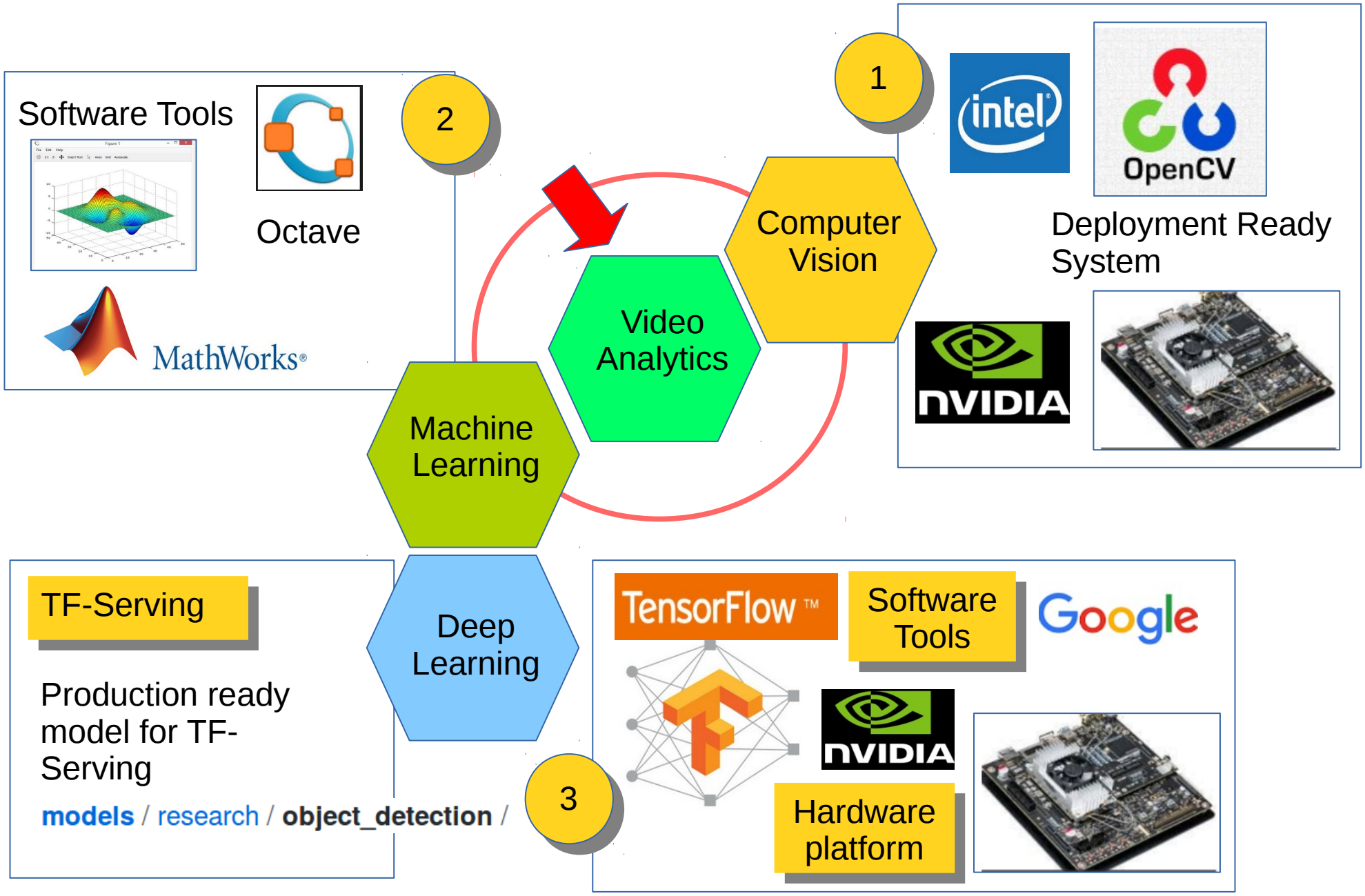
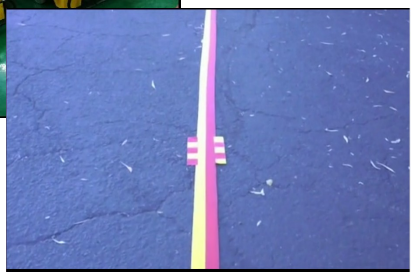
The Scope

