

计算机视觉班

从CV基础到深度学习实战

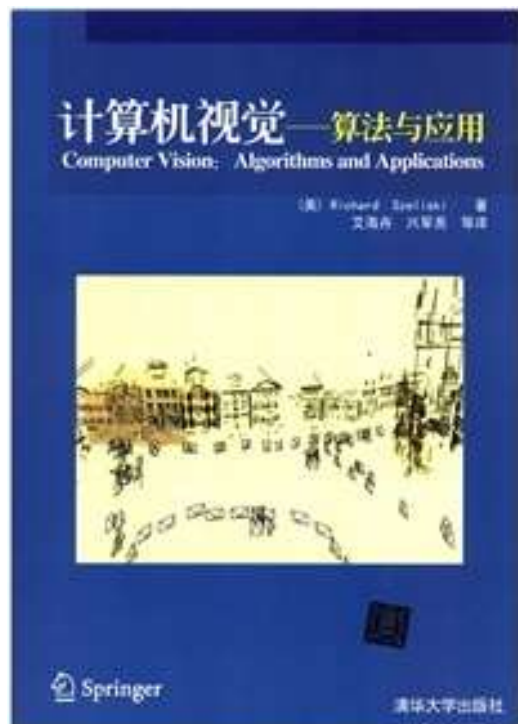
Syllabus

- 图像处理基础
- 图像的特征提取
- 机器视觉中的几何学：
 - 坐标变换与视觉测量
 - 3D计算机视觉
- 机器视觉中的机器学习方法与数据处理
 - 图像识别
 - 图像搜索

Text

□ Computer Vision: Algorithms and Applications

Richard Szeliski



Lecture 1&2

Fundamental of Computer Vision

“工欲善其事必先利其器”

Outline

- 1. CV背景介绍
- 2. OpenCV完全解析基础
- 3. 图像的基本操作：遍历图像，ROI选取等
- 4. Python环境搭建+语法
- 5. 机器学习在CV中的应用：Kmeans与KNN

What is Computer Vision?

What is Computer Vision?

- Goal of computer vision is to write computer programs that can interpret images



Why computer vision matters



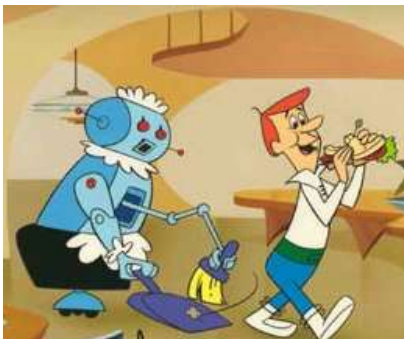
Safety



Health



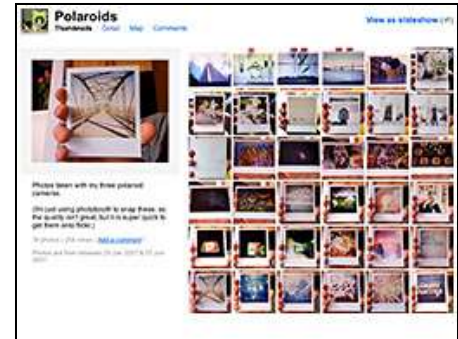
Security



Comfort



Fun



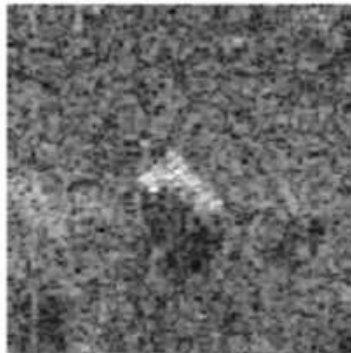
Access

Examples of application areas

- Target recognition

- Find enemy vehicles (which are trying not to be found!)

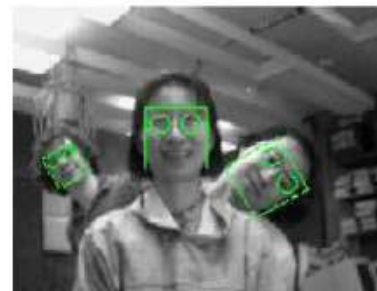
FLIR image from
sdvision.kaist.ac.kr/



SAR image from
web.mit.edu/6.003/courseware/

- Human Interfaces

- Detect faces and identify people
- Recognize gestures, activities



Examples of application areas

- Augmented reality

- Augment the real world with virtual objects



- Robotics

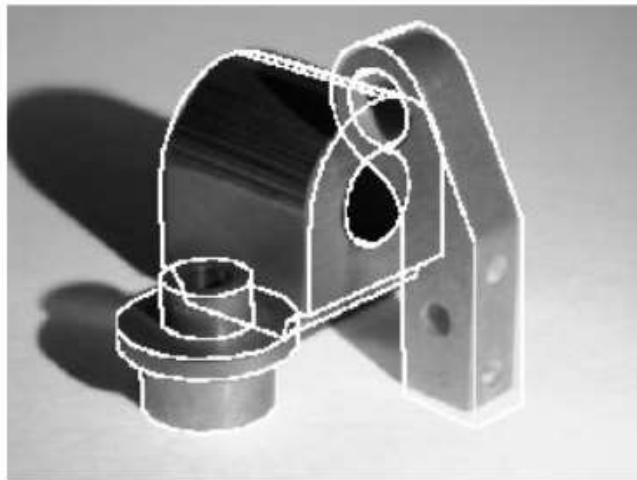
- Recognize objects
- Estimate motion and position



<http://www.robocup.org/>

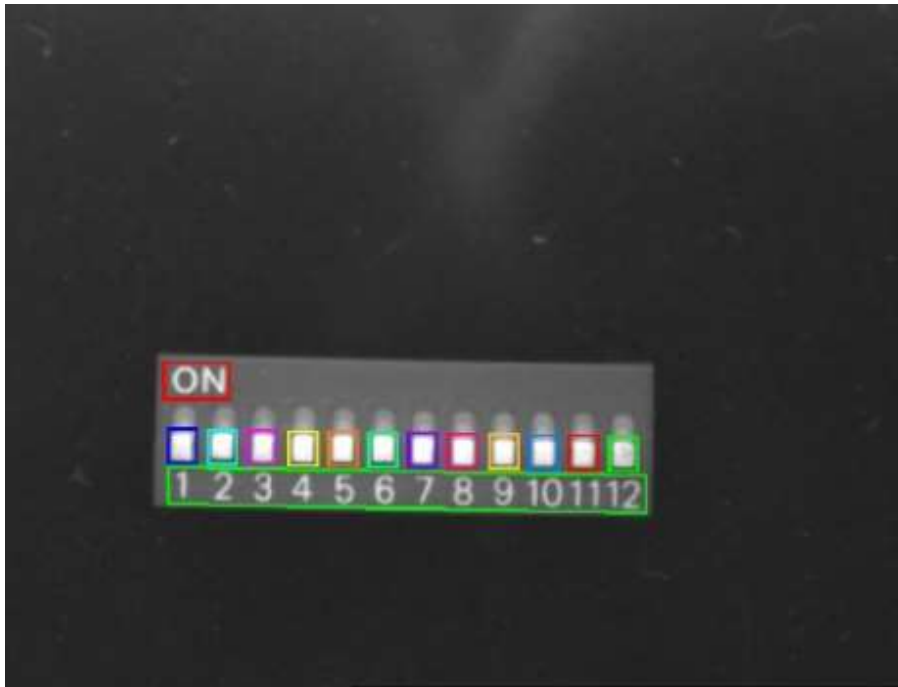
Examples of application areas

- Industrial inspection
 - Find known objects in the scene
 - Measure dimensions, verify features
- Optical character recognition
 - Processing scanned text pages
 - Detect and identify characters

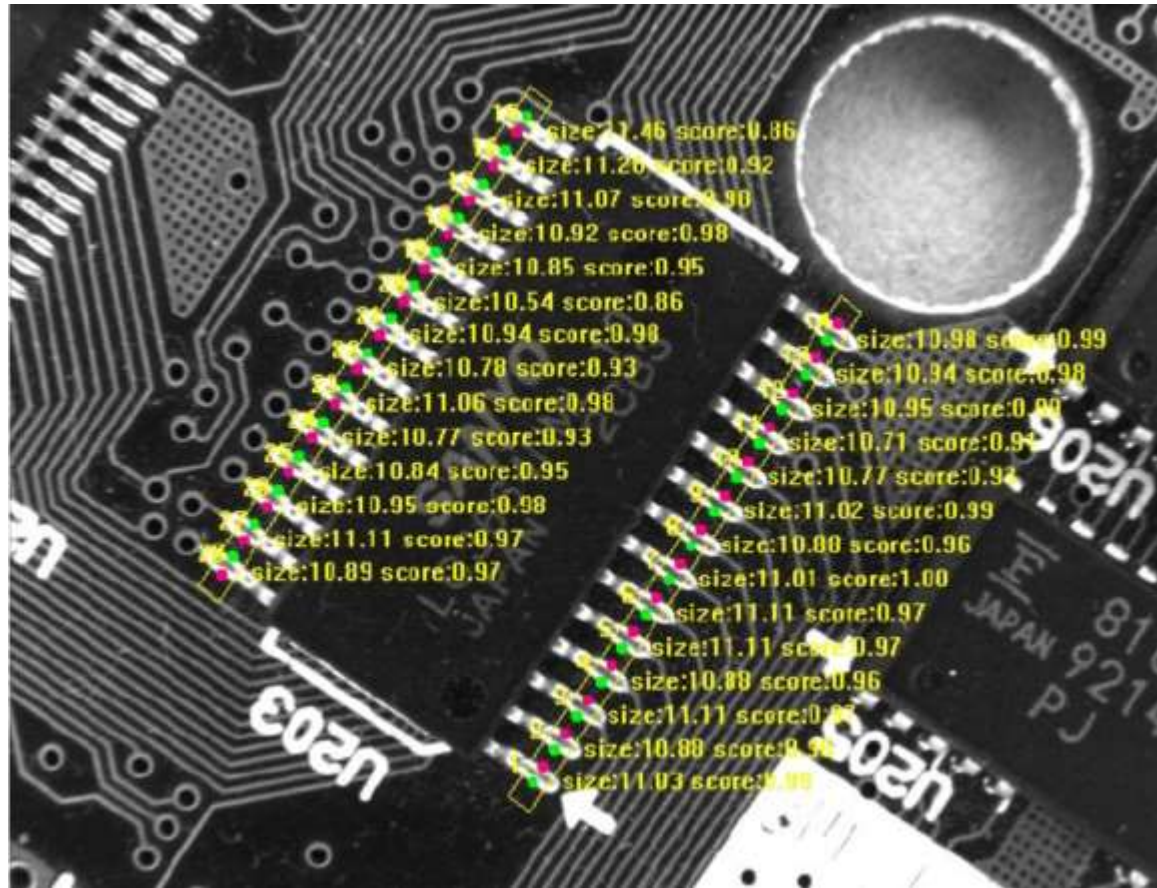


儘眼望遠極，
佰程無窮哩。
壹物明域現，
以迺吾後脊！

Examples of application areas



Examples of application areas



- Recognize objects
 - people we know
 - things we own
 - Locate objects in space
 - to pick them up
 - Track objects in motion
 - catching a baseball
 - avoiding collisions with cars on the road
 - Recognize actions
 - walking, running, pushing
-

Applications

- Geometric reconstruction: modeling, forensics, special effects (**ILM**, RealVis)
- Image and video editing (Avid, **Adobe**)
- Scientific / medical applications (GE)

ILM

CREDITS



2015



CAREERS | [f](#) [t](#) [i](#) [g+](#)

IN PRODUCTION

WARCRAFT

[LEGENDARY/UNIVERSAL]

An epic fantasy/adventure based on the popular video game series.

