贾志刚

# 特征和图像识别工程案例实践

# 主要内容

- 数据预处理与特征工程
- 训练模型与调优
- 模型导出与使用
- 案例-猫狗大战

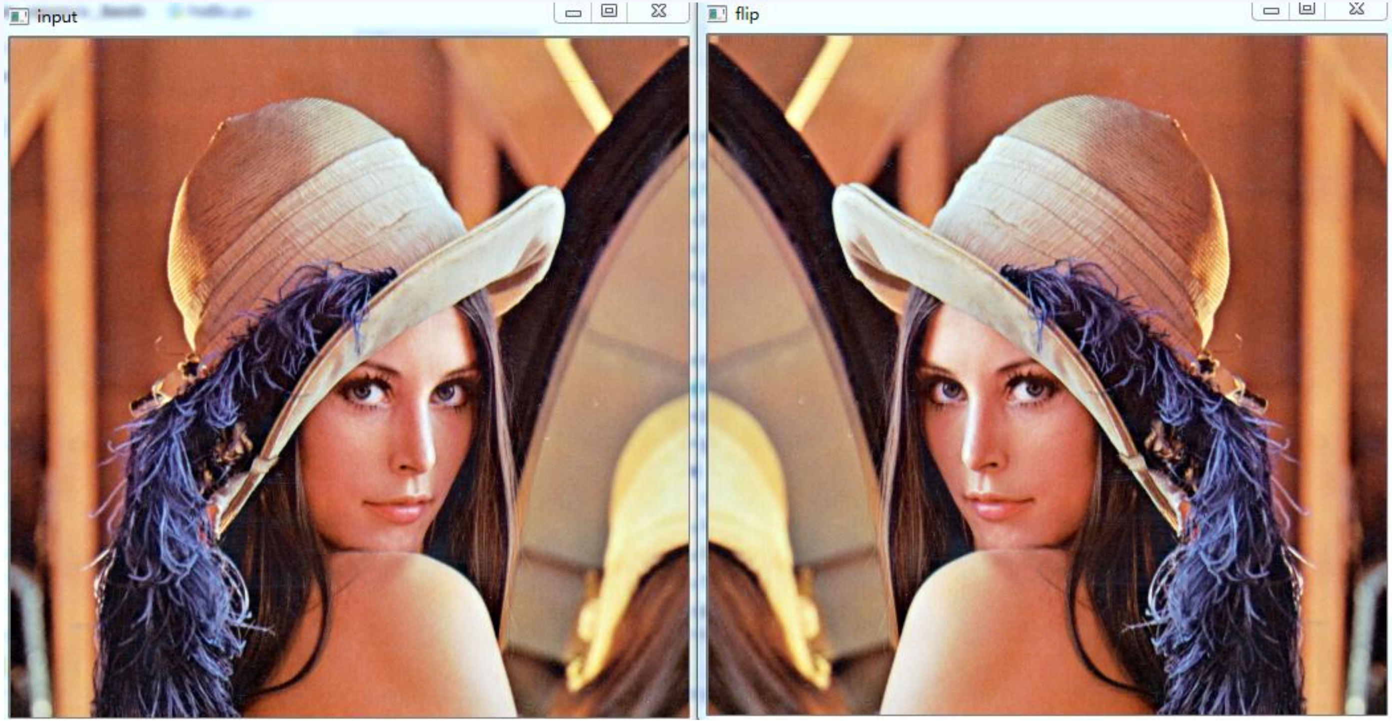
# 数据预处理与特征工程

- 数据增强与标注
- 数据处理
- 特征工程

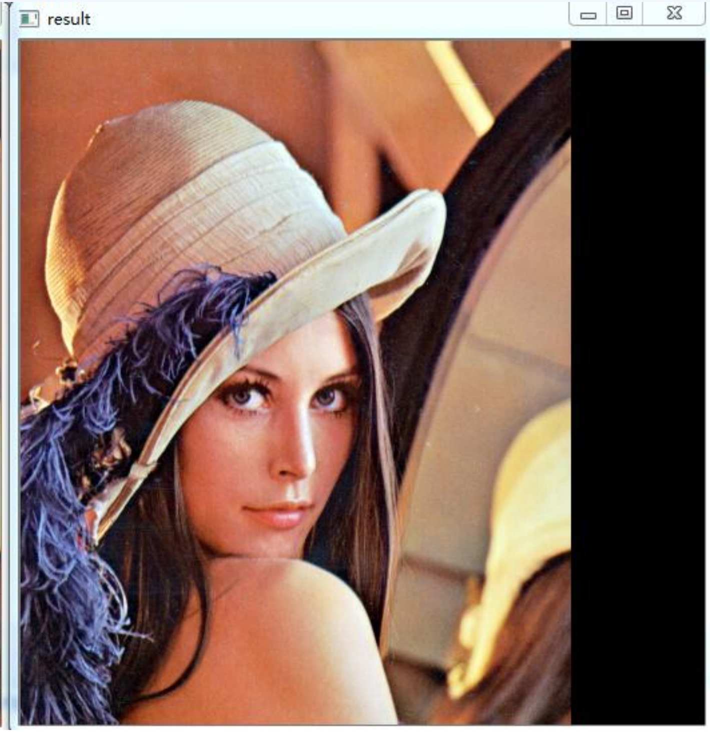
# 图像数据增强与标注

- 图像翻转
- 图像平移
- 图像crop
- 透视变换
- Gabor变换
- 图像数据标注

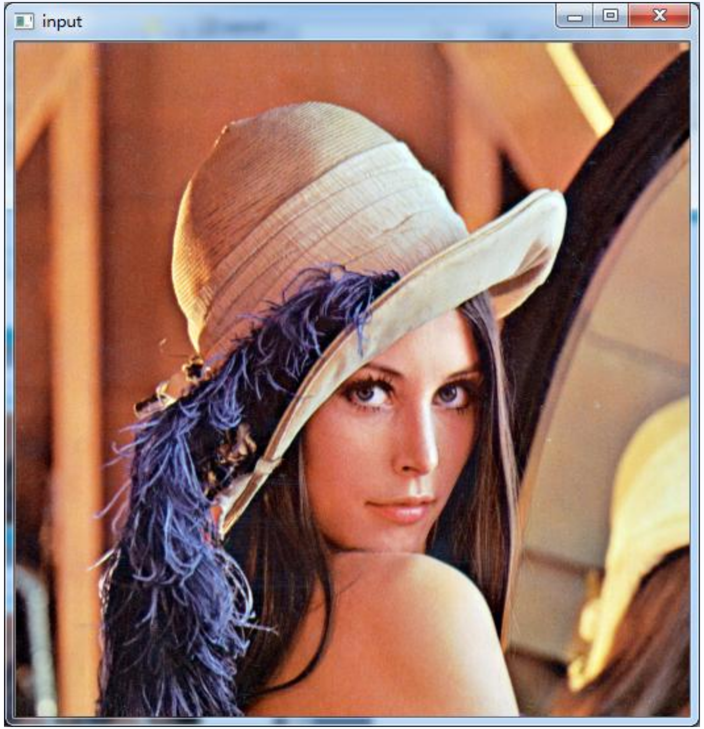
# 图像翻转

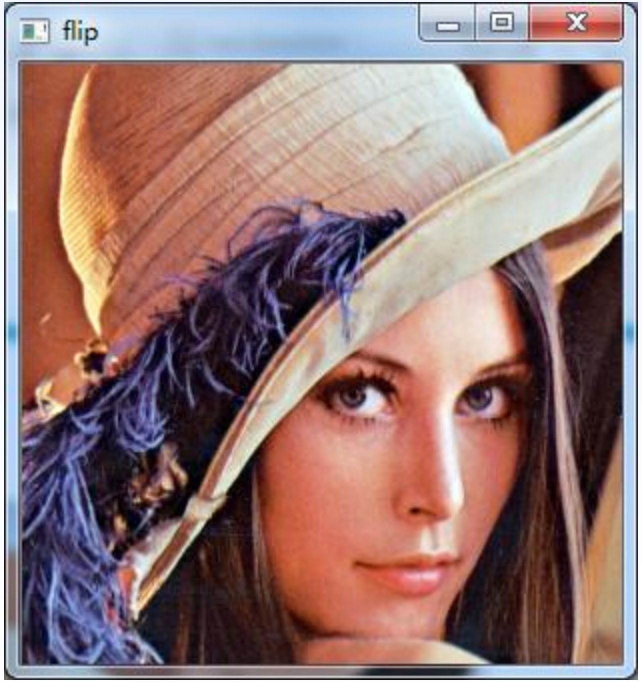


# 图像平移

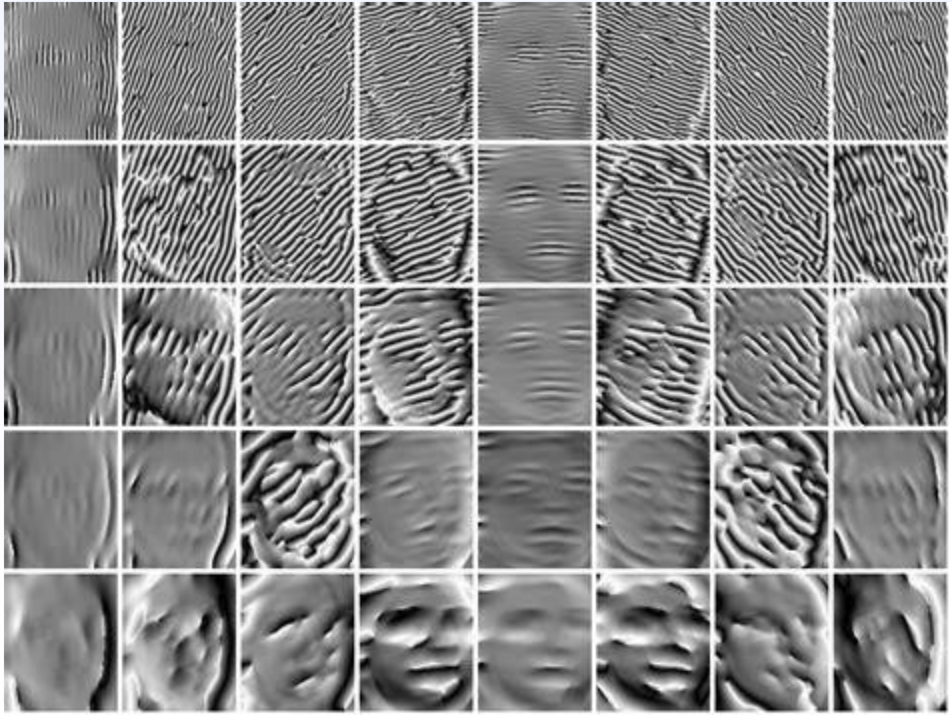
# 图像crop



# 透视变换

# Gabor变换



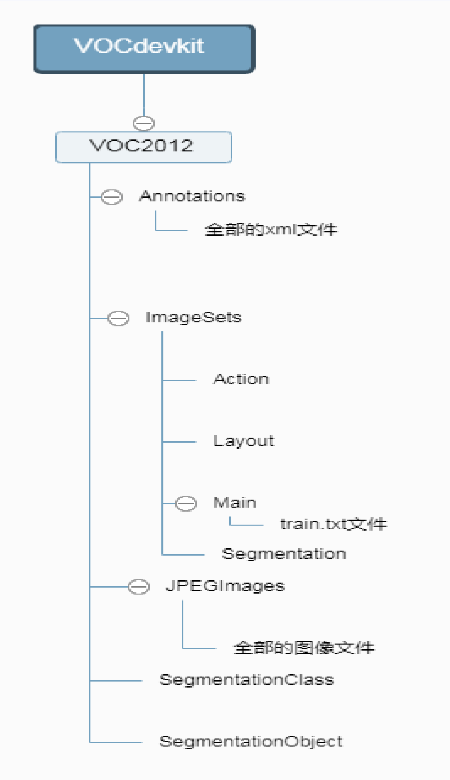
# 更多图像数据增强方法

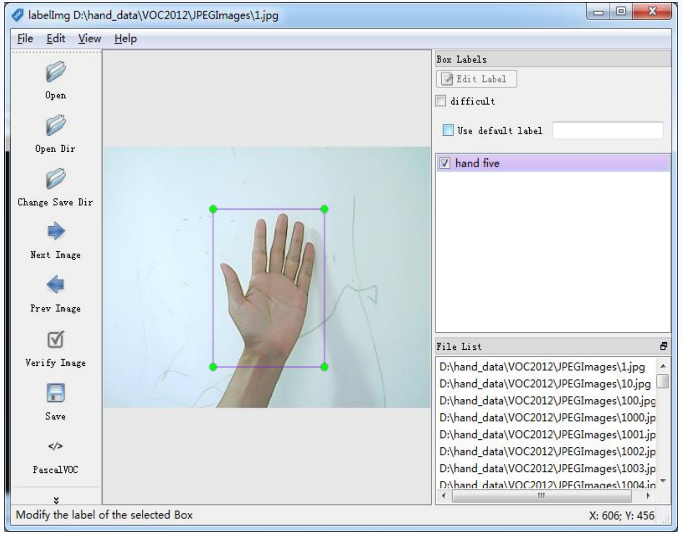
- 亮度调整
- 对比度调整
- 饱和度调整
- 图像采样插值放缩
- 色彩空间转换
- Gamma校正

# 数据标注

- 人工打码平台
- 自动标注技术
- 手工标注工具
- https://github.com/tzutalin/labelImg
- PSCAL VOC2012格式

# PASCAL VOC2012数据格式



# 数据预处理与初始化

- 归一化
- 标准化
- 正则化

# 数据归一化

```
NORM_L1 - L1归一化，依据和为1
// Norm to probability (total count)
// sum(numbers) = 20.0
// 2.0       0.1      (2.0/20.0)
// 8.0       0.4      (8.0/20.0)
// 10.0      0.5      (10.0/20.0)

NORM_L2 ，  单位向量为1
// Norm to unit vector: ||positiveData|| = 1.0
// 2.0       0.15
// 8.0       0.62
// 10.0      0.77

NORM_INF，  根据最大值
// Norm to max element
// 2.0       0.2      (2.0/10.0)
// 8.0       0.8      (8.0/10.0)
// 10.0      1.0      (10.0/10.0)

NORM_MINMAX，  根据delta = max-min
// Norm to range [0.0;1.0]
// 2.0       0.0      (shift to left border)
// 8.0       0.75     (6.0/8.0)
