Computer Vision Foundations

Girish Gore
www.linkedin.com/in/datascientistgirishgore

About Me

Professional Experience

- 10+ Years of Industry Experience (5+ Y: SAS Analytics ,3+ Y : Cognizant , 2+: Start Ups)
- Artificial Intelligence,Algorithms , Analytics & Data Products
- Data Scientist by passion
- Founder two companies
  - ThinkBiggerAnalytics in India (Data Science Consulting Firm)
  - Recent Reomnify in Singapore (Data Product in Location Intelligence)

Training Experience

250 + Learners

- Mentor in Data Science at SpringBoard, SF
- Adjunct Faculty : Machine & Deep Learning Using Python at Aegis School Of Data Science
- Adjunct Faculty : Advanced Deep Learning at Great Learning
- 5+Y Training Corporates & Executives with experience ranging from 5 years to 25+ years

Hobbies

- Playing Tennis , Foodie (btw a pure Vegetarian :))
- Meditation & Vedic Philosophy

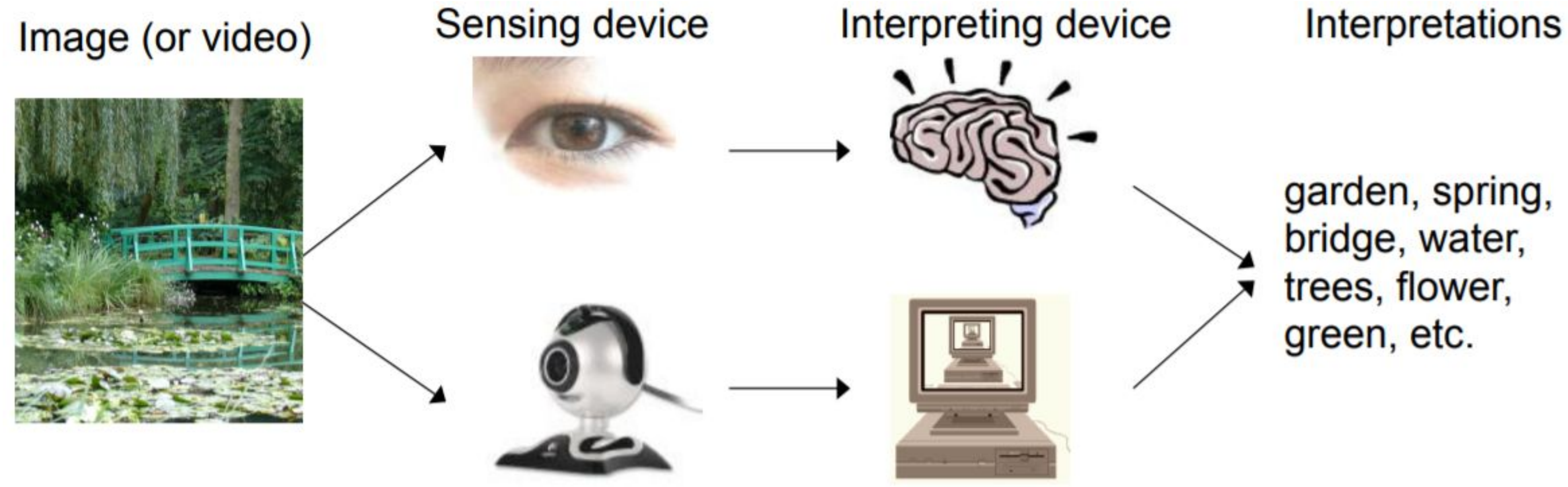
Computer Vision Foundations

- Basic Computer Vision
  - Computer Vision an Introduction
  - Fundamentals of Image Processing
- Convolutional Neural Networks
- CNN Architectures
- Transfer Learning & Applications

Pre Requisites

- Machine Learning Overall Process Understanding
- Deep Learning Foundations

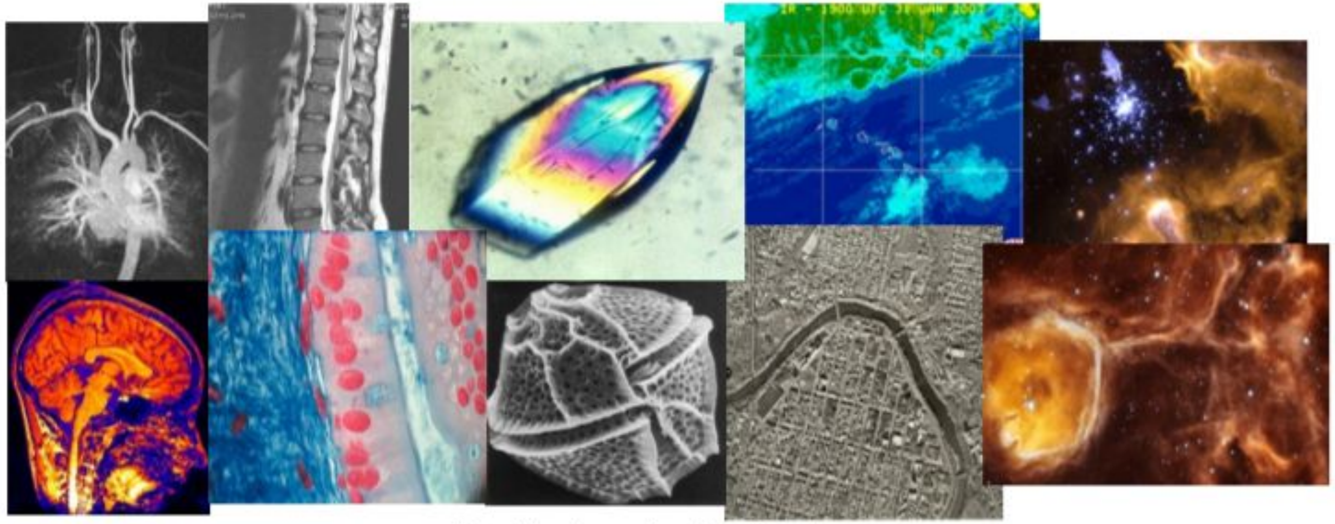
Computer Vision is the science & technology of computers that can see



Why Now ?