DIRECTOR: DUNCAN JONES

ILM VISUAL EFFECTS SUPERVISOR: JEFF WHITE, JASON SMITH, NIGEL SUMNER

ILM ANIMATION DIRECTOR: HAL HICKEL

ILM VISUAL EFFECTS ART DIRECTOR: CHRISTIAN ALZMANN

ILM VISUAL EFFECTS PRODUCER: JILL HUGHES

ILM EXECUTIVE VISUAL EFFECTS PRODUCER: JACQUI LOPEZ

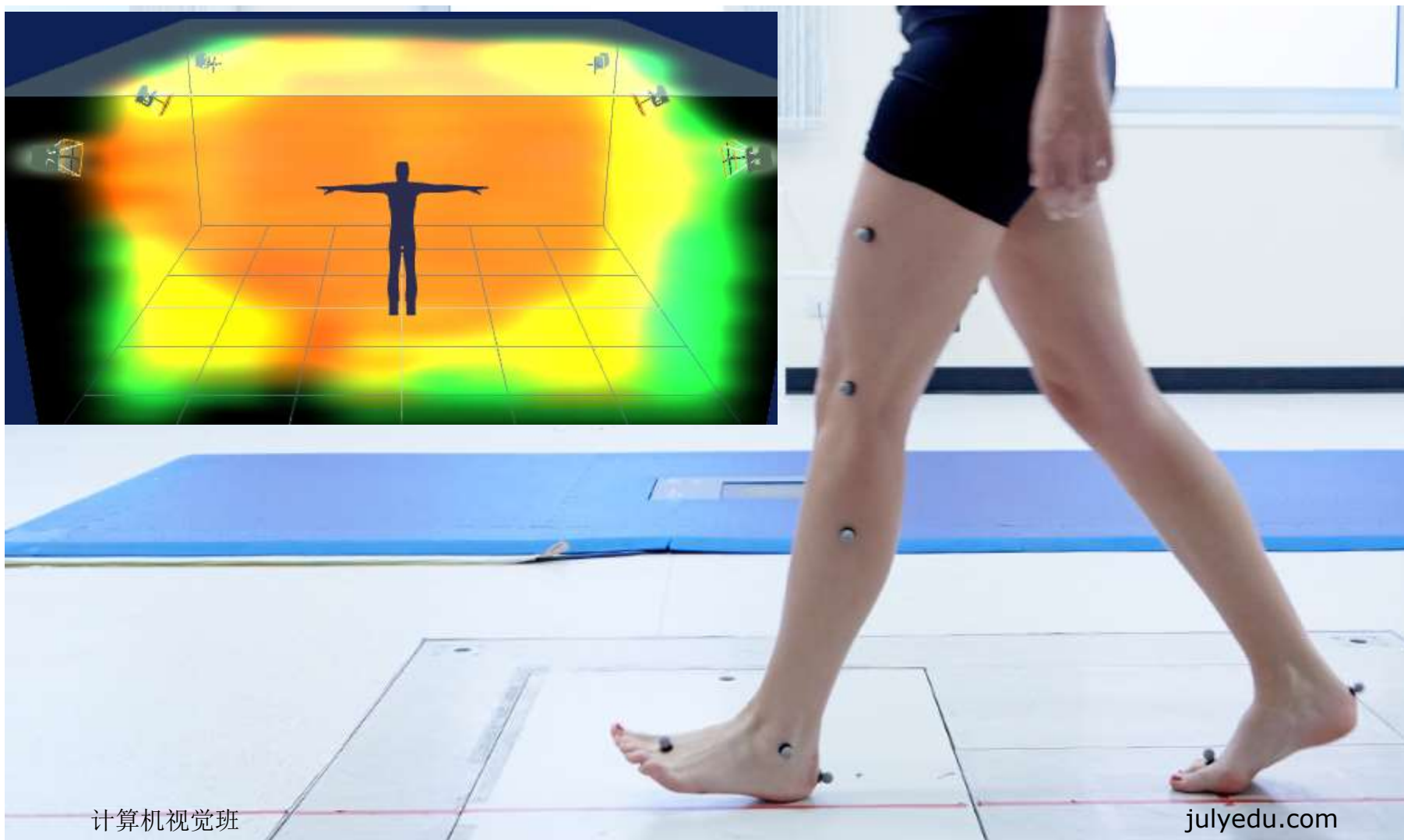
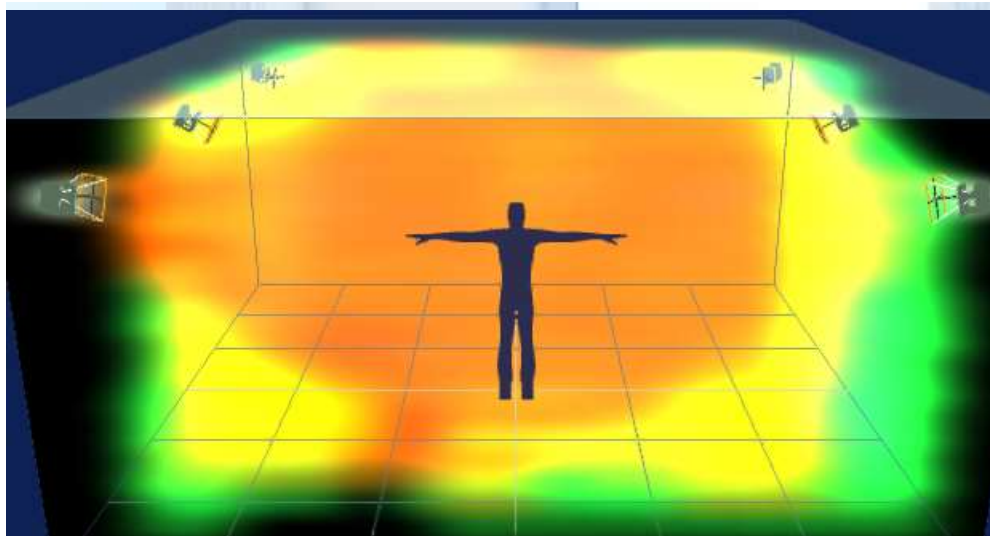
ILM STUDIOS: SAN FRANCISCO, SINGAPORE, VANCOUVER

ilmedu.com

Applications

- Tracking and surveillance (Sarnoff)
- Fingerprint recognition (Digital Persona)
- Biometrics / iris scans (Iridian Technologies)
- Vehicle safety (**MobilEye**)
- Optical motion capture (Vicon)

Vicon

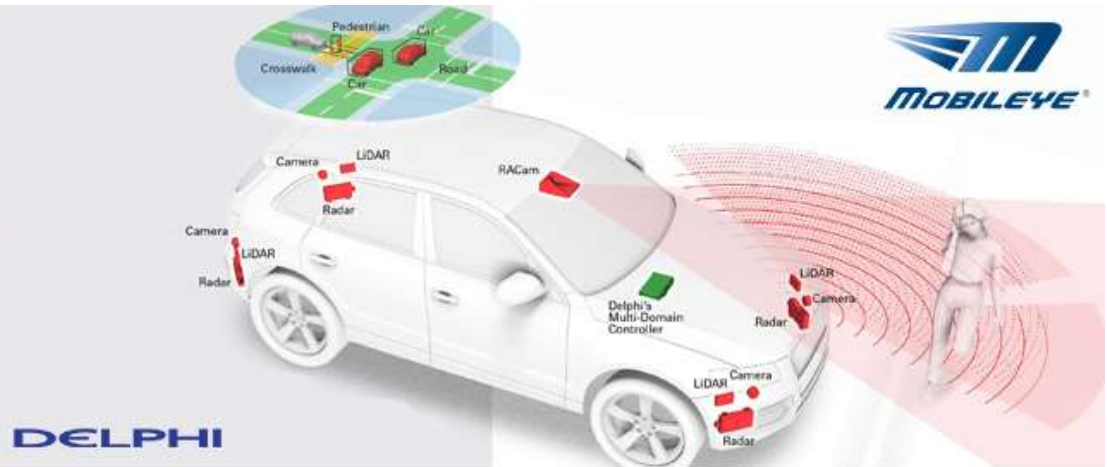


MobileEye

Announcing the Market's 1st Turnkey Level 4/5 Automated Driving Solution

Mobileye and Delphi partner to produce the "Central Sensing Localization and Planning" (CSLP) platform to accelerate the time to market for a complete automated driving solution.

1 2 3 4 5 6



Open Libraries/projects

□ OpenCV <http://opencv.org/>

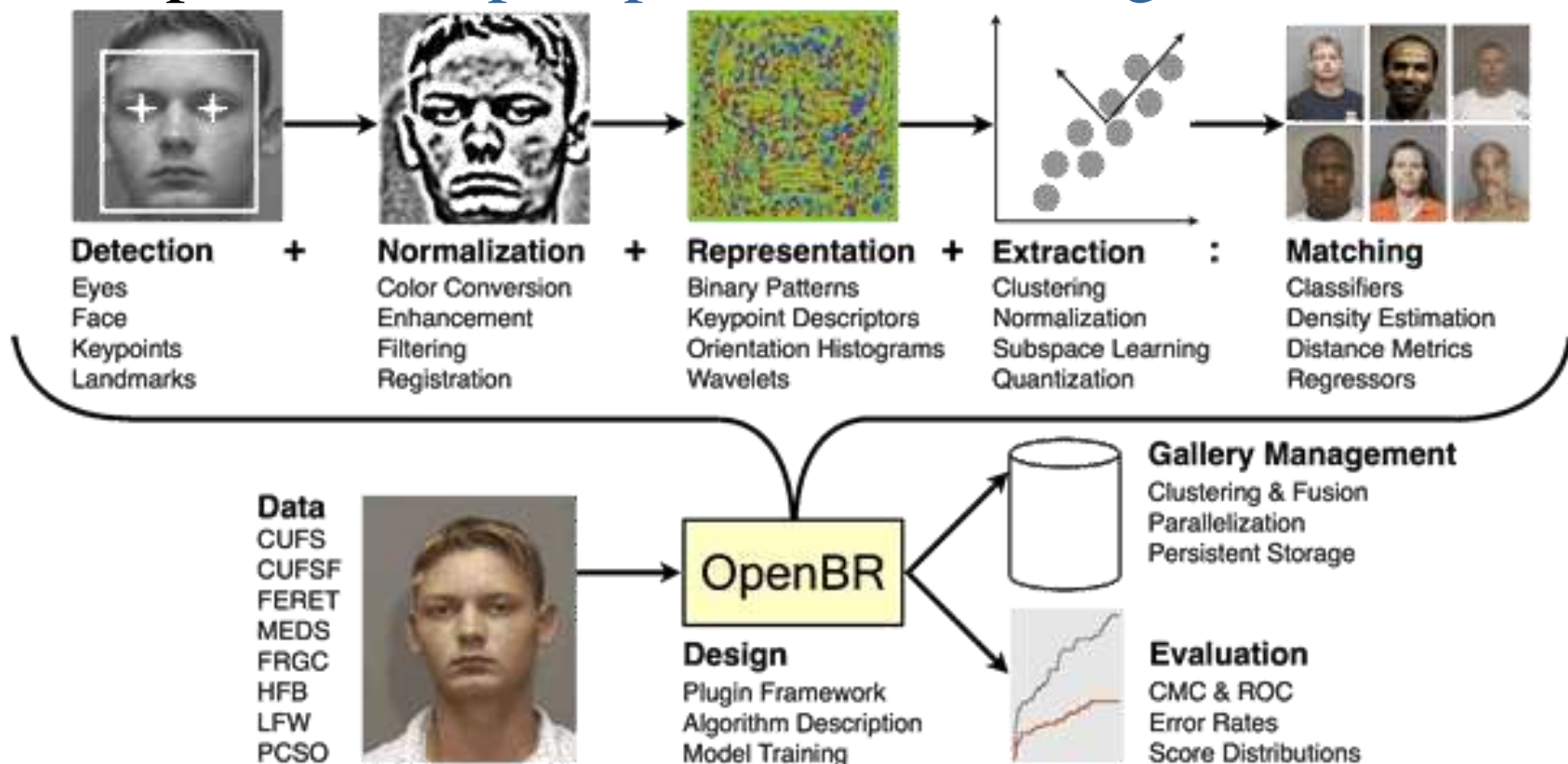
- C++, C, Python and Java interfaces
- Windows, Linux, Mac OS, iOS and Android

Open Libraries/projects

- HALCON <http://www.halcon.com/>
 - Easy programming in C++, C, C# or VB.NET
 - Available for Windows, Linux, Mac OS

