台灣人工智慧學校

卷積神經網路 Convolutional Neural Network
&
電腦視覺 Computer Vision
Part1

林彥宇 & 教研處

「版權聲明頁」

本日課程內容

本日課程:

1. 電腦視覺入門
2. CNN原理與介紹

延伸閱讀 (Optional):

1. Before LeNet
2. CNN Application
3. OpenCV

本次課程結束後你 (妳) 應該會什麼?

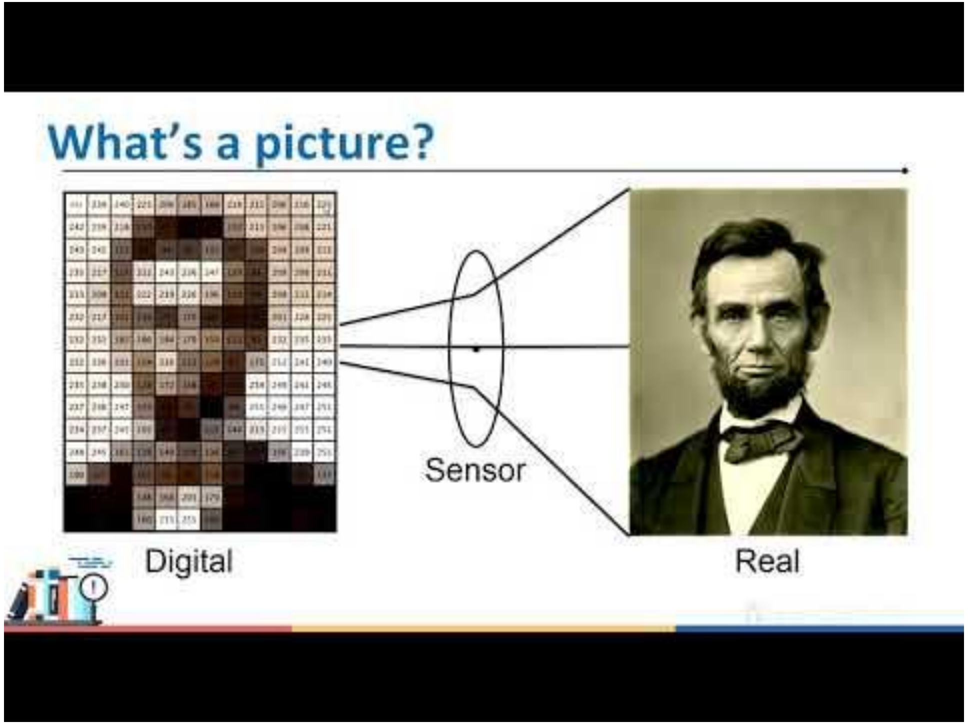
- **軟實力**
  - 了解 Convolution (卷積) 背後的原理及為何其有效
  - 了解Filter的工作原理與技法
- **硬底子**
  - 如何用 TensorFlow 寫出基本的 CNN
  - (optional)使用OpenCV來進行圖片的前處理

Code / Data 放在 hub 中的 courses 內

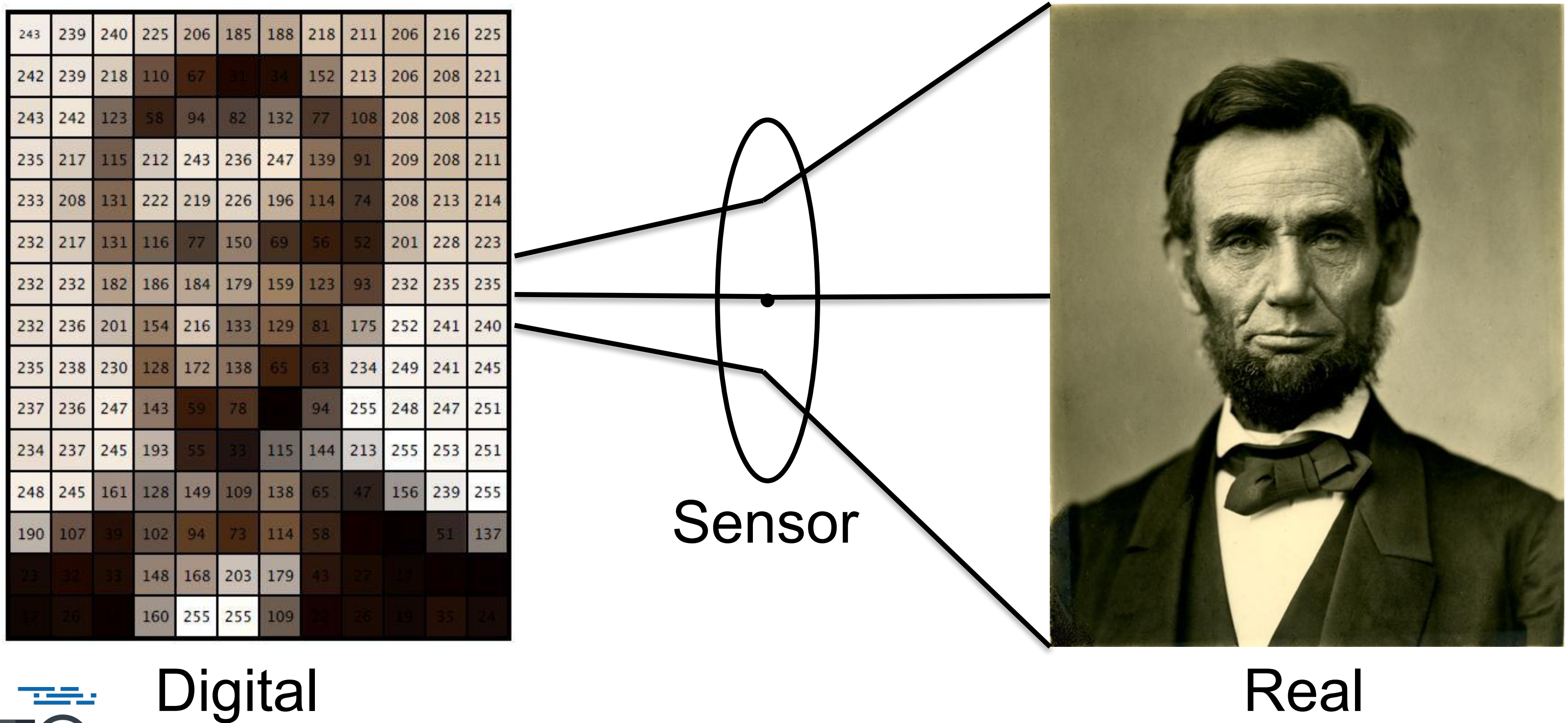
- 為維護課程資料 , courses 中的檔案皆為 read-only, 如需修改請 cp 至自身的環境中
- 打開 terminal, 輸入
    - cp -r courses-tpe/CVCNN/part1/ <存放至本機的名稱>

電腦視覺入門

Computer Vision



What's a picture?



What’s a picture?

$$f: R^2 \to R$$

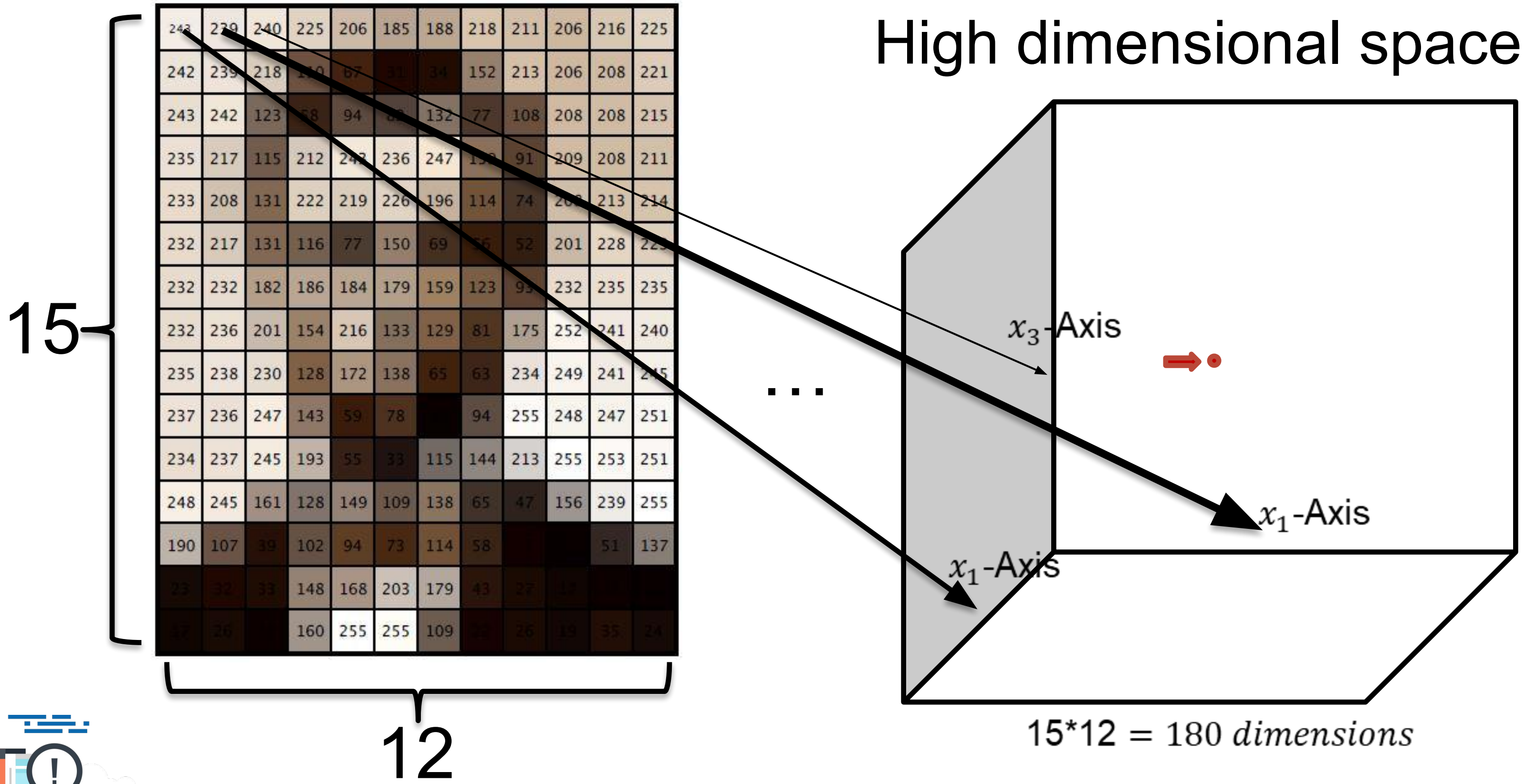
$$f(x, y) =$$

243	239	240	225	206	185	188	218	211	206	216	225
242	239	218	110	67	31	34	152	213	206	208	221
243	242	123	58	94	82	132	77	108	208	208	215
235	217	115	212	243	236	247	139	91	209	208	211
233	208	131	222	219	226	196	114	74	208	213	214
232	217	131	116	77	150	69	56	52	201	228	223
232	232	182	186	184	179	159	123	93	232	235	235
232	236	201	154	216	133	129	81	175	252	241	240
235	238	230	128	172	138	65	63	234	249	241	245
237	236	247	143	59	78	[illegible]	94	255	248	247	251
234	237	245	193	55	33	115	144	213	255	253	251
248	245	161	128	149	109	138	65	47	156	239	255
190	107	39	102	94	73	114	58	[illegible]	[illegible]	51	137
23	32	33	148	168	203	179	43	27	[illegible]	[illegible]	[illegible]
[illegible]	26	[illegible]	160	255	255	109	[illegible]	26	19	35	24

Intensity

GRAYSCALE		
255	230	205
180	155	115
80	40	0

What’s a picture?