// 10.0      1.0      (shift to right border)
```

$norm = \frac{x_i - \min(x)}{\max(x) - \min(x)}$ 其中 $x_i$ 表示图像像素点值，$\min(x), \max(x)$ 分别表示图像像素的最大与最小值

# 数据归一化

```python
# 转换为浮点数类型数组
gray = cv.cvtColor(src, cv.COLOR_BGR2GRAY)
gray = np.float32(gray)
print(gray)

# scale and shift by NORM_MINMAX
dst = np.zeros(gray.shape, dtype=np.float32)
cv.normalize(gray, dst=dst, alpha=0, beta=1.0, norm_type=cv.NORM_MINMAX)
print(dst)
cv.imshow("NORM_MINMAX", np.uint8(dst*255))

# scale and shift by NORM_INF
dst = np.zeros(gray.shape, dtype=np.float32)
cv.normalize(gray, dst=dst, alpha=1.0, beta=0, norm_type=cv.NORM_INF)
print(dst)
cv.imshow("NORM_INF", np.uint8(dst*255))
```

# 标准化

$$\text{image\_standardization} = \frac{x - \mu}{\text{adjusted\_stddev}}$$ 其中$\mu$是图像的均值，$X$表示图像矩阵

其中

$$\text{adjusted\_stddev} = \max(\sigma, \frac{1.0}{\sqrt{N}})$$，$\sigma$表示标准方差、N表示图像X的像素数目


# 正则化

$$\min_{\mathbf{w}} \quad \frac{1}{n}\sum_{i=1}^{n}\ell(y, f_{\mathbf{w}}(x)) \ + \ \lambda \ \text{count}\{w_j \neq 0\}$$

L0正则化

$$\min_{\mathbf{w}} \quad \frac{1}{n}\sum_{i=1}^{n}\ell(y, f_{\mathbf{w}}(x)) \ + \ \lambda|\mathbf{w}|_1$$

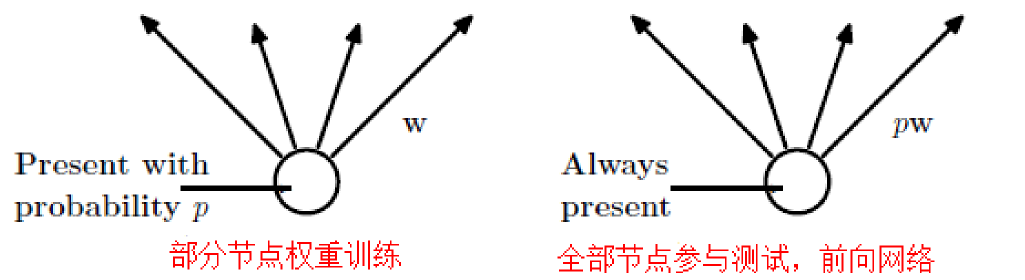
L1正则化

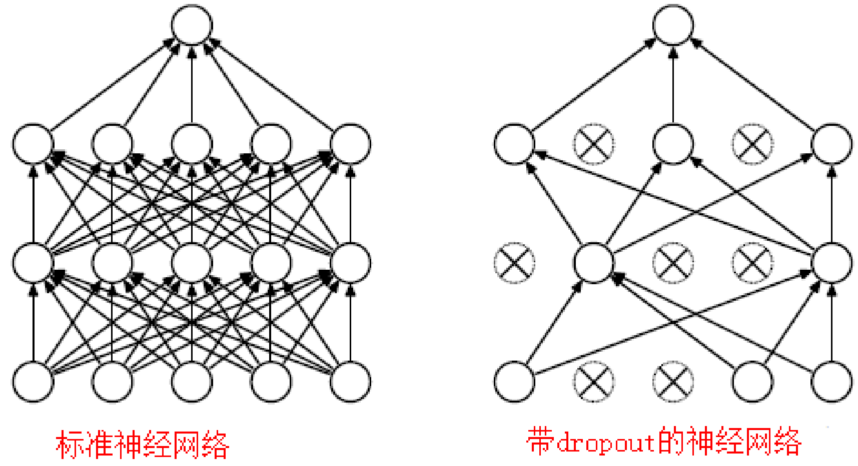
$$\min_{\mathbf{w}} \quad \frac{1}{n}\sum_{i=1}^{n}\ell(y, f_{\mathbf{w}}(x)) \ + \ \lambda\|\mathbf{w}\|_2$$