Open Libraries/projects

□ OpenBR <http://openbiometrics.org/>



Open Libraries/projects

□ EasyPR

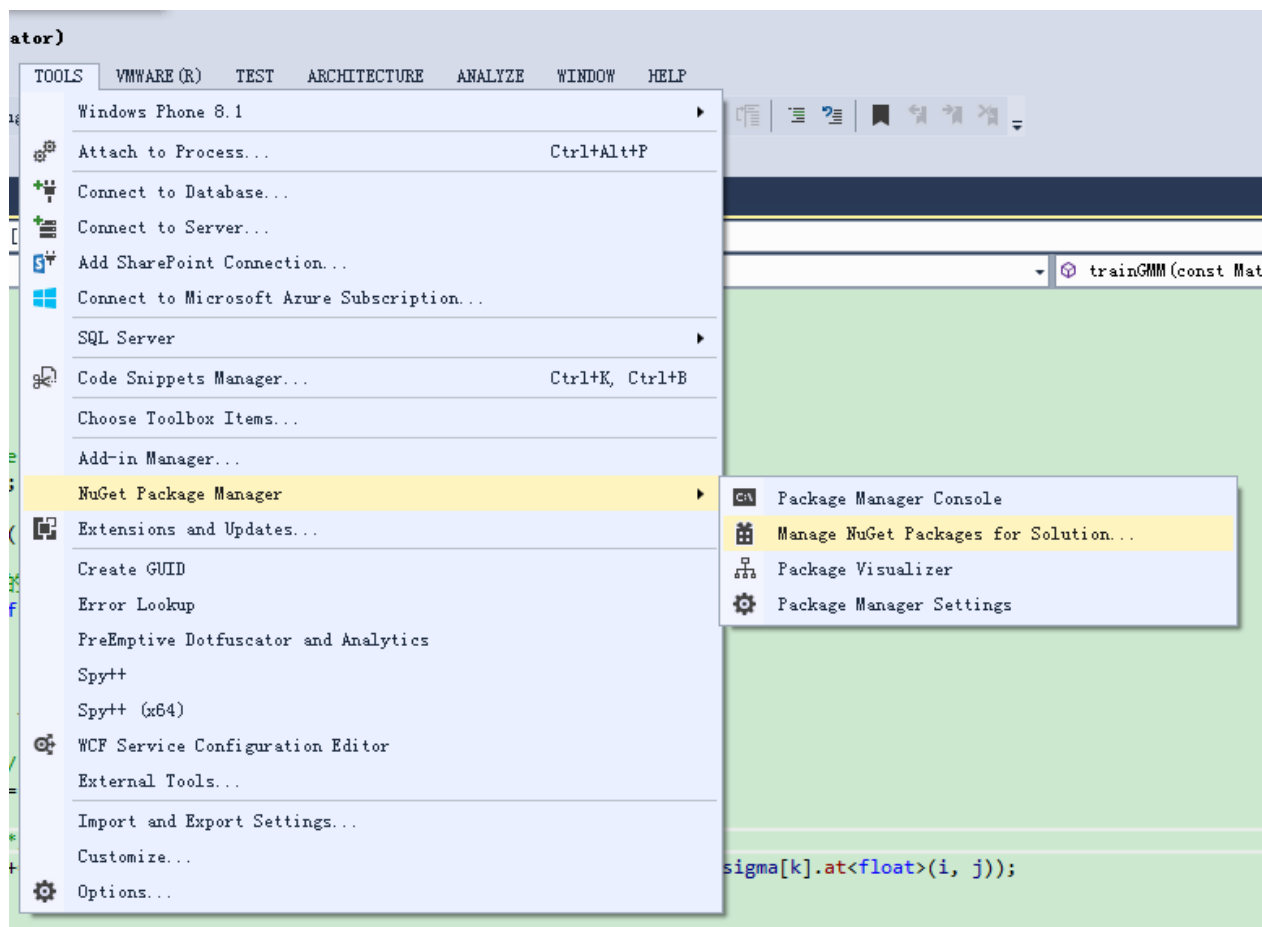
<http://git.oschina.net/easypr/EasyPR>



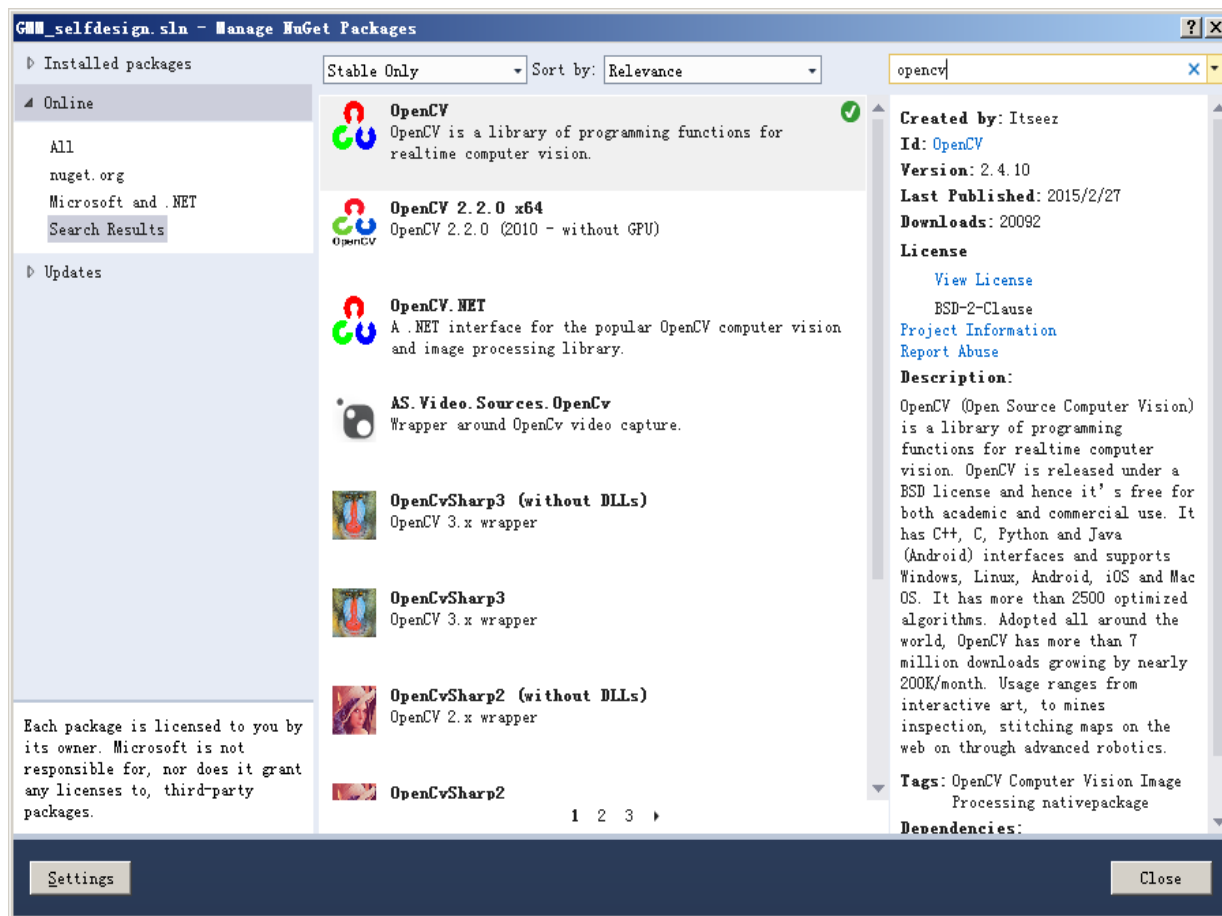
CV之旅从这里开始

建议：vs2013+opencv2.4.10
或以上版本

免安装方法



免安装方法



Opencv安装配置 (opencv2.4&3.0)

□ 下载:

■ <http://opencv.org/> (最新发布)



■ <https://sourceforge.net/projects/opencvlibrary/>



■ <https://github.com/opencv>

OpenCV文件

doc
include
java
python
share
x64
x86

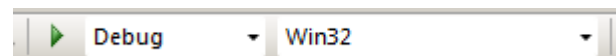
cv.h
cv.hpp
cvaux.h
cvaux.hpp
cwwimage.h
cxcore.h
cxcore.hpp
cxeigen.hpp
cxmisc.h
highgui.h
ml.h

calib3d
contrib
core
features2d
flann
gpu
highgui
imgproc
legacy
ml
nonfree
objdetect
ocl
photo
stitching
superres
ts
video
videostab
opencv.hpp
opencv_modules.hpp

Opencv安装配置 (opencv2.4&3.0)

□ 环境变量 PATH

■ opencv\build\x86\vc10\bin



■ opencv\build\x64\vc10\bin (64位编译)

```
%OPENCVROOT%\cmakeOpenCV\X64VS12CV3\install\x64\vc12\bin  
%OPENCVROOT%\cmakeOpenCV\X86VS12CV300\install\x86\vc12\bin  
%OPENCVROOT%\opencv2.4.10\build\x64\vc10\bin  
%OPENCVROOT%\opencv2.4.10\build\x86\vc10\bin  
  
OpenCV\opencv2411\build\x86\vc12\bin  
OpenCV\opencv2411\build\x86\vc10\bin
```


Opencv安装配置 (opencv2.4&3.0)

☐ VS工程“属性管理器”中添加：

■ **【通用属性】 -> 【VC++目录】 -> 【包含目录】**

- ☐ build\include
- ☐ build\include\opencv
- ☐ build\include\opencv2

■ **【通用属性】 -> 【VC++目录】 -> 【库目录】**

- ☐ opencv\build\x86\vc12\lib

■ **【通用属性】 -> 【链接器】 -> 【输入】 -> 【附加的依赖项】**

- ☐ opencv_nonfree2411d.lib opencv_nonfree2411.lib....
- ☐ opencv_ts300d.lib opencv_world300d.lib

Image Watch

- ❑ 1、Image Watch 的下载[链接](#)。
- ❑ 2、OpenCV 关于Image Watch的[介绍页面链接](#)。
- ❑ 3、OpenCV2.4 在线文档关于Image Watch的[介绍文档](#)。
- ❑ 4、更详细的信息参见Image Watch的[官方网站](#)。

view → other windows → Image watch

程序模板

```
□ #include <iostream>
   #include "opencv2/opencv.hpp"
   using namespace std;
   using namespace cv;
```

学习OpenCV

□ Model:

- Core → opencv_core.lib
- Imgproc → opencv_imgproc.lib
- Highgui → opencv_highgui.lib

认识图像

- 二值图像
- 灰度图像
- 彩色图像 (RGB)
- 多种颜色空间



10	11	10	9	9	9	10	11	12	16	18	10	9	9	9	10
10	10	10	10	10	11	10	16	26	59	69	16	10	11	9	10
10	10	10	11	16	27	49	62	89	134	147	34	12	11	15	15
10	10	11	20	43	109	153	162	165	175	171	110	22	47	73	39
9	10	37	117	166	184	187	193	180	170	171	166	65	84	65	14
10	43	165	186	185	185	189	181	158	115	135	154	123	92	16	16
35	159	183	178	174	155	118	90	77	44	28	77	138	45	51	88
79	176	186	174	150	102	78	56	35	19	14	43	102	47	146	102
89	177	186	179	175	139	104	47	25	36	90	140	141	34	135	33
98	171	181	185	189	188	158	95	68	172	198	186	188	48	84	39
114	155	177	188	192	198	193	164	154	201	209	204	210	151	43	114
142	144	167	173	178	174	172	166	178	190	202	208	209	208	115	35
150	154	161	168	168	162	176	177	175	172	183	189	203	210	171	39
155	151	162	170	164	177	186	183	167	138	173	190	193	209	175	40

digital images

== arrays of numbers

== matrix

Data in memory

表 3-1 灰度图像的存储示意图

I_{00}	I_{01}	...	I_{0N-1}
I_{10}	I_{11}	...	I_{1N-1}
...
I_{M-10}	I_{M-11}	...	I_{M-1N-1}

	Column 0			Column 1			Column ...			Column m		
Row 0	0,0	0,0	0,0	0,1	0,1	0,1	0, m	0, m	0, m
Row 1	1,0	1,0	1,0	1,1	1,1	1,1	1, m	1, m	1, m
Row,0	...,0	...,0	...,1	...,1	...,1, m	..., m	..., m
Row n	n,0	n,0	n,0	n,1	n,1	n,1	n,...	n,...	n,...	n, m	n, m	n, m

```

class CV_EXPORTS Mat
{
public:
    //一系列函数
    ...
    /* flag 参数中包含许多关于矩阵的信息，如：
        -Mat 的标识
        -数据是否连续
        -深度
        -通道数目
    */
    int flags;
    //矩阵的维数，取值应该大于或等于 2
    int dims;
    //矩阵的行数和列数，如果矩阵超过 2 维，这两个变量的值都为-1
    int rows, cols;
    //指向数据的指针
    uchar* data;

    //指向引用计数的指针
    //如果数据是由用户分配的，则为 NULL
    int* refcount;

    //其他成员变量和成员函数
    ...
};

```

```

Mat M(3,2, CV_8UC3, Scalar(0,0,255));
cout << "M = " << endl << " " << M << endl;

```

```

M =
[0, 0, 255, 0, 0, 255;
 0, 0, 255, 0, 0, 255;
 0, 0, 255, 0, 0, 255]

```


常用构造函数

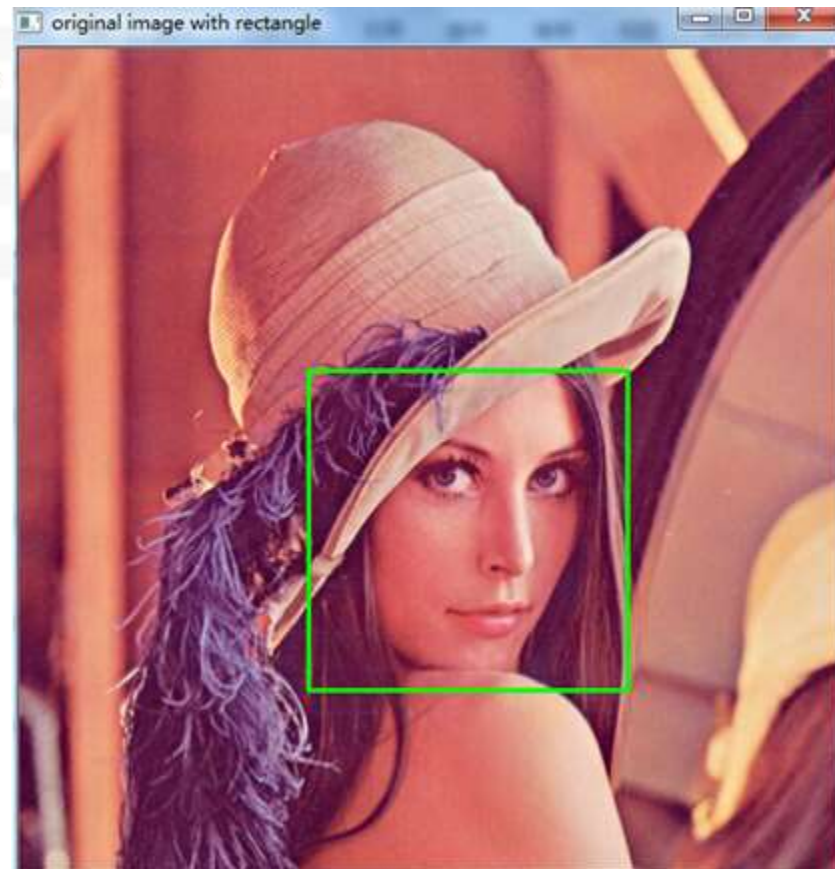
- ❑ `Mat::Mat()`
- ❑ `Mat::Mat(int rows, int cols, int type)`
- ❑ `Mat::Mat(Size size, int type)`
- ❑ `Mat::Mat(int rows, int cols, int type, const Scalar& s)`
- ❑ `Mat::Mat(Size size, int type, const Scalar& s)`
- ❑ `Mat::Mat(const Mat& m)`
- ❑ `Mat::Mat(int rows, int cols, int type, void* data, size_t step=AUTO_STEP)`
- ❑ `Mat::Mat(Size size, int type, void* data, size_t step=AUTO_STEP)`
- ❑ `Mat::Mat(const Mat& m, const Range& rowRange, const Range& colRange)`
- ❑ `Mat::Mat(const Mat& m, const Rect& roi)`

`Mat M(2,2, CV_8UC3); //构造函数创建图像`

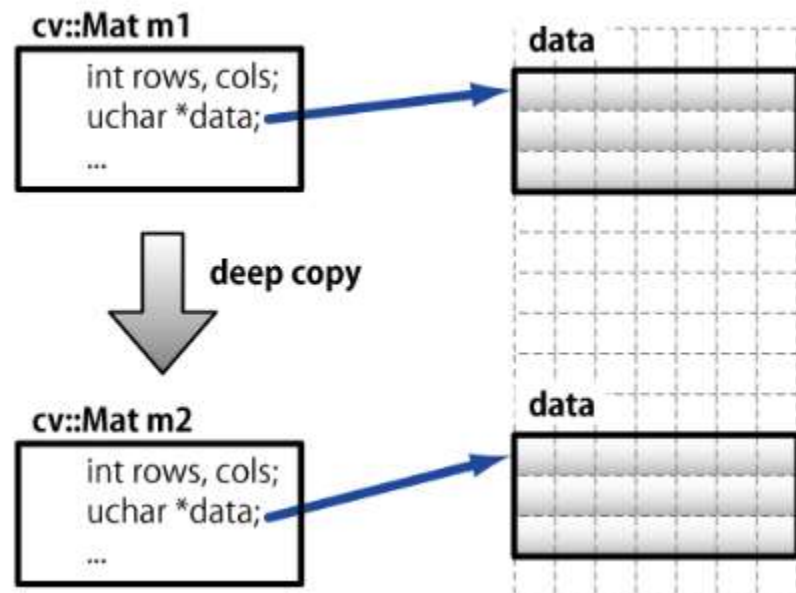
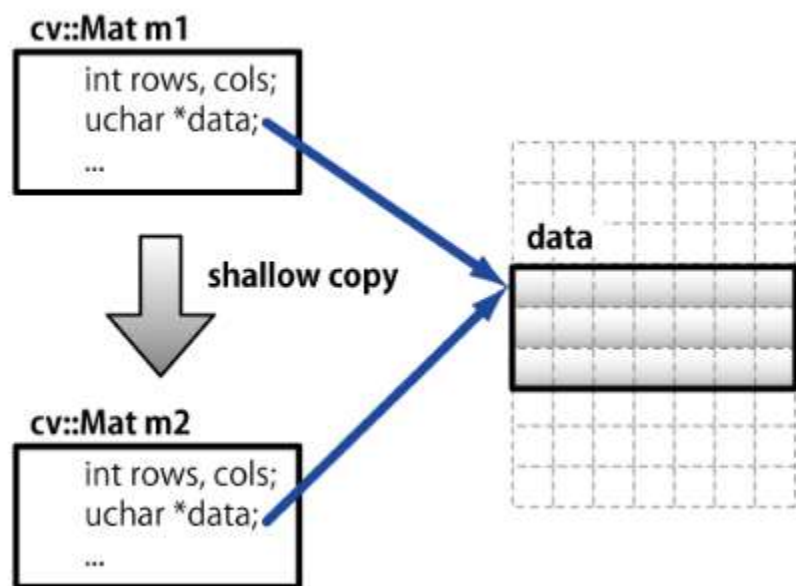
`M.create(3,2, CV_8UC2); //释放内存重新创建图像`