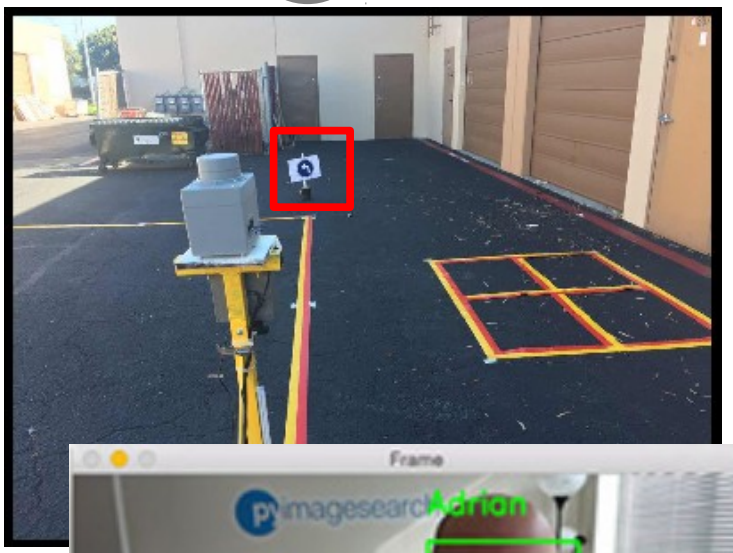

Software Tools
Octave
MathWorks
2
1
intel
OpenCV
Deployment Ready System
Computer Vision
Video Analytics
Machine Learning
NVIDIA
Deep Learning
TF-Serving
Production ready model for TF-Serving
models / research / object_detection /
3
TensorFlow ™
Software Tools
Google
NVIDIA
Hardware platform