Channels

GRAYSCALE		
255	230	205
180	155	115
80	40	0

GRAY = 1 SET OF DIGITS		
11111111	11100110	11001101
10110100	10011011	01110011
01010000	00101000	00000000

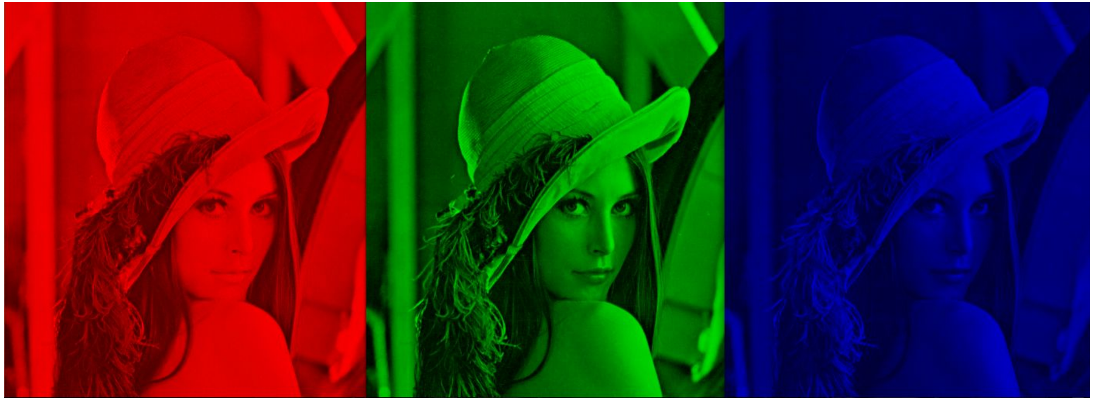
NUMBERS		
R 255 G 0 B 0	R 102 G 102 B 255	R 51 G 204 B 153
R 255 G 255 B 102	R 255 G 0 B 204	R 51 G 204 B 255
R 51 G 51 B 0	R 51 G 51 B 153	R 255 G 153 B 153

'RGB' = 3 SETS OF DIGITS		
11111111 00000000 00000000	01100110 01100110 11111111	00110011 11001100 10011001
11111111 11111111 01100110	11111111 00000000 11001100	00110011 11001100 11111111
00110011 00110011 00000000	00110011 00110011 10011001	11111111 10011001 10011001

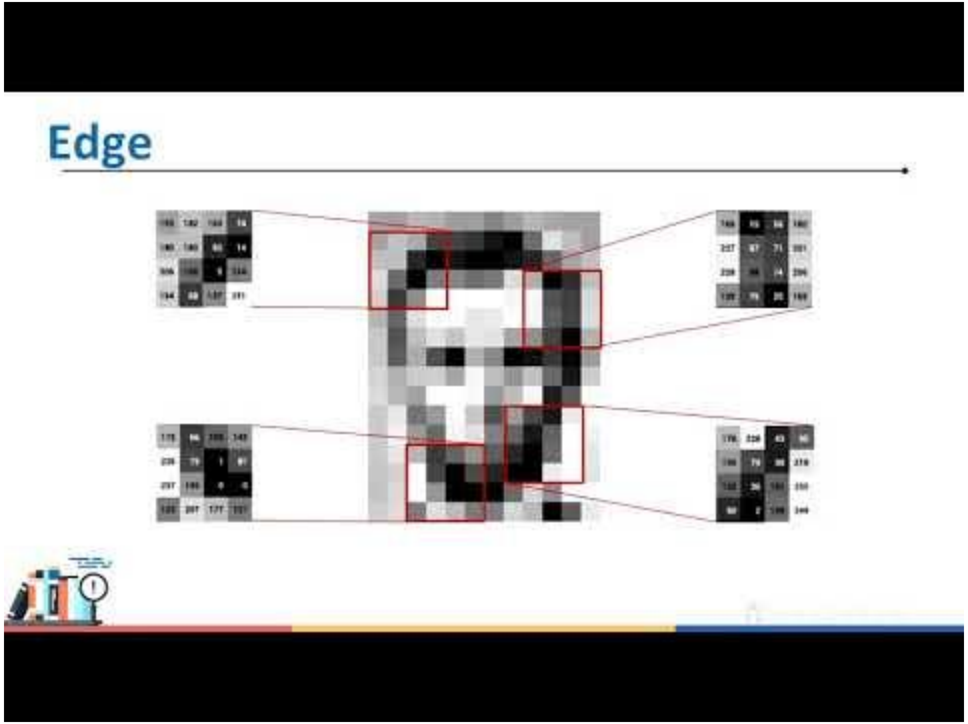
Grayscale

$$f(x,y) = [R(x,y), G(x,y), B(x,y)]$$

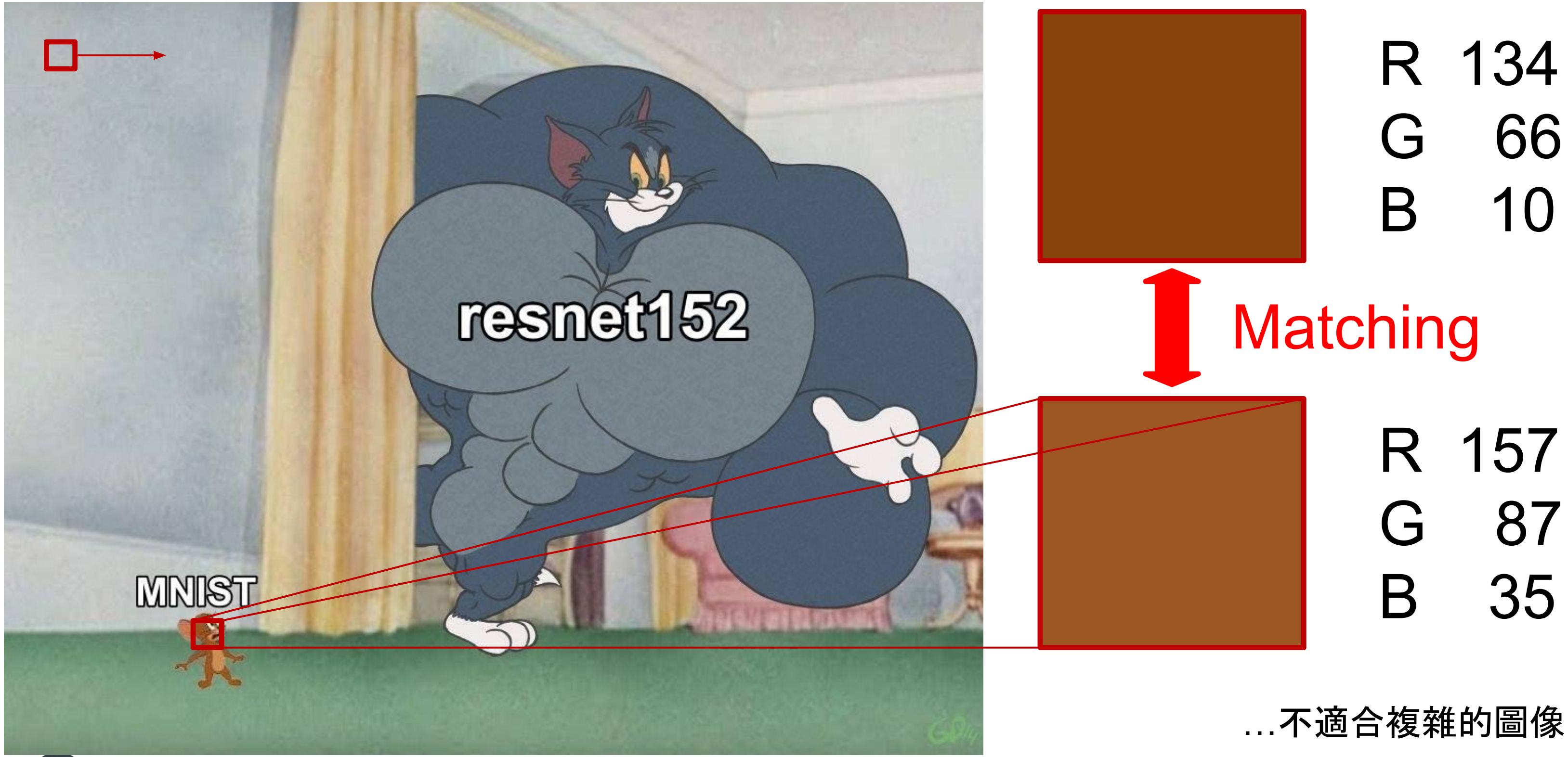
RGB 3 channels (not layers)