ROI

```
// Rect
cv::Mat pImg = imread("Lena.jpg",1);
cv::Rect rect(180,200,200,200);//(x,y)=(180,200),w=200,height=200
cv::Mat roi = cv::Mat(pImg, rect);
cv::Mat pImgRect = pImg.clone();
cv::rectangle(pImgRect,rect,cv::Scalar(0,255,0),2);
cv::imshow("original image with rectangle",pImgRect);
cv::imshow("roi",roi);
cv::waitKey();
```



Mat的赋值和拷贝问题



Similar to Matlab

```
□ Mat Z = Mat::zeros(2,3, CV_8UC1);  
  cout << "Z = " << endl << " " << Z << endl;  
  Mat O = Mat::ones(2, 3, CV_32F);  
  cout << "O = " << endl << " " << O << endl;  
  Mat E = Mat::eye(2, 3, CV_64F);  
  cout << "E = " << endl << " " << E << endl;
```

```
□ E = [ ? ]
```

像素值的读写 1

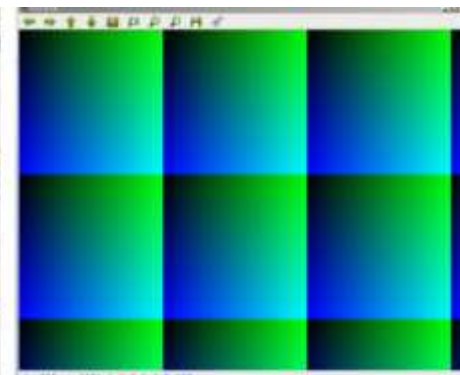
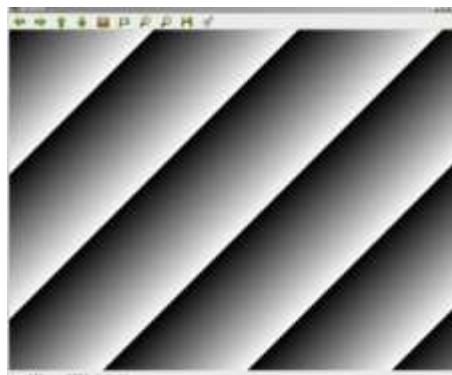
```
□ uchar value = grayim.at<uchar>(i,j);  
  for( int i = 0; i < grayim.rows; ++i)  
    for( int j = 0; j < grayim.cols; ++j )  
      grayim.at<uchar>(i,j) = (i+j)%255;
```

```
for( int i = 0; i < colorim.rows; ++i)  
  for( int j = 0; j < colorim.cols; ++j )  
  {  
    Vec3b pixel;  
    pixel[0] = i%255; //Blue  
    pixel[1] = j%255; //Green  
    pixel[2] = 0; //Red  
    colorim.at<Vec3b>(i,j) = pixel;  
  }
```

像素值的读写 1

```
□ uchar value = grayim.at<uchar>(i,j);  
  for( int i = 0; i < grayim.rows; ++i)  
    for( int j = 0; j < grayim.cols; ++j )  
      grayim.at<uchar>(i,j) = (i+j)%255;
```

```
for( int i = 0; i < colorim.rows; ++i)  
  for( int j = 0; j < colorim.cols; ++j )  
  {  
    Vec3b pixel;  
    pixel[0] = i%255; //Blue  
    pixel[1] = j%255; //Green  
    pixel[2] = 0; //Red  
    colorim.at<Vec3b>(i,j) = pixel;  
  }
```



像素值的读写 2

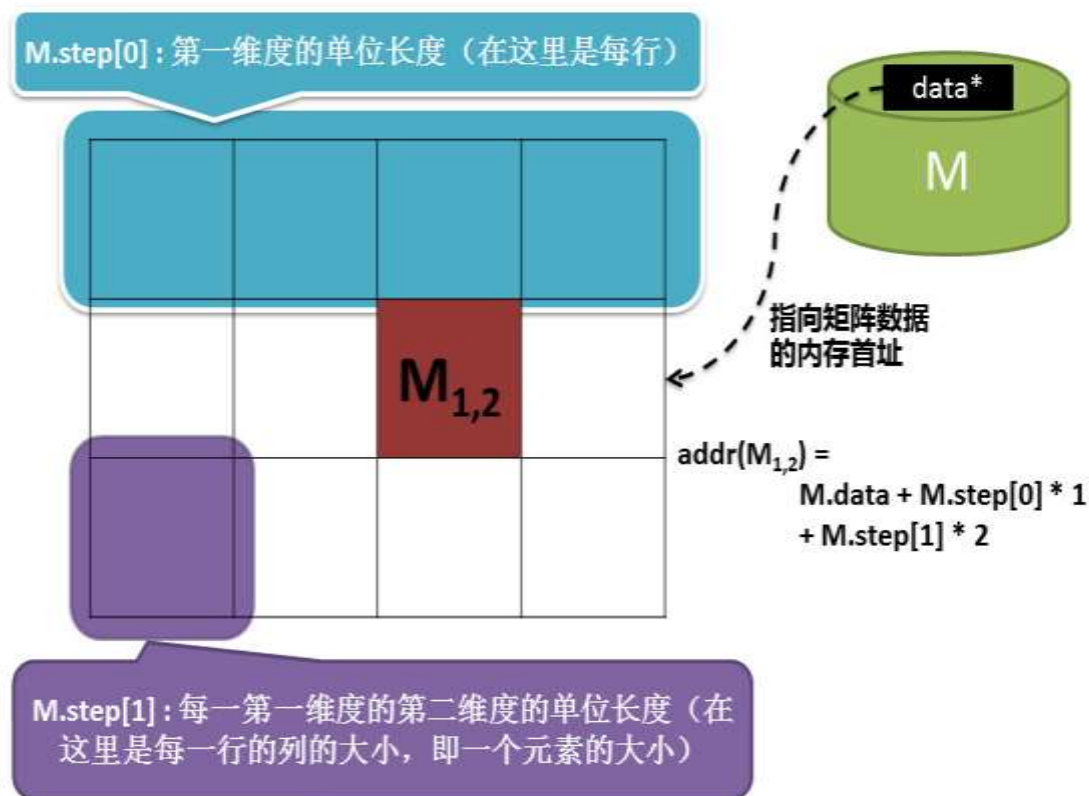
□ `cv::Mat Iterator_<uchar> grayit, grayend;`
`for(grayit = grayim.begin<uchar>(), grayend =`
`grayim.end<uchar>(); grayit != grayend; ++grayit)`
`*grayit = rand()%255;`

```
MatIterator_<Vec3b> colorit, colorend;
for( colorit = colorim.begin<Vec3b>(), colorend =
colorim.end<Vec3b>(); colorit != colorend; ++colorit)
{
    (*colorit)[0] = rand()%255; //Blue
    (*colorit)[1] = rand()%255; //Green
    (*colorit)[2] = rand()%255; //Red
}
```


像素值的读写 3

```
□ for( int i = 0; i < grayim.rows; ++i)
{
    // 获取第 i 行首像素指针
    uchar * p = grayim.ptr<uchar>(i);
    // 对第 i 行的每个像素(byte)操作
    for( int j = 0; j < grayim.cols; ++j )
        p[j] = (i+j)%255;
}
```

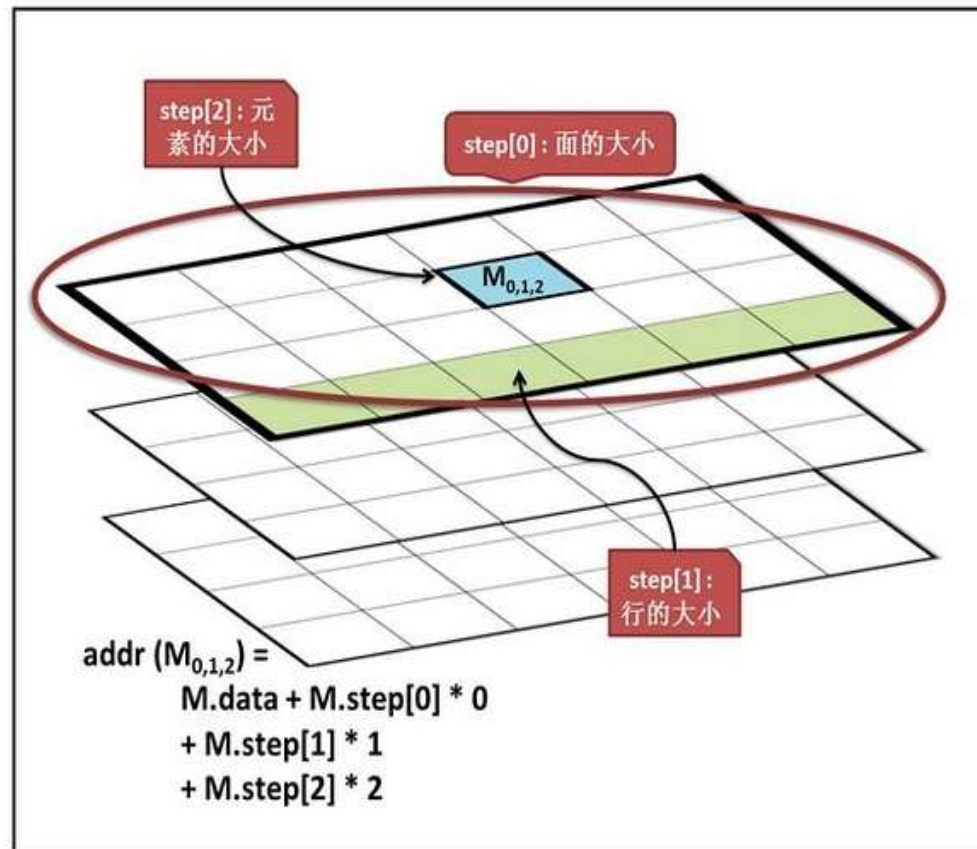
像素值的读写 4



$$addr(M_{i_0, i_1, \dots, i_{m-1}}) = M.data + M.step[0] * i_0 + M.step[1] * i_1 + \dots + M.step[m-1] * i_{m-1} \quad (\text{其中 } m = M.dims \text{ 的维度})$$

像素值的读写 4

图片分析2：考虑三维情况 (stored plane by plane) 按面存储



```

int main()
{
    //新建一个 uchar 类型的单通道矩阵 (grayscale image 灰度图)
    Mat m(400, 400, CV_8U, Scalar(0));
    for (int col = 0; col < 400; col++)
    {
        for (int row = 195; row < 205; row++)
        {
            cout << (int)(*(m.data + m.step[0] * row + m.step[1] * col)) << " ==> ";
            //获取第[row,col]个像素点的地址并用 * 符号解析
            *(m.data + m.step[0] * row + m.step[1] * col) = 255;
            cout << (int)(*(m.data + m.step[0] * row + m.step[1] * col)) << endl;
        }
    }
    imshow("canvas", m);
    cvWaitKey();
    return 0;
}

```

```

int main()
{
    //读入一个彩色图像
    Mat m = imread("../data/HappyFish.jpg");
    int *p_address;
    Vec3i color;
    for (int col = 20; col < 40; col++)
    {
        for (int row = 2; row < 20; row++)
        {
            color[0] = (int)(*(m.data + m.step[0] * row + m.step[1] * col));
            color[1] = (int)(*(m.data + m.step[0] * row + m.step[1] * col + m.elemSize1()));
            color[2] = (int)(*(m.data + m.step[0] * row + m.step[1] * col + m.elemSize1()*2));
            //获取第[row,col]个像素点的地址并用 * 符号解析
            cout << color[0]<<","<<color[1]<<","<<color[2] << " ==> ";
            color[0] = 255;
            color[1] = 0;
            color[2] = 0;
            *(m.data + m.step[0] * row + m.step[1] * col) = color[0];
            *(m.data + m.step[0] * row + m.step[1] * col + m.elemSize1()) = color[1];
            *(m.data + m.step[0] * row + m.step[1] * col + m.elemSize1()*2) = color[2];
            cout << (int)*(m.data + m.step[0] * row + m.step[1] * col) << (int)*(m.data +
m.step[0] * row + m.step[1] * col + 1) << (int)*(m.data + m.step[0] * row + m.step[1] * col + 2) <<
endl;
        }
    }
    imshow("canvas", m); cvWaitKey();
    return 0;
}

```

像素值的读写 5 Mat_类

```
Mat M(600, 800, CV_8UC1);
```

```
for( int i = 0; i < M.rows; ++i)  
{
```

```
    uchar * p = M.ptr<uchar>(i);  
    for( int j = 0; j < M.cols; ++j )  
    {
```

```
        double d1 = (double) ((i+j)%255);  
        M.at<uchar>(i,j) = d1;  
        double d2 = M.at<double>(i,j);
```

```
    }
```

```
}
```

Mat_类

```
Mat_<uchar> M1 = (Mat_<uchar>&)M;
for( int i = 0; i < M1.rows; ++i)
{
    uchar * p = M1.ptr(i);
    for( int j = 0; j < M1.cols; ++j )
    {
        double d1 = (double) ((i+j)%255);
        M1(i,j) = d1;
        double d2 = M1(i,j);

    }
}
```

