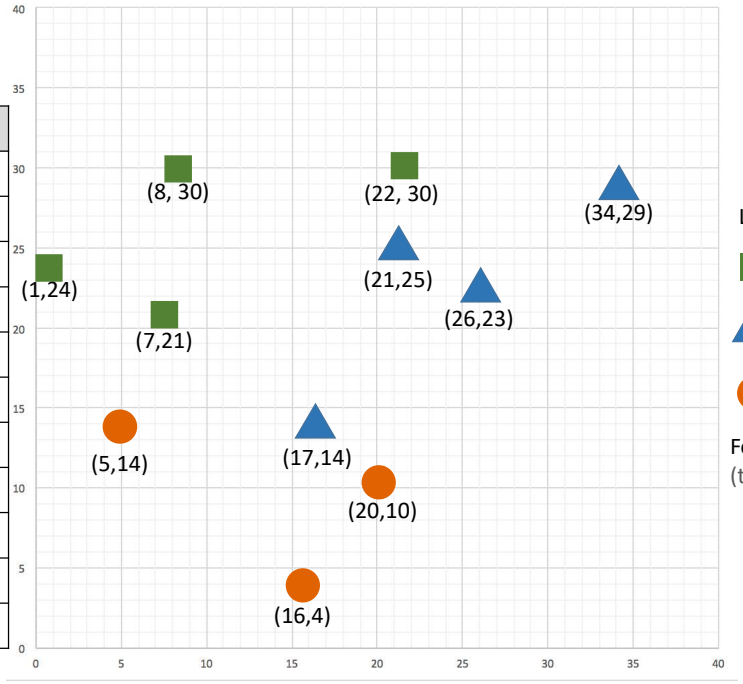
If we have a

temp	humidity	weather
21	17	?

44 of 69

Training

temp	humidity	weather
1	24	snowy
8	30	snowy
7	21	snowy
22	30	snowy
5	14	sunny
20	10	sunny
16	4	sunny
26	223	rainy
21	25	rainy
17	14	rainy
34	29	rainy

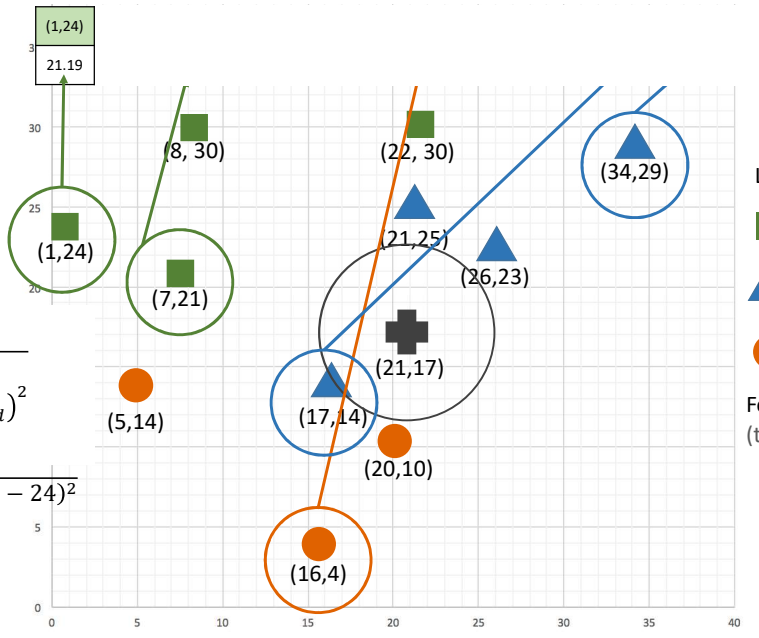


45 of 69

Testing

Distances of each training data to (21,17)

