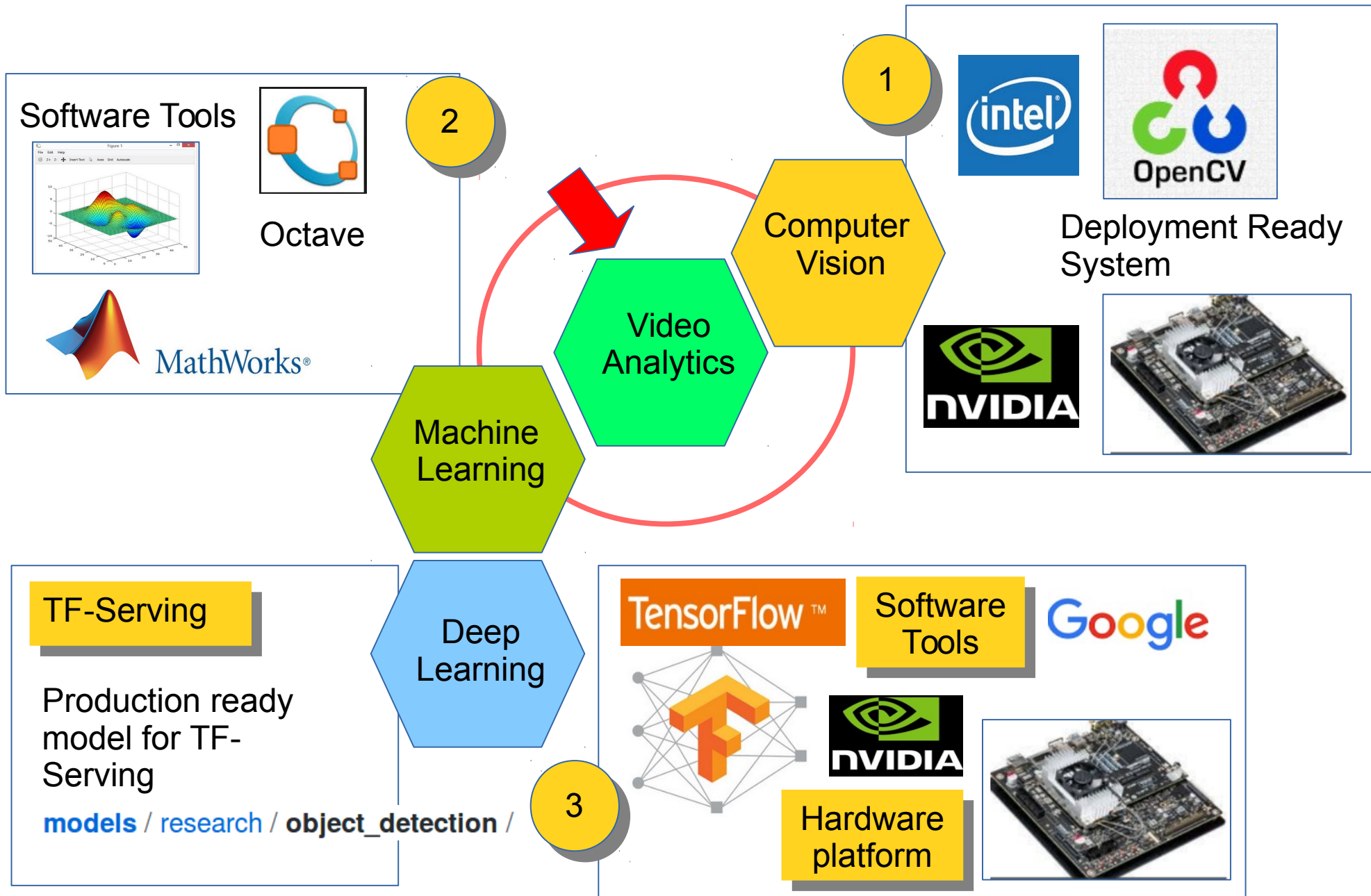


The Scope





Objective Is To Build Video Search Engine



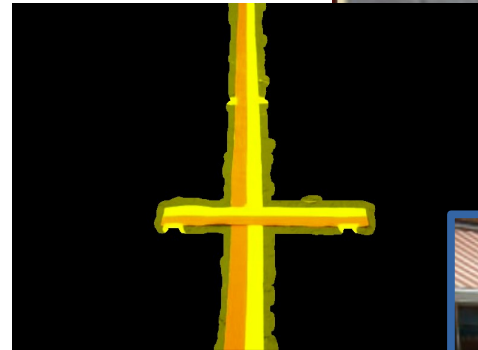
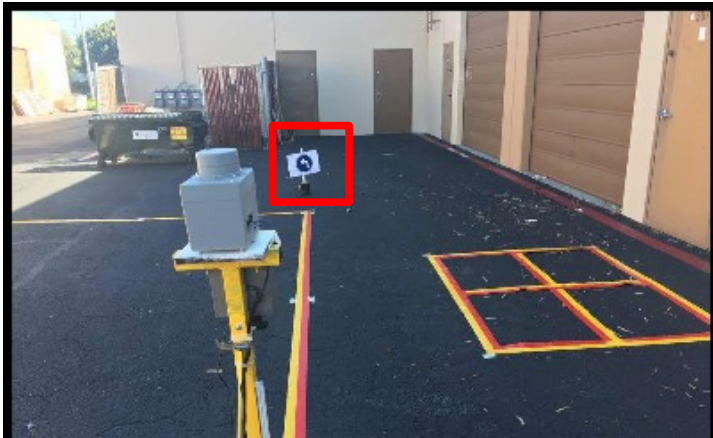
1



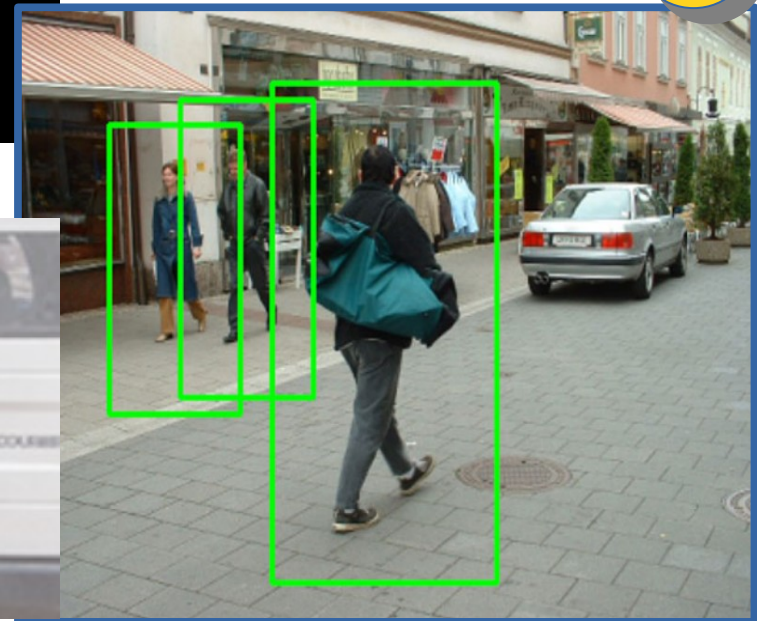
2



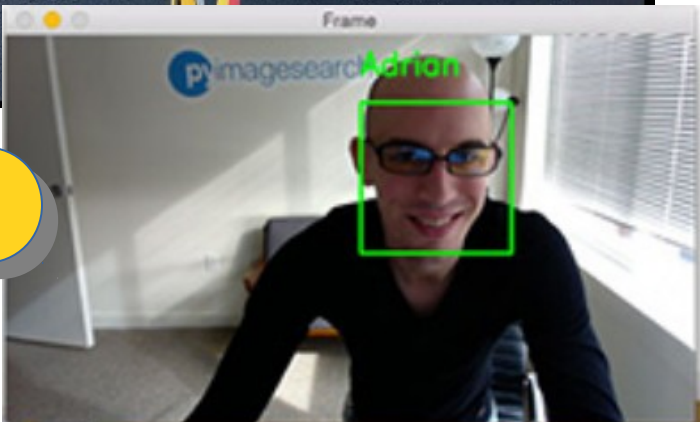
3



5



4





To Prepare For Interview (1)

OpenCV 2.1 Cheat Sheet (C++)

The OpenCV C++ reference manual is here:

<http://opencv.willowgarage.com/documentation/cpp/>.