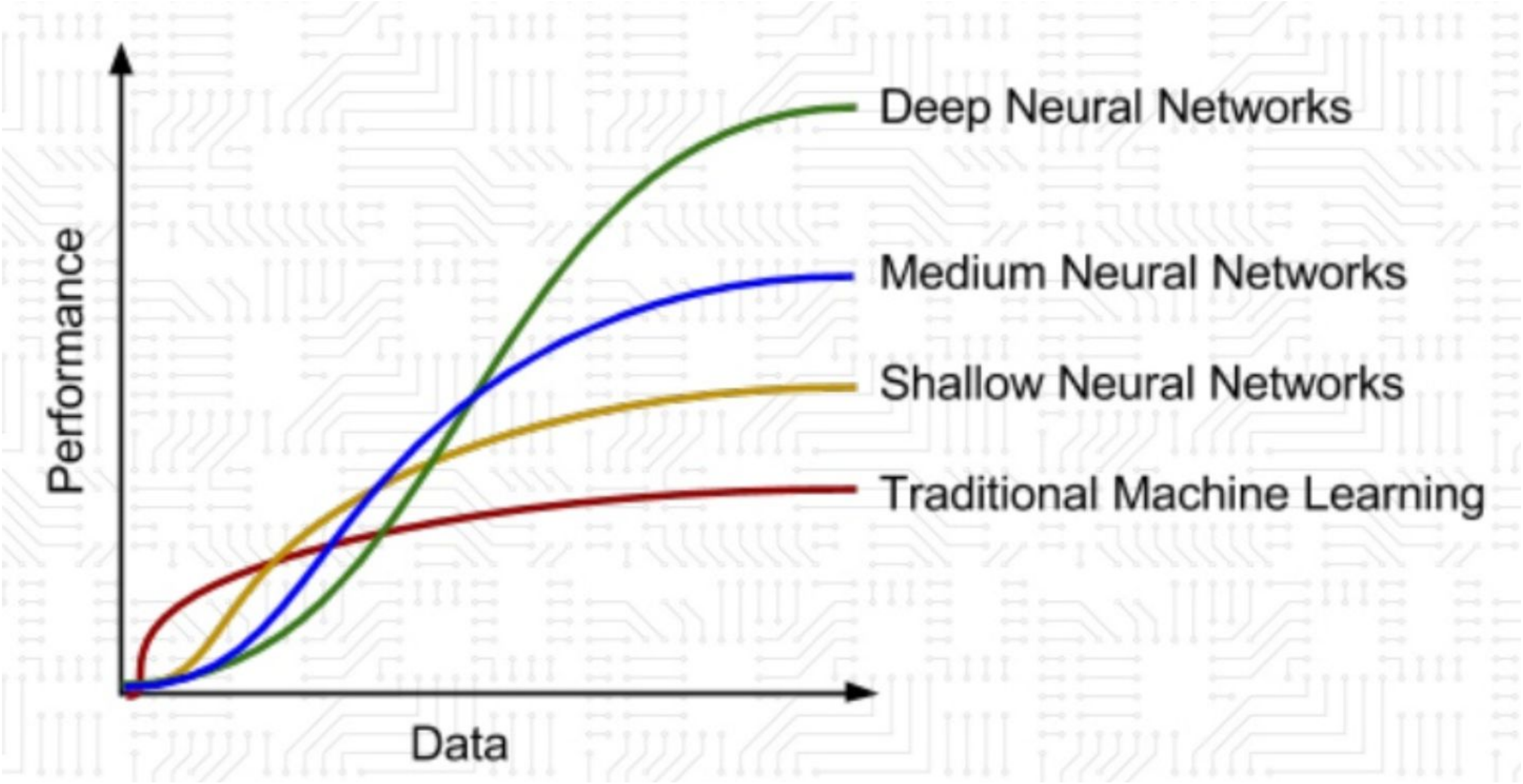
Google Image Search Picasa flickr GAMMA webshots beta picsearch YouTube Broadcast Yourself