CV methods

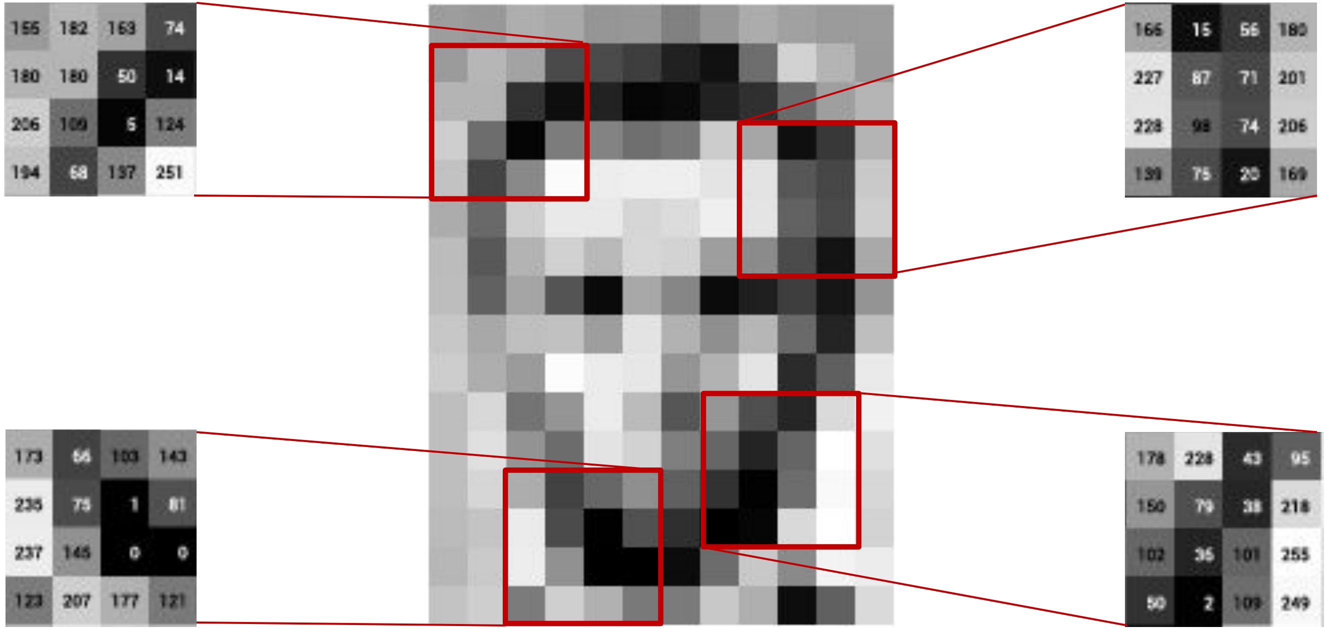


Pixel by pixel



…不適合複雜的圖像

Edge


155 182 163 74
180 180 50 14
206 109 5 124
194 68 137 251
166 15 56 180
227 87 71 201
228 98 74 206
139 75 20 169
173 66 103 143
236 75 1 81
237 145 0 0
123 207 177 121
178 228 43 95
150 79 38 218
102 36 101 255
50 2 109 249

Kernel

150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150
150	150	150	180	250	250	30	120	150	150

Filter or Kernel

-1	0	1
-1	0	1
-1	0	1

0

$$150 * -1 + 150 * 0 + 150 * 1$$
$$+150 * -1 + 150 * 0 + 150 * 1$$
$$+150 * -1 + 150 * 0 + 150 * 1$$

210

$$180 * -1 + 250 * 0 + 250 * 1$$
$$+180 * -1 + 250 * 0 + 250 * 1$$
$$+180 * -1 + 250 * 0 + 250 * 1$$

Convolution

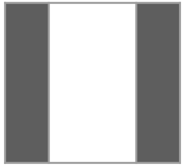
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

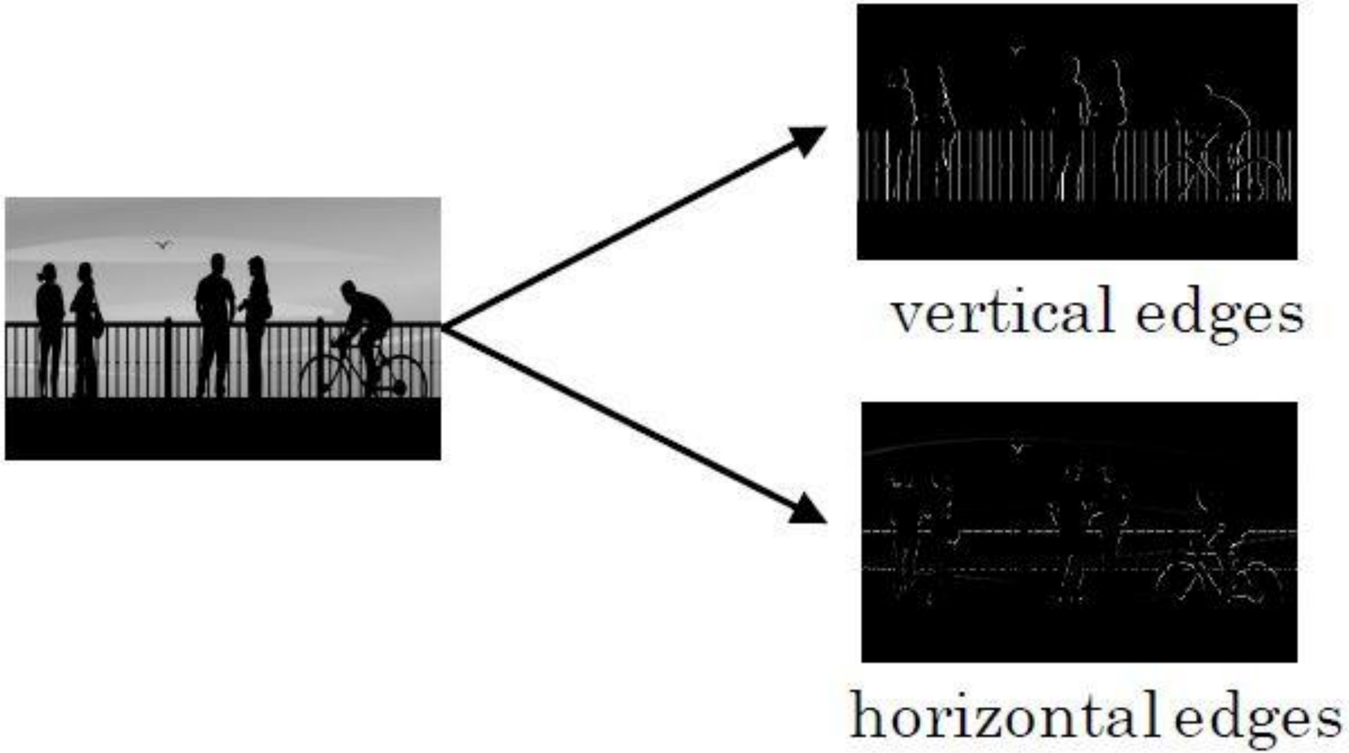
-1	0	1
-1	0	1
-1	0	1

=

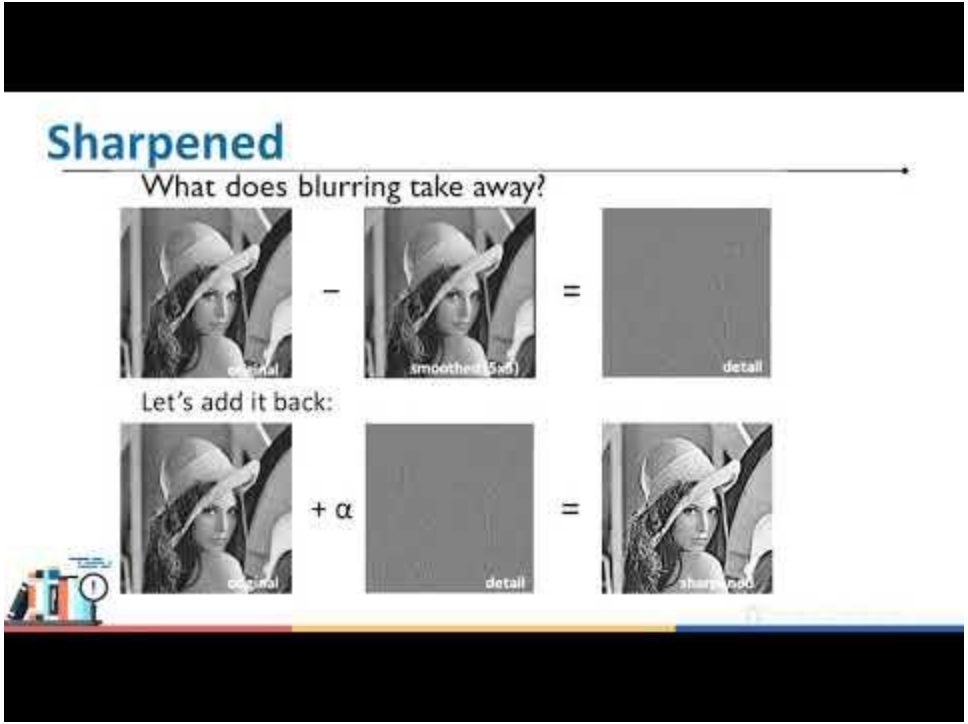
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



Different filters



Filters


Sharpened
What does blurring take away?
original
−
smoothed (5x5)
=
detail
Let's add it back:
original
+ α
detail
=
sharpened

Filters

original

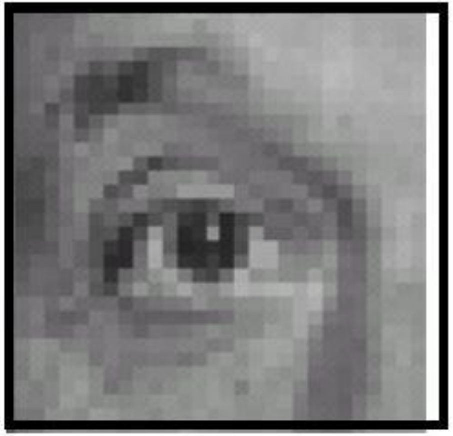
0	0	0
0	1	0
0	0	0

Filtered
(no change)

Shift

original

0	0	0
0	0	1
0	0	0

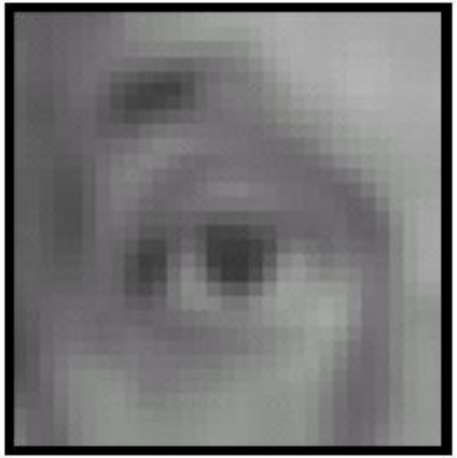


Shifted left
By 1 pixel

Blur(Mean filter)

original

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



Blur (with a box filter)