L2正则化

# dropout正则化

- 神经网络（多层感知器MLP）
- dropout




作用：防止过拟合，增加了神经网络的训练层数

# 特征工程

- 原始特征
  - 图像像素、轮廓、纹理等
  - 信号采样、光谱
  - 时间序列
  - 生物数据-DNA、Marker、Gene
  - 文本数据-词组、语法、关系
- 组合特征
  - 线性特征组合

# 特征工程

- Hessian矩阵与角点检测
- SIFT/SURF图像特征检测不变性问题
- 自动特征提取

# Hessian矩阵求解

高斯模糊与高斯梯度求解

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

X方向偏导数 Gx

$$\frac{\partial G(x,y,\sigma)}{\partial x} = -\frac{x}{2\pi\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Gxx --

$$\frac{\partial^2 G(x,y,\sigma)}{\partial^2 x} = \left(-1 + \frac{x^2}{\sigma^2}\right) \frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4}$$

Gyy --

$$\frac{\partial^2 G(x,y,\sigma)}{\partial^2 y} = \left(-1 + \frac{y^2}{\sigma^2}\right) \frac{e^{-\frac{(x^2+y^2)}{2\sigma^2}}}{2\pi\sigma^4}$$

Gxy --

$$\frac{\partial^2 G(x,y,\sigma)}{\partial xy} = \frac{xy}{2\pi\sigma^6} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

# Hessian矩阵求解

Hessian求解需要计算图像的二阶偏导数 $D_{xx}$ $D_{xy}$ $D_{yy}$
基于高斯核计算

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$f(x, y) = x^3 - 2xy - y^6$ at the point $(1, 2)$:

$$f_x(x, y) = \frac{\partial}{\partial x}(x^3 - 2xy - y^6) = 3x^2 - 2y$$

$$f_y(x, y) = \frac{\partial}{\partial y}(x^3 - 2xy - y^6) = -2x - 6y^5$$

# Hessian矩阵求解

$$f_{xx}(x,y) = \frac{\partial}{\partial x}(3x^2 - 2y) = 6x$$

$$f_{xy}(x,y) = \frac{\partial}{\partial y}(3x^2 - 2y) = -2$$

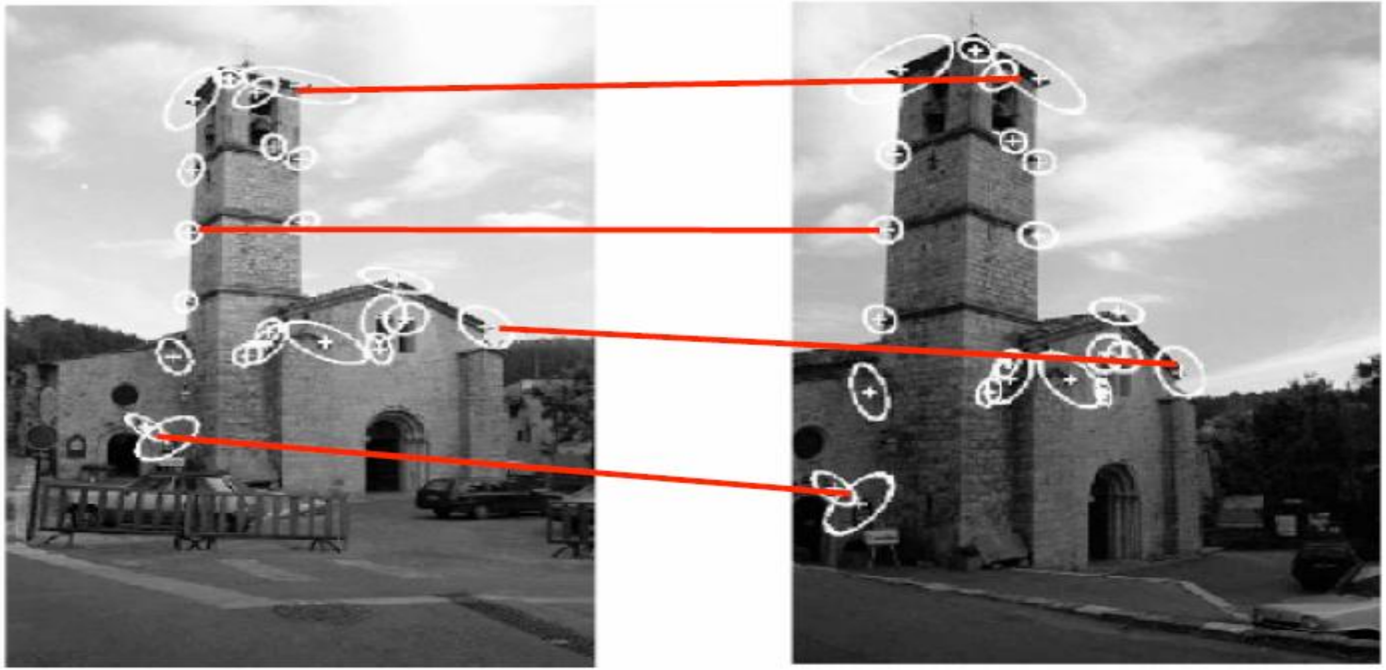
$$f_{yx}(x,y) = \frac{\partial}{\partial x}(-2x - 6y^5) = -2$$

$$f_{yy}(x,y) = \frac{\partial}{\partial y}(-2x - 6y^5) = -30y^4$$

$$\mathbf{H}f(x,y) = \begin{bmatrix} f_{xx}(x,y) & f_{xy}(x,y) \\ f_{yx}(x,y) & f_{yy}(x,y) \end{bmatrix} = \begin{bmatrix} 6x & -2 \\ -2 & -30y^4 \end{bmatrix}$$

$$\mathbf{H}f(1,2) = \begin{bmatrix} 6(1) & -2 \\ -2 & -30(2)^4 \end{bmatrix} = \begin{bmatrix} 6 & -2 \\ -2 & -480 \end{bmatrix}$$

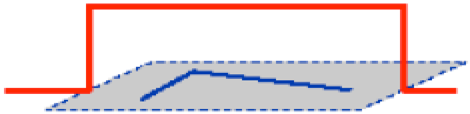
# Harris角点检测器



# Harris角点检测器

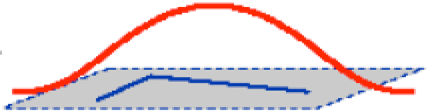
$$E(u,v)=\sum_{x,y}\underbrace{w(x,y)}_{\text{Window function}}\Big[\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}}-\underbrace{I(x,y)}_{\text{Intensity}}\Big]^2$$

Window function $w(x,y) =$



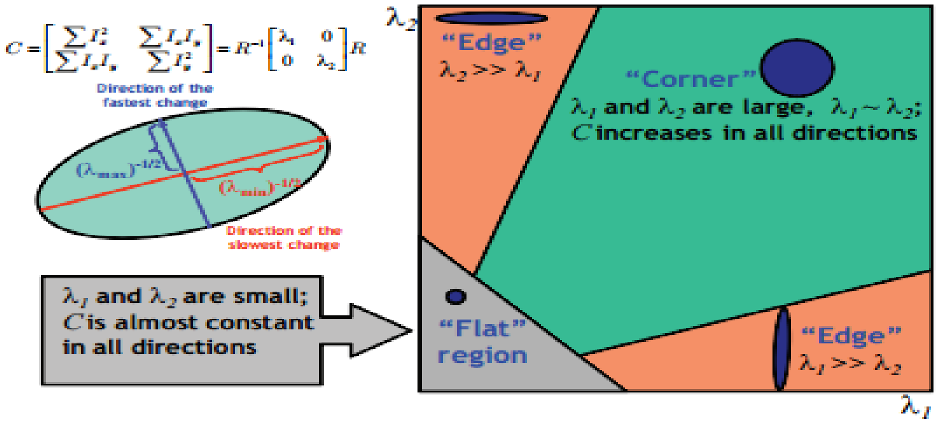
1 in window, 0 outside

or



Gaussian

# 图像Hessian特征




$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

k=0.04~0.06

# Harris角点检测器

矩阵的特征值与特征向量

$$A = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} \qquad \det(A - \lambda I)$$

$$A - \lambda I = \begin{bmatrix} 3 & 6 & -8 \\ 0 & 0 & 6 \\ 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 3-\lambda & 6 & -8 \\ 0 & -\lambda & 6 \\ 0 & 0 & 2-\lambda \end{bmatrix}$$