Surveillance and security

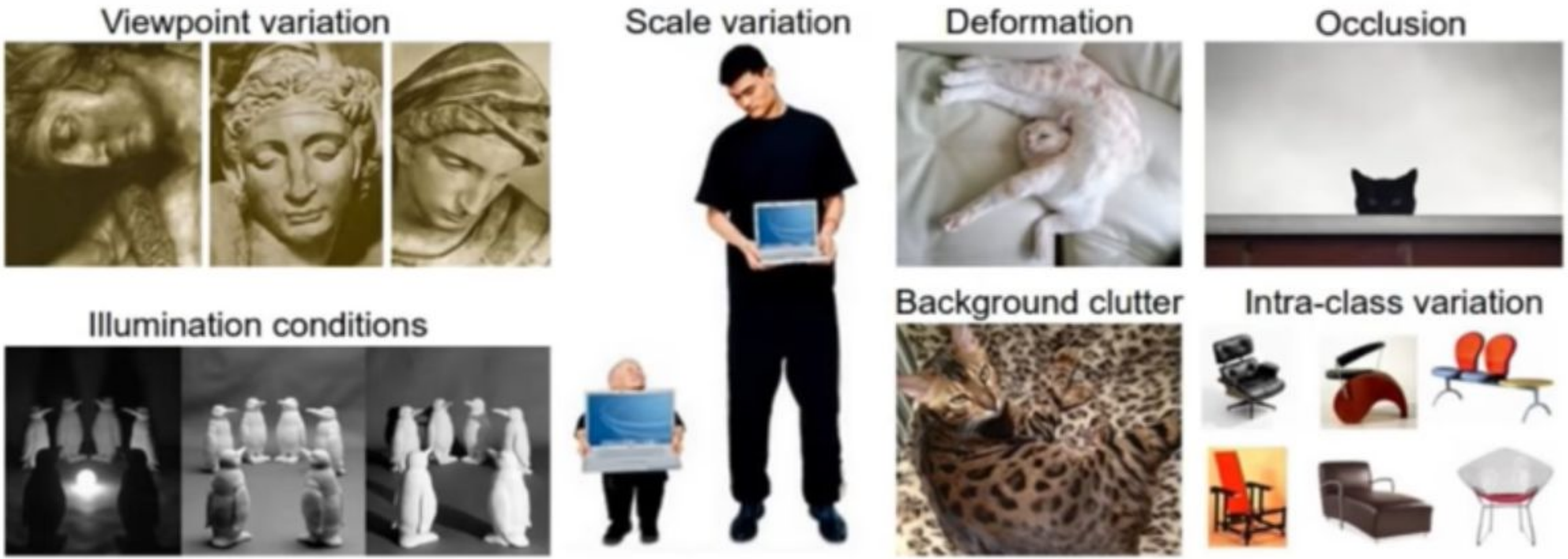


Medical and scientific images

Data vs Performance



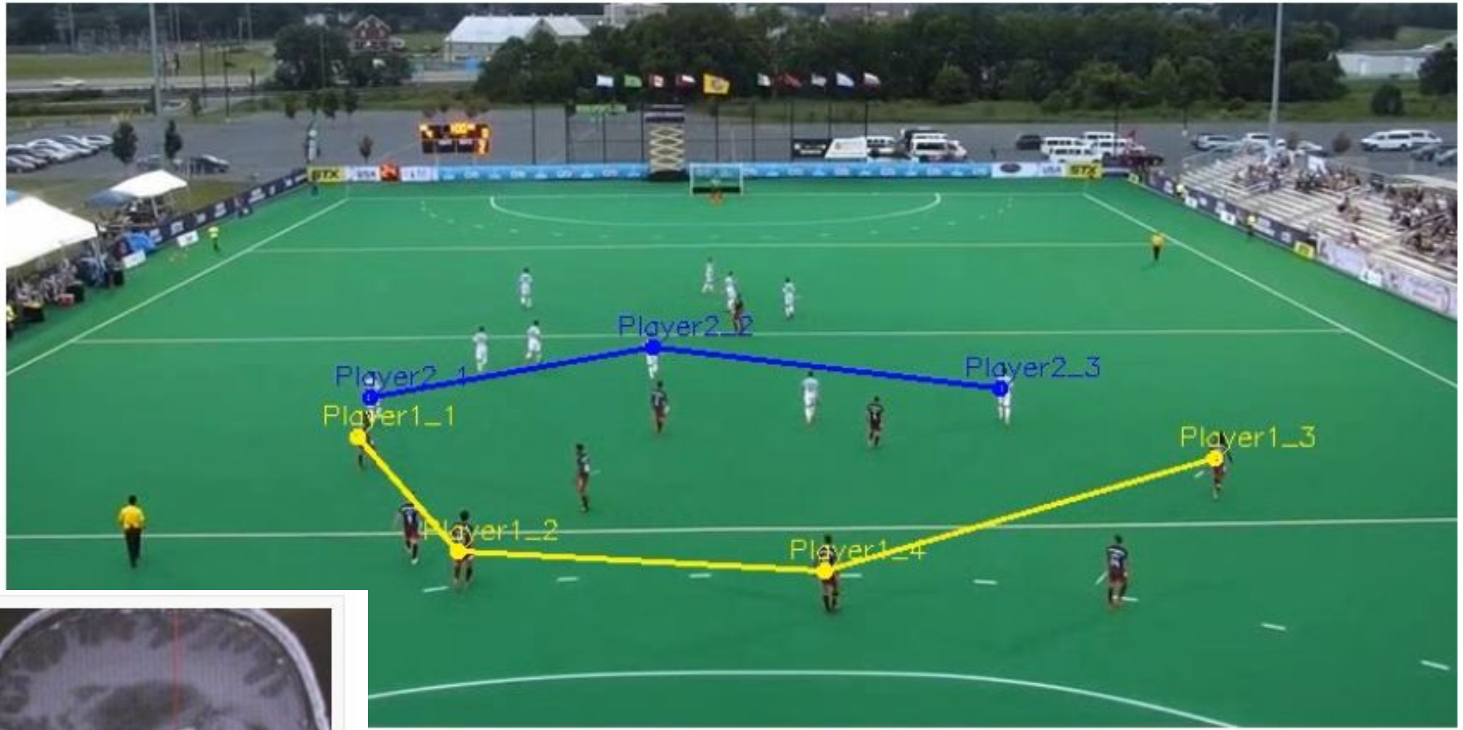
Challenges in feature extraction in images