像素值的读写 5

```
m = imread("../data/HappyFish.jpg");
Mat_<Vec3b> m2 = m;

// for 循环画一个红色的实心圆
for (int y = 21; y < 42; y++)
{
    for (int x = 2; x < 21; x++)
    {
        if (pow(double(x - 11), 2) + pow(double(y - 31), 2) - 64.0 < 0.000000000001)
        {
            // Mat_ 模板类实现了对()的重载, 可以定位到一个像素
            m2(x, y) = Vec3b(0, 0, 255);
        }
    }
}

imshow("CircleImage", m2);|
```


像素值的读写 6

- ☐ `int divideWith=10;`
- ☐ `uchar table[256];`
- ☐ `for (int i = 0; i < 256; ++i)`
- ☐ `table[i] = divideWith* (i/divideWith);`

- ☐ `Mat lookUpTable(1, 256, CV_8U);`
- ☐ `uchar* p = lookUpTable.data;`
- ☐ `for(int i = 0; i < 256; ++i)`
- ☐ `p[i] = table[i];`
- ☐ `LUT(I, lookUpTable, Out);`

Mat 与 IplImage 和 CvMat 的转换

❑ `void mycvOldFunc(IplImage * p, ...);`

❑ `Mat img(Size(320, 240), CV_8UC3);`

`...`

`IplImage iplimg = img;`

`//CvMat cvimg = img;`

`mycvOldFunc(& iplimg, ...);`

IplImage 和 CvMat 格式转为 Mat

`Mat::Mat(const CvMat* m, bool copyData=false)`

`Mat::Mat(const IplImage* img, bool copyData=false)`

`IplImage * iplimg = cvLoadImage("lena.jpg");`

`Mat im(iplimg, true);`

数据获取与存储

□ imread()

Mat imread(const string& filename, int flags=1)

Windows 位图文件 - BMP, DIB;

JPEG 文件 - JPEG, JPG, JPE;

便携式网络图片 - PNG;

便携式图像格式 - PBM, PGM, PPM;

Sun rasters - SR, RAS;

TIFF 文件 - TIFF, TIF;

OpenEXR HDR 图片 - EXR;

JPEG 2000 图片- jp2。

imwrite()

```
bool imwrite(const string& filename, InputArray  
image, const vector<int>& params=vector<int>())
```

推荐资料

□ 手册：<http://docs.opencv.org/>

□ 教程：
<http://docs.opencv.org/doc/tutorials/tutorials.html>



□ 进阶：<https://github.com/opencv/opencv/wiki>

Summary

- 开源的库与开源项目
- Opencv安装配置 (windows)
- 像素的读取, 地址分配, ROI

NEXT

Opencv视频设备

常见错误分析

Python 入门

机器学习知识介绍与实战

作业1

- 1, CV的第一个程序
read & save images
- 2, 读入“字符”数据

```
1 00000000000000011111110000000000
2 00000000001111111111110000000000
3 00000000001111111111111000000000
4 00000000111111111111111000000000
5 00000001111111111111111000000000
6 00000001111111000011111000000000
7 00000011111000000011111000000000
8 00000001100000000111100000000000
9 00000000000000000111100000000000
0 00000000000000001111000000000000
1 00000000000000001111100000000000
2 00000000000000011111100000000000
3 00000000000000111111100000000000
4 00000000000001111111100000000000
5 00000000000001111111110000000000
6 00000000000001111111111000000000
7 00000000000001111111111000000000
8 00000000000000000111111000000000
9 00000000000000000001111100000000
0 00000000000000000001111000000000
1 00000000000000000001111000000000
2 00000000000000000001111000000000
3 00000000000000000001111000000000
4 00000000000000000001111000000000
5 00000000000000000001111000000000
6 00000000000000000001111100000000
7 000000000000000000011111100000000
8 000000000000000000011111100000000
9 000000000000000000011111100000000
0 000000000000000000011111000000000
1 000000000000000000011111000000000
2 000000000000000000011111000000000
3 000000000000000000011111000000000
4 000000000000000000011111000000000
5 000000000000000000011111000000000
6 000000000000000000011111100000000
7 000000000000000000011111100000000
8 000000000000000000011111100000000
9 000000000000000000011111100000000
0 000000000000000000011111000000000
1 000000000000000000011111000000000
2 0000000000000000000100000000000000
```

感谢大家！

恳请大家批评指正！

Lecture 1&2

Fundamental of Computer Vision

“工欲善其事必先利其器”

Outline

- 1, OpenCV完全解析基础
- 2, 图像的基本操作：遍历图像，ROI选取等
- 3, Python环境搭建+语法
- 4, 机器学习在CV中的应用

Kmeans与KNN

Video读写类

```
□ VideoCapture cap(0);
□ VideoCapture cap("video.short.raw.avi");
  if(!cap.isOpened())
  {
    cerr << "Can not open a camera or file." << endl;
    return -1;
  }
  Mat edges;
  namedWindow("edges",1);
  for(;;)
  {
    Mat frame;
    cap >> frame;
    if(frame.empty())
      break;
    ....
    if(waitKey(30) >= 0)
      break;
  }
```

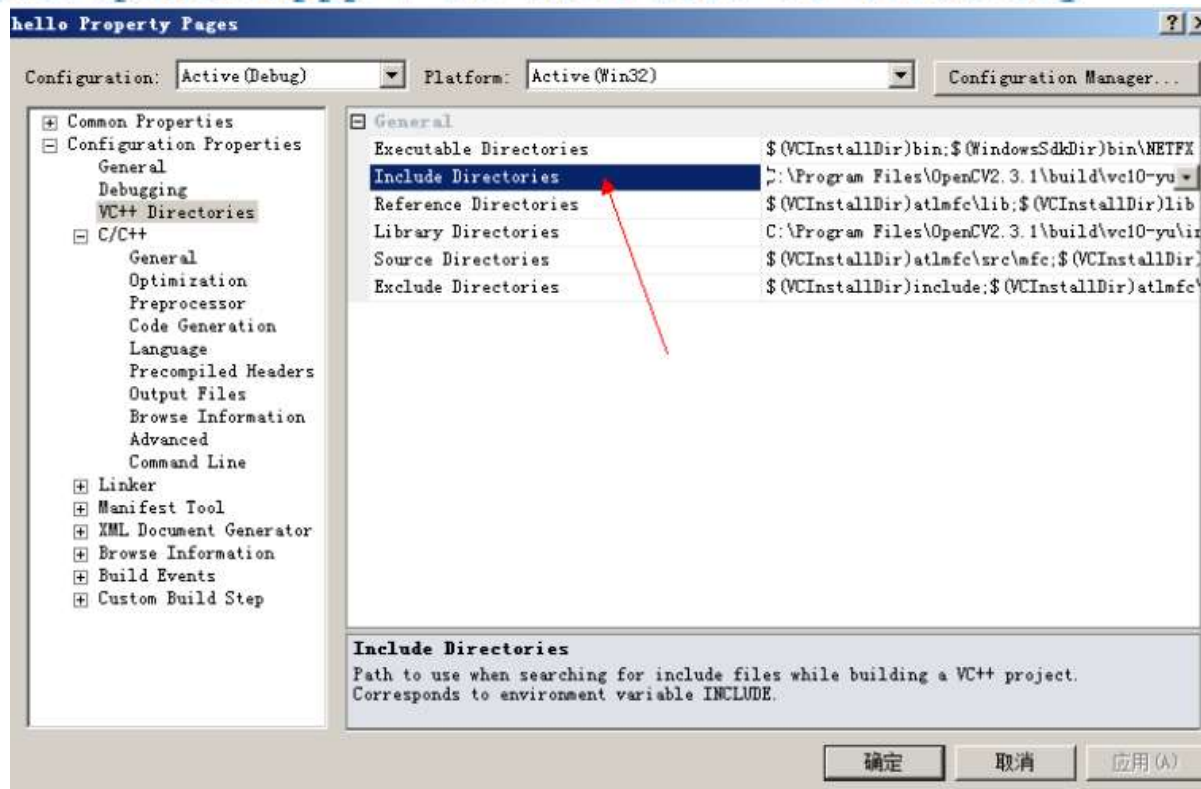
<http://www.fourcc.org/codecs.php>

Video读写类

```
Size s(320, 240);
VideoWriter writer = VideoWriter("myvideo.avi",
CV_FOURCC('M','J','P','G'), 25, s);
if(!writer.isOpened())
{
    cerr << "Can not create video file.\n" << endl;
    return -1;
}
//视频帧
Mat frame(s, CV_8UC3);
for(int i = 0; i < 100; i++)
{
    writer << frame;
}
```

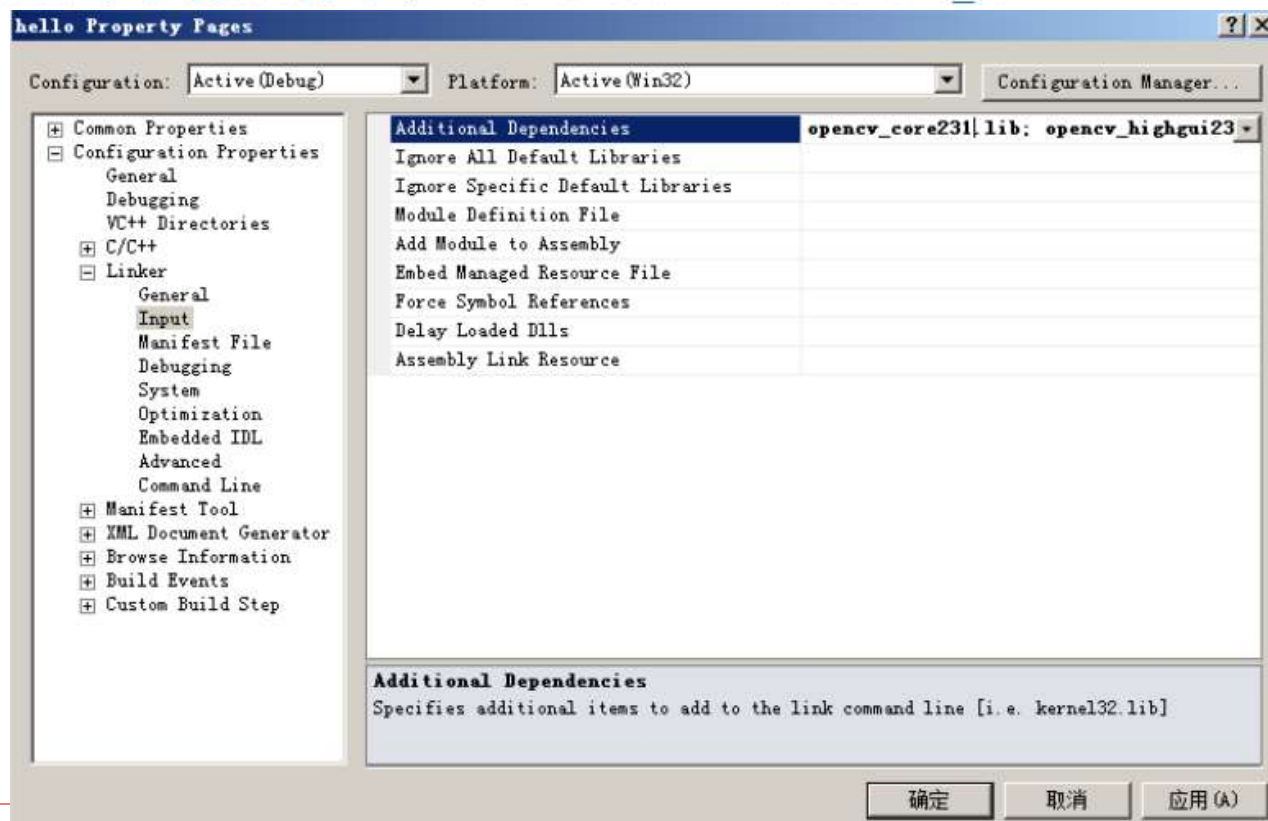
Common mistakes

```
hello.cpp(2): fatal error C1083: Cannot open include file:  
'opencv2/opencv.hpp': No such file or directory
```

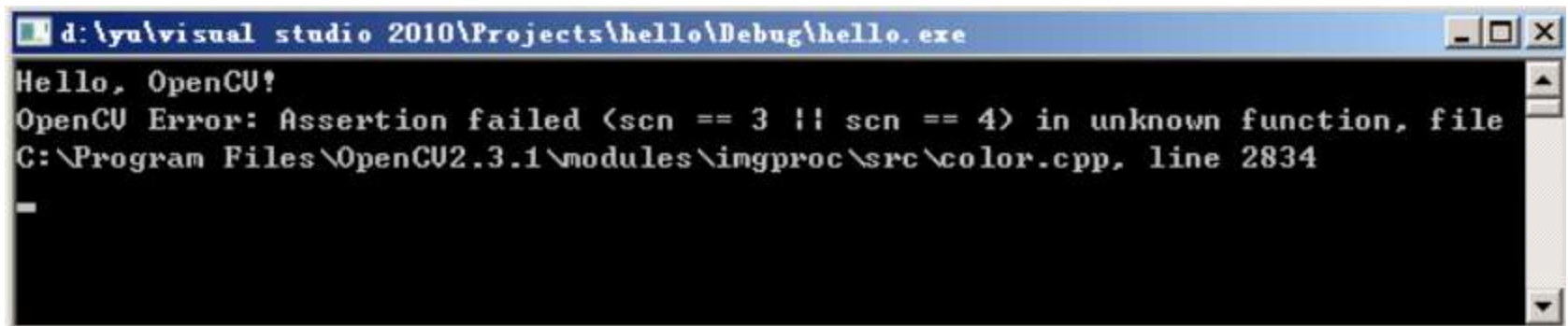


Common mistakes

```
1>hello.obj : error LNK2019: unresolved external symbol "class cv::Mat __cdecl cv::imread(class std::basic_string<char,struct std::char_traits<char>,class std::allocator<char> > const &,int)" (?imread@cv@@YA?AVMat@1@ABV?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@2@@@std@@H@Z) referenced in function _main
```