$k = 1$



$$dist(tr_i, te_j) = \sqrt{\sum_{d=1}^{dim} (tr_{i,d} - te_{j,d})^2}$$

$$= \sqrt{(21-1)^2 + (17-24)^2}$$

$$= 21.19$$

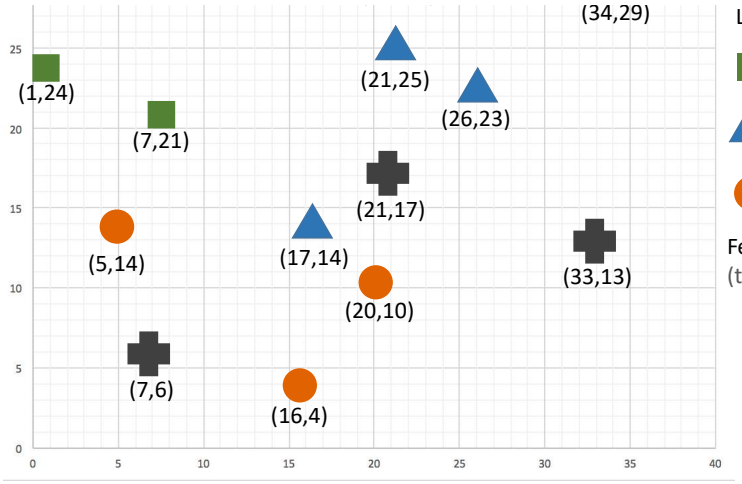
46 of 69

Testing

Distances of each training data to ✚'s

	(1,24)	(8, 30)	(7,21)	(22, 30)	(5,14)	(20,10)	(16,4)	(26,23)	(21,25)	(17,14)	(34,29)
(21,17)	21.19	18.38	14.56	13.03	16.28	7.07	13.93	7.81	8.00	5.00	17.69

$k = 1$



Labels (y):

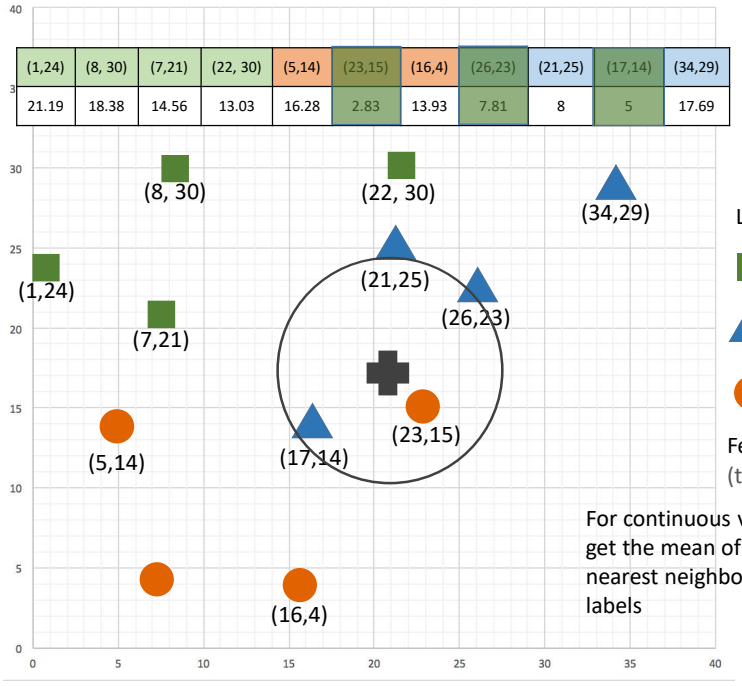
- Snowy
- Rainy
- Sunny

Features/dimensions (X):
(temperature, humidity)

47 of 69

Testing

	(1,24)	(8, 30)	(7,21)	(22, 30)	(5,14)	(23,15)	(16,4)	(26,23)	(21,25)	(17,14)	(34,29)
	21.19	18.38	14.56	13.03	16.28	2.83	13.93	7.81	8	5	17.69



Labels (y):

- Snowy
- Rainy
- Sunny

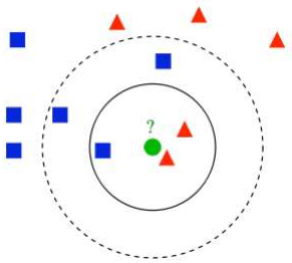
Features/dimensions (X):
(temperature, humidity)