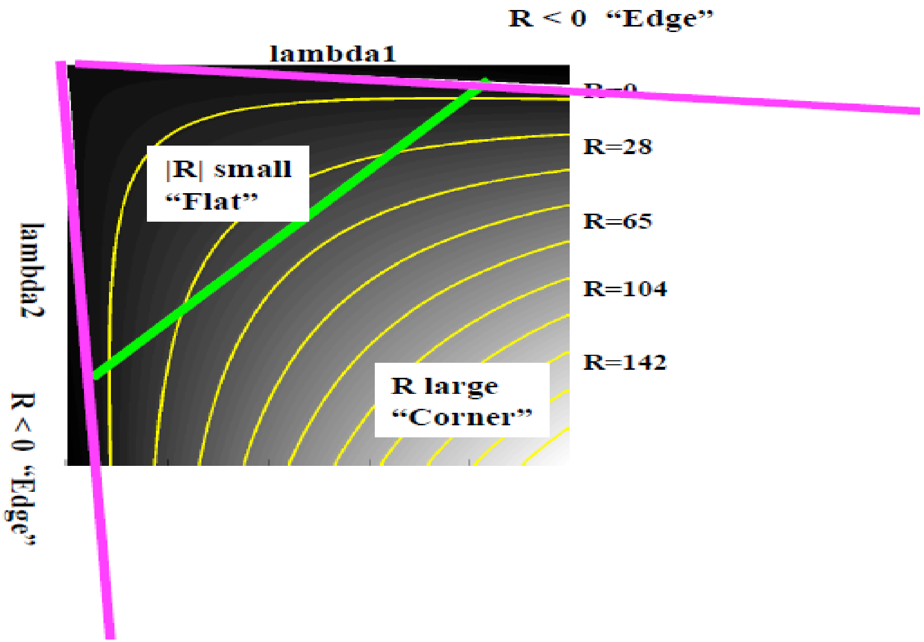
$$\begin{aligned} \det(A - \lambda I) &= (3-\lambda)(-\lambda)(2-\lambda) + (6)(6)(0) + (-8)(0)(0) \\ &\quad - (0)(-\lambda)(-8) - (0)(6)(3-\lambda) - (-\lambda)(0)(6) \\ &= -\lambda(3-\lambda)(2-\lambda) \end{aligned}$$

特征值有三个 $\lambda = 0,\ 2,\ \text{or}\ 3$

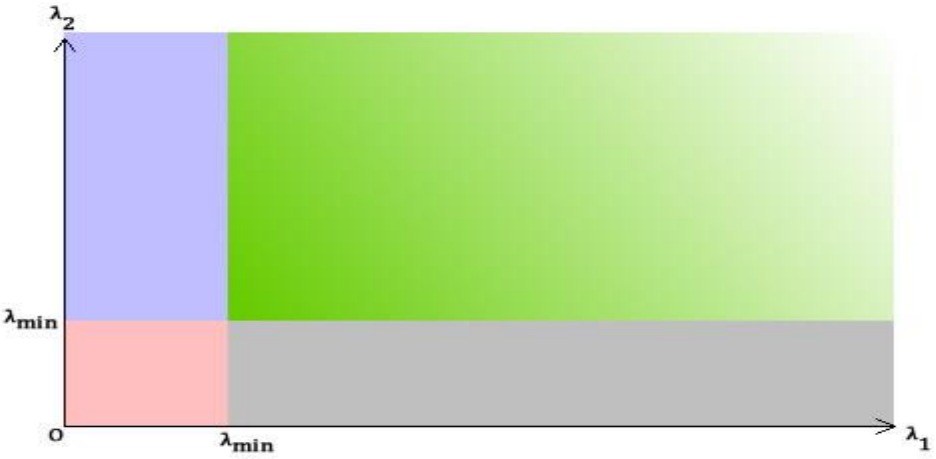
**Eigenvalues of the Hessian matrix and image structure orientation (L low, H+ high positive, H- high negative)**

| $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | structure orientation |
|---|---|---|---|
| L | L | L | noise (no preferred structure) |
| L | L | H– | bright sheet-like structure |
| L | L | H+ | dark sheet-like structure |
| L | H– | H– | bright tubular structure |
| L | H+ | H+ | dark tubular structure |
| H– | H– | H– | bright blob-like structure |
| H+ | H+ | H+ | dark blob-like structure |

# Harris角点检测器

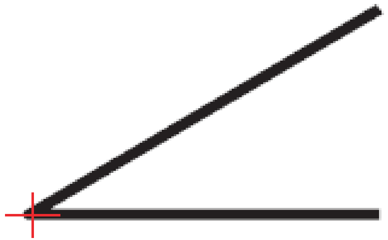


# Shi-Tomas角点检测




$$R = \min(\lambda_1, \lambda_2)$$

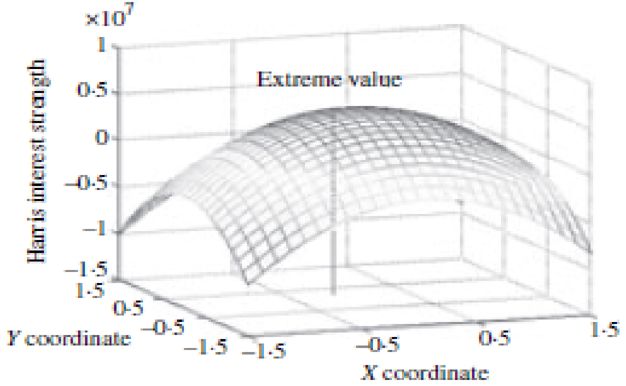
# 亚像素级别角点检测



(a)

| | | |
|---|---|---|
| 216244 | 696907 | 1495135 |
| 397338 | 4043951 | 1095007 |
| 167681 | 1211299 | 1246681 |

(b)




(c)

# 亚像素级别角点检测

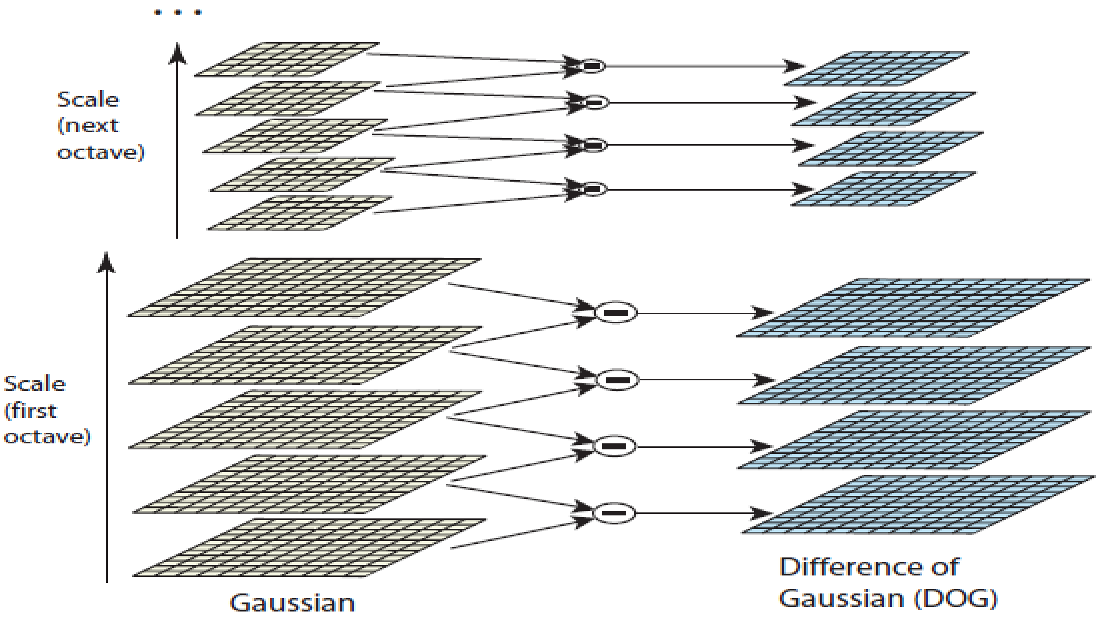
- 插值方法
- 基于图像矩计算
- 曲线拟合方法 -（高斯曲面、多项式、椭圆曲面）

# SIFT特征提取

SIFT(**S**cale-**I**nvariant **F**eature **T**ransform)特征检测关键特性：

- **建立尺度空间，寻找极值**
- **关键点定位(寻找关键点准确位置与删除弱边缘)**
- **关键点方向指定**
- **关键点描述子**

# SIFT-金字塔尺度空间

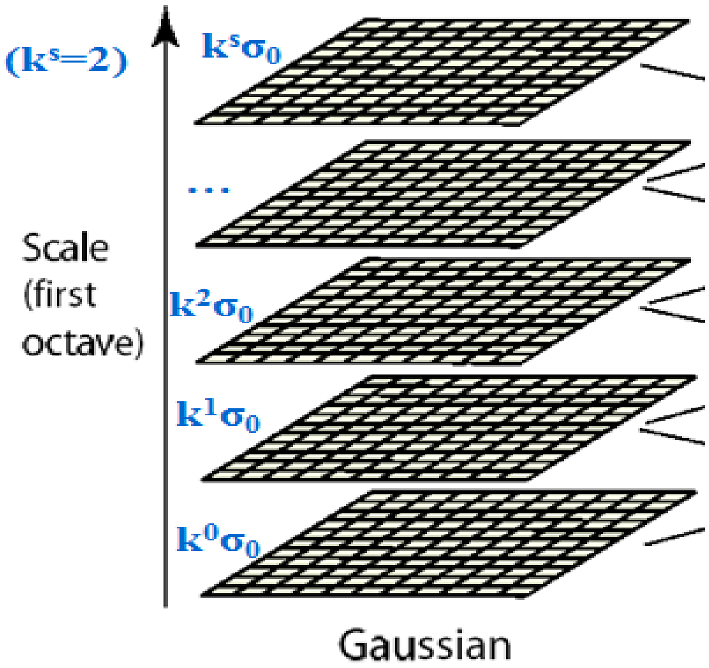


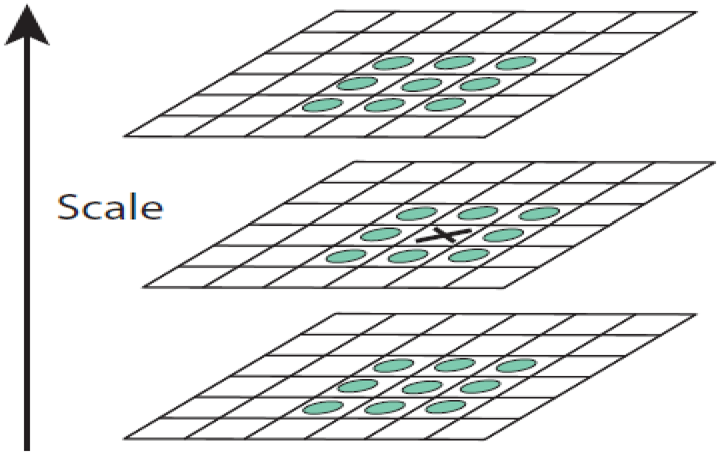

$$
\begin{aligned}
D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\
&= L(x,y,k\sigma) - L(x,y,\sigma).
\end{aligned}
$$