Common mistakes



Python in ONE page

□ Python 基础教程: <http://www.runoob.com/python/python-tutorial.html>

□ **Anaconda + Ipython + (PyScripter)**

■ pip install numpy

■ pip install scipy

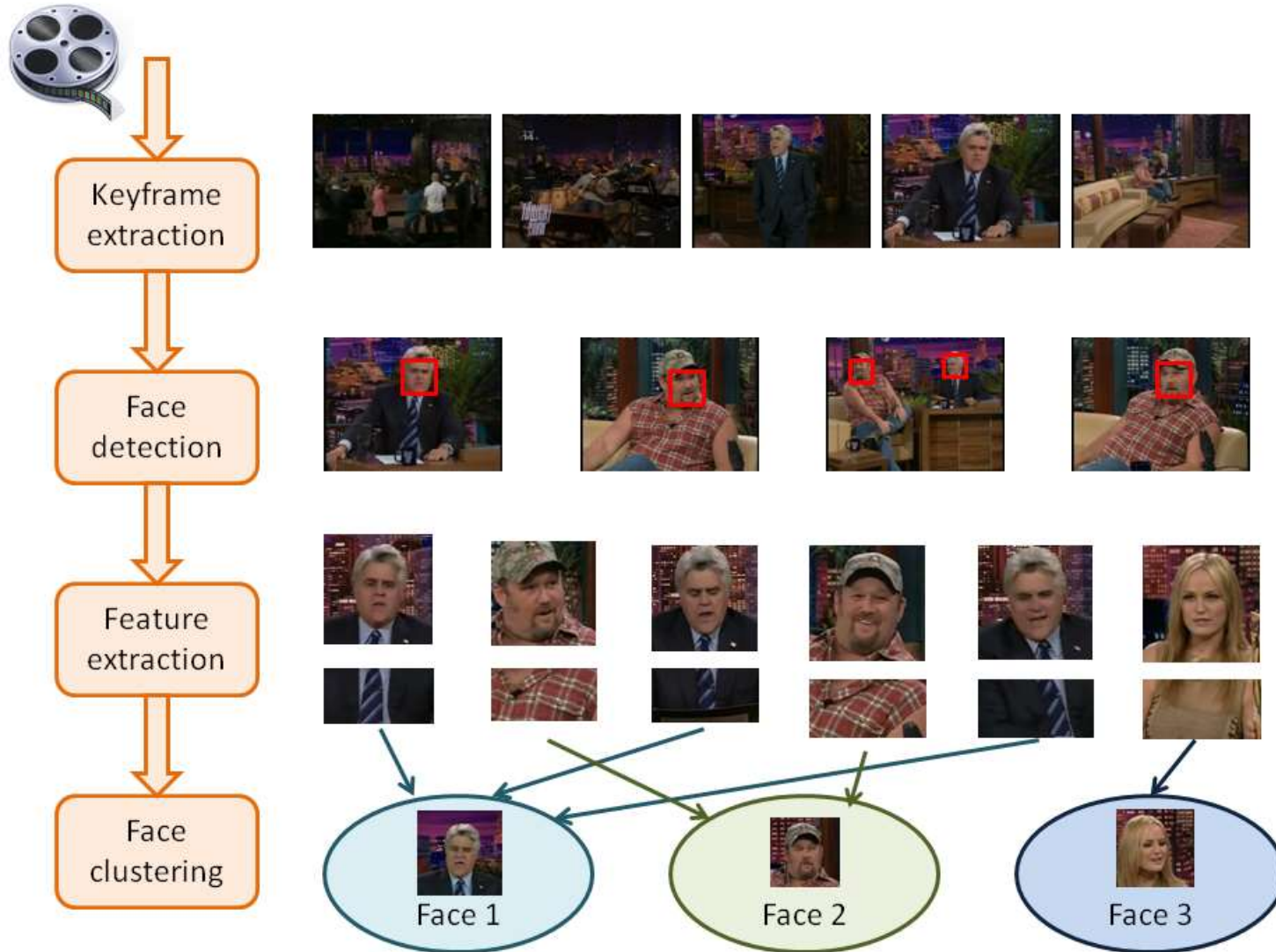
■ pip install matplotlib

■

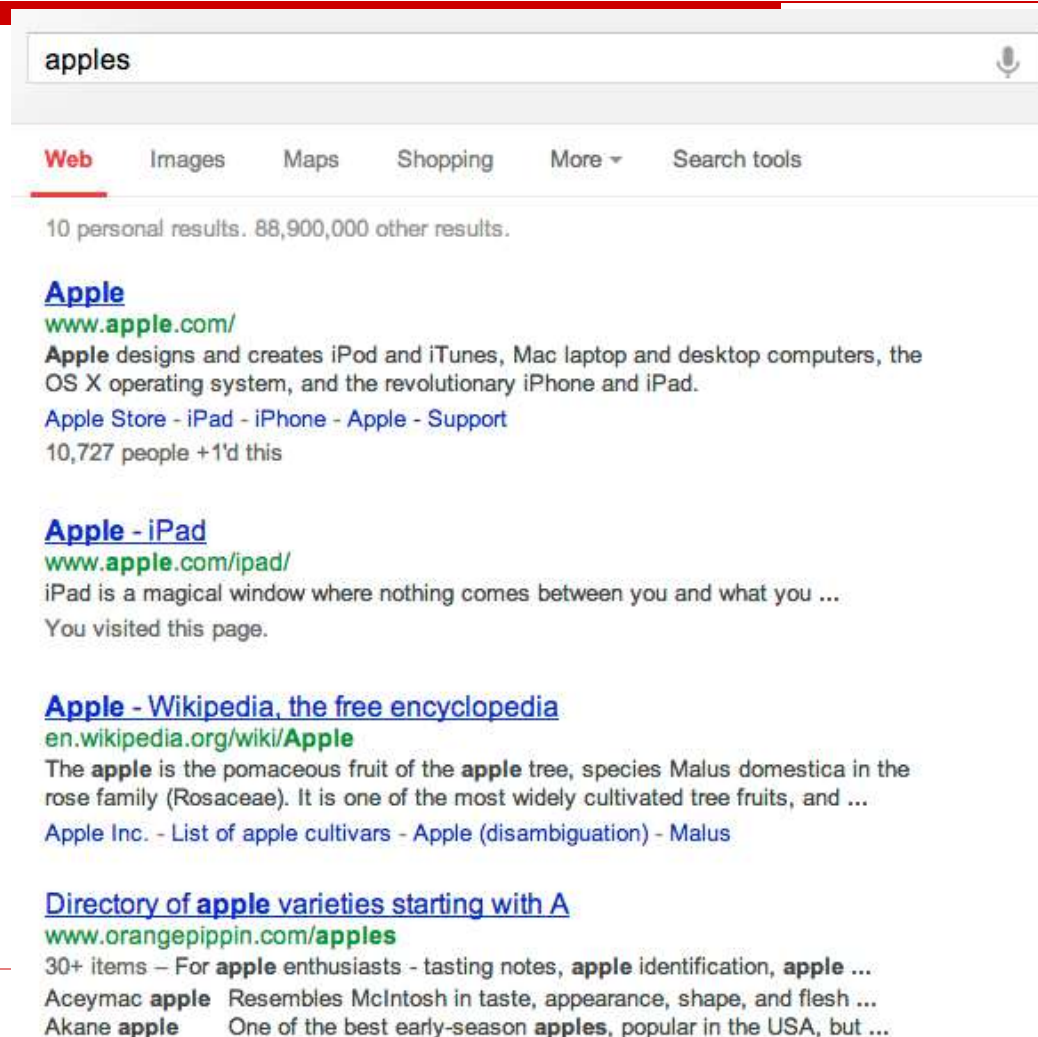
Machine Learning

- Machine learning is about learning some properties of a data set and applying them to new data
 - supervised learning
 - unsupervised learning

Face Clustering



Search result clustering



apples

Web Images Maps Shopping More Search tools

10 personal results. 88,900,000 other results.

[Apple](#)
www.apple.com/
Apple designs and creates iPod and iTunes, Mac laptop and desktop computers, the OS X operating system, and the revolutionary iPhone and iPad.
[Apple Store](#) - [iPad](#) - [iPhone](#) - [Apple](#) - [Support](#)
10,727 people +1'd this

[Apple - iPad](#)
www.apple.com/ipad/
iPad is a magical window where nothing comes between you and what you ...
You visited this page.

[Apple - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Apple
The **apple** is the pomaceous fruit of the **apple** tree, species *Malus domestica* in the rose family (Rosaceae). It is one of the most widely cultivated tree fruits, and ...
[Apple Inc.](#) - [List of apple cultivars](#) - [Apple \(disambiguation\)](#) - [Malus](#)

[Directory of apple varieties starting with A](#)
www.orangeippin.com/apples
30+ items – For **apple** enthusiasts - tasting notes, **apple** identification, **apple** ...
Aceymac apple Resembles McIntosh in taste, appearance, shape, and flesh ...
Akane apple One of the best early-season **apples**, popular in the USA, but ...



Google News

Google

News

Top Stories

Iran
Xbox One
Tarun Tejpal
Manny Pacquiao
Ukraine
Kabul
New England Patriots
Latvia
Derrick Rose
Doctor Who



+ Xbox One



E! Online

See realtime coverage

Console Wars 2013: Microsoft's Xbox One vs. Sony's PlayStation 4

E! Online - 1 hour ago

The future is now! Last week, Sony released its next generation console, PlayStation 4. This weekend, Microsoft drops the much touted all-in-one media device, Xbox One. We've been geeking out over the two new systems, and compiling a report on the new ...

[Xbox One sales exceed one million in first 24 hours](#) Joystiq - by David Hinkle

[Xbox One vs. PS4: A Guide to Making the Toughest Gaming Decision in Years](#)

ABC News - by Joanna Stern

Related

[Microsoft »](#)



Wall Street Journal



YouTube



YouTube



Washingto...



RedOrbit



Guardian ...



The Guardian

Xbox One and Microsoft websites marred by problems on launch day

The Guardian
9 hours ago



Written by
Jemima Kiss

Microsoft's Xbox One launch was marred by problems with its online services early on Friday which took down the official website Xbox.

Consumers line up for Xbox One

USA TODAY - Nov 23, 2013

Eager video game players lined up at stores across the country awaiting the arrival of Microsoft's Xbox One, a week to the day after rival Sony introduced its PlayStation 4. The console, available for sale tonight at 12:01 a.m.



Polygon

Here are all the Xbox One voice commands

Polygon
9 hours ago



Written by
Megan Farokhmanesh

Microsoft posted a guide to Xbox One voice commands, including how to navigate menus, control volume and multitask, on its Tumblr.

聚类算法

- ❑ “相似度” 评定对于聚类至关重要
- ❑ “相似度” 与 “距离” 负相关



K-means

Most well-known and popular clustering algorithm:

Start with some initial cluster centers

Iterate:

- Assign/cluster each example to closest center
- Recalculate centers as the mean of the points in a cluster



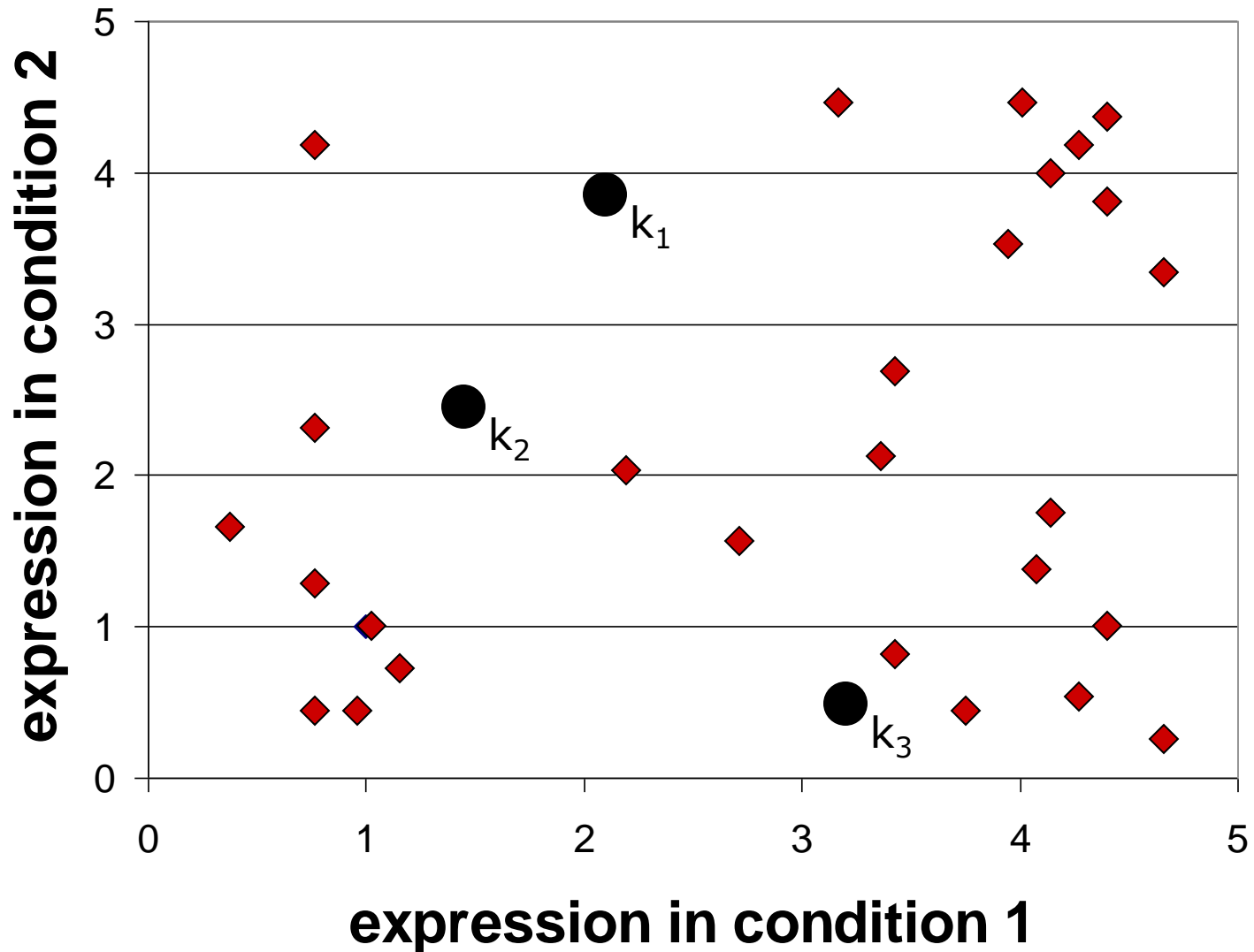
K-means algorithm

- 1) Pick a number (k) of cluster centers
- 2) Assign every gene to its nearest cluster center
- 3) Move each cluster center to the mean of its assigned genes
- 4) Repeat 2-3 until convergence