Sharpening

original

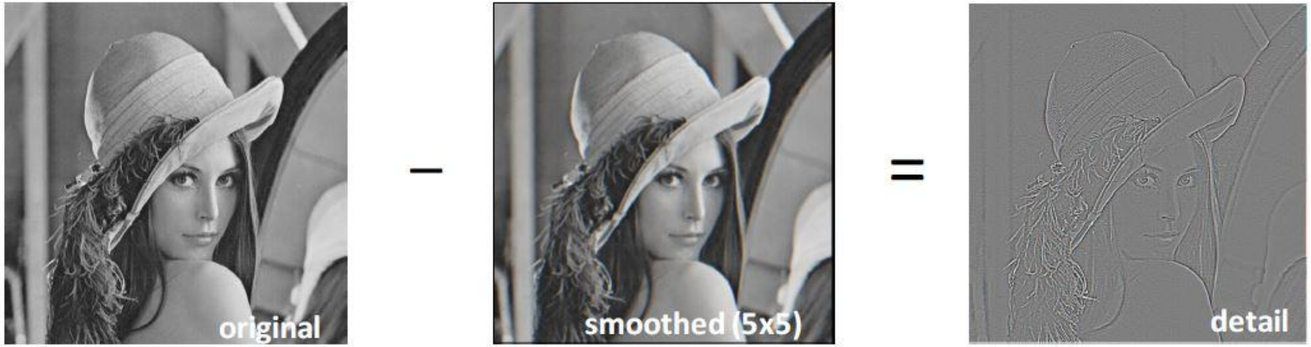
$$\begin{bmatrix}0&0&0\\0&2&0\\0&0&0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Sharpening filter

- Accentuates differences with local average

Sharpened

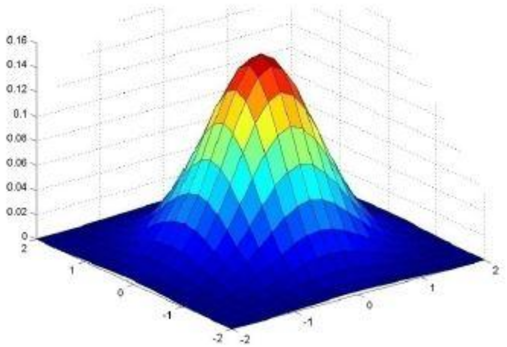
What does blurring take away?



Let's add it back:

Gaussian Filter

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



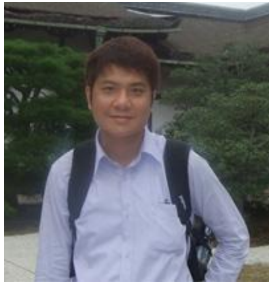
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, σ = 1

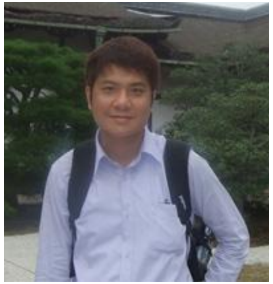
- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)

CNN的原理與介紹

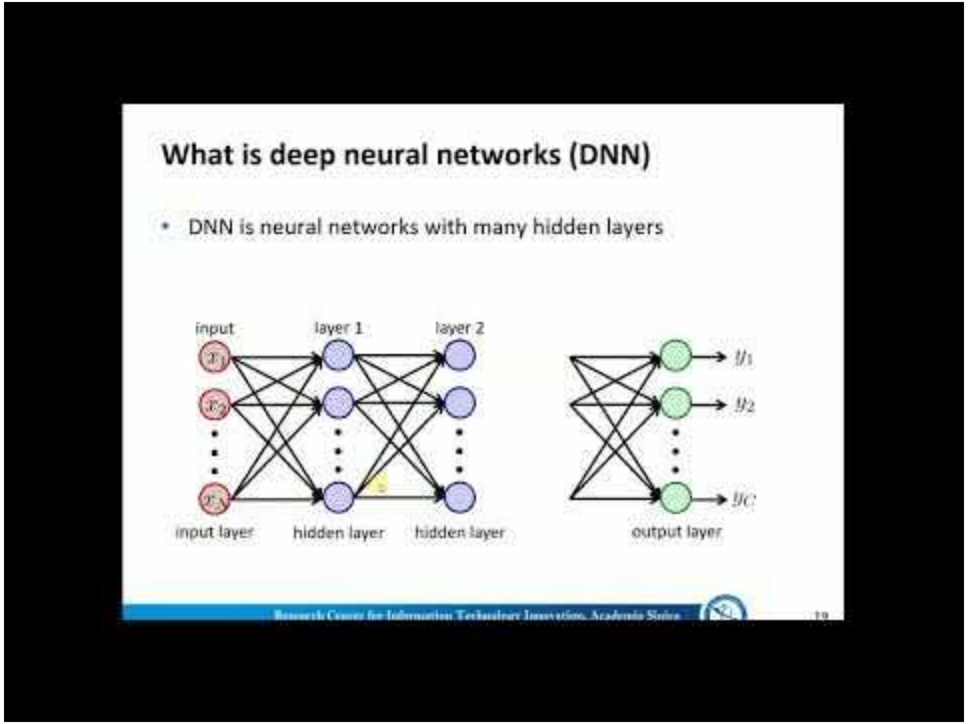
理論講授01 - Self Introduction

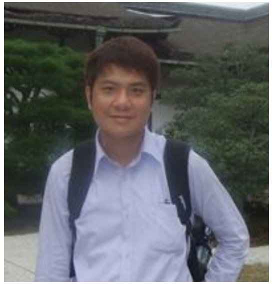


理論講授02 - Conventional Approach v.s Deep Learning

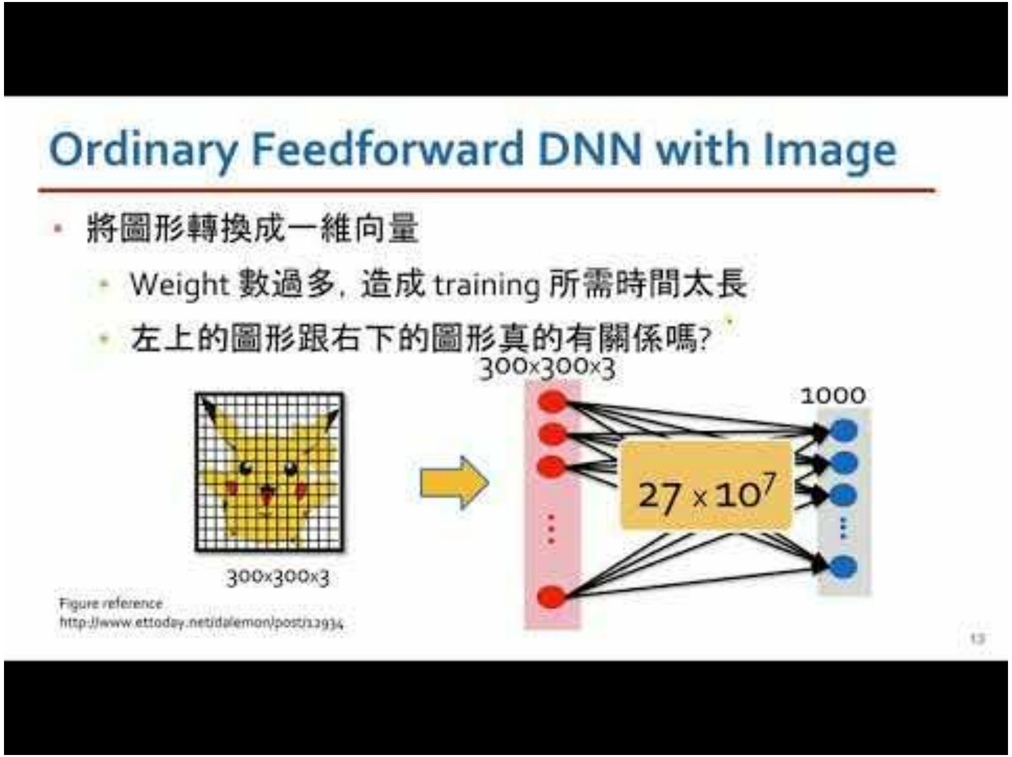


理論講授03 - Neural Network





Before CNN


Ordinary Feedforward DNN with Image
將圖形轉換成一維向量
Weight 數過多, 造成 training 所需時間太長
左上的圖形跟右下的圖形真的有關係嗎?
300x300x3
1000
27 x 10^7
300x300x3
Figure reference
http://www.ettoday.net/dalemon/post/12934

What’s this?

```
decode_predictions(preds, top=3)
```

```
[[('n02111889', 'Samoyed', 0.8701604),
  ('n02120079', 'Arctic_fox', 0.12416725),
  ('n02114548', 'white_wolf', 0.0036498504)]]
```

What's this?

```
decode_predictions(preds, top=3)
```

```
[[('n01531178', 'goldfinch', 0.99720144),
  ('n01537544', 'indigo_bunting', 0.0014747247),
  ('n01530575', 'brambling', 0.00056995713)]]
```

What’s this?

```
decode_predictions(preds, top=3)
```

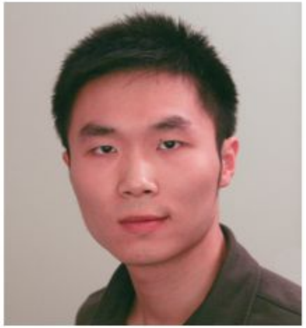
```
[[('n02787622', 'banjo', 0.99902868),
  ('n02676566', 'acoustic_guitar', 0.000544385),
  ('n03272010', 'electric_guitar', 0.0001290191)]]
```

Introduction to IMAGENET

- Large Scale Visual Recognition Challenge (ILSVRC)
- **1000 object classes**
- **1,431,167 images**


Olga Russakovsky
(Stanford U.)


Sean Ma
(Stanford U.)


Jia Deng
(U. of Michigan)


Jonathan Krause
(Stanford U.)


Alexander Berg
(UNC Chapel Hill)


Fei-Fei Li
(Stanford U.)

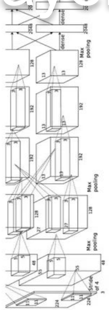
Variety of object classes in ILSVRC

Deep Neural Networks

Human yields 5.1 % error!!

http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf

8 layers



16.4%

AlexNet (2012)

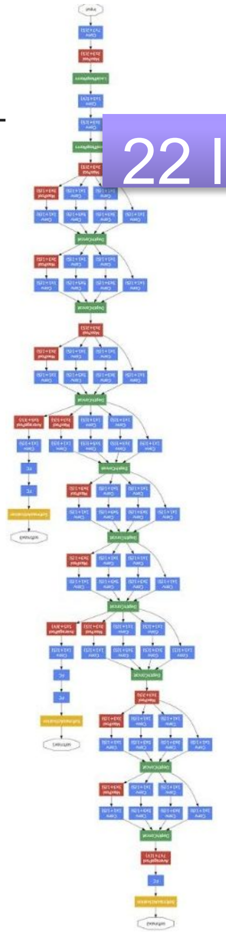
19 layers

7.3%

VGG (2014)

22 layers

6.7%



GoogleNet (2014)

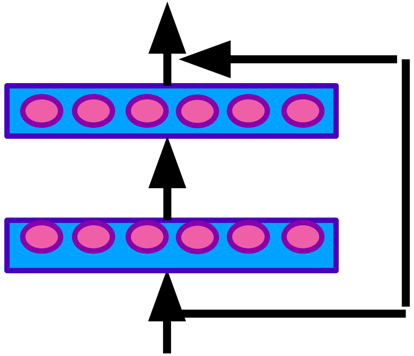
(Slide Credit: Hung-Yi Lee)

(Slide Credit: Hung-Yi Lee)

Deep Neural Networks

Human yields 5.1 % error!!