Use **Quick Search** to find descriptions of the particular functions and classes

Image Processsing

Filtering

`filter2D()`

Non-separable linear filter

`sepFilter2D()`

Separable linear filter

`boxFilter()`,

Smooth the image with one of the linear or non-linear filters

`GaussianBlur()`,

`medianBlur()`,

`bilateralFilter()`

`Sobel()`, `Scharr()`

Compute the spatial image derivatives

`Laplacian()`

compute Laplacian: $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$

`erode()`, `dilate()`

Erode or dilate the image

Key OpenCV Classes

`Point_`

Template 2D point class

`Point3_`

Template 3D point class

`Size_`

Template size (width, height) class

`Vec`

Template short vector class

`Scalar`

4-element vector

`Rect`

Rectangle

`Range`

Integer value range

`Mat`

2D dense array (used as both a matrix or an image)

`MatND`

Multi-dimensional dense array

`SparseMat`

Multi-dimensional sparse array

`Ptr`

Template smart pointer class

Set Up OpenCV

http://docs.opencv.org/2.4/doc/tutorials/introduction/table_of_content_introduction/table_of_content_introduction.html

How to set up openCV

http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation



Title: *Installation in Linux*

Compatibility: > OpenCV 2.0

Author: Ana Huamán

We will learn how to setup OpenCV in your computer!

How to compile and build

http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_gcc_cmake/linux_gcc_cmake.html#linux-gcc-usage



Title: *Using OpenCV with gcc and CMake*

Compatibility: > OpenCV 2.0

Author: Ana Huamán

We will learn how to compile your first project

Using Eclipse

http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_eclipse/linux_eclipse.html#linux-eclipse-usage



Title: *Using OpenCV with Eclipse (plugin CDT)*

Compatibility: > OpenCV 2.0

Author: Ana Huamán

Optional but better

Introduction 6-18-18

<https://github.com/hualili/opencv/tree/master/IP110-Summer18>

IP110 Computer Vision (I)
Jun 20, 2018. 1/.

I. Introduction.

- 1) Theory (Reference: ① MIT. Robot Vision. 1982.
- ② Learning OpenCV. 3rd Edition. pdf Version.) → C/C++ Examples. ✓
- OpenCV. Toolkit (2.1 Version) C++
- Cheat Sheet (pdf) 32
③ Tx2 Board / Laptop Computer
Deployment Purpose.
- ④ Schedule: Wednesday 4:00-7:00 PM (Last Hour for Hands-on Show & Tell, presentation)
Friday 4:00-7:00 PM. Machine + A.I.

2) Background on Computer Vision. → Learning
Biological Perspectives: 60% Neurons. "Early Vision"

~1 million Abstract Form Cognitive 50%
~10 million Photo Receptors.
50% "Early" Retina
Color / Intensity / Derivatives of Changes of the Intensity
~10 Layers
1. Rods.
2. Cone
3. Bipolar Cells
2D Convolution → Filtering
Pyramid Images.
USB CAM
Original Image $I(x,y)$
Electronic Beam
~50 yrs. B/W T.V. CRT.
Electronic Gun
N x N
(1024 x 1024) (1024 x 768) pixel (picture element)
1° Find Resolution $I(x,y)$ Col. Rows → $i=0, 1,$

Example 1:
Example 2:

1. how and why is the image coordinate defined in such as a way? 2. CRT display device? 3. What is pyramid image? Calculation of pyramid resolution?

Pyramid And Convolution 6-18-18

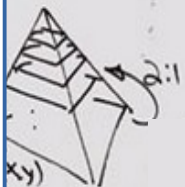
$I_1(x,y) : 512 \times 512 \cdot (2^9 \times 2^9)$
 $I_2(x,y) : 256 \times 256 \cdot (2^8 \times 2^8)$
 2^0 Mapping:

$I_{ij}^{(1)} = \frac{1}{4} (I_{ij} + I_{i+1,j} + I_{i,j+1} + I_{i+1,j+1})$... (1)
 → Low Pass Filter. * "Average" ops
 "Blur"
 High Freq. Comp. → Removed.
 Sharp Boundaries (Edges)