Computer Vision Applied

Image Source: Amazon.com

Computer Vision Applied



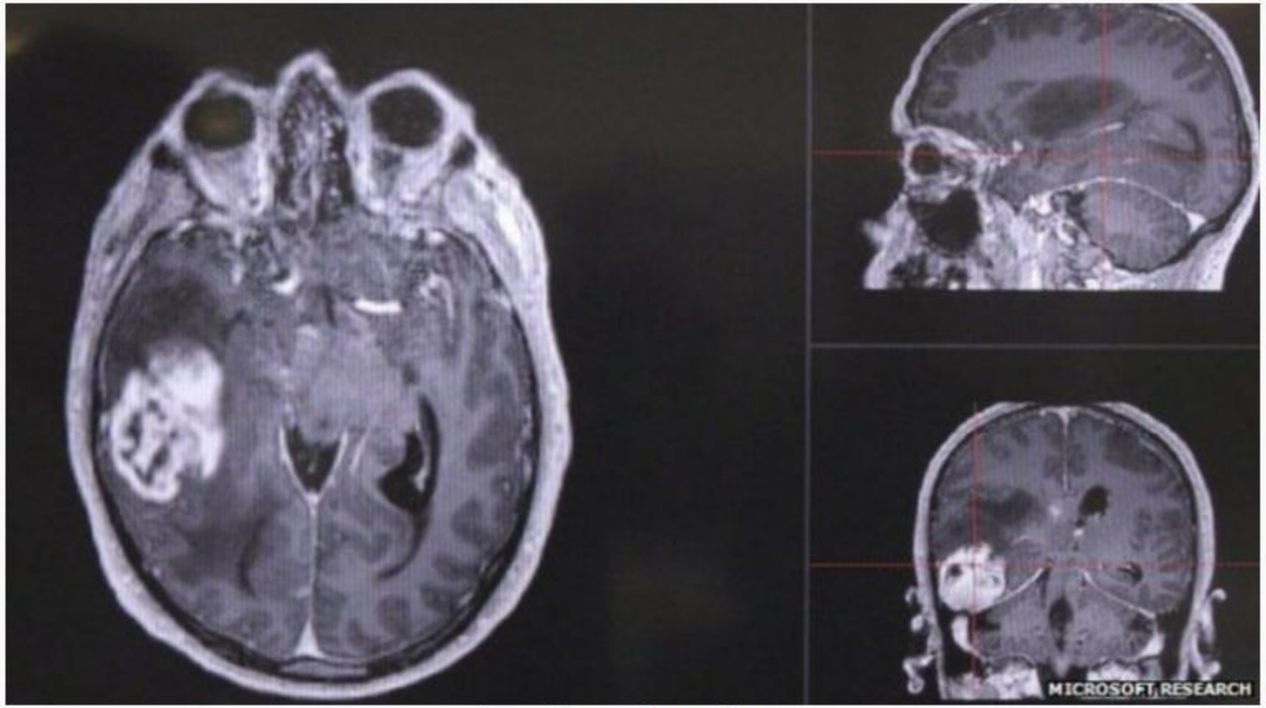
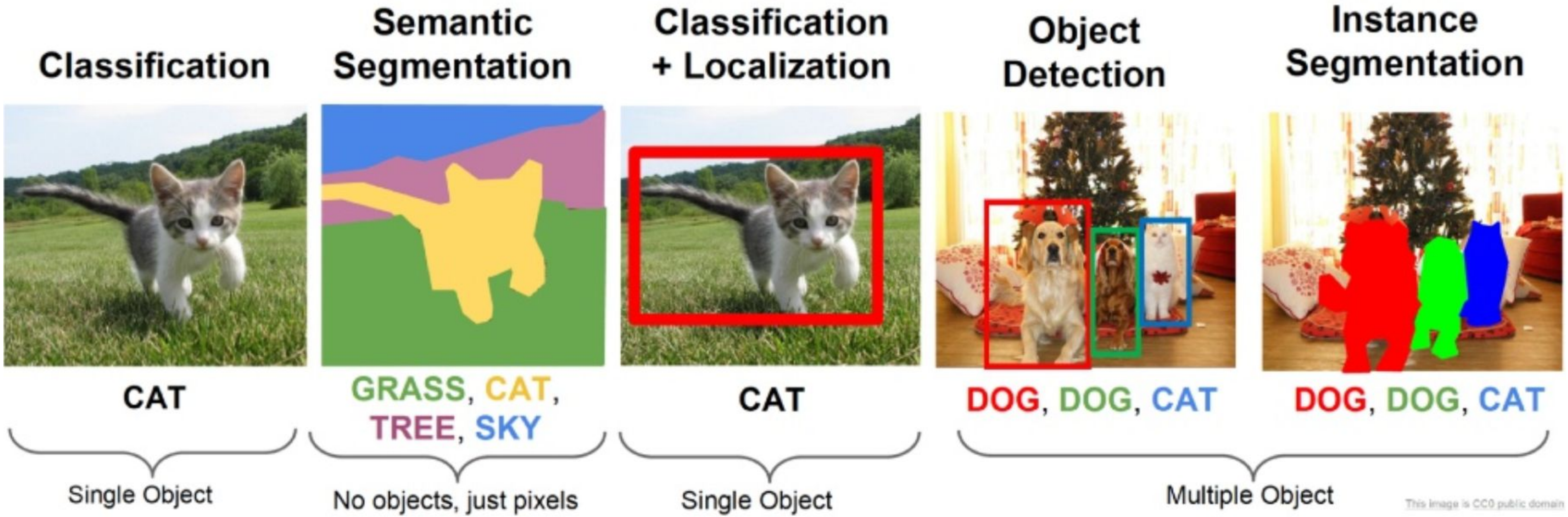