Special structure



Ref: https://www.youtube.com/watch?v=dxB6299gpvI

16.4% AlexNet (2012)

7.3%  VGG (2014)

6.7%  GoogleNet (2014)

152 layers

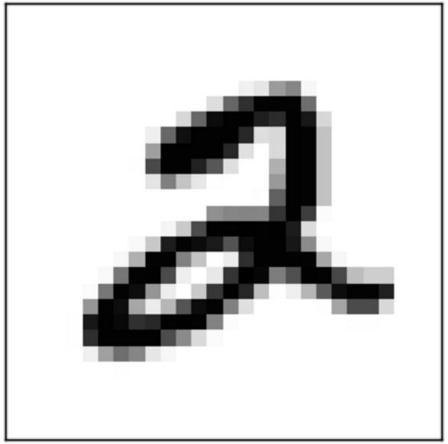
3.57%  Residual Net (2015)

101 layers

Taipei 101

2 Dimensional Inputs

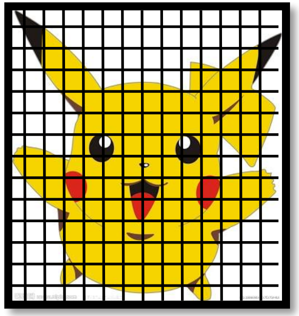
- DNN 的輸入是一維的向量，那二維的矩陣呢? 例如圖形資料



Figures reference
https://twitter.com/gonainlive/status/507563446612013057

Ordinary Feedforward DNN with Image

- 將圖形轉換成一維向量
  - Weight 數過多，造成 training 所需時間太長
  - 左上的圖形跟右下的圖形真的有關係嗎?



300x300x3

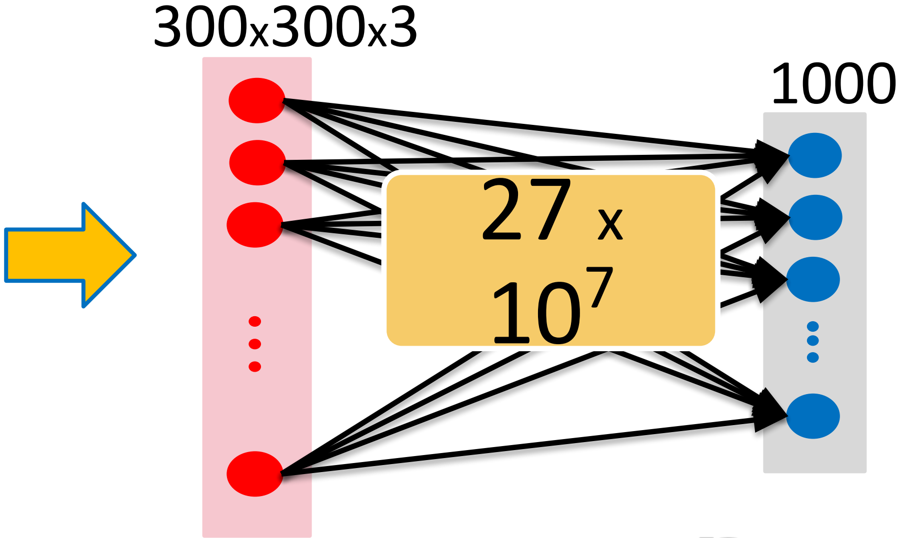


Figure reference
http://www.ettoday.net/dalemon/post/12934

Characteristics of Image

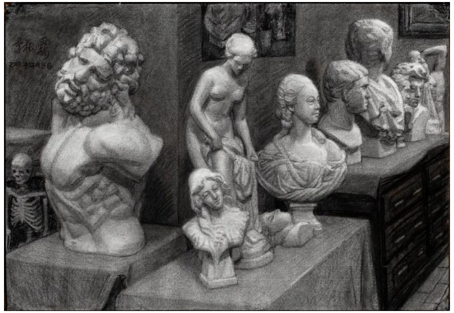
- 圖的構成：線條 → 圖案 (pattern)→物件 →場景

Line Segment

Pattern

Object

Scene



Figures reference
http://www.sumiaozhijia.com/touxiang/471.html
http://122311.com/tag/su-miao/2.html

Patterns

- 猜猜看我是誰
- 辨識一個物件只需要用幾個特定圖案

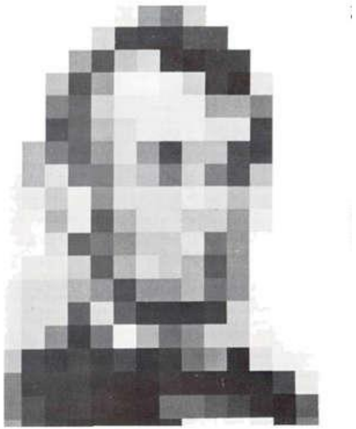
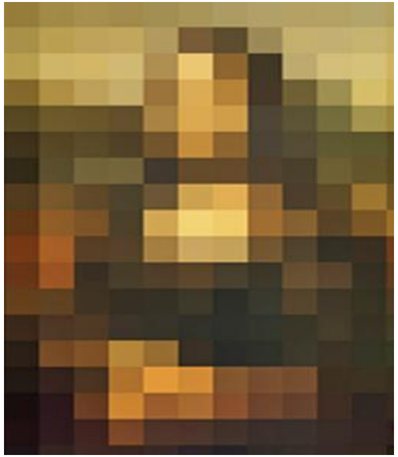
皮卡丘

小火龍

Figures reference
http://arcadesushi.com/charmander-drunken-pokemon-tattoo/#photogallery-1=1

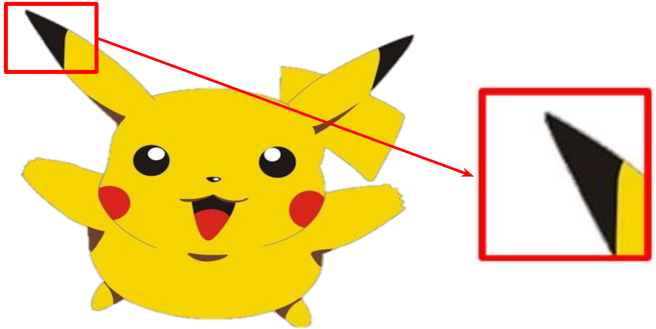
Patterns

- 猜猜看我是誰
- 甚至解析度不需要太高，有輪廓也行!


Property 1: What

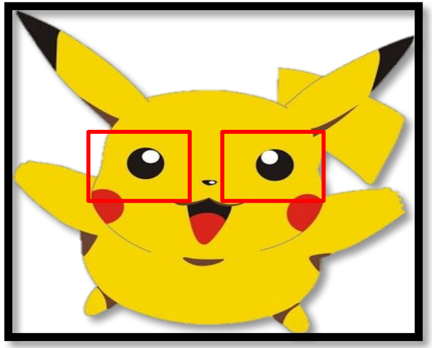
- 圖案的類型



皮卡丘
的耳朵

Property 2: Where

- 重複的圖案可能出現在很多不同的地方



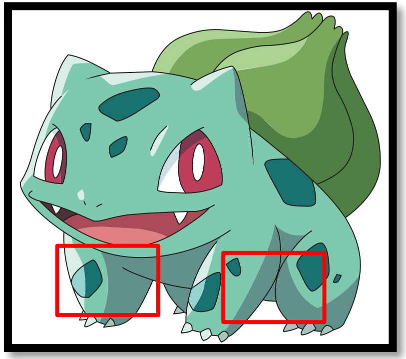
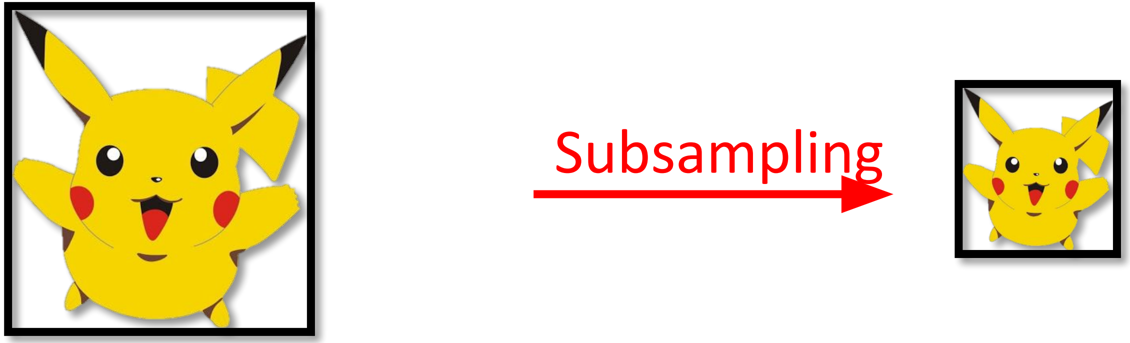


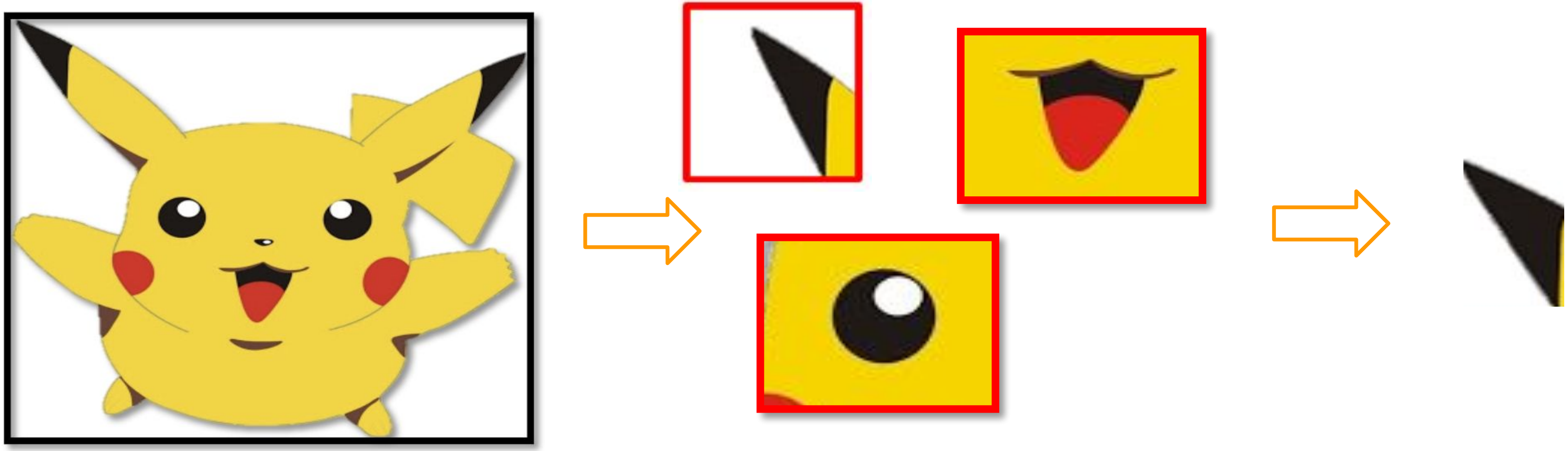
Figure reference
https://www.youtube.com/watch?v=NN9LaU2NlLM