3^0 $I_{ij}^{(3)}$ Significantly Reduced Resolution.
 1024×1024
 Find Total Number of Bits per Second to process Video Streams.
 30 F.P.S.
 $CV-8UC3$

$I(x,y) \in [0, 255]$
 $2^8 = 256$
 $1024 \times 1024 \times 30 \times 24$
 $= \frac{2^{10} \cdot 2^{10} \cdot 2^5 \cdot 2^5}{1K}$
 $= 2^{10} \cdot 2^{10} \cdot 2^{10}$
 $2^4 = 16$
 $2^5 = 32$
 If more up 2 levels in Pyramid, then Resolution is Reduced from 1 Gbps to $2^{30}/16 = 2^{30}/2^4 = 2^{26} = 2^6 \cdot 2^{20}$
 1982 David Marr
 "Vision"
 \log , Laplace of Gaussian.

* Filtering Operations.
 To Extract Features. Edge(s)
 Removal of Random Noise.
 2D Convolution.
 $g(x) \rightarrow P \rightarrow R(x) \rightarrow g(x) * h(x) \dots (2)$
 $h(x)$, plant(System)
 $g(x) * h(x) = \int_{-\infty}^{+\infty} g(\tau) h(x-\tau) d\tau = \int_{-\infty}^{+\infty} h(\tau) g(x-\tau) d\tau \dots (2*)$
 1) Linear System: If $x(t) \rightarrow$ then $y(x)$
 $\sum_{n=-\infty}^{+\infty} h(n) g(k-n)$ then, $ax_1(t) + bx_2(t) \rightarrow y_1(t) + y_2(t) \mid y_1(t) = ax_1(t) \mid y_2(t) = bx_2(t)$
 $h(k) * g(k)$
 $I(x,y) \cdot K(x,y) = \iint_{-\infty}^{+\infty} I(u,v) K(x-u, y-v) du dv$... (3)
 Kernel
 $\sum_{v \in \mathbb{R}_+} \sum_{u \in \mathbb{R}_+} I(u,v) K(x-u, y-v) \dots (3*)$
 Difference of Gaussian

Filtering


4. How to form pyramid image? And why is it LPF (low pass filter)? 5. Convolution definition?

LoG(x,y; mu, sigma)
Laplace of Gaussian

DoG(x,y; mu1, mu2, sigma1, sigma2)
Difference of Gaussian

Convolution Example And Kernel Concept 6-18-18

IP110 Computer Vision (I)
Jun 20, 2018. 2/.

Diagram illustrating the convolution process:

- An input image $I(x,y)$ of size $M \times N$ is convolved with a kernel $K(x,y)$ of size $K \times K$.
- The result is a feature map of size $(M-2) \times (N-2)$.
- The process involves multiplication ($*$) and summation.

Handwritten notes and examples:

Example 3: Christoff Koch + Hualu "padding"

Input image $I(x,y)$ (4x4):

0	0	0	0
100	100	100	0
0	100	0	0
0	100	0	0

Kernel $K(x,y)$ (3x3):

-1	1	1
1	1	1
1	1	1

Next, calculate the convolution result:

Calculation 1:

$$-1 \times 0 + 0 \times 0 + 1 \times 0 = 0$$

$$-1 \times 0 + 0 \times 0 + 1 \times 0 = 0$$

$$-1 \times 0 + 0 \times 100 + 1 \times 100 = 100$$

Calculation 2:

$$-1 \times 0 + 0 \times 0 + 1 \times 0 = 0$$

$$-1 \times 0 + 0 \times 0 + 1 \times 0 = 0$$

$$-1 \times 100 + 0 \times 100 + 1 \times 100 = 0$$

Result: 0, 0, 100, 0, 0, 0, 0, 0, 0

Derivation of Kernels:

Derivatives:

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

$$= \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

Homework 1:

- 1° 5-8 Second Video Clips.
- 2° Display Video
- 3° Para (1/2), disp.
- 4° Recover Original Image
- 5° for K times

Forward Difference:

Calculation:

$$= f(x+1) - f(x)$$

$$= 1 \times f(x+1) + (-1) \times f(x)$$

$$= K(x+1)f(x+1) + K(x)f(x)$$