Image Source: Microsoft.com

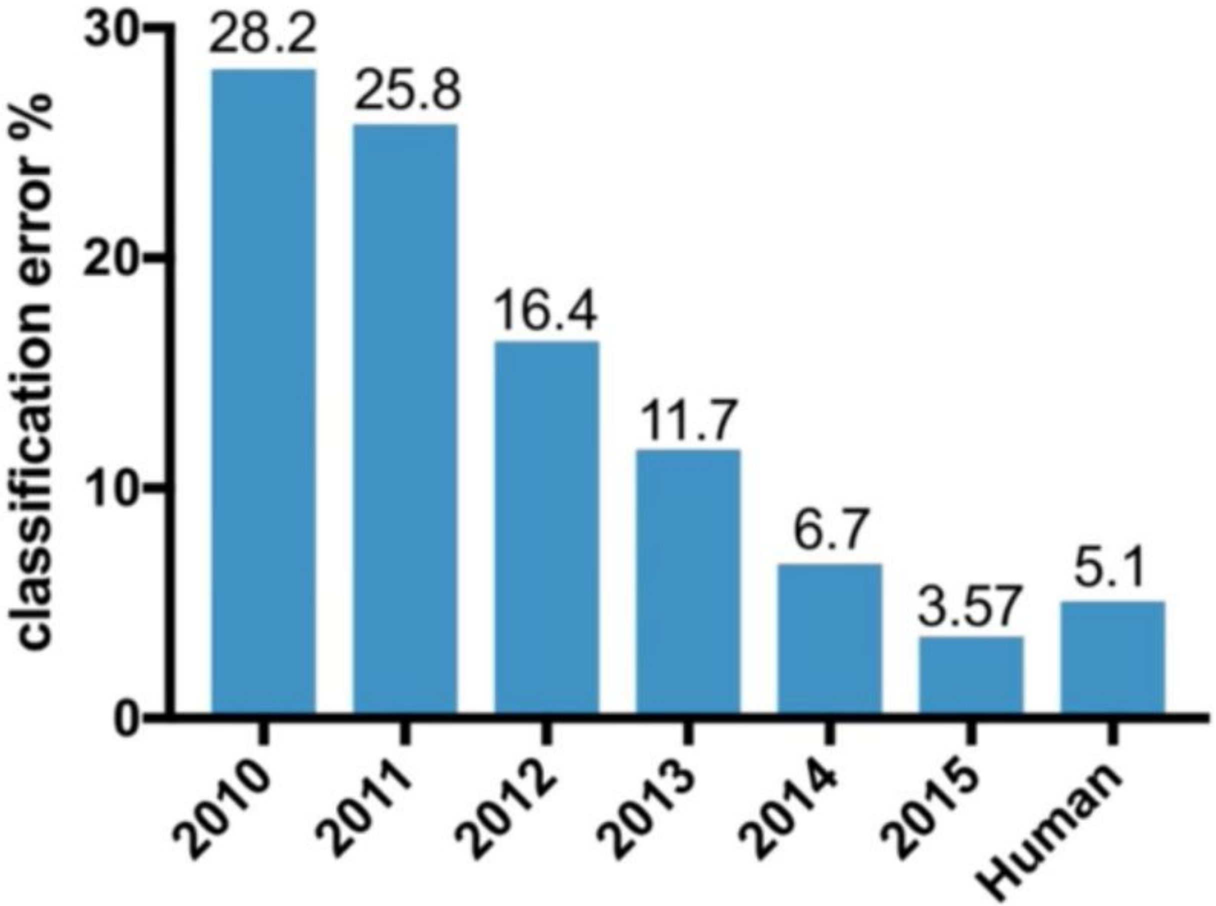
Computer Vision Tasks



Computer Vision Tasks … more …

- Style Transfer
- Object Tracking
- Image ReConstruction
- Image Synthesis ….

ImageNet Competition- Classification Task History



2012: AlexNet. First CNN to win.
- 8 layers, 61 million parameters

2013: ZFNet
- 8 layers, more filters

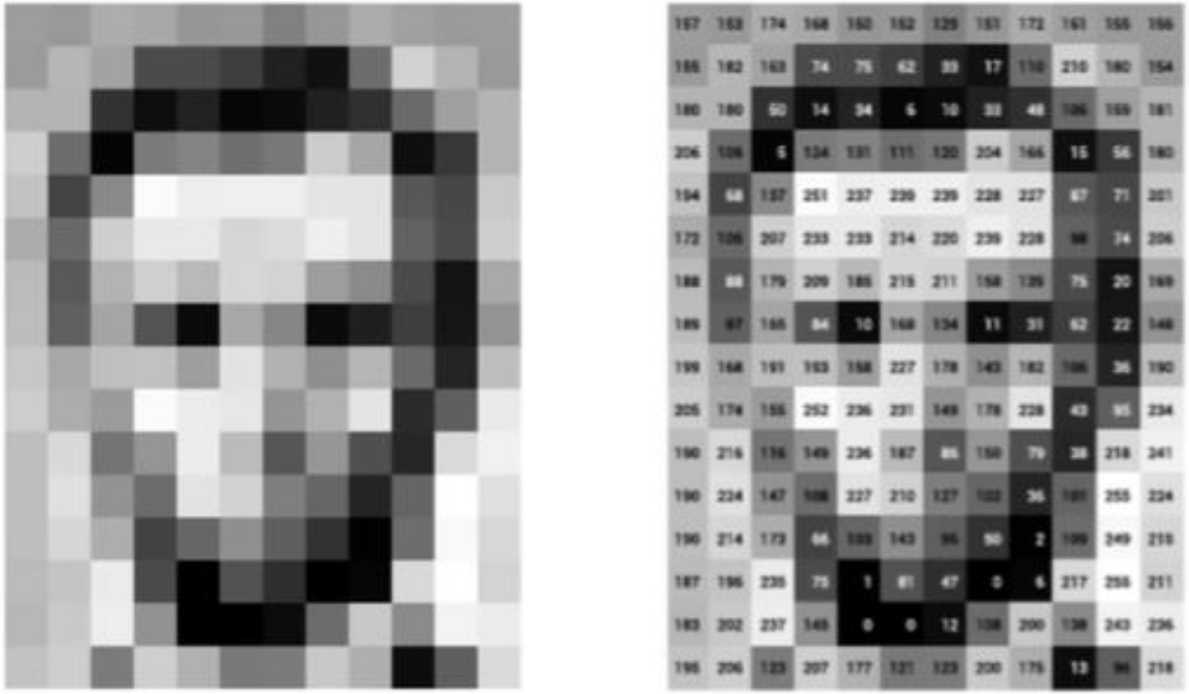
2014: VGG
- 19 layers

2014: GoogLeNet
- "Inception" modules
- 22 layers, 5million parameters

2015: ResNet
- 152 layers

How computers view images?

greatlearning
Learning for Life



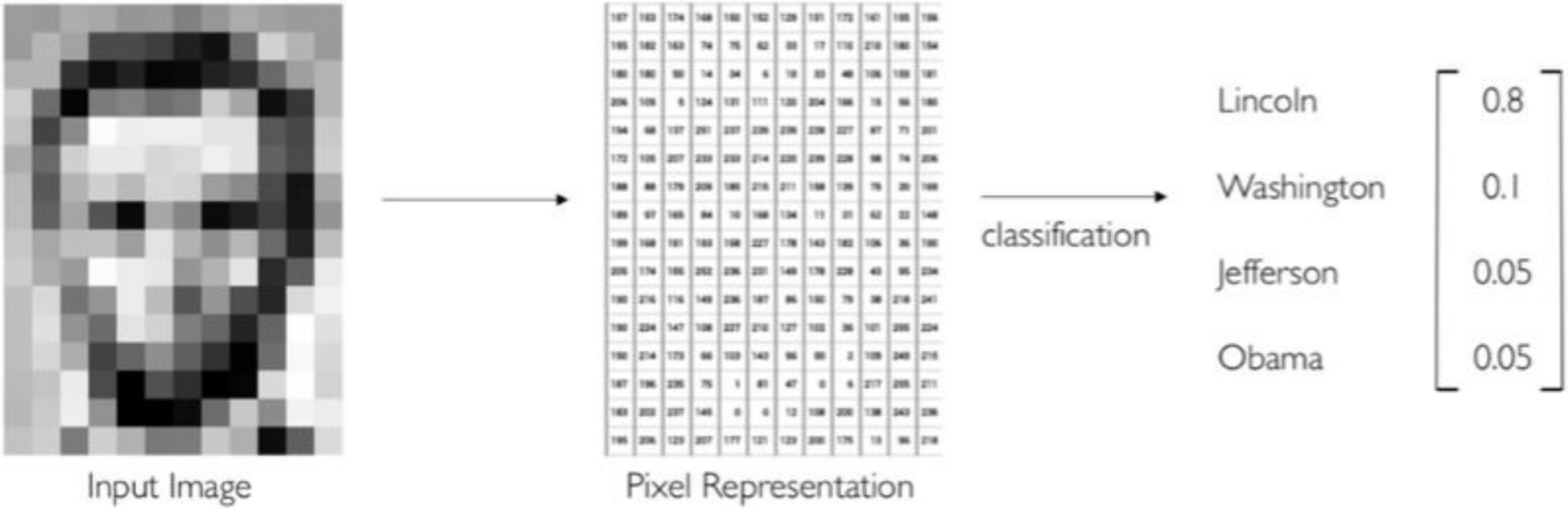
What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