Computer Vision Techniques and Deep Learning For Video Search Engine

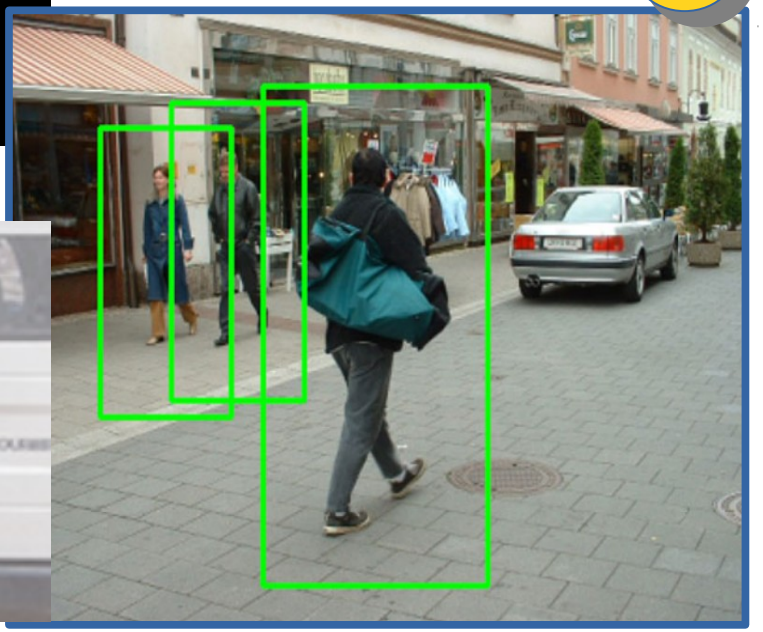
1

2

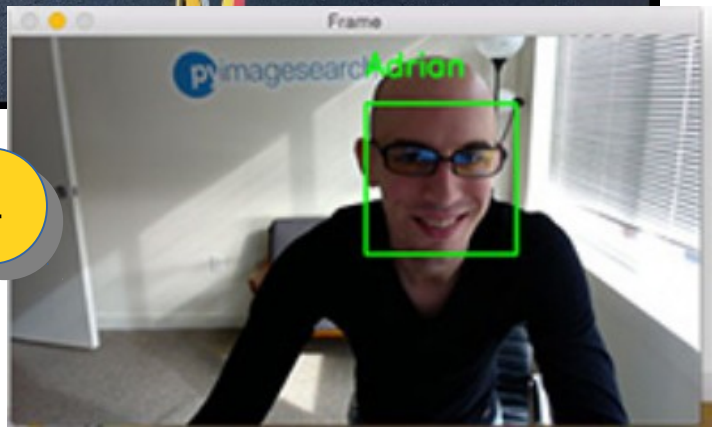




3



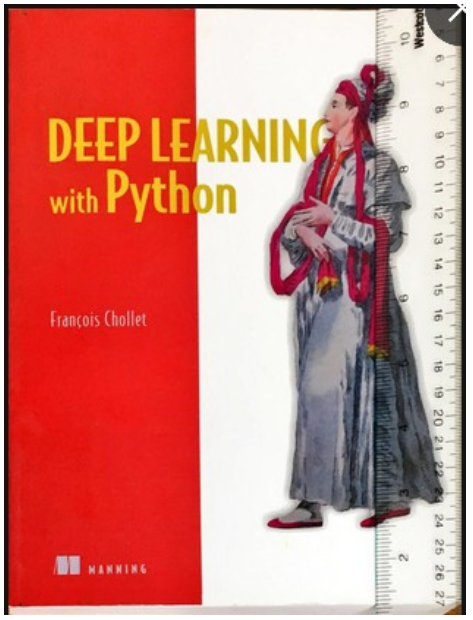
4

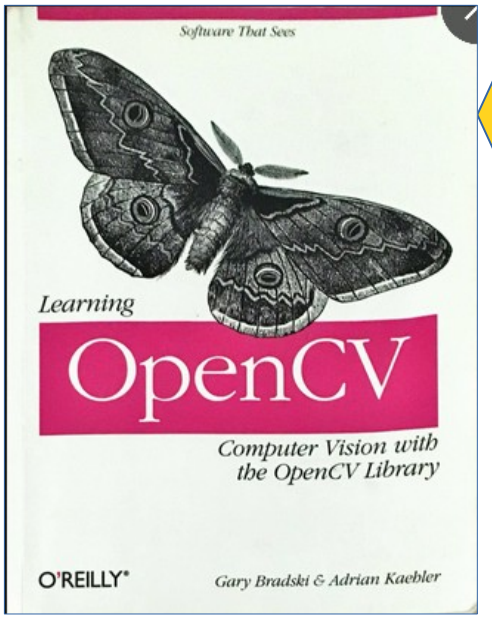


5

Harry Li, Ph.D.