Property 3: Size

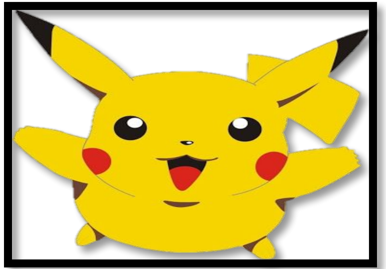
- 大小的變化並沒有太多影響



Why do humans know that this is PICACHU



CNN Structure



Convolutional Layer
&
Activation Layer

New
image

Pooling Layer

New
image

Convolutional Layer
&
Activation Layer

New
image

Pooling Layer

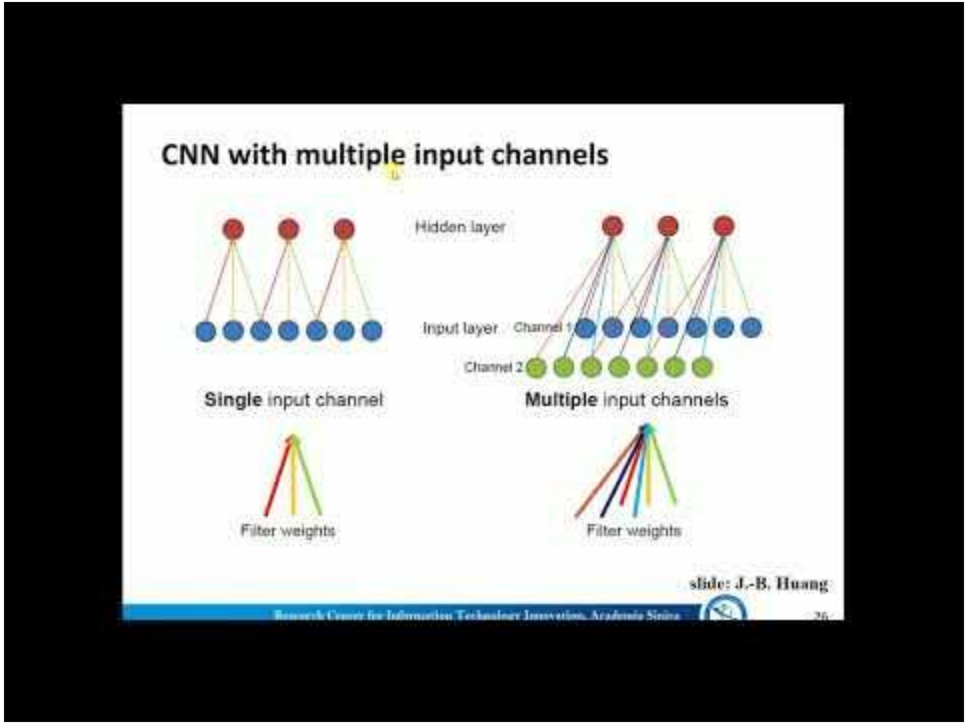
Can be performed
many times

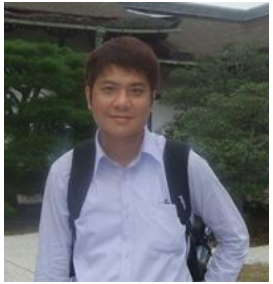
Flatten

Feedforward DNN

PICACHU!!!

理論講授04 - Convolutional Neural Networks





推薦閱讀1

- 聽聽李宏毅老師怎麼說CNN: [影片連結]

推薦閱讀2

- Write CNN from scratch:
  CNN 包含 convolution, pooling, backpropagation... 等操作，Siraj 教你如何用 numpy 手刻 CNN！ [影片連結]

推薦閱讀3

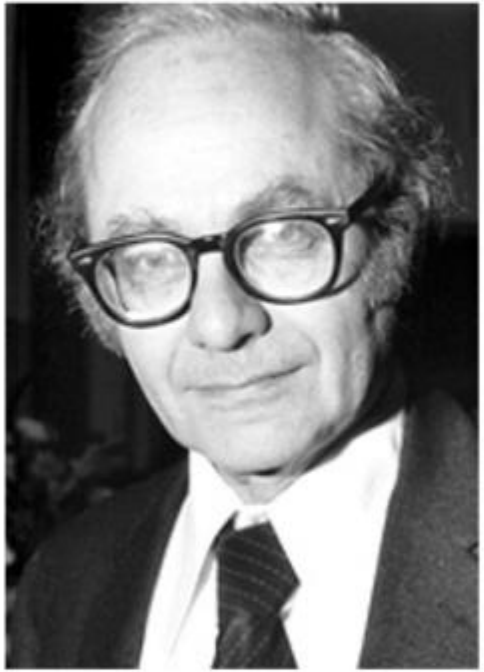
- 強烈推薦 Stanford CS231n 的課程內容，清楚且完整的教學 deep learning 技術與原理，如果覺得助教講的不清楚，來這邊看就對了！
    - 課堂筆記
    - 投影片
    - 課堂錄影
- 給初學者的 CNN 原理

延伸閱讀-1

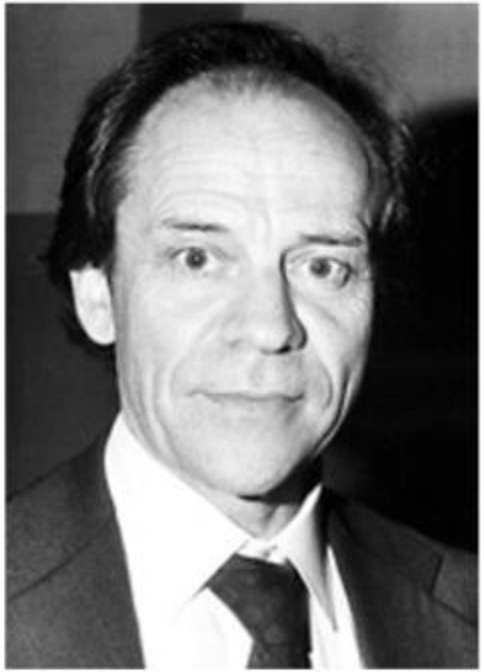
Before LeNet

Hubel & Wiesel

Receptive fields, binocular interaction and functional architecture in the cat's visual cortex
J Physiol. 1962 Jan; 160(1): 106–154.2.

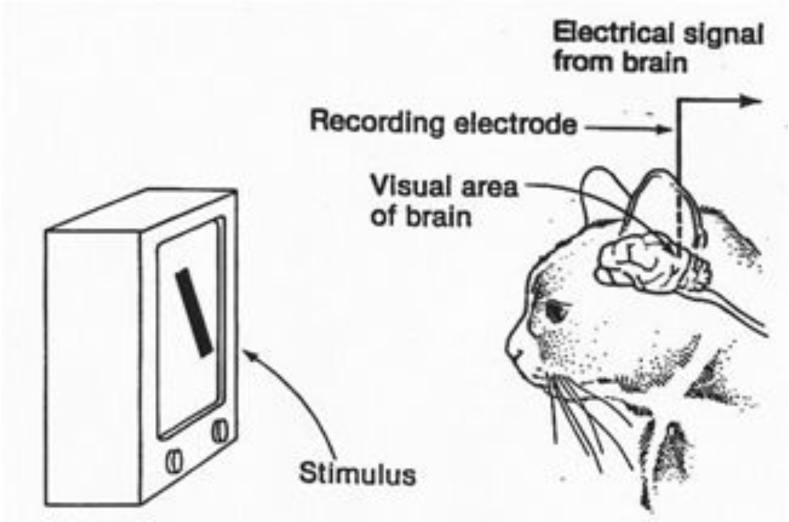


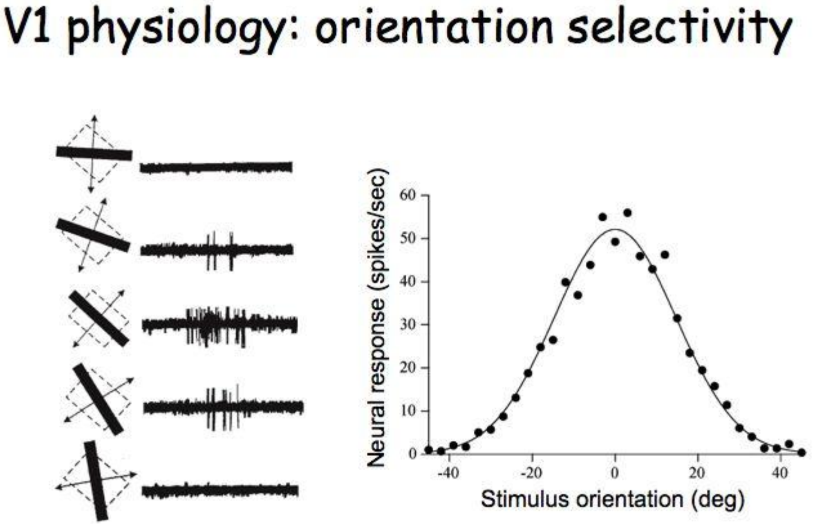
D.H.Hubel



T.N.Wiesel

Visual cortex



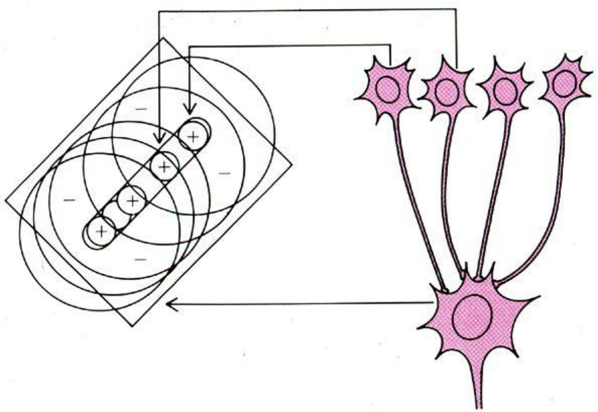


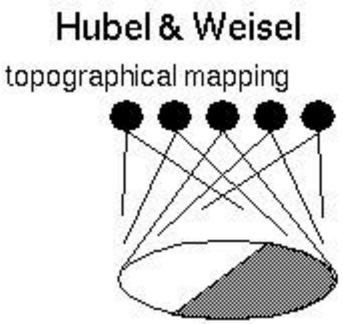
Hubel & Wiesel discovered that cat's visual cortex respond strongly to lines, bars, or edges of a particular orientation (e.g., vertical) but not to the orthogonal orientation (e.g., horizontal).