For continuous values, get the mean of the nearest neighbors's labels

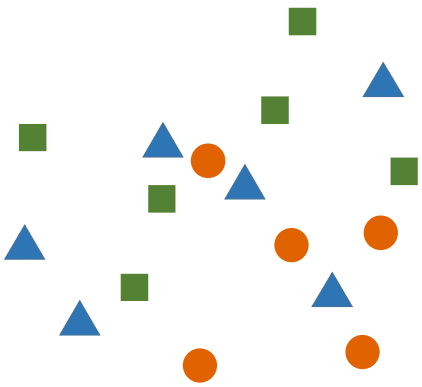
48 of 69

k Nearest Neighbour

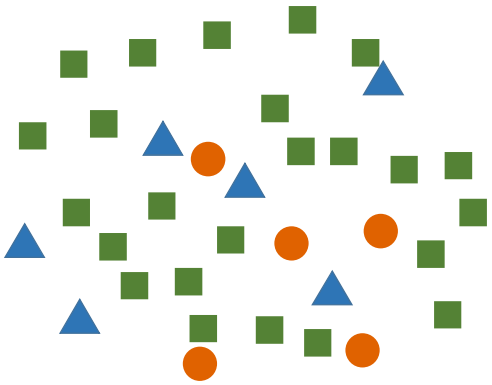
- Assumes all instances correspond to points in the n-dimensional space $\boldsymbol{R^n}$
- Values may be discrete or continuous.



49 of 69

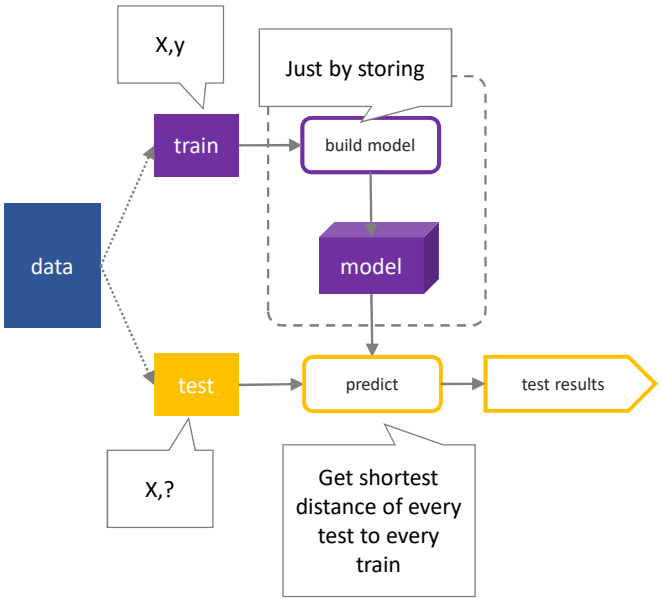


50 of 69



51 of 69

Pipeline



52 of 69

A deeper look

53 of 69

Some Notations

- y – denotes the label / output variable / target variable
- $X \in \mathbb{R}^{N \times D}$ - your whole data represented as a matrix
  - N – denotes the total number of examples
  - D – denotes the dimensions / or the number of features in a data point
- $x \in \mathbb{R}^{D}$ - a particular data point
  - It is represented as a vector with D dimensions / entries / elements
  - Each entry / element / dimension corresponds to your features
  - A particular element is usually denoted by a subscript
    - $x = [x_1, x_2, \dots, x_D]$
    - In the our previous kNN example x_1 would correspond to temperature and x_2 corresponds to humidity and $D = 2$ since we only have two features.
- Note: the subscripts can be confusing / inconsistent but unfortunately it is also what is used in papers. You really have to look at the context
  - For example, $X = [x_1, x_2, \dots, x_N]$ in this context the subscript would denote the i^{th} data point in your whole data.