The goal of computer vision is to bridge the gap between pixels & "meaning" they convey

Common Problems we can solve



- **Regression**: output variable takes continuous value
- **Classification**: output variable takes class label. Can produce probability of belonging to a particular class

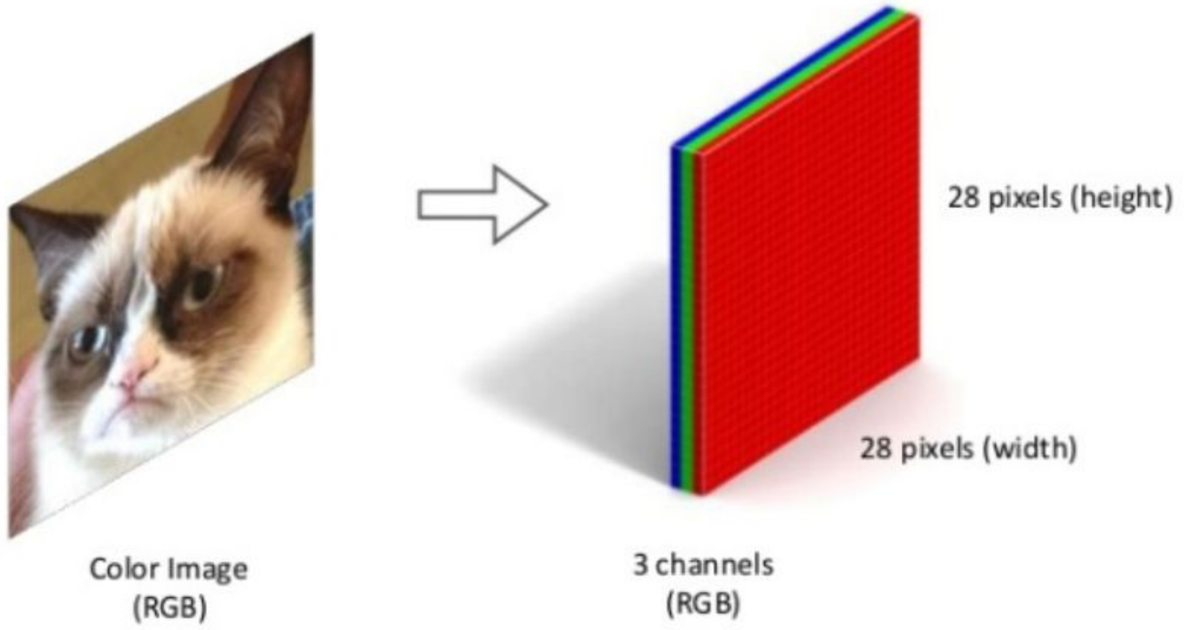
Images as Tensors

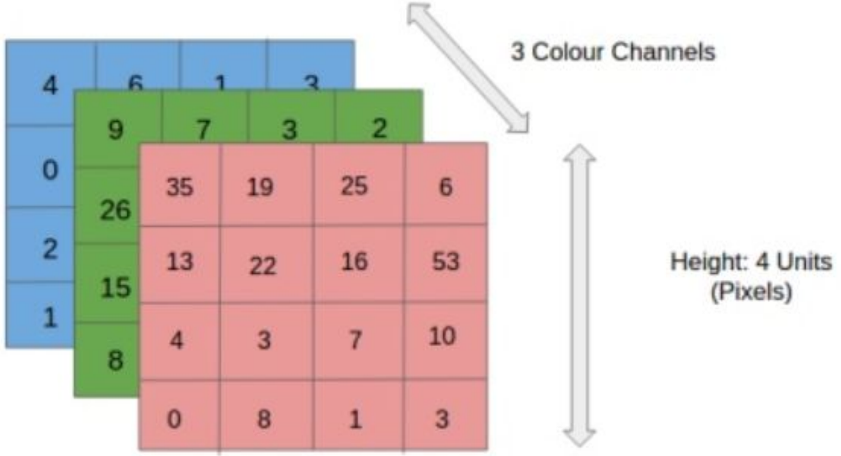
- A scalar image has 2^a - 1 integer values

$$u\ 2\ \{0,\ 1,\ ...,\ 2^a — 1\}$$

- a: level (bit)

- Ex. If 8 bit (a=8), image spans from 0 to
  - 0 black
  - 255 white
- Ex. If 1 bit (a=1), it is binary image, 0 and