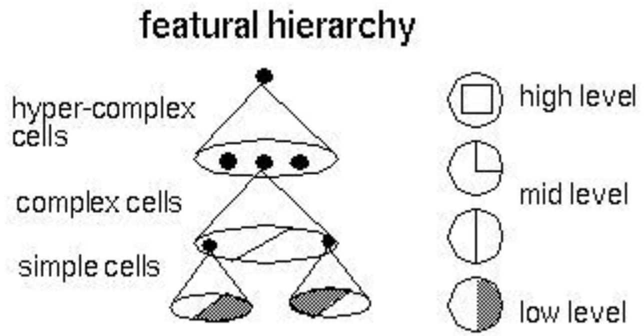
Hubel & Wiesel, 1968

Hierarchy

Simple cell sums LGN inputs







Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

Neocognitron

[Fukushima, Biological Cybernetics 1980](#)

The neocognitron is proposed by Fukushima in 1980, it was inspired by the model proposed by Hubel & Wiesel. They found two types of cells in the visual primary cortex called simple cell and complex cell, and also proposed a cascading model of these two types of cells for use in pattern recognition tasks.

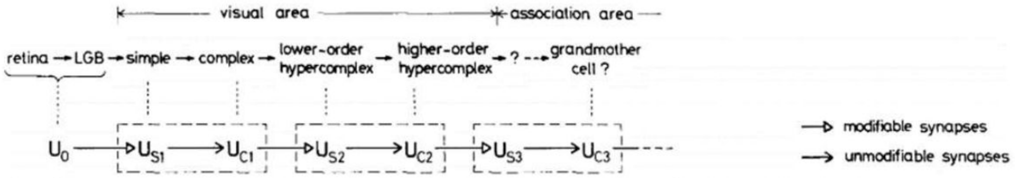
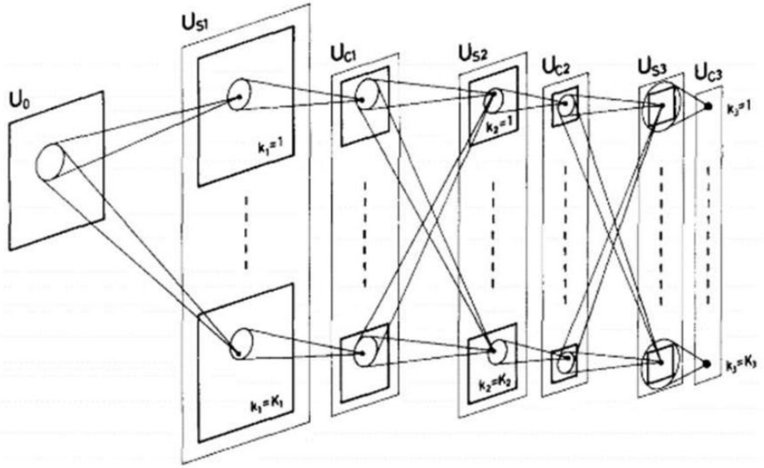
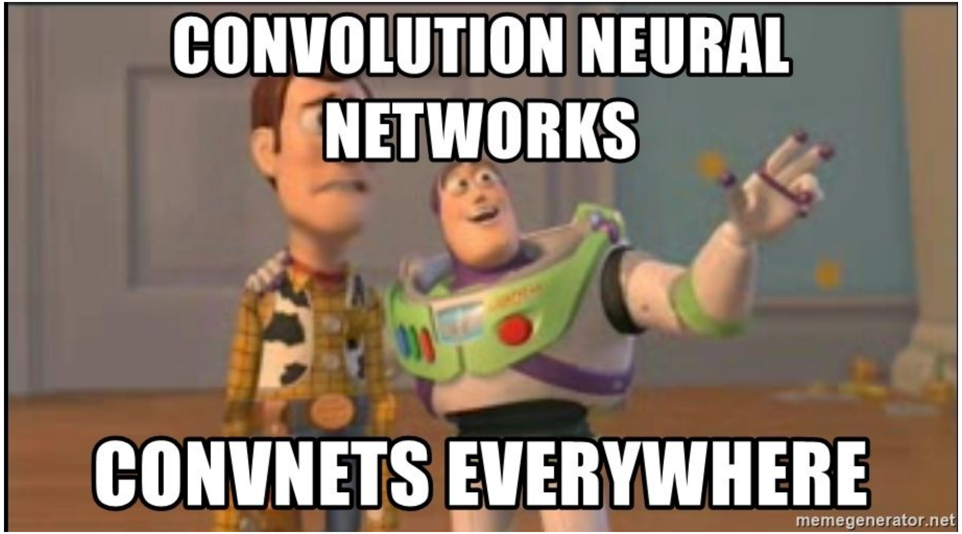


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron



延伸閱讀-2

影像問題，優先考慮 CNN



CNN Applications

Object detection

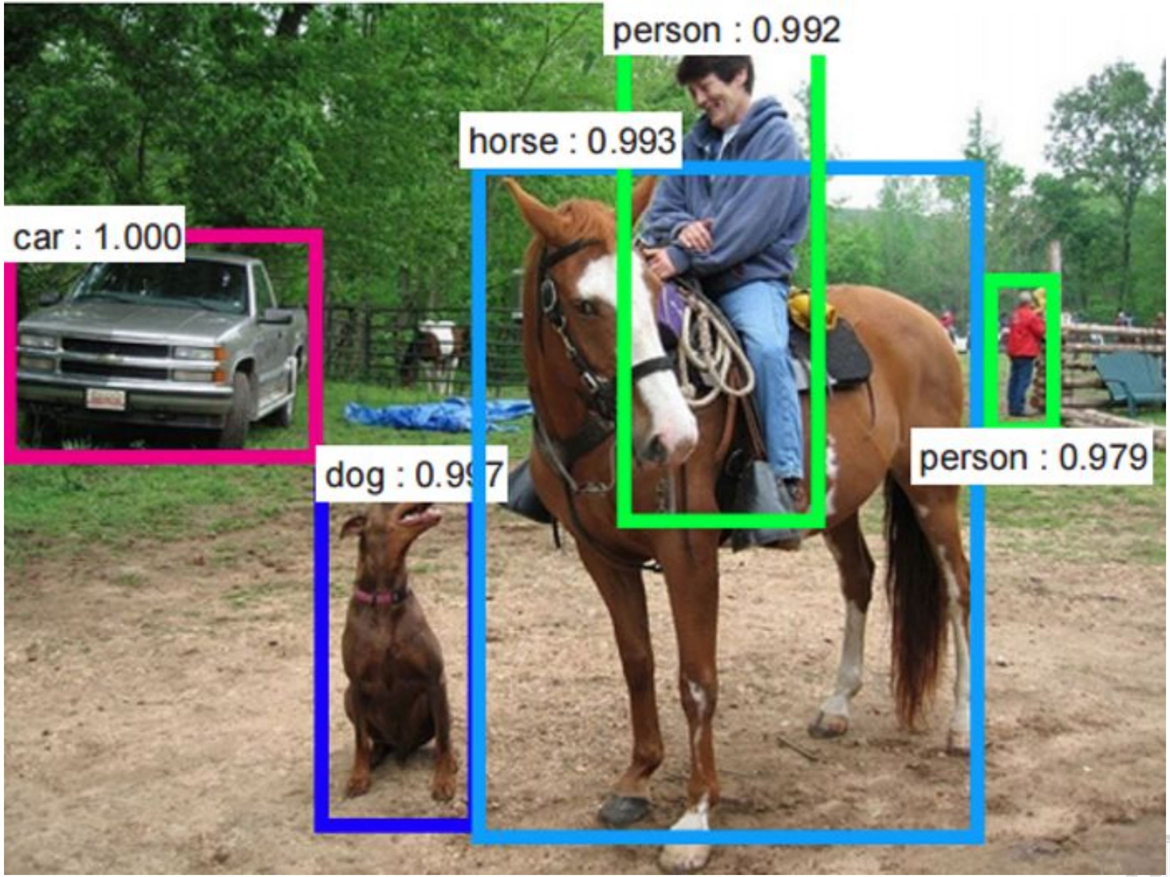
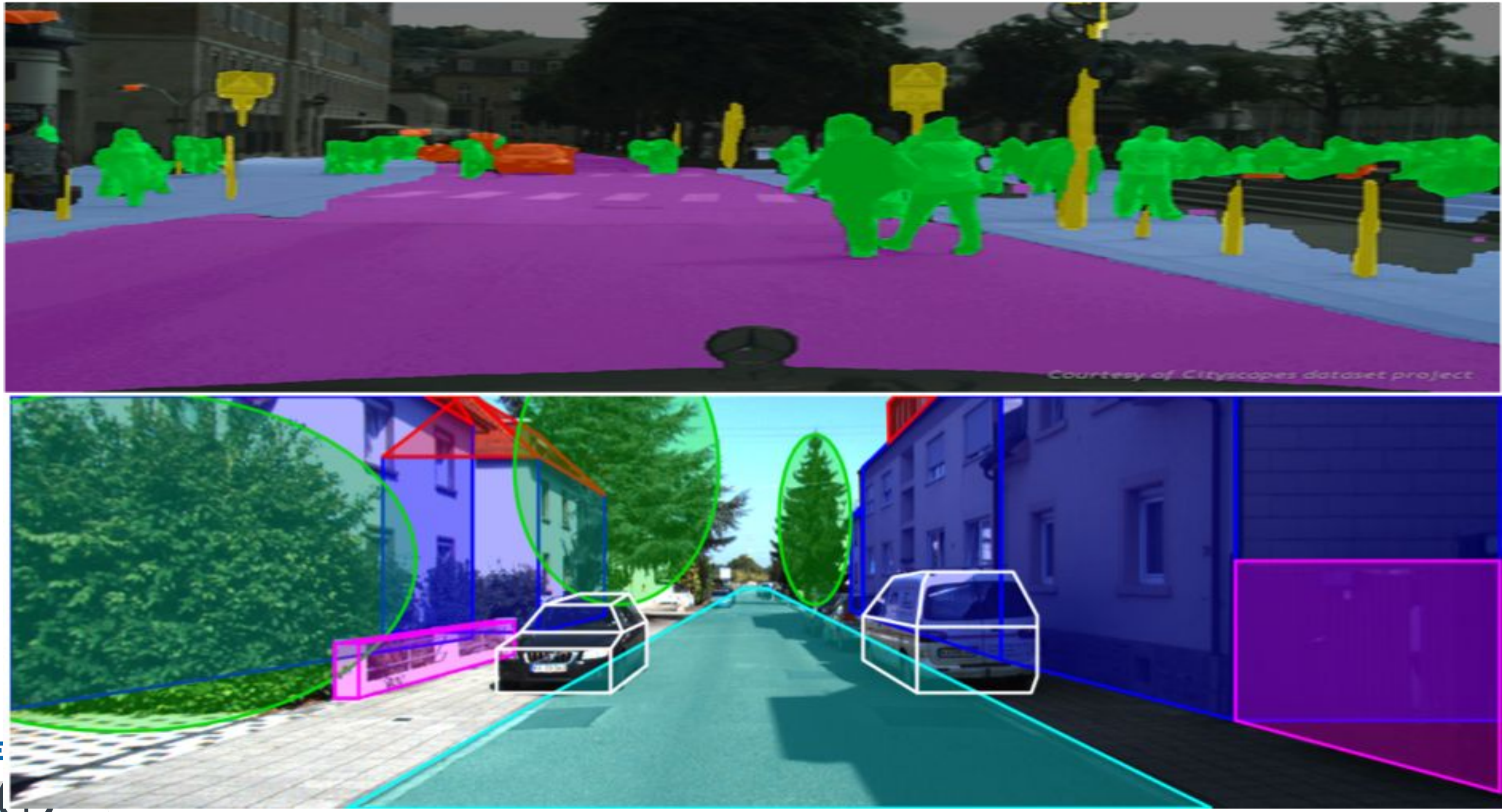