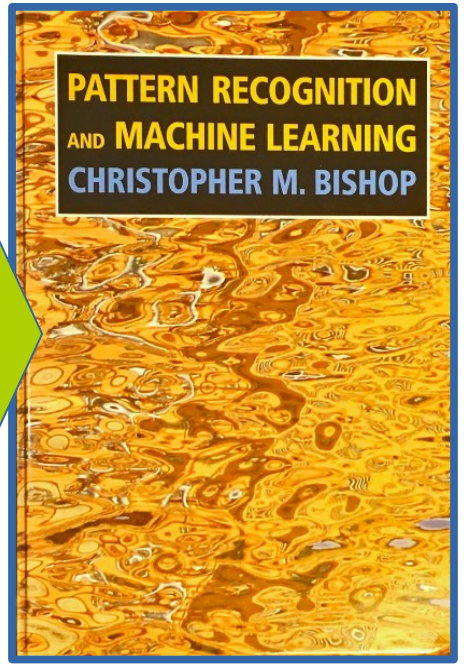
Reference For This Class

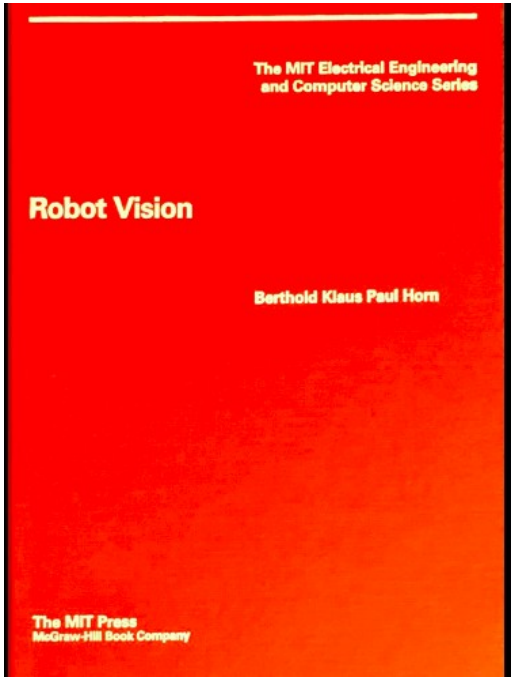




Computer Vision

Machine Learning





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

Tensor Flow Installation on Ubuntu 14.04 (1)

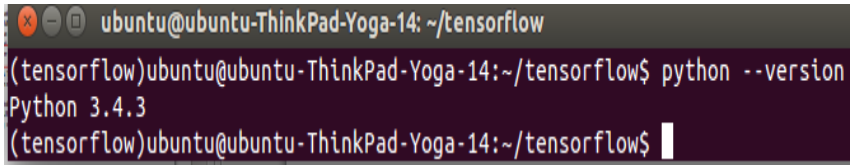
https://www.tensorflow.org/tutorials/

After all the prerequisite, now follow the tensor flow installation recommendation, go with virtualenv installation for isolated python environment.5 Steps from the tensor flow tutorial:
Step 1: sudo apt-get install python3-pip python3-dev python-virtualenv

Step 2: virtualenv --system-site-packages -p python3 ~/tensorflow

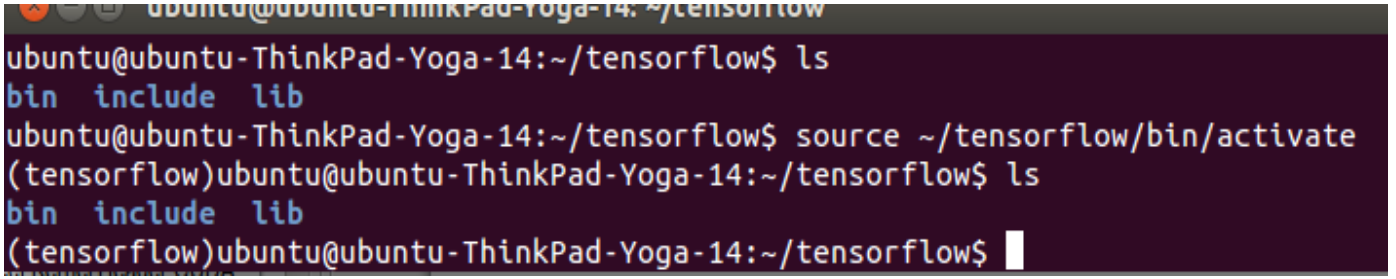
Step 3: $source ~/tensorflow/bin/activate # bash, sh, ksh, or zsh (note, you can use the following for your choice:
$ source ~/tensorflow/bin/activate.csh # csh or tcsh) if it is successful, then The preceding source command should change your prompt to the following:
(tensorflow)$