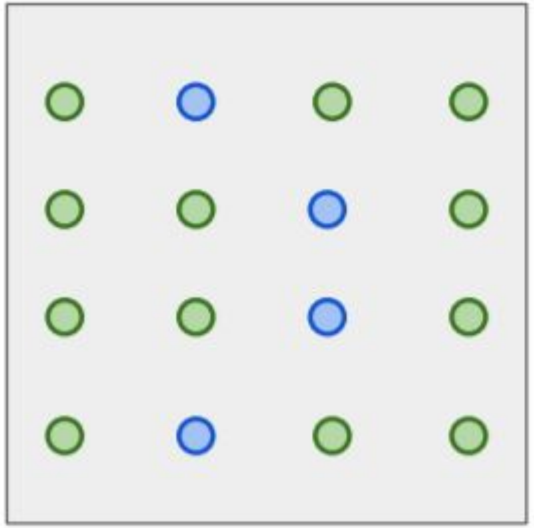


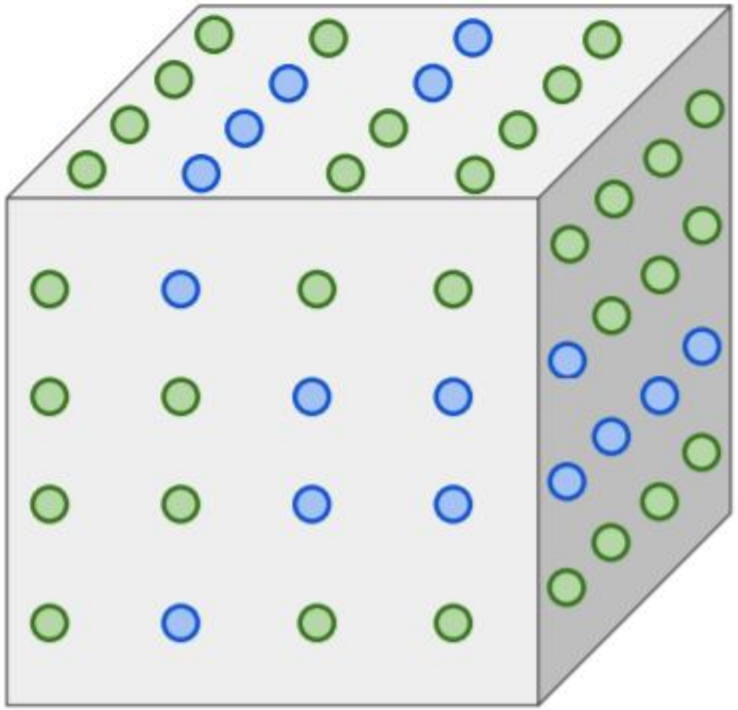


Curse of Dimensionality

Dimensions = 1
Points = 4

Dimensions = 2
Points = 4^2





Images as Functions

- **An Image** as a function f from R^2 to R^M:
  - $f(x, y)$ gives the **intensity** at position (x, y)
  - Defined over a rectangle, with a finite range:

$$f: \underbrace{[a,b] \times [c,d]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$

- A color image: $f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$

Filtering

Form a new image whose pixels are a combination of original pixel values

Goals:

- Extract useful information from the images
    - Features (edges, corners, blobs...)
- Modify or enhance image properties
    - super-resolution;de-noising

Filtering Ex: Moving Average

In summary

- Replaces each pixel with an average of its neighborhood.

- Achieve smoothing effect (remove sharp features)

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filtering Ex: Moving Average

A 2D moving average over a 3 * 3 window of neighbourhood

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$(f * h)[m,n] = \frac{1}{9} \sum_{k,l} f[k,l]\, h[m-k, n-l]$$

Filtering Ex: Image Segmentation

Image segmentation based on very basic threshold

$$g[n,m] = \begin{cases} 255, & f[n,m] > 100 \\ 0, & \text{otherwise.} \end{cases}$$

Derivative : Rate Of Change

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) = f_x$$

$$\frac{df}{dx} = \frac{f(x) - f(x-1)}{1} = f'(x)$$

Backward Difference

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x)$$

Forward Difference

$$\frac{df}{dx} = f(x) - f(x+1) = f'(x)$$

Central Difference

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x)$$

Derivative Mask

$f(x) =$	10	15	10	10	25	20	20	20
$f'(x) =$	0	5	-5	0	15	-5	0	0
$f''(x) =$	0	5	-10	5	15	20	5	0

Backward Difference Mask [-1,1]

Forward Difference Mask [1,-1]

Central Difference Mask [-1,0,1]

Derivative 2 Dimension

Given function $f(x,y)$

Gradient vector $$\nabla f(x,y) = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

Gradient magnitude $$|\nabla f(x,y)| = \sqrt{f_x^2 + f_y^2}$$

Gradient direction $$\theta = \tan^{-1} \frac{f_x}{f_y}$$

Derivative 2 Dimension

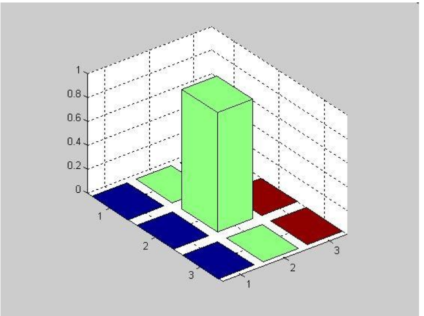
Given function $f(x, y)$

Gradient vector $\nabla f(x, y) = \begin{bmatrix} \dfrac{\partial f(x, y)}{\partial x} \\ \dfrac{\partial f(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$

Gradient magnitude $|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$

Gradient direction $\theta = \tan^{-1} \dfrac{f_x}{f_y}$

Understanding Filters

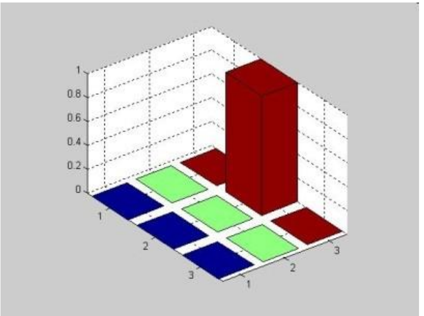


*

0	0	0
0	1	0
0	0	0

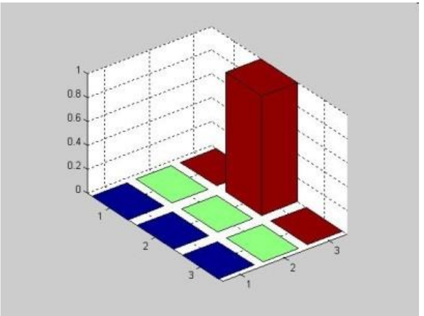
=

Filtering examples



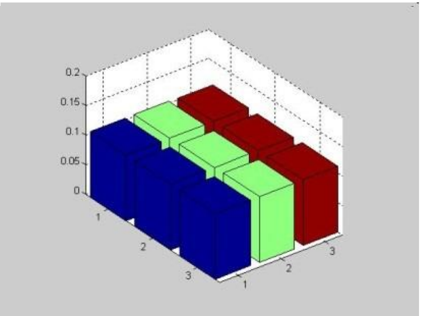
$$* \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} =$$

Filtering examples



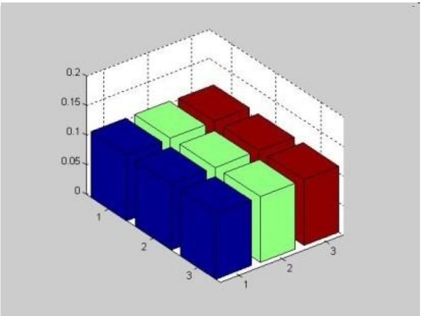
$$*\begin{bmatrix}0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} =$$