**工作原理**

1. 构建图像高斯金字塔，求取DOG，发现最大与最小值在每一级

2. 构建的高斯金字塔，每一层根据sigma的值不同，可以分为几个等级，最少有4个。

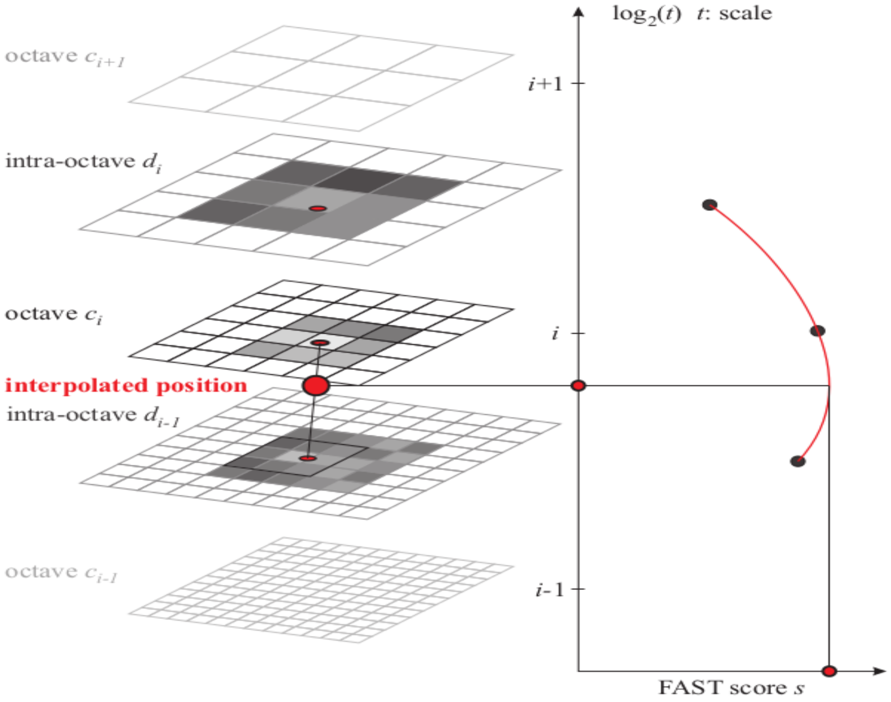

$(k^s=2)$
$k^s\sigma_0$
...
Scale
(first
octave)
$k^2\sigma_0$
$k^1\sigma_0$
$k^0\sigma_0$
Gaussian


Scale

# 关键点定位

- 我们在像素级别获得了极值点的位置，但是更准确的值应该在亚像素位置，如何得到 – 这个过程称为关键点(准确/精准)定位。
- 删除弱边缘- 通过Hessian 矩阵特征值实现，小于阈值自动舍弃

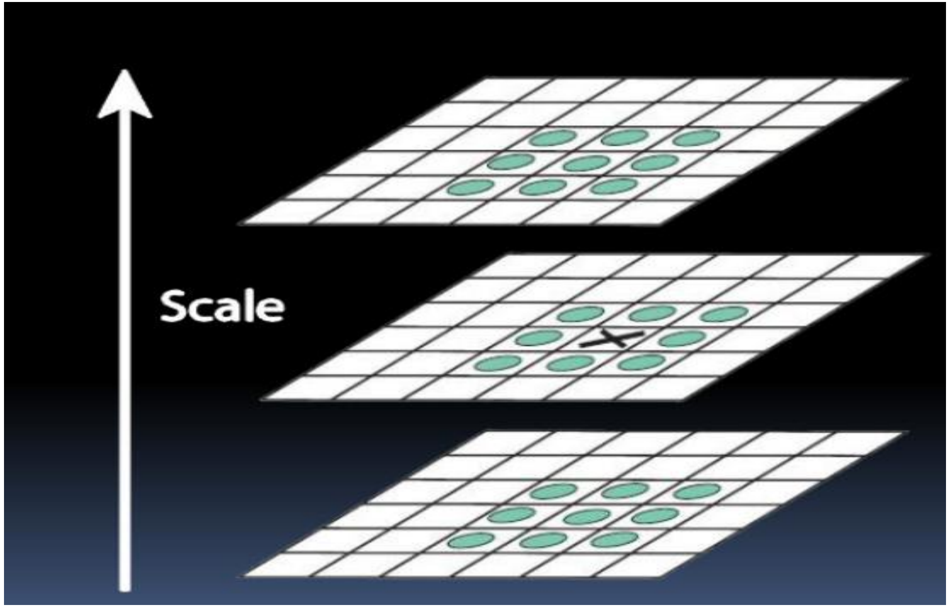
# SIFT-关键点定位




$$H(\mathbf{x}) = H + \frac{\partial H}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 H}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\hat{x} = -\frac{\partial^2 H}{\partial \mathbf{x}^2}^{-1} \frac{\partial H}{\partial \mathbf{x}}$$

# SIFT-关键点定位




$$\frac{\partial^2 H}{\partial \mathbf{x}^2} = \begin{bmatrix} d_{xx} & d_{yx} & d_{sx} \\ d_{xy} & d_{yy} & d_{sy} \\ d_{xs} & d_{ys} & d_{ss} \end{bmatrix}$$

$$\frac{\partial H}{\partial \mathbf{x}} = \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix}.$$

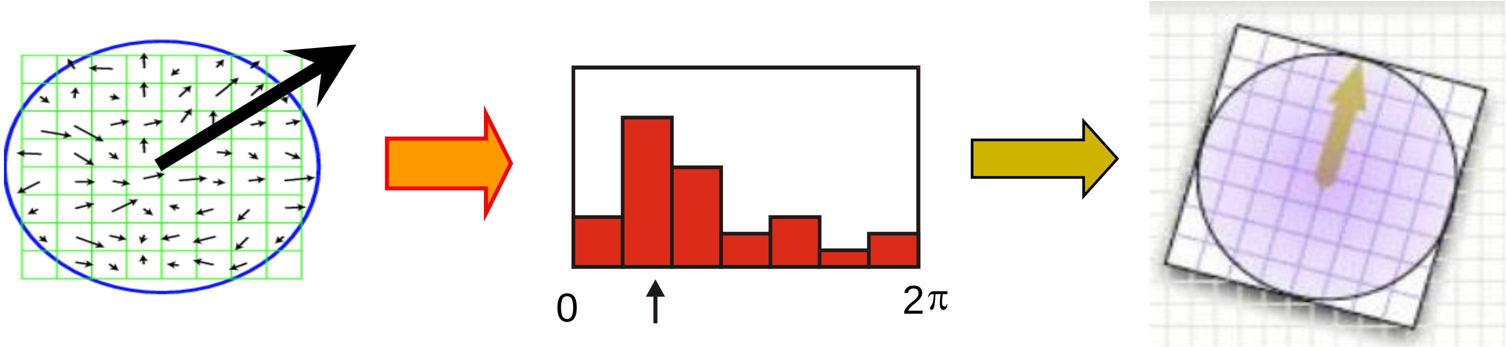
# SIFT-角度方向指定

- 求得每一层对应图像的梯度，根据给定的窗口大小
- 计算每个高斯权重，sigma=scalex1.5，0~360之间建立36个直方图Bins
- 找最高峰对应的Bin， 大于max*80% 的都保留
- 这样就实现了旋转不变性，提高了匹配时候的稳定性。
- 大约有15%的关键点会有多个方向。