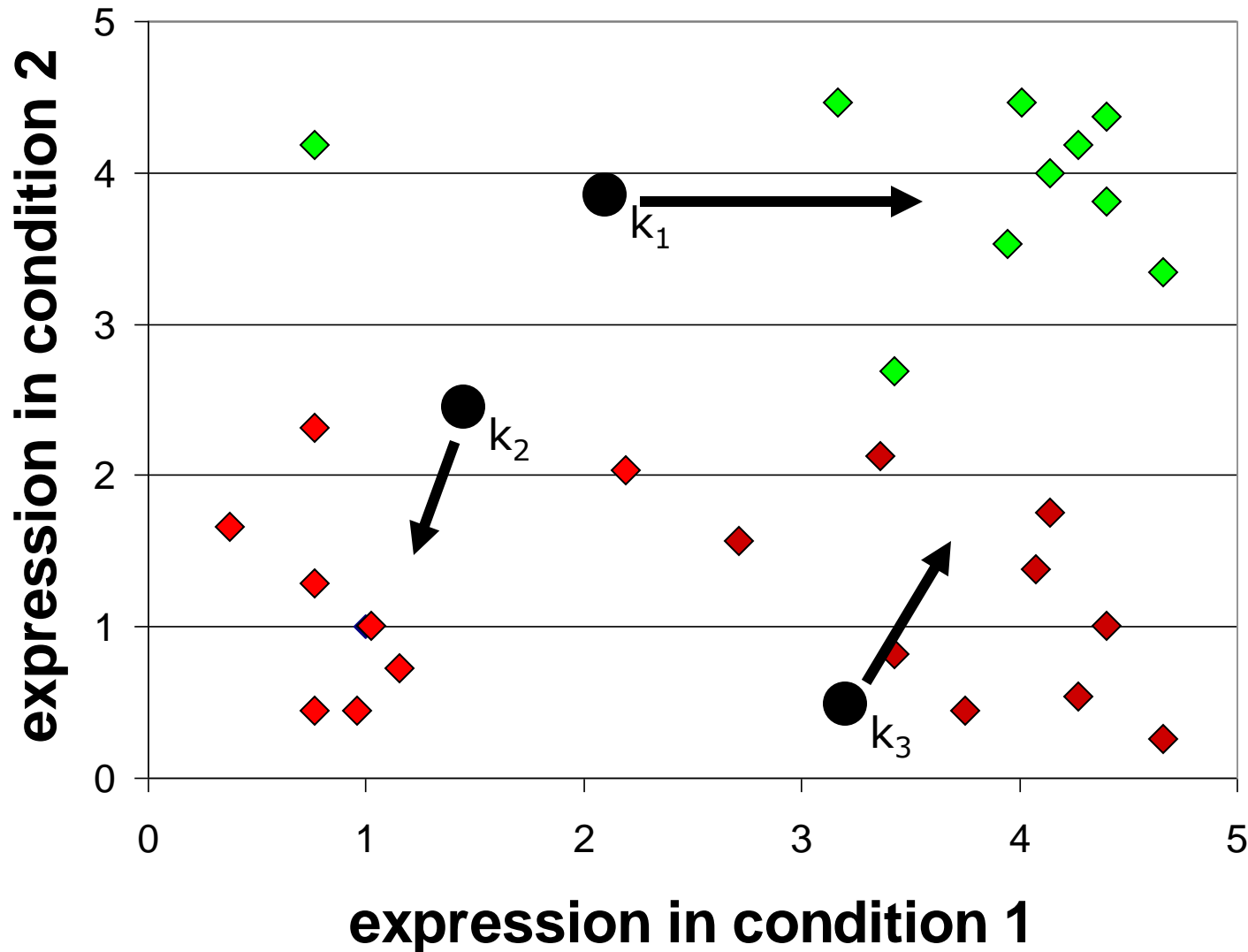
Clustering: Example 2, Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



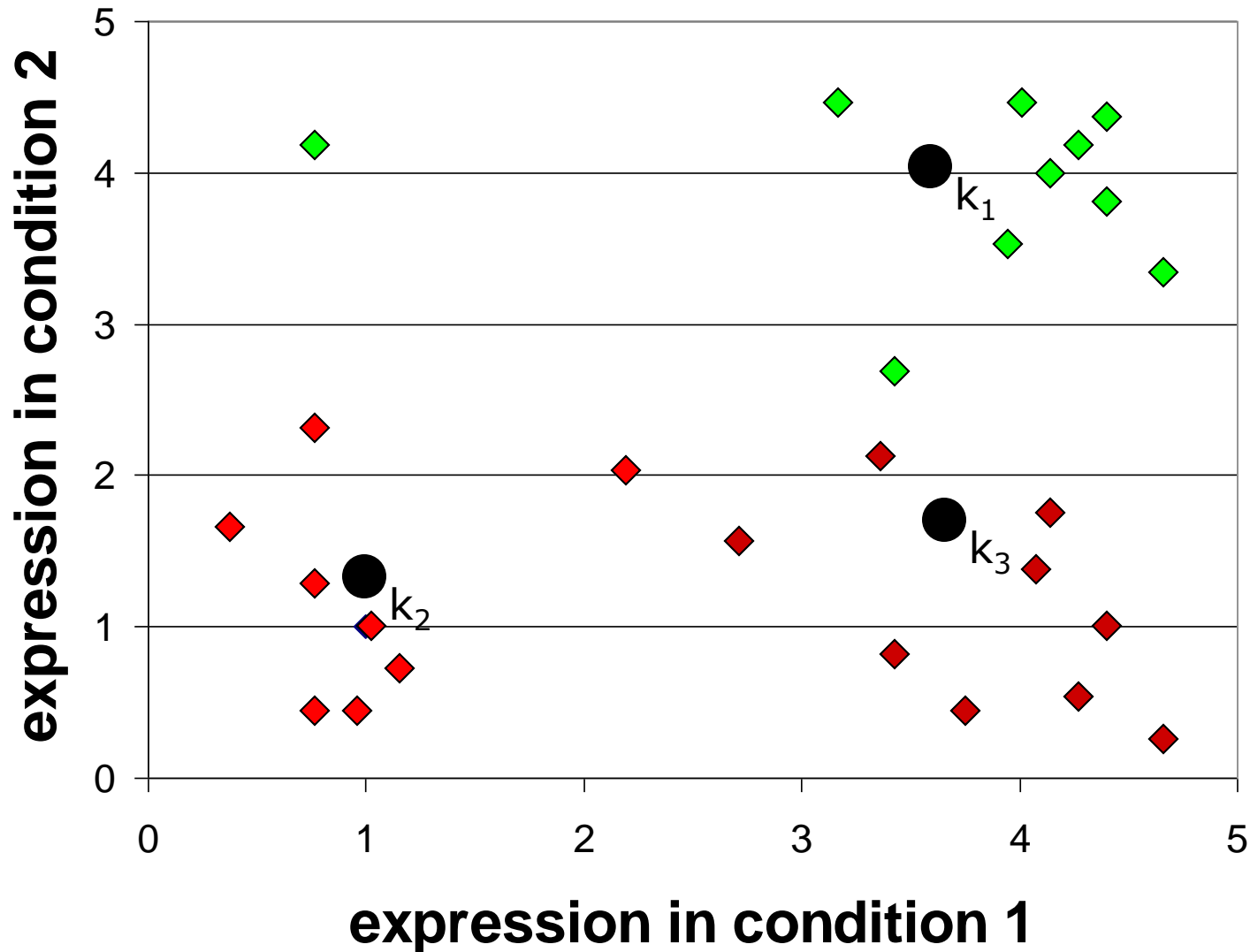
Clustering: Example 2, Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



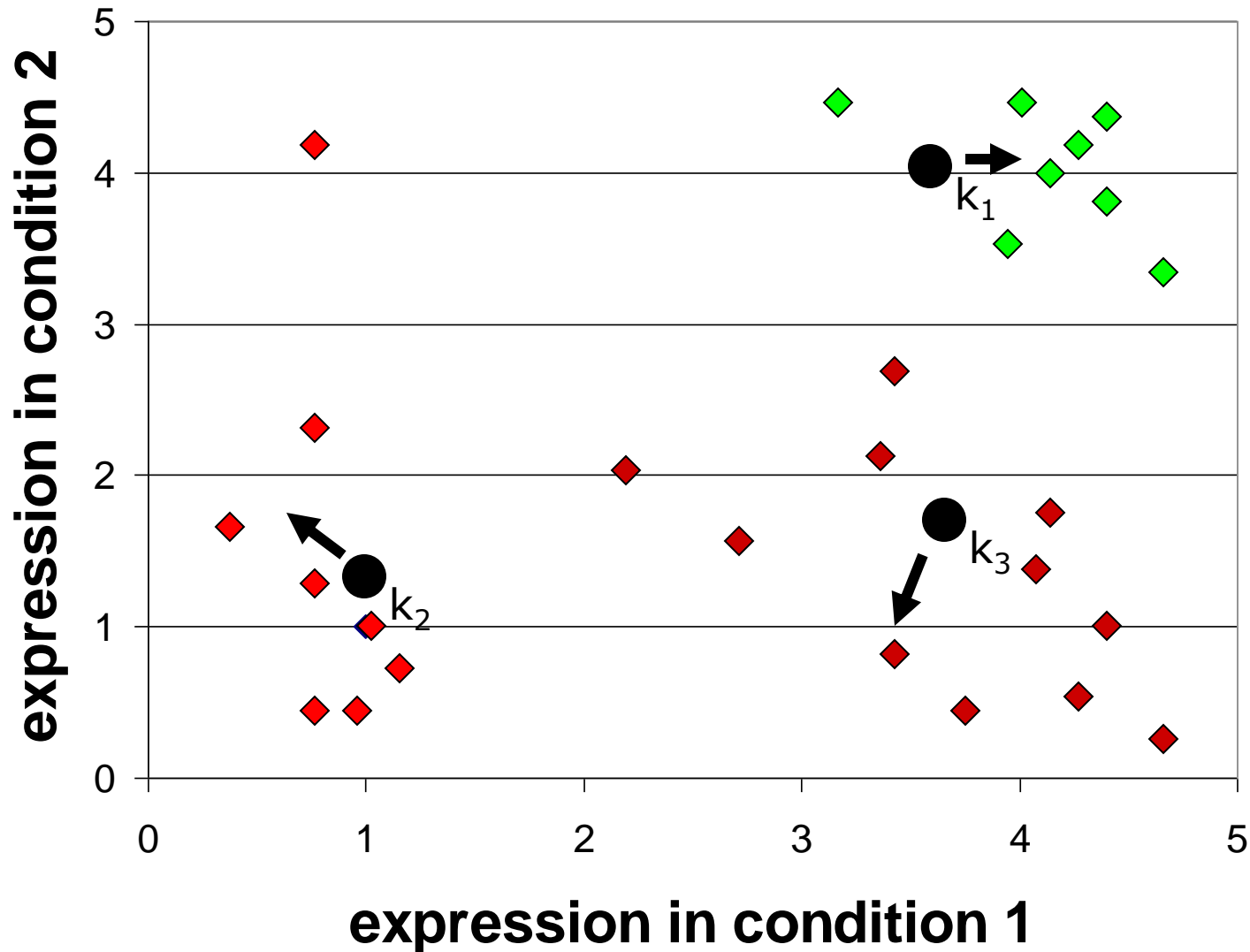
Clustering: Example 2, Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



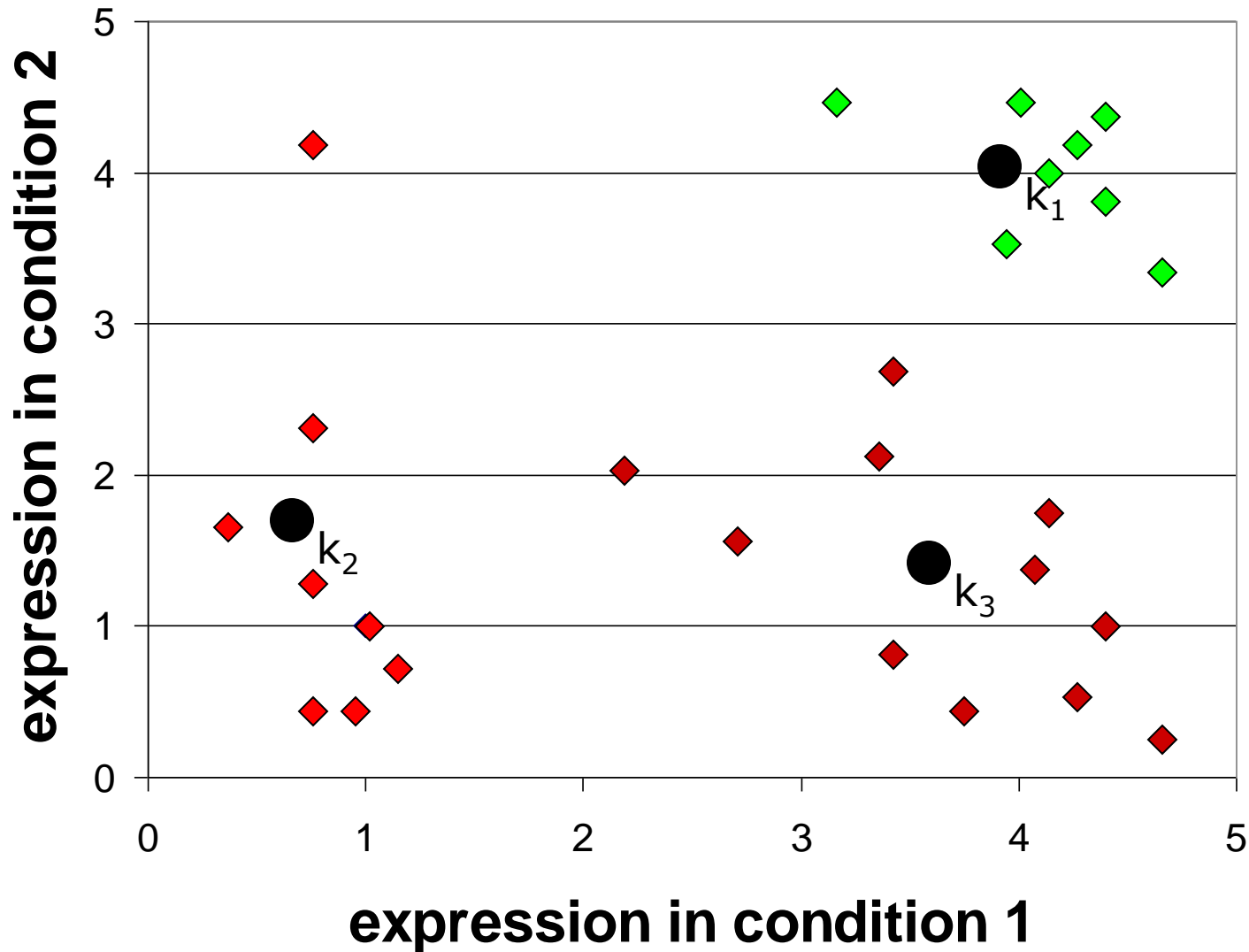
Clustering: Example 2, Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



Clustering: Example 2, Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means variations/parameters

Initial (seed) cluster centers

Convergence

- A fixed number of iterations
- partitions unchanged
- Cluster centers don't change

K!