54 of 69

Distance metric

Euclidean Distance:

$$D(a,b) = \sqrt{\sum_i (a_i - b_i)^2}$$

Or equivalently:

$$D(a,b) = \|a-b\|_2 = \sqrt{(a-b)^T(a-b)}$$

Other common metrics:
- L_1 norm (also called Manhattan distance)
- L_∞ norm
- Mahalanobis distance
- Angle (cosine distance)

55 of 69

Distance in relation to norms

- Norm of a vector is informally a measure of the "length" of the vector.
- The length of the difference between two points can be thought of as the "distance" between those two points
- Most commonly used is the Euclidean norm or L_2 norm.
  - $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
  - $\|x\|_2^2 = x^T x$

More formally, a norm is any function $f : \mathbb{R}^n \to \mathbb{R}$ that satisfies 4 properties:

1. For all $x \in \mathbb{R}^n$, $f(x) \geq 0$ (non-negativity).
2. $f(x) = 0$ if and only if $x = 0$ (definiteness).
3. For all $x \in \mathbb{R}^n$, $t \in \mathbb{R}$, $f(tx) = |t|f(x)$ (homogeneity).
4. For all $x, y \in \mathbb{R}^n$, $f(x + y) \leq f(x) + f(y)$ (triangle inequality).

56 of 69

- L_0 norm
  - $\|x\|_0 = \sum_{i=1}^{n} |x_i|^0$ (where $0^0 = 0$)
- L_1 norm
  - $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- L_∞ norm
  - $\|x\|_\infty = \max_i |x_i|$
- L_p norm
  - $\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$

- Frobenius norm (for matrices)

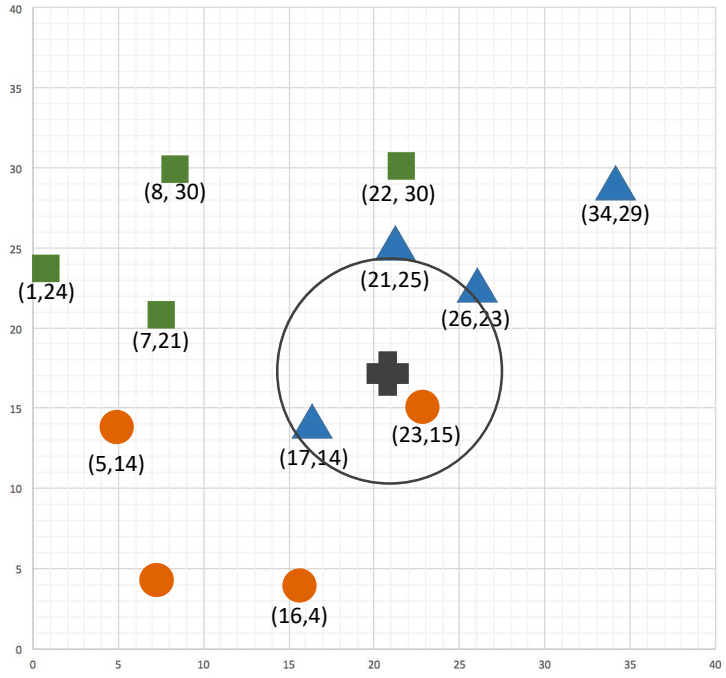
$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2} = \sqrt{\mathrm{tr}(A^T A)}.$$

57 of 69

If I change the distance metric, what difference does it make?