# SIFT-角度方向指定

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$

$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$



# SIFT-局部图像描述子

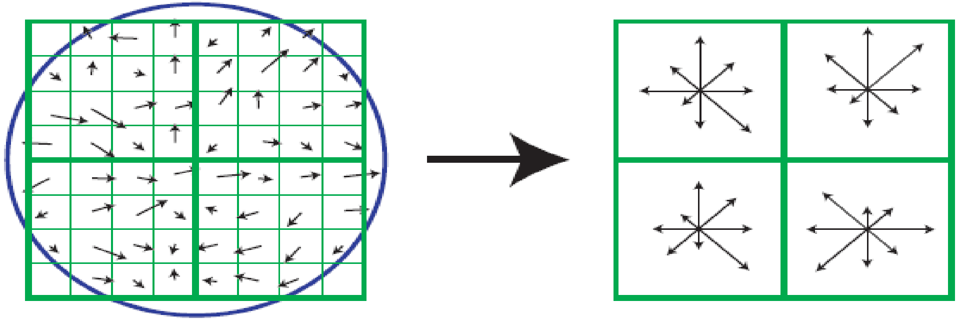
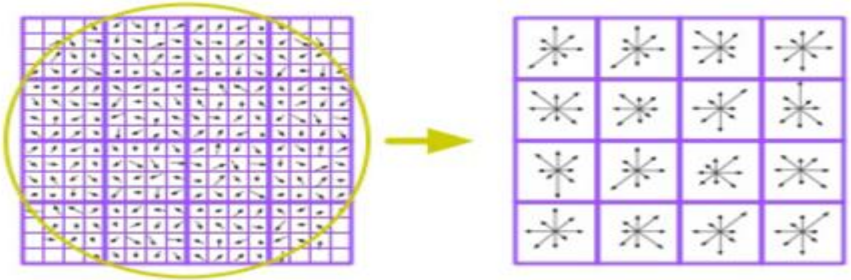


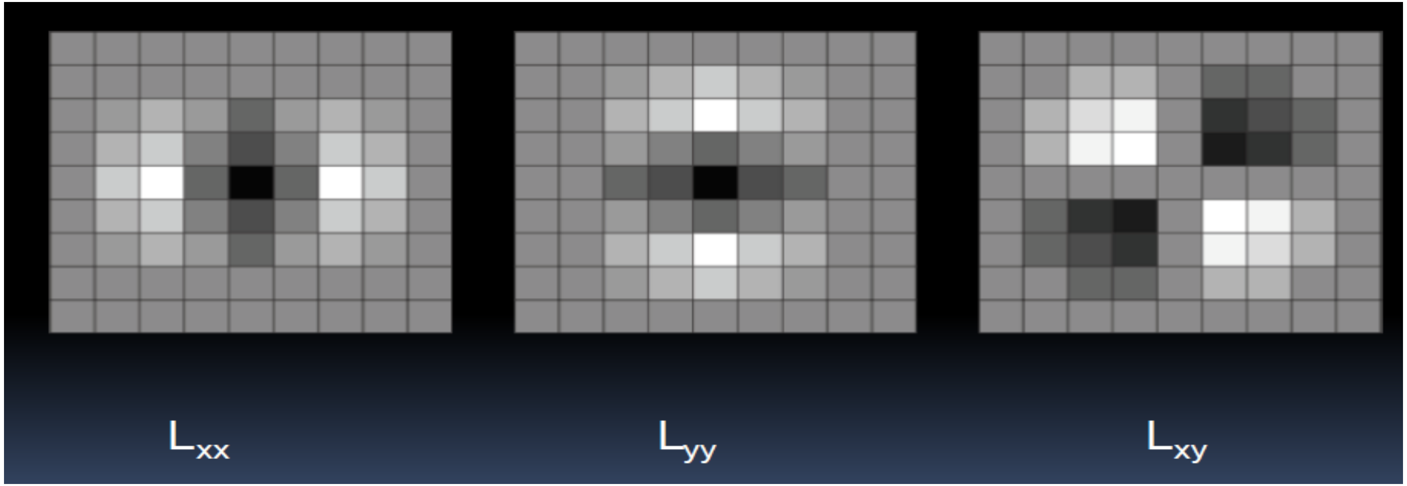
Image gradients　　Keypoint descriptor



16x16 局部图像区域　　4x4x8=128个描述子

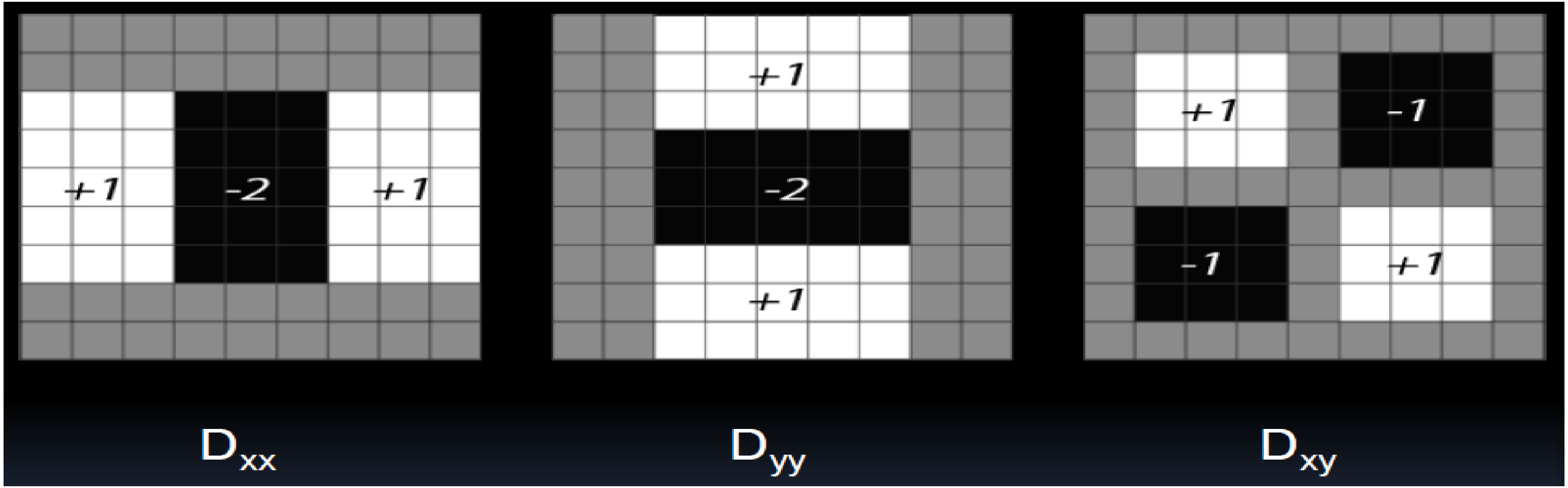
# SIFT-角度方向指定

- 选择图像中POI（Points of Interest） Hessian Matrix
- 在不同的尺度空间发现关键点，非最大信号压制
- 发现特征点方法、旋转不变性要求
- 生成特征向量

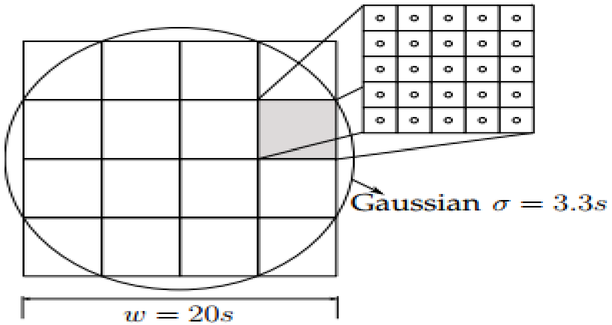



$$\begin{bmatrix} \dfrac{\partial I^2}{\partial x^2} & \dfrac{\partial I^2}{\partial x \partial y} \\ \dfrac{\partial I^2}{\partial y \partial x} & \dfrac{\partial I^2}{\partial y^2} \end{bmatrix}$$

# SURF-HAAR特征-积分图计算



# SURF-描述子





$$v = \{\sum dx, \sum |dx|, \sum dy, \sum |dy|\}$$

2x2 Haar，对于5x5的区域得到一个向量V， 总计16个向量

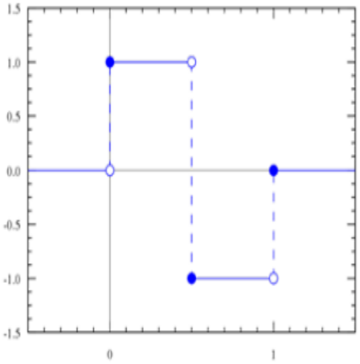
# 更多图像特征

- HOG特征描述
- LBP、HAAR特征描述
- ORB/BRISK/KAZE

# HAAR特征

- 高类间变异性
- 低类内变异性
- 局部强度差
- 不同尺度
- 计算效率高

# HAAR特征

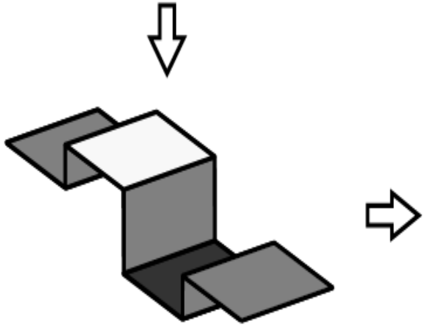


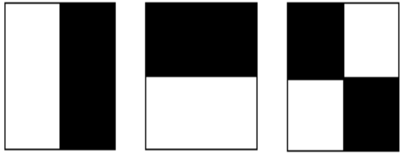

**haar小波函数**

$$\psi(t)=\begin{cases}1 & 0\le t<1/2\\ -1 & 1/2\le t<1\\ 0 & \text{otherwise}\end{cases}$$

**尺度函数**

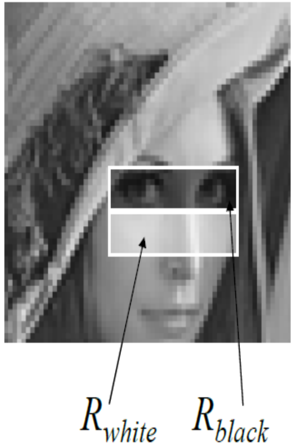
$$\phi(t)=\begin{cases}1 & 0\le t<1\\ 0 & \text{otherwise}\end{cases}$$