Example: Color Quantization using K-Means



手写字符识别 MNIST database of handwritten digits

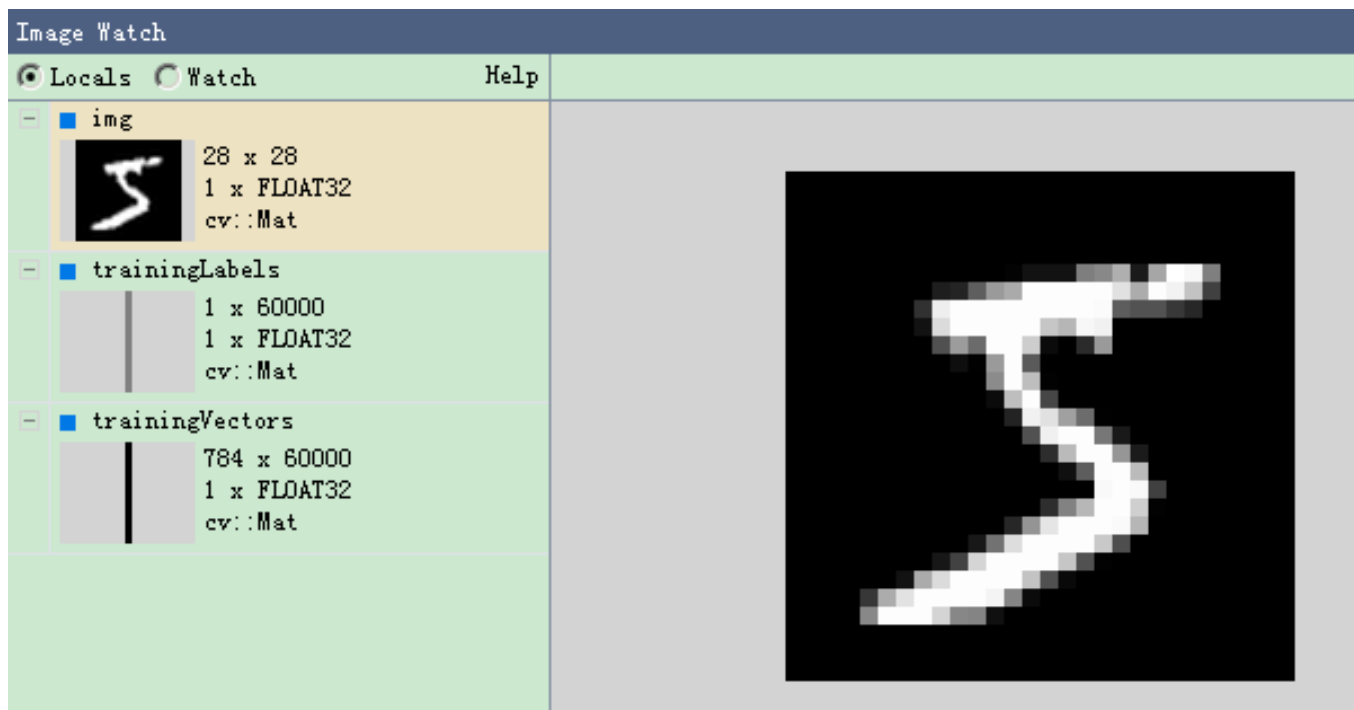
□ <http://yann.lecun.com/exdb/mnist/>

□ 60,000 training

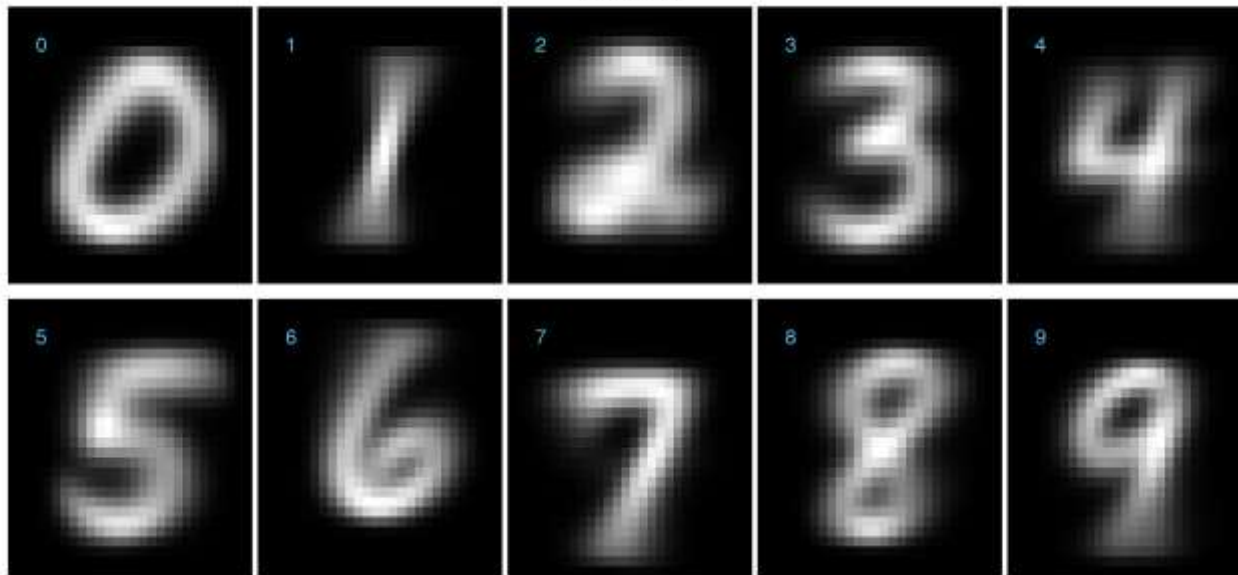
□ 10,000 test



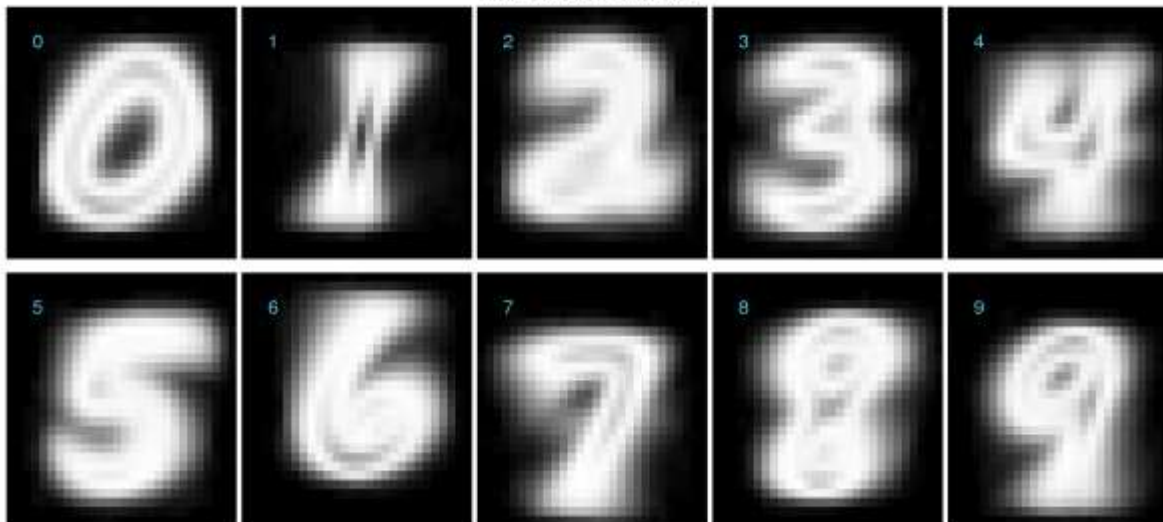
Example in memory



Pixel Means

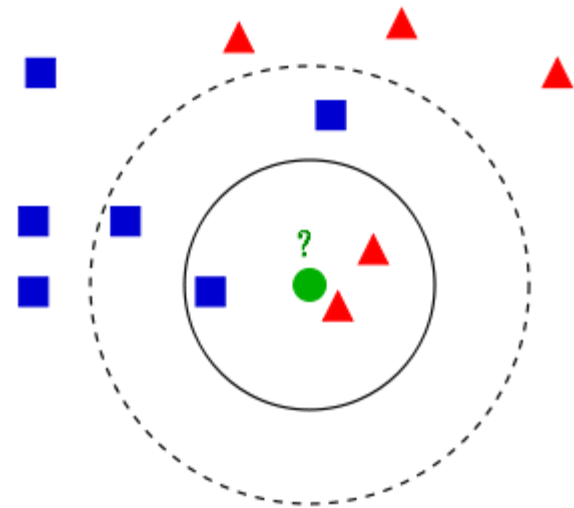


Pixel Standard Deviations



Knn – Grassroots Democracy

- 目标：分类未知类别案例。
- 输入：待分类未知类别案例项目。已知类别案例集合 D ，其中包含 j 个已知类别的案例。
- 输出：项目可能的类别。



Knn in OpenCV3

```
Ptr<ml::KNearest> knn = ml::KNearest::create();  
Ptr<ml::TrainData> trainData =  
ml::TrainData::create(train_features, ml::SampleTypes::ROW_SAMPLE,  
labels);  
knn->train(trainData);
```

```
Mat predictedLabels;  
knn->findNearest(sample, K,  
predictedLabels);
```

```
float prediction = predictedLabels.at<float>(0,0);
```

Knn in OpenCV 2.4

- ☐ `int size = numRows*numCols;`
- ☐ `Mat trainingVectors(numImages, size, CV_32FC1);`
- ☐ `Mat trainingLabels(numImages, 1, CV_32FC1);`

- ☐ `KNearest knn(trainingVectors, trainingLabels);`

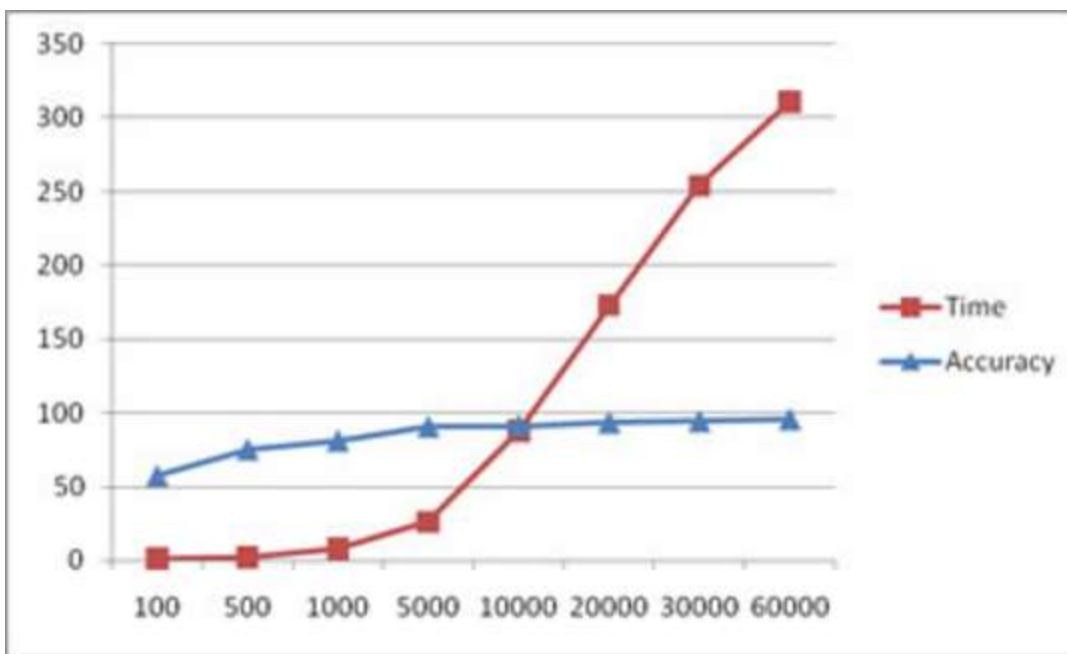
- ☐ `CvMat *currentTest = cvCreateMat(1, size, CV_32FC1);`
- ☐ `CvMat *currentLabel = cvCreateMat(1, 1, CV_32FC1);`
- ☐ `knn.find_nearest(currentTest, 5, currentLabel);`

Results

□ 60000 train + 10000 test 96.88%

□ $K = 5$

Training samples	Time to test 1000 samples (sec)	Memory used (KB)	Accuracy (%)
100	1 s	4,328K	57.70 %
500	2 s	6,800K	75.50 %
1000	8 s	9,868K	81.50 %
5000	26 s	34,480 K	91.00 %
10000	88 s	65, 220 K	91.60 %
20000	173 s	128, 608 K	93.70 %
30000	254 s	188, 192 K	94.70 %
60000	311 s	372, 648 K	96.10 %



Summary of KNN

□ 优缺点:

■ (1) 优点:

算法简单，易于实现，不需要参数估计，不需要事先训练。

■ (2) 缺点:

kNN计算量特别大，而且训练样本必须存储在本机，内存开销也特别大。

□ K的取值:

□ 参数k的取值一般通常不大于20

作业1

■ 1, 论坛帖子 (有奖励)

名称: 手写字符识别资源汇总

- ☐ 汇总: 手写字符识别的数据源
- ☐ 每种数据的读取方法及数据结构

Datasets of recognizing hand-written digits