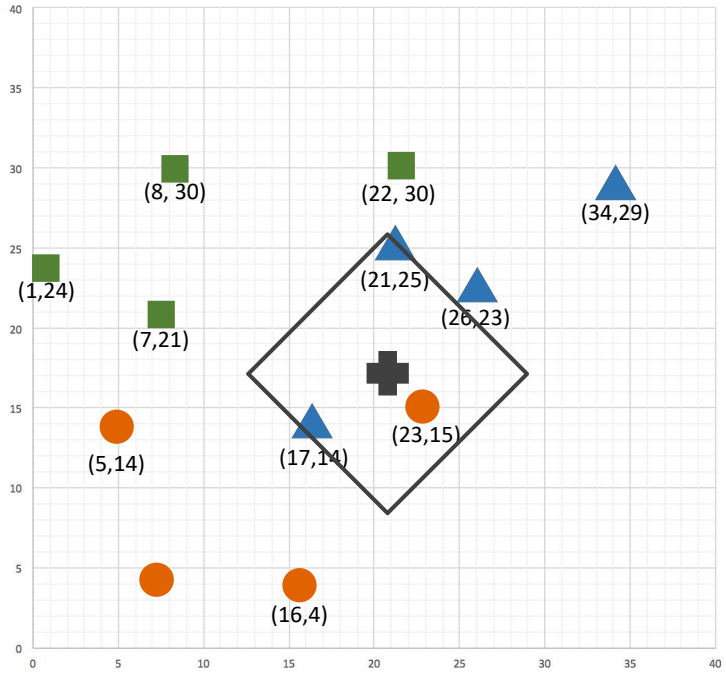
58 of 69

This is what using L_2 norm / distance would look like



59 of 69

What if we use L_1 instead?



60 of 69

(Digression) Norms in relation to losses

Suppose we are looking for a single point a, that has the smallest distance to all other points in your dataset.

- If we define our distance to be $L_0 = \sum$ optimal value of a?
  - Answer: Mode (Can you think of why?)
- If we define our distance to be $L_1 = \sum$ value of a?
  - Answer: Median (Can you think of why?
- If we define our distance to be $L_2 = \sum$ optimal value of a?
  - Answer: Mean
  - This is the reason why the formula of va

$\left(var = \frac{1}{n}\sum(x_i - \mu)^2\right)$

Derivation for ℓ_2 loss minimization:

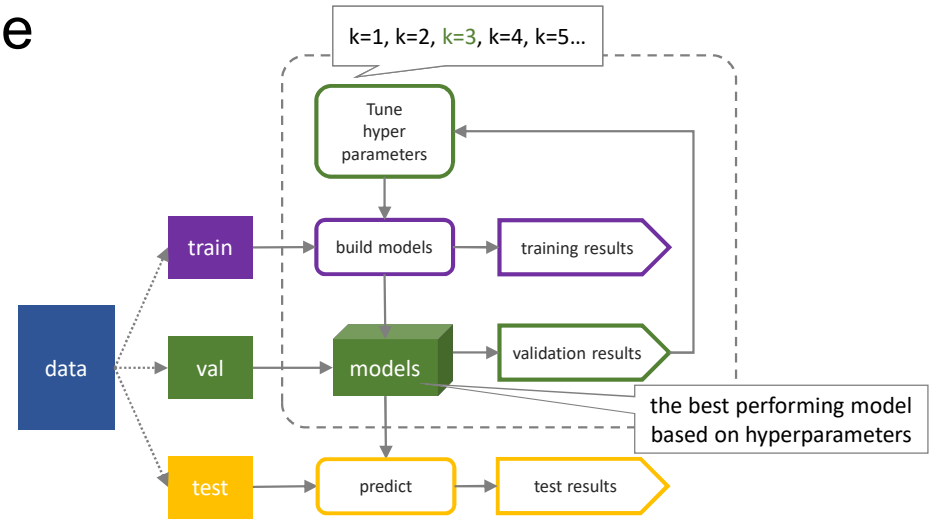
- $\ell_2\ loss = \sum_{i=1}^{n}(x_i - a)^2$
- $\frac{d}{dx}\sum_{i=1}^{n}(x_i - a)^2 = 0$
- $2\sum_{i=1}^{n}(x_i - a) = 0$
- $\sum_{i=1}^{n}(x_i - a) = 0$
- $\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} a = 0$
- $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} a$
- $\sum_{i=1}^{n} x_i = a\sum_{i=1}^{n} 1$
- $\sum_{i=1}^{n} x_i = an$
- $\frac{1}{n}\sum_{i=1}^{n} x_i = a$

61 of 69

How do we choose the best value for k?