Filtering examples



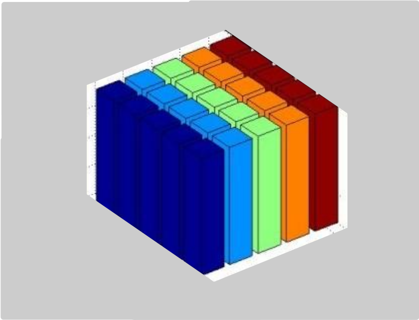
$$* \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Filtering examples



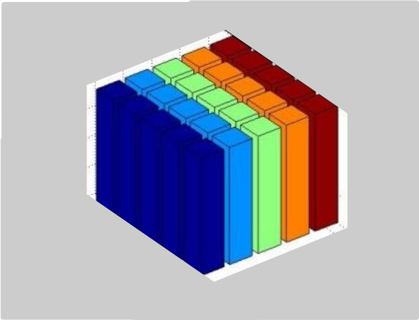
$$* \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$

Filtering examples



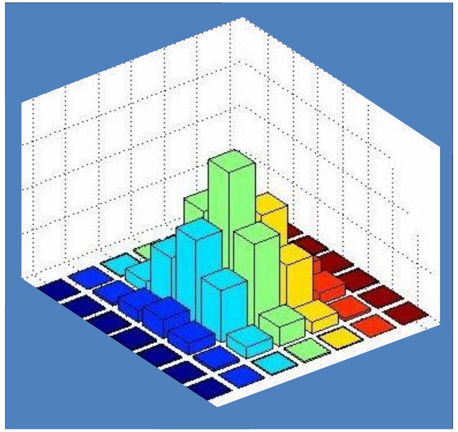
$$* \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Filtering examples



$$* \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

Filtering examples - Gaussian



Filtering example – Gaussian vs. Smoothing

Gaussian Smoothing

Smoothing by Averaging

Filtering example – Noise filtering

Gaussian Smoothing

Smoothing by Averaging

Filtering example – Noise filtering

Gaussian Noise

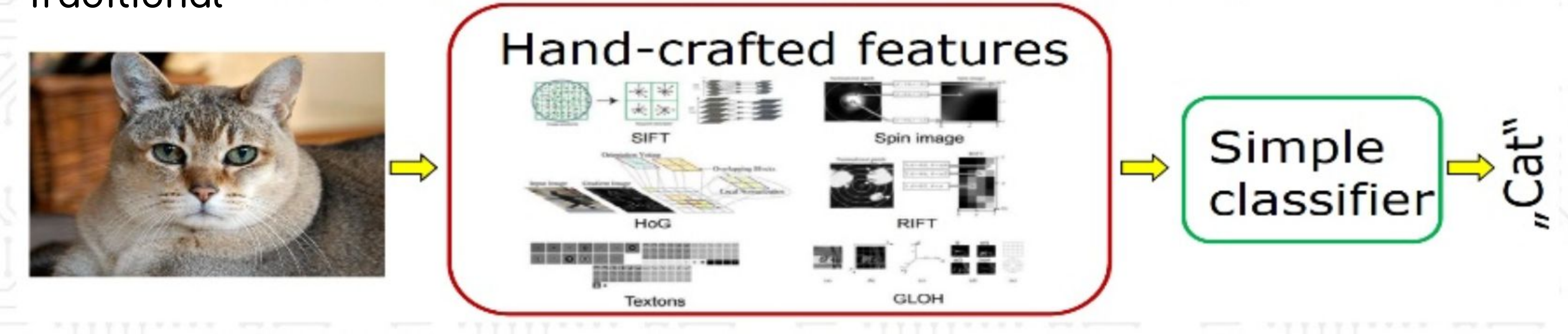
After averaging

After Gaussian Smoothing

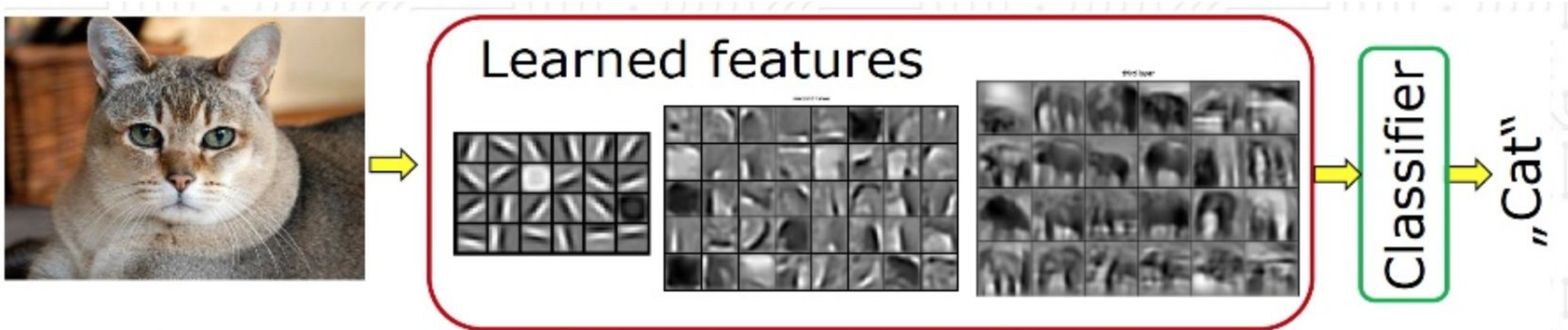
greatlearning
Learning for Life

Computer Vision Yesterday & Today

Traditional



Deep Learning



References

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

He, Kaiming, et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition." arXiv preprint arXiv:1406.4729 (2014). [PDF]

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

Fei-Fei, Li, et al. "What do we perceive in a glance of a real-world scene?." Journal of vision 7.1 (2007):

Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection."

Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.•

Felzenszwalb, Pedro, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model."

Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008

Everingham, Mark, et al. "The pascal visual object classes (VOC) challenge." International Journal of Computer Vision 88.2 (2010): 303-338.

Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.
Russakovsky, Olga, et al. "Imagenet Large Scale Visual Recognition Challenge." arXiv:1409.0575. [PDF]