Note: check your python version
$python --version



Step 4: pip3 install --upgrade tensorflow-gpu
But had error message of no download was found, so upgrade pip and tensorflow as follows, then install again, the error was gone.

pip install --upgrade pip
pip install --upgrade tensorflow



Step 5. Just Download cuDNN 5.1 click Here and follow the steps (Tested on Ubuntu 16.04, CUDA toolkit 8.0)

Note: see cuDNN installation slide.

I have had an error after installation of Tensorflow, “ImportError: libcudnn.Version: cannot open shared object file: No such file or director“, after reinstall cuDNN, the error is gone.

Tensor Flow Installation on Ubuntu 14.04(2)

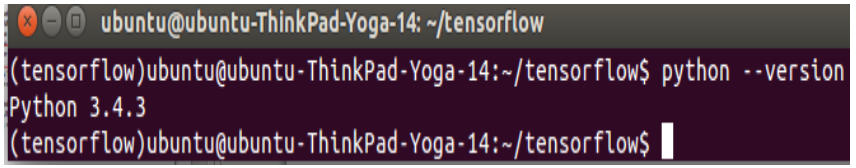
https://www.tensorflow.org/tutorials/

After all the prerequisite, now follow the tensor flow installation recommendation, go with virtualenv installation for isolated python environment.5 Steps from the tensor flow tutorial:
Step 1: sudo apt-get install python3-pip python3-dev python-virtualenv

Step 2: virtualenv --system-site-packages -p python3 ~/tensorflow

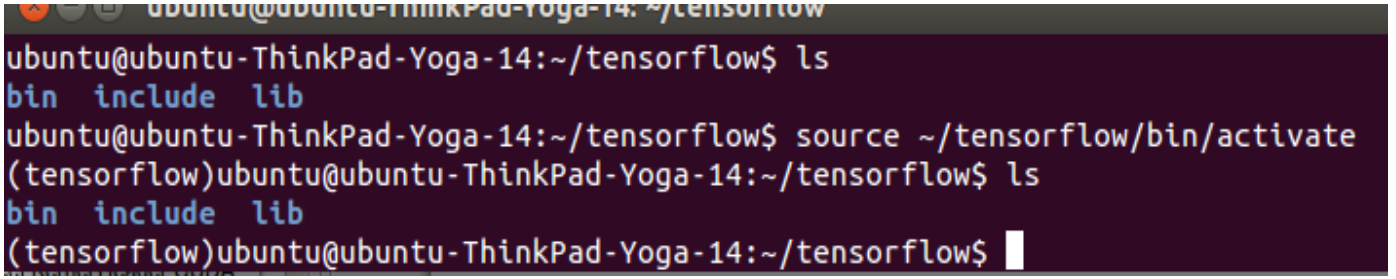
Step 3: $source ~/tensorflow/bin/activate # bash, sh, ksh, or zsh (note, you can use the following for your choice:
$ source ~/tensorflow/bin/activate.csh # csh or tcsh) if it is successful, then The preceding source command should change your prompt to the following:
(tensorflow)$

Note: check your python version
$python --version



Step 4: pip3 install --upgrade tensorflow-gpu
But had error message of no download was found, so upgrade pip and tensorflow as follows, then install again, the error was gone.

pip install --upgrade pip
pip install --upgrade tensorflow



Step 5. Just Download cuDNN 5.1 click Here and follow the steps (Tested on Ubuntu 16.04, CUDA toolkit 8.0)

$ tar xvzf cudnn-8.0-linux-x64-v5.1-ga.tgz
$ sudo cp -P cuda/include/cudnn.h /usr/local/cuda/include
$ sudo cp -P cuda/lib64/libcudnn* /usr/local/cuda/lib64
$ sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*