62 of 69

Pipeline



63 of 69

Setting / Calibrating Hyperparameters

Idea #1 Choose hyperparameters that work best on the data

BAD: $k = 1$ always works perfectly on the training data

Idea #2 Split data into **train** and **test**, choose hyperparameters that work best on the **test** data

BAD: No idea how the algorithm will perform on new data

Idea #3 Split data into **train**, **validation**, and **test** then choose hyperparameters on **validation** and **evaluate** on test

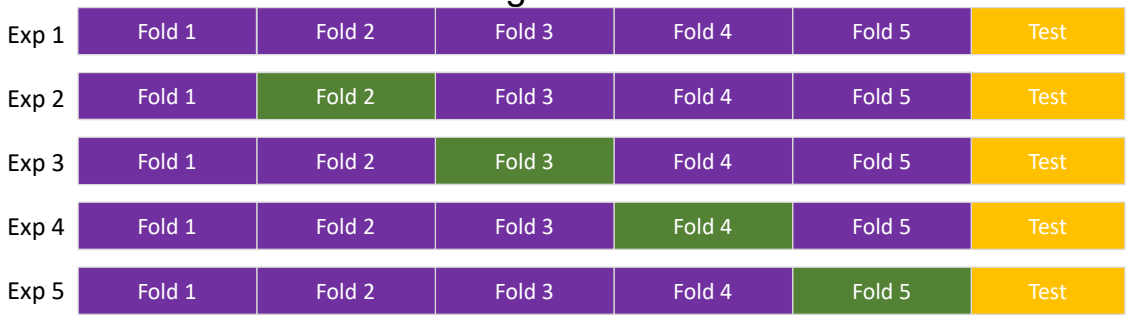
Good!

64 of 69

Setting / Calibrating Hyperparameters

Useful for small datasets, but not used too frequently in deep learning.

Idea #4 Cross-validation: Split data into folds, try each fold as validation and average the results



65 of 69

Disadvantages of kNN

- Computations happen at test-time rather than during training time
- Considers all the attributes all the time – and they all have equal weights

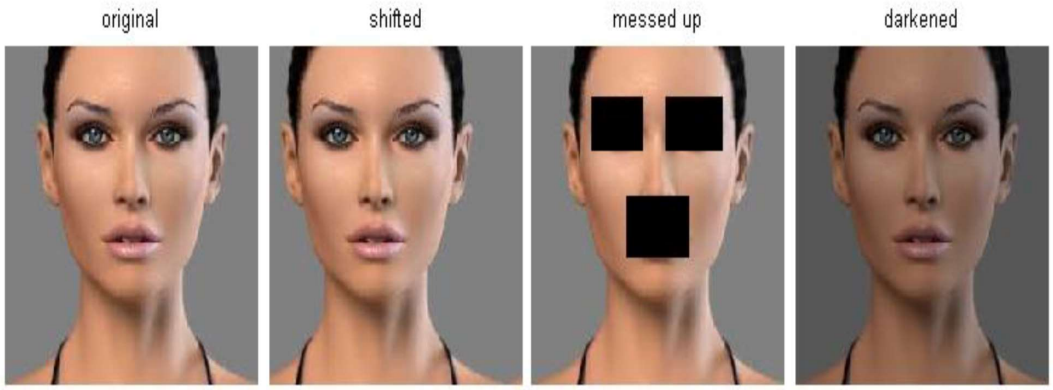
Advantages of kNN

- Fast training time
- Easy to implement

66 of 69

Is k-Nearest Neighbor good for image data?

- No! Distance metrics on pixels are not informative! All 3 images below have the same L_2 distance to the original image.



67 of 69

Slide credit: cs231n Fei-Fei Li, Justin Johnson, Serena Yeung

Programming Assignment 1

- For your first programming assignment you will be going through the basic classification pipeline by implementing the k-Nearest Neighbor algorithm to classify images.
  - You will need Anaconda 3.6 in order to do the assignment.
- First you will implement it using two for-loops which computes for the distance of every test point to every other training point.
- Then you will implement it without using any loops.
  - Hint: You will have to rely on your linear algebra (matrices / vector operations)
- Lastly, you will implement cross-fold validation and find the best value for the hyperparameter k.

68 of 69

References / Slide Credits

- Fei-Fei Li et al
- Derek Hoiem
- Jia-Bin Huang
- Svetlana Lazebnik