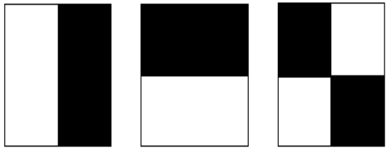




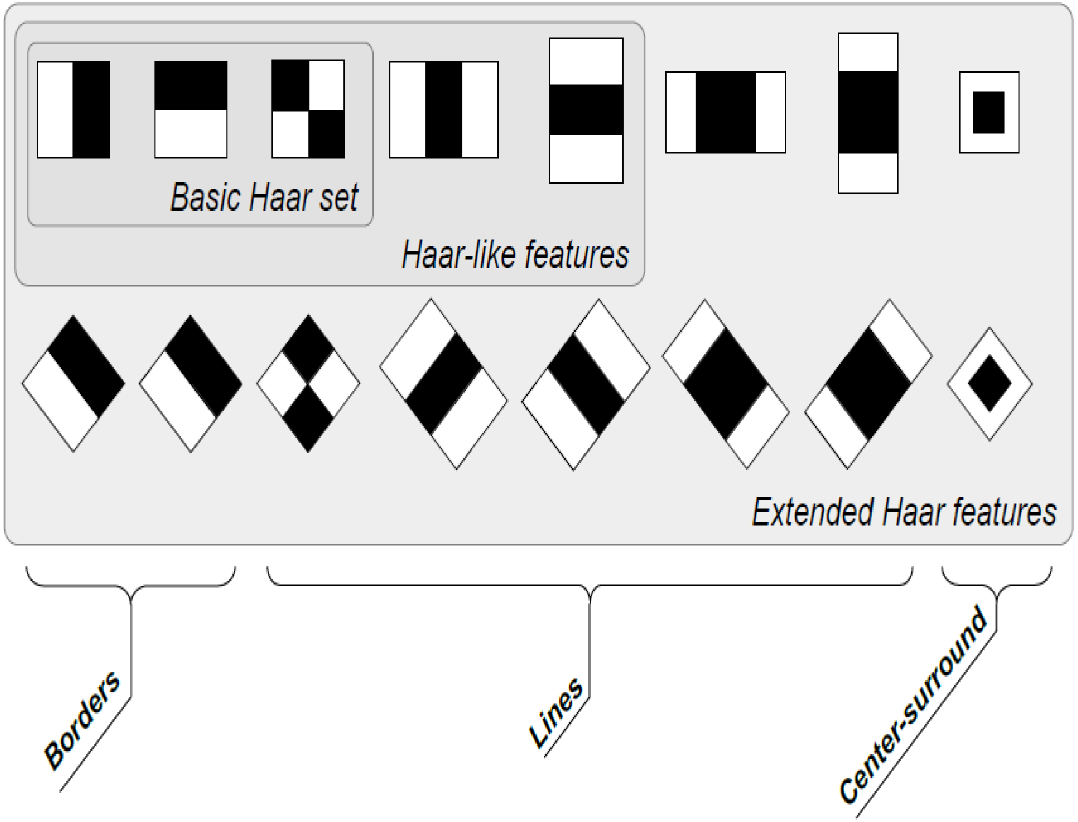
$\sim \partial x$ $\sim \partial y$ $\sim \partial x \partial y$

## Haar特征介绍（Haar Like Features）

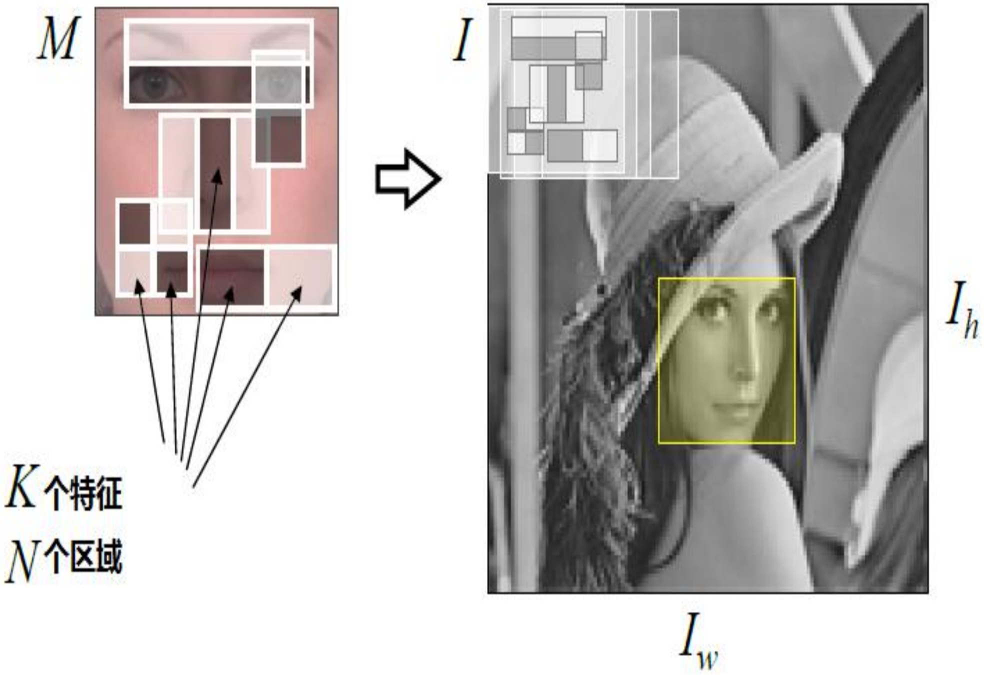


$$F_{Haar} = E(R_{white}) - E(R_{black})$$



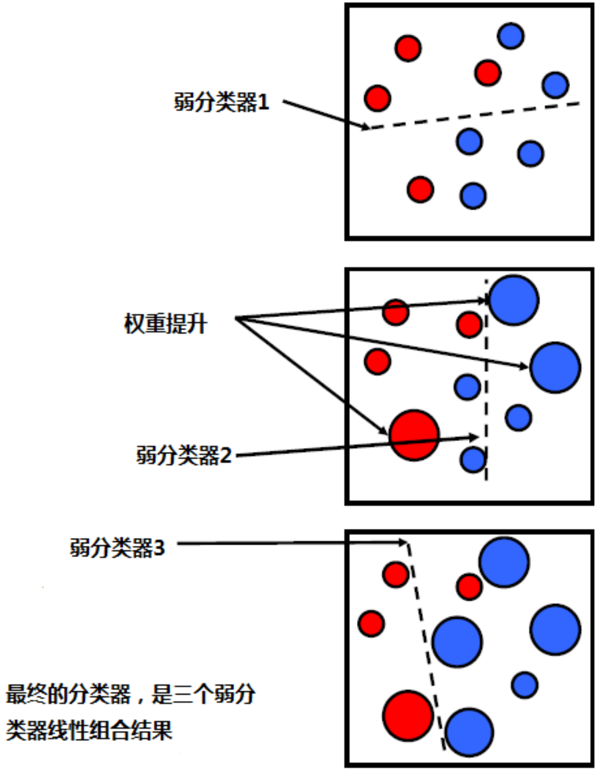
## Haar特征介绍人脸检测

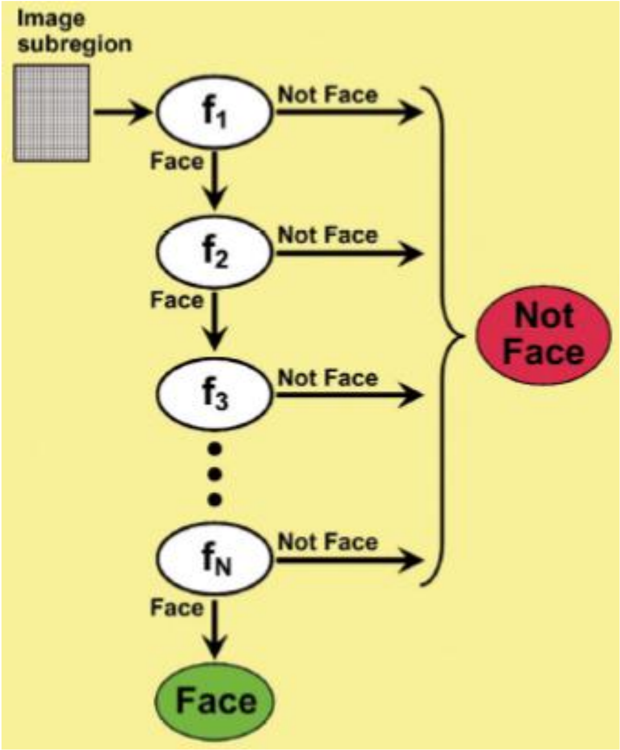



$$F_{Haar}=\left|\frac{E(R_{black})-E(R_{white})}{w\cdot h\cdot\sqrt{\left|E(R_{\mu})^{2}-E(R_{\mu}{}^{2})\right|}}\right|$$