First OpenCV Sample Program (1)

Bitwise.cpp

```
#include <iostream>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

using namespace cv;
using namespace std;

int main(  )
{
    Mat src1, src2, dst;
    cout<<" Bitwise Operations "<<endl;

    src1 = imread("1image.jpg");
    src2 = imread("2image.jpg");

    namedWindow("AND", CV_WINDOW_NORMAL);
    namedWindow("OR",  CV_WINDOW_NORMAL);
    namedWindow("XOR", CV_WINDOW_NORMAL);
    namedWindow("NOT", CV_WINDOW_NORMAL);

    Mat res;
    bitwise_and(src1,src2,res);     imshow("AND",res);
    bitwise_or(src1,src2,res);      imshow("OR",res);
    bitwise_xor(src1,src2,res);     imshow("XOR",res);
    bitwise_not(src1,res);          imshow("NOT",res);

    waitKey(0);
    return(0);
}
```

First OpenCV Sample Program (2)

VideoContourSegm.cpp

```
int main(int argc, char** argv)
{
    VideoCapture cap;
    bool update_bg_model = true;

    if (input.empty())
        cap.open(0);
    else
        cap.open(input);
    if( !cap.isOpened() )
    {
        printf("\nCan not open camera or video file\n");
        return -1;
    }
    Mat tmp_frame, bgmask, out_frame;
    cap >> tmp_frame;
    if(tmp_frame.empty())
    {
        printf("can not read data from the video source\n");
        return -1;
    }
    namedWindow("video", 1);
    namedWindow("segmented", 1);
    Ptr<BackgroundSubtractorMOG2> bgsubtractor=createBackgroundSubtractorMOG2();
    bgsubtractor->setVarThreshold(10);
```

OpenCV Classes and Function Modules (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(), GaussianBlur(), medianBlur(), bilateralFilter()	Smooth the image with one of the linear or non-linear filters
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

Accessing Pixels

https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html

Example: load image

```
import cv2
import numpy as np
img = cv2.imread('messi5.jpg')
```

Example: reading pixel

```
>>> px = img[100,100]
>>> print px
[157 166 200]

# accessing only blue pixel
>>> blue = img[100,100,0]
>>> print blue
157
```

Example: writing pixel

```
>>> img[100,100] = [255,255,255]
>>> print img[100,100]
[255 255 255]
```

Better pixel accessing and editing method :

```
# accessing RED value
>>> img.item(10,10,2)
59
# modifying RED value
>>> img.itemset((10,10,2),100)
>>> img.item(10,10,2)
100
```

```
>>> print img.shape
(342, 548, 3)
>>> print img.size
562248
>>> print img.dtype
uint8
```

Accessing ROI

https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html

```
>>> ball = img[280:340, 330:390]
>>> img[273:333, 100:160] = ball
```

```
>>> b,g,r = cv2.split(img)
>>> img = cv2.merge((b,g,r))
```

cv2.split() is a costly operation (in terms of time). So do it only if you need it. Otherwise go for Numpy indexing.

```
>>> b = img[:,:,0]
```

To make all the red pixels to zero, simply use Numpy indexing and more faster.

```
>>> img[:,:,2] = 0
```

C++ Accessing ROI

Read ROI

```
#include <stdio.h>
#include <opencv2/opencv.hpp>
using namespace cv;
int main(int argc, char** argv ) {
    if ( argc != 2 )
    {
        printf("usage: DisplayImage.out <Image_Path>\n");
        return -1;
    }
    Mat image;
    image = imread( argv[1], 1 );
    if ( !image.data )
    {
        printf("No image data \n");
        return -1;
    }
    int x=20, y=20, w=40, h=60;
    Rect region_of_interest = Rect(x, y, w, h);
    Mat image_roi = image(region_of_interest);
    imshow("Image", image);
    imshow("ROI", image_roi);
    waitKey(0);
    return 0;
}
```