Image segmentation



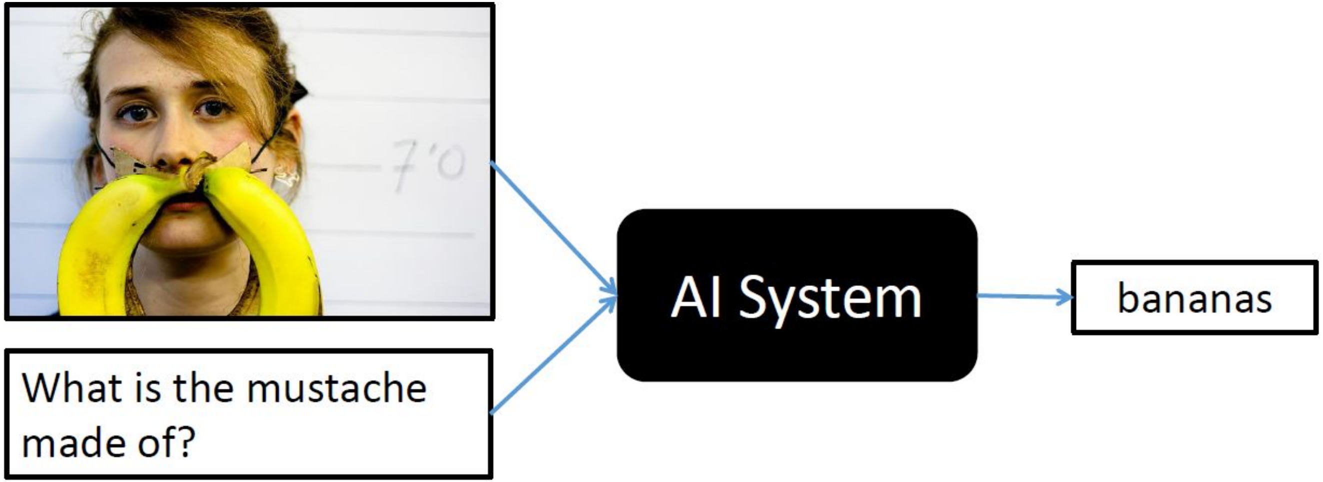
Auto Coloring

https://paintschainer.preferred.tech/index_zh.html
https://zhuanlan.zhihu.com/p/24712438

Colorful Image Colorization

Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." *European Conference on Computer Vision*. Springer Internationa Publishing, 2016.

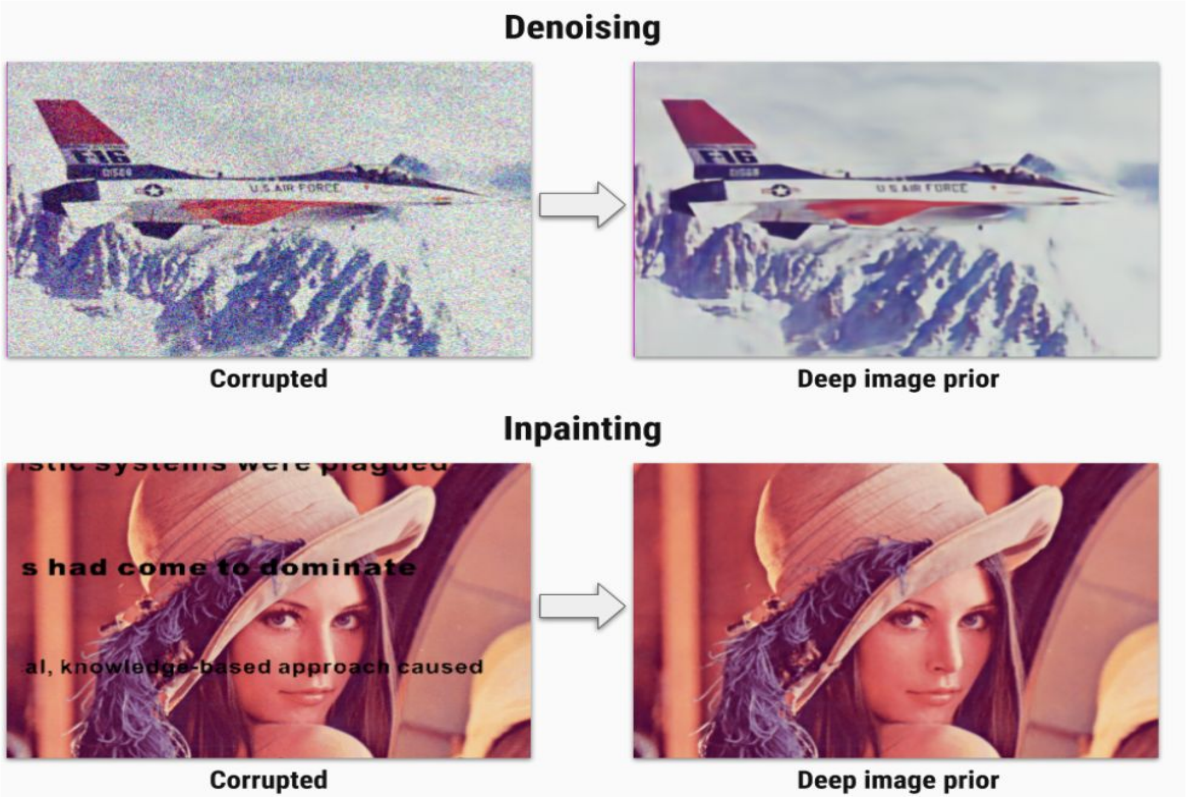
Visual Question Answering



source: http://visualqa.org/

(Slide Credit: Hung-Yi Lee)

Image denoising & inpainting



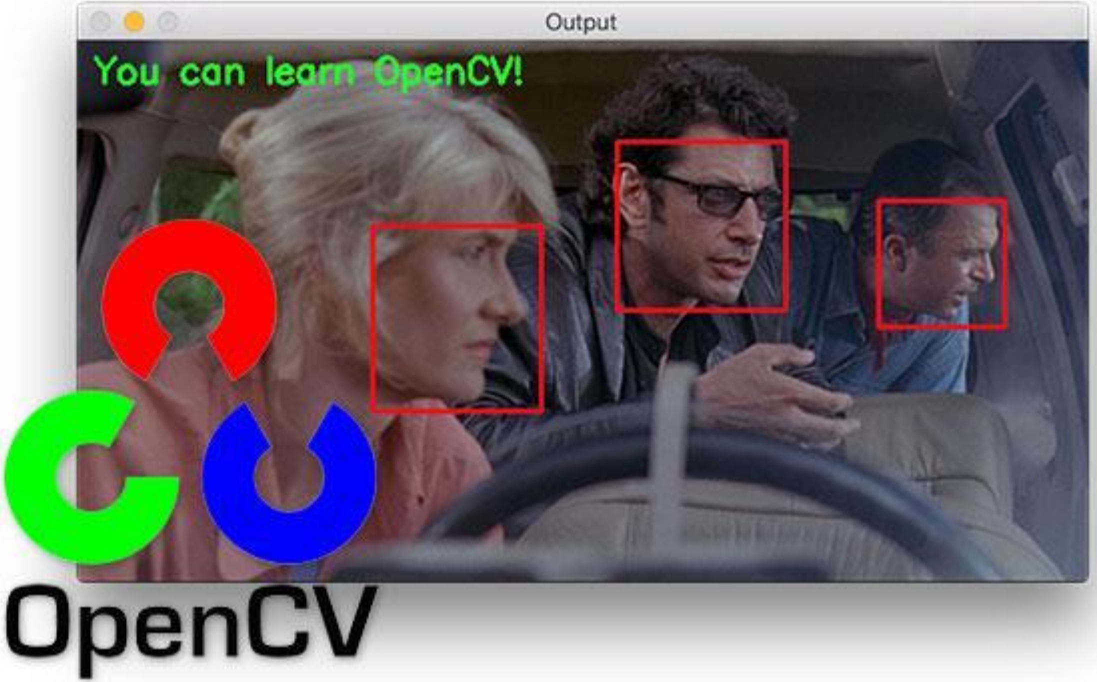
https://dmitryulyanov.github.io/deep_image_prior

延伸閱讀-3

OpenCV

OpenCV

OpenCV



OpenCV

(Open source computer vision)

is a library of programming functions mainly aimed at real-time computer vision.

It supports the deep learning frameworks **TensorFlow**, **Torch/PyTorch** and **Caffe**.

OpenCV

範例Code位置：

courses-tpe/CVCNN/part1/00_computer_vision