# Haar特征介绍人脸检测

**AdaBoost**
-积分图技术
-级联分类器
-人脸检测与物体检测
-高拒绝率低通过率



# HAAR特征

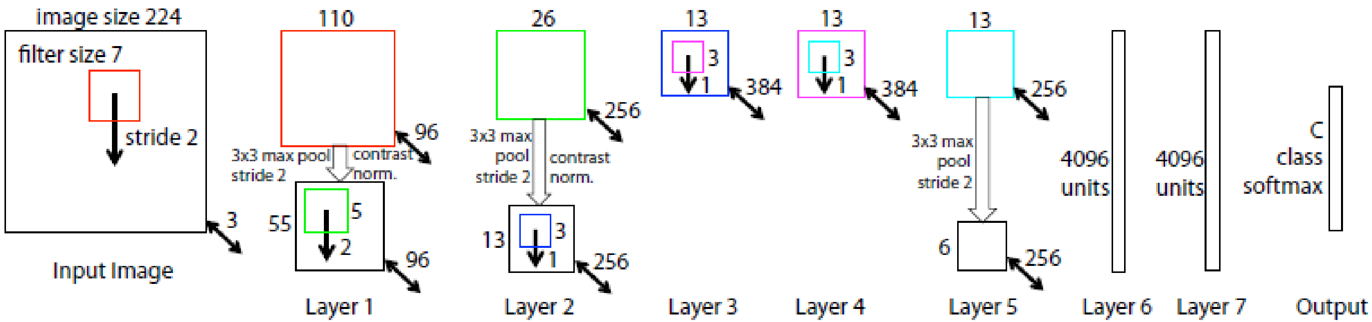
- 高类间变异性
- 低类内变异性
- 局部强度差
- 不同尺度
- 计算效率高

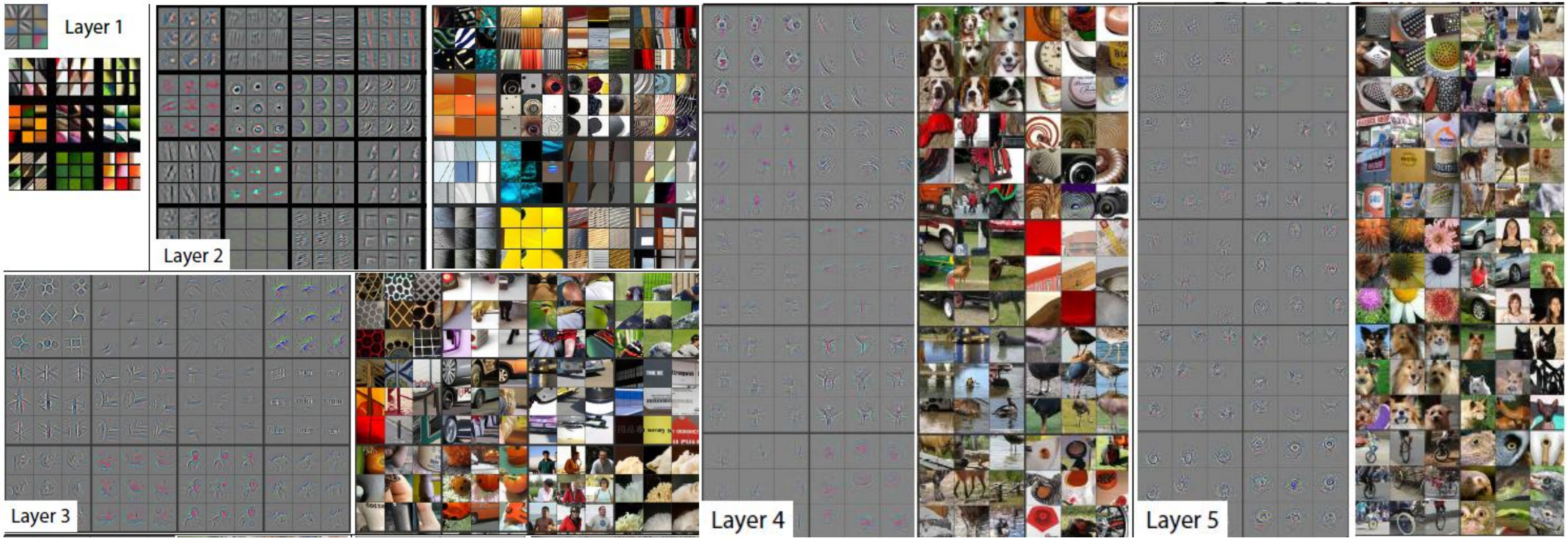
# 人工提取特征的局限性

- 基于先验模型与数学推导
- 无法描述一些数学模型无法建模的特征
- 人的计算能力远不如计算机

# 自动特征提取

- 卷积神经网络CNN – 监督学习
- 自编码学习 – 无监督学习






卷积层自动特征提取

# 迁移学习

- 基于原有训练好的模型
- Finetuning原有模型
- 大大提高了训练速度

# 实验 - 基于卷积神经网络的图像识别

- 下载安装 - https://github.com/tensorflow/models/tree/master/research/object_detection
- pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
- 下载数据-http://www.robots.ox.ac.uk/~vgg/data/pets/data/images.tar.gz
- 下载数据-http://www.robots.ox.ac.uk/~vgg/data/pets/data/annotations.tar.gz
- http://download.tensorflow.org/models/object_detection/ssd_mobilenet_v1_coco_2018_01_28.tar.gz

```
pip install --user Cython
pip install --user contextlib2
pip install --user pillow
pip install --user lxml
pip install --user jupyter
pip install --user matplotlib
```

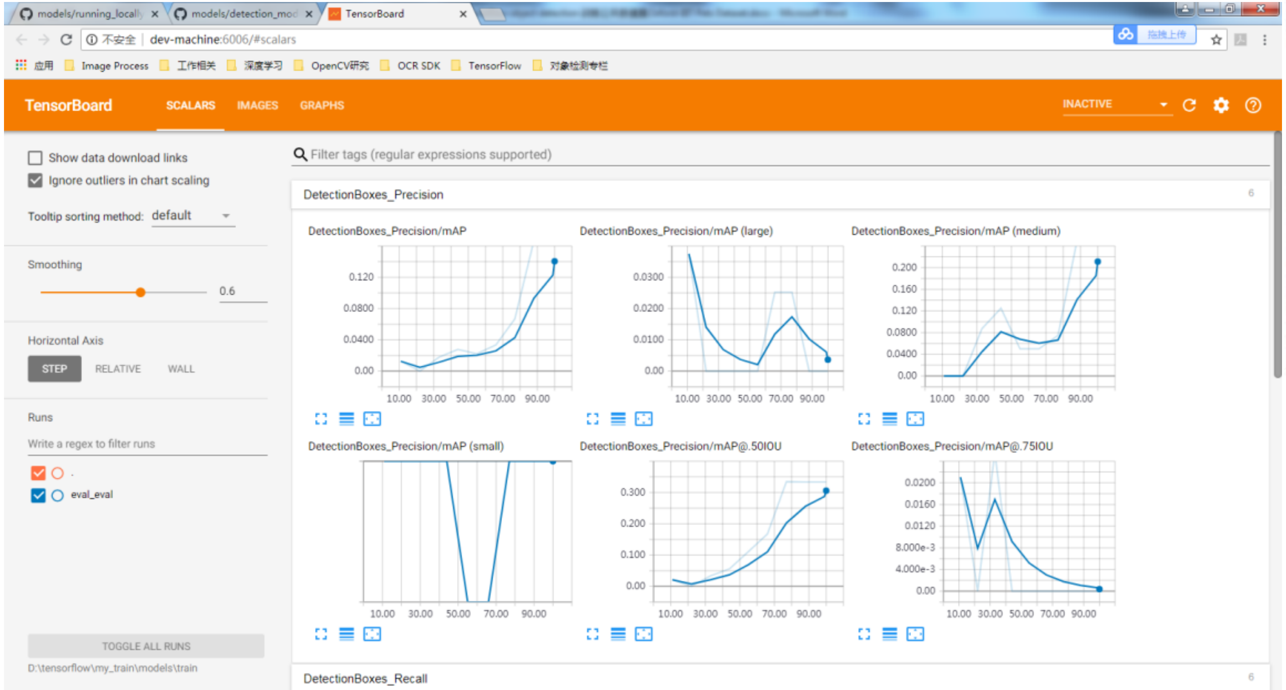
# 实验 － 基于卷积神经网络的图像识别

```
管理员: C:\Windows\system32\cmd.exe - tensorboard  --logdir=D:\tensorflow\my_train\models\train
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\zhigang>cd\

C:\>tensorboard --logdir=D:\tensorflow\my_train\models\train
W0819 16:20:35.533990 Reloader tf_logging.py:120] Found more than one graph event per run, or there was a metagraph containing a graph_def, as well as
 one or more graph events.  Overwriting the graph with the newest event.
W0819 16:20:35.533989 11832 tf_logging.py:120] Found more than one graph event per run, or there was a metagraph containing a graph_def, as well as on
e or more graph events.  Overwriting the graph with the newest event.
W0819 16:20:35.565991 Reloader tf_logging.py:120] Found more than one metagraph event per run. Overwriting the metagraph with the newest event.
W0819 16:20:35.565991 11832 tf_logging.py:120] Found more than one metagraph event per run. Overwriting the metagraph with the newest event.
TensorBoard 1.10.0 at http://dev-machine:6006 (Press CTRL+C to quit)
```

# 实验 - 基于卷积神经网络的图像识别




课程问题咨询微信

